

# Développement d'applications FLASH<sup>®</sup> LITE<sup>™</sup> 2.x et 3.0

© 2008 Adobe Systems Incorporated. Tous droits réservés.

Développement d'applications Flash® Lite™ 2.x et 3.x d'Adobe®

S'il est distribué avec un logiciel comprenant un contrat de licence, ce manuel, ainsi que le logiciel qui y est décrit, sont cédés sous licence et ne peuvent être utilisés ou copiés que conformément à la présente licence. Sauf lorsque cela est prévu par la licence, aucune partie de ce manuel ne peut être reproduite, conservée sur un support de stockage ou transmise par un moyen ou sous une forme quelconque (électronique, mécanique, enregistrée ou autre), sans l'autorisation écrite préalable d'Adobe Systems Incorporated. Veuillez noter que le contenu de ce manuel est protégé par des droits d'auteur, même s'il n'est pas distribué avec un logiciel comprenant un contrat de licence.

Les informations contenues dans ce manuel sont fournies à titre purement indicatif et ne doivent pas être considérées comme un engagement de la part d'Adobe Systems Incorporated, qui se réserve le droit de les modifier sans préavis. Adobe Systems Incorporated décline toute responsabilité en cas d'éventuelles erreurs ou inexactitudes relevées dans le contenu informationnel de ce manuel.

Nous attirons votre attention sur le fait que les illustrations ou images que vous pouvez être amené à inclure dans vos projets sont peut-être protégées par des droits d'auteur. L'exploitation de matériel protégé sans l'autorisation de l'auteur constitue une violation de droit. Assurez-vous d'obtenir les autorisations requises avant de procéder.

Toutes les références à des noms de sociétés utilisés dans les modèles sont purement fictives et ne renvoient à aucune entreprise existante.

Adobe, the Adobe logo, ActionScript, Creative Suite, Dreamweaver, Flash, Flash Lite, and Macromedia are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

ActiveX and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Macintosh is a trademark of Apple Inc., registered in the United States and other countries. Symbian and all Symbian based marks and logos are trademarks of Symbian Limited. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. SVG is a trademark of the World Wide Web Consortium; marks of the W3C are registered and held by its host institutions MIT, INRIA and Keio. ITC American Typewriter is a registered trademark of International Typeface Corporation. Arial is a trademark of The Monotype Corporation registered in the U.S. Patent and Trademark Office and certain other jurisdictions. Helvetica is a trademark of Heidelberger Druckmaschinen AG exclusively licensed through Linotype Library GmbH, and may be registered in certain jurisdictions. All other trademarks are the property of their respective owners.

Sorenson Spark™ video compression and decompression technology licensed from Sorenson Media, Inc.

MPEG Layer-3 audio compression technology licensed by Fraunhofer IIS and Thomson Multimedia (<http://www.iis.fhg.de/amm/>).

Portions licensed from Nellymoser, Inc. (<http://www.nelly-moser.com>).

Adobe Flash 9.2 video is powered by On2 TrueMotion video technology. © 1992-2005 On2 Technologies, Inc. All Rights Reserved. <http://www.on2.com>.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Notice to U.S. Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

# Sommaire

## Chapitre 1 : Présentation de Flash Lite 2.x et 3.x

Généralités sur Flash Lite .....	1
Fonctionnalités de Flash Lite 3.1 .....	2
Fonctionnalités de Flash Lite 3.0 .....	7
Fonctionnalités de Flash Lite 2.x .....	11

## Chapitre 2 : Prise en main de Flash Lite

Application Flash Lite Hello World .....	15
Emulateur Adobe Device Central et base de données de périphériques .....	15
Flux de travail lors de la création d'applications Flash Lite .....	16
A propos des types de contenus Flash Lite .....	17

## Chapitre 3 : Didacticiel : Création d'une application Flash Lite

Affichage de l'application terminée .....	19
Création de l'application .....	20

## Chapitre 4 : Création d'interactivité et d'éléments de navigation

Interaction de l'utilisateur et touches prises en charge .....	35
Navigation par défaut .....	37
Evénements de touche et de bouton .....	43

## Chapitre 5 : Utilisation du texte et des polices

Texte .....	53
Saisie de texte .....	54
Rendu des polices .....	63
Défilement du texte .....	66

## Chapitre 6 : Utilisation du son

Son de périphérique .....	68
Son natif .....	75

## Chapitre 7 : Utilisation de la vidéo

Travailler avec les fichiers FLV .....	79
Travailler avec les vidéos de périphérique .....	82

## Chapitre 8 : Développement d'applications Flash Lite pour BREW

Généralités sur BREW .....	95
Configuration de votre système pour BREW .....	99
Programmation de fichiers Flash Lite pour BREW .....	100
Publication de fichiers Flash Lite pour BREW .....	103
Téléchargement des fichiers sur les périphériques BREW .....	112

## Chapitre 9 : Test du contenu Flash Lite

Présentation du test .....	117
Utilisation de l'émulateur .....	118

**Chapitre 10 : Présentation du code ActionScript Flash Lite 2.x et 3.x**

Eléments de langage et classes ActionScript non pris en charge, et pris en charge intégralement ou partiellement	125
Classes non prises en charge ou partiellement prises en charge : détails	174
Eléments ActionScript non pris en charge ou pris en charge de façon partielle: Détails	191

**Chapitre 11 : Messages d'erreur et d'avertissement**

Codes d'erreur de périphérique	194
Messages d'informations et d'erreur de l'émulateur Adobe Device Central	196

<b>Index</b>	204
--------------	-----

# Chapitre 1 : Présentation de Flash Lite 2.x et 3.x

Adobe® Flash® Lite™ est une version de Flash Player® conçue spécifiquement pour les périphériques. Ce document porte sur les logiciels d'Adobe Macromedia® Flash® Lite™ 2.0 et Macromedia Flash Lite 2.1, qui sont regroupés sous l'appellation Flash Lite 2.x. Il couvre également Adobe Flash Lite 3.0 et 3.1, regroupés sous l'appellation Flash Lite 3.x.

## Généralités sur Flash Lite

### A propos de Flash Lite

Macromedia® Flash Lite 1.0 et Macromedia Flash Lite 1.1 s'appuient sur Macromedia Flash Player 4. Les versions 2.0 et 2.1 de Flash Lite (dénommées collectivement 2.x) s'appuient sur Macromedia Flash Player 7 et prennent en charge la plupart des fonctionnalités de Flash Player 7. Flash Lite 2.x propose également des fonctionnalités propres au développement de périphériques mobiles et qui ne sont pas disponibles dans Flash Player 7. Par exemple, dans Flash Lite 2.x vous pouvez charger des types de fichiers multimédias propres à chaque périphérique (image, son, vidéo) qui ne sont pas pris en charge par Flash Lite en mode natif. Flash Lite 2.x présente également des fonctionnalités d'intégration des périphériques, telles que la possibilité d'effectuer des appels téléphoniques et d'envoyer des messages texte.

Flash Lite 3.0 s'appuie sur Flash Player 7, prend en charge Video Flash (FLV) et la navigation dans la plupart des contenus Flash 8, et propose de nombreuses améliorations en termes de performance. Il intègre enfin un nouveau modèle de sécurité similaire à celui utilisé dans Flash Player 8.

Flash Lite 3.1 s'appuie aussi sur Flash Player 7. Il offre les fonctions pour améliorer la navigation sur le Web des utilisateurs de périphériques mobiles. De nombreuses fonctions sont intégrées dans Flash® Player 8, mais ne sont pas prises en charge par Flash Lite ; elles ne sont pas prises en charge par Flash Lite 3.1. Flash Lite 3.1 offre la capacité de lire le contenu ActionScript™ 2.0 détecté sur des sites Web Flash 9. En outre, Flash Lite 3.1 ne prend pas en charge le traitement et la lecture de ActionScript™ 3.0, introduit dans Flash Player 9. Reportez-vous à « [Fonctionnalités de Flash Lite 3.1](#) » à la page 2 pour de plus amples informations.

### A propos des composants de Flash Lite

Les composants installés avec Flash (par exemple, DataGrid et Accordion) ont été conçus pour les applications de bureau Flash. La mémoire et la puissance de traitement requises rendent généralement leur exploitation impossible pour les applications Flash Lite. Adobe déconseille l'utilisation de composants d'interface utilisateur standard dans les applications Flash Lite.

### Optimisation du contenu pour de meilleures performances et davantage d'espace mémoire

Pour optimiser du contenu Flash, il est nécessaire de respecter quelques principes de base. Les développeurs Flash, par exemple, doivent souvent éviter les graphismes extrêmement complexes, les interpolations à outrance et une utilisation abusive de l'effet de transparence.

Même si les précédentes versions de Flash ont pratiquement résolu tous les problèmes de performances pour les ordinateurs de bureau, les développeurs d'Adobe Flash Lite rencontrent encore quelques difficultés dues aux limites des appareils portables : certains périphériques sont (parfois nettement) plus performants que d'autres, et comme la programmation pour périphériques portables sous-entend en général une publication sur de nombreux types de périphériques, les développeurs doivent quelquefois concevoir leurs applications en se basant sur le plus petit dénominateur commun.

L'optimisation de contenu pour des appareils mobiles nécessite donc souvent de faire des compromis. Ainsi, une technique donnée permettra d'obtenir un meilleur rendu, mais une autre offrira de meilleures performances. Pour trouver le meilleur compromis à vos yeux, vous serez donc amené à effectuer une série de tests à répétition dans l'émulateur et sur le ou les périphériques cibles.

Dans Flash 8, l'émulateur prévu pour Flash Lite 2.x était intégré à l'environnement de programmation de Flash. Dans Flash CS3 et Flash Lite 3.0 puis avec Flash CS4 et Flash Lite 3.1, la fonctionnalité d'émulateur fait partie d'Adobe® Device Central. Device Central permet de simuler l'exécution de vos projets Flash Lite sur divers appareils en émulant l'affichage du périphérique, l'utilisation de la mémoire et les performances de certains périphériques particuliers. Pour toute information sur l'utilisation de l'émulateur afin d'optimiser votre contenu Flash Lite sur des périphériques portables, consultez la section « Meilleures pratiques relatives au contenu de périphériques mobiles » dans la documentation Device Central.

## Fonctionnalités de Flash Lite 3.1

### Navigation Web améliorée

Flash Lite 3.1 propose un ensemble de fonctionnalités permettant d'améliorer la navigation sur le Web des utilisateurs de périphériques mobiles. Nombre de ces fonctionnalités étaient présentes sur Flash® Player 8 mais pas prises en charge par Flash Lite. Elles sont désormais prises en charge par Flash Lite 3.1. Ces nouvelles fonctionnalités sont décrites ci-après.

### Navigation Flash 9

Flash Lite 3.1 a été publiée avec la prise en charge de Flash Player 8 et ne prend donc pas en charge ActionScript™ 3.0, qui a été introduite sur Flash Player 9. Néanmoins, de nombreux sites Web Flash 9 ont du contenu ActionScript 2.0, qui ne pouvait pas être lu dans Flash Lite 3.0. Flash Lite 3.1 introduit la capacité de lire ce contenu lorsqu'il est rencontré sur des sites Web Flash 9. Le lecteur Flash Lite 3.1 se présente lui-même comme un lecteur capable de traiter du contenu Flash 9. Lorsque des sites Web comportant du contenu Flash 9 sont chargés, Flash Lite 3.1 évalue le fichier SWF et lit le contenu Flash 9 utilisant ActionScript 2.0 uniquement. Si le contenu Flash 9 utilise Actionscript 3, Flash Lite 3.1 ne le lit pas, mais affiche à la place une icône d'erreur conviviale (identique sur Flash Lite 3.0).

### Objet LocalConnection

La classe LocalConnection permet de développer un ou plusieurs fichiers SWF qui peuvent échanger des instructions entre eux sans utiliser `fscommand()` ou JavaScript. Les objets LocalConnection peuvent uniquement communiquer entre des fichiers SWF exécutés sur le même client, mais ils peuvent concerner différentes applications. Les objets LocalConnection peuvent également être utilisés pour envoyer et recevoir des données dans un seul fichier SWF, mais cette fonctionnalité n'est pas amplement utilisée. La mise en œuvre Flash Lite 3.1 de l'objet LocalConnection prend en charge la communication parmi les fichiers SWF s'exécutant dans le même processus (par exemple dans la même fenêtre de navigateur).

## Gestion du texte HTML

Flash Lite 3.1 introduit la prise en charge de plusieurs balises HTML de gestion de texte qui n'étaient pas prises en charge dans Flash Lite 3.0. Flash Lite 3.1 inclut la prise en charge pour l'utilisation de texte comme liens, la prise en charge de la balise `<img>` et la prise en charge de la gestion de feuilles de style en cascade (CSS).

**Texte sous forme de liens** La balise `<a>` est utilisée pour créer un lien hypertexte et est désormais prise en charge dans Flash Lite 3.1. Voir la documentation Flash Lite pour obtenir des informations complètes concernant l'utilisation et les attributs de la balise `<a>`.

**Images intégrées externes** Les champs de texte dans Flash Lite 3.1 prennent en charge la balise `<img>`. Cette balise peut pointer des images stockées dans le système de fichiers local ou sur le réseau. L'image peut être au format JPG, GIF ou PNG et sera restituée si la plateforme prend en charge ce type d'image. Le champ de texte lui-même peut avoir différents attributs, comme multiligne et retour automatique à la ligne, qui affecte le rendu du texte. En plus d'images, la balise `<img>` peut également pointer un fichier SWF ou un objet MovieClip. Lorsqu'ils sont intégrés dans ce champ de texte, ces supports sont traités comme des enfants du champ de texte et peuvent donc être contrôlés et manipulés via `ActionScript`.

**Prise en charge CSS** Le formatage de texte avec des styles CSS n'était pas pris en charge dans Flash Lite 3.0. Le contenu Flash 8 visualisé dans Flash Player sur le bureau prend en charge le formatage style CSS — chargé sous forme de fichier de feuilles de style CSS distincts ou en ajoutant des styles incorporés via `ActionScript`. La fonction CSS permet aux développeurs d'ajuster l'aspect et le rendu de leurs applications. Dans Flash Lite 3.1, cette fonctionnalité est prise en charge.

## Prise en charge de WMODE

Flash Lite 3.1 introduit la prise en charge du paramètre `wmode`. En utilisant ce paramètre dans le contexte du module d'extension ActiveX et un navigateur, vous pouvez manipuler l'arrière-plan de la page HTML qui contient le film Flash pour contrôler si l'image ou la couleur d'arrière-plan transparait ou non, permettant d'ajouter le contenu rich media au contenu DHTML. La valeur de `wmode` peut être `window` (par défaut), `opaque` ou `transparent` :

- `Window` lit l'application dans sa propre fenêtre rectangulaire sur une page Web. `Window` indique que l'application n'a aucune interaction avec les calques HTML et qu'elle est toujours l'élément le plus visible.
- `Opaque` masque tout élément de la page derrière l'application.
- `Transparent` rend l'arrière-plan de la page HTML visible sur toutes les parties transparentes de l'application et peut ralentir les performances des effets animés.

## Prise en charge des propriétés scale, salign et align

Flash Lite 3.1 introduit la prise en charge des paramètres JavaScript `scale`, `salign` et `align`, qui contrôlent de quelle manière le contenu rich media est mis à l'échelle, positionné et aligné lorsqu'il est intégré dans une page HTML. Le paramètre `scale` définit de quelle manière le fichier SWF est placé dans la fenêtre du navigateur lorsque les valeurs de largeur et de hauteur sont des pourcentages. Ses valeurs peuvent être `showall`, `noborder` ou `exactfit`. L'attribut d'alignement détermine de quelle manière le fichier SWF est positionné dans la fenêtre du navigateur. Ses valeurs peuvent être `L` (gauche), `R` (droite), `T` (haut) ou `B` (bas). Le paramètre d'alignement spécifie l'endroit où un fichier SWF redimensionné est placé à l'intérieur de la zone définie par les paramètres `width` et `height`. Ses valeurs peuvent être `TL` (haut gauche), `TR` (haut droite), `BL` (bas gauche) ou `BR` (bas droite).

## Paramètre GetURL \_target

`_target` est un paramètre optionnel de la fonction `ActionScript` global `getURL` qui spécifie la fenêtre ou le cadre HTML dans lequel le document devrait se charger. Le paramètre `_target` n'était pas pris en charge par les applications Flash Lite, car les navigateurs de périphériques mobiles ne prenaient pas en charge avant récemment plusieurs fenêtres. Vous pouvez entrer une chaîne vide ou choisir parmi les noms de cible réservés suivants :

`_self` spécifie le cadre actif de la fenêtre en cours d'utilisation.

`_blank` crée une fenêtre.

`_parent` appelle le parent du cadre actif.

`_top` sélectionne le cadre de plus haut niveau de la fenêtre active. Si aucun `_target` n'est spécifié, le paramètre par défaut est alors `_self`. La fenêtre de navigateur doit posséder plusieurs cadres pour que tous les paramètres `_target` puissent fonctionner normalement. Si le navigateur ne comporte pas de cadres, `_blank` fonctionne comme prévu. Il ouvre l'URL dans une nouvelle fenêtre vierge. Néanmoins, les paramètres `_self`, `_parent` et `_top` ouvrent l'URL dans la même fenêtre.

## Modifications du comportement de SharedObject

Les règles de stockage des objets partagés locaux dans Flash Lite 3.1 sont désormais pleinement compatibles avec celles de Flash Player 8 (bureau). Avant Flash Lite 3.1, l'emplacement d'un objet partagé était issu du nom du fichier SWF. Ainsi, un objet partagé stocké par A.swf ne pouvait pas être ouvert par B.swf, où que se trouvent ces fichiers SWF. Dans Flash Lite 3.1, l'emplacement d'un objet partagé est issu de son URL au lieu de son nom, ainsi /data/A.swf situé sur un périphérique mobile peut accéder aux objets partagés stockés par /B.swf. Une autre différence importante entre Flash Lite 3.1 et les versions précédentes repose sur le fait que dans lesdites versions si un fichier SWF était modifié, même si son nom ne changeait pas, ses objets partagés étaient traités comme étant différents de ceux d'origine. Dans Flash Lite 3.1, l'objet partagé d'un fichier SWF est traité comme étant le même aussi longtemps que l'URL/le sandbox est identique, même si le fichier SWF lui-même a été modifié ou renommé. En outre, avant la version 3.1, deux versions différentes d'un fichier SWF ne pouvaient pas accéder réciproquement aux objets partagés de l'autre ; dans la version 3.1, cet accès est possible. Flash Lite 3.1 intègre d'importantes modifications de comportement qui furent introduites avec Flash Lite 3.0. Avant la version Flash Lite 3.0, les objets partagés de Flash Lite n'étaient disponibles qu'avec des fichiers SWF stockés en local. Les fichiers SWF lus dans un navigateur activé par réseau ne pouvaient pas utiliser les objets partagés Flash Lite. Dans la version Flash Lite 3.0 et versions ultérieures, les fichiers SWF exécutant le sandbox de sécurité réseau peuvent accéder aux et stocker les objets partagés. En outre, depuis la version Flash Lite 3.0, tous les fichiers SWF réseau peuvent stocker et accéder aux objets partagés en local, mais l'accès est limité à un fichier SWF. Dans Flash Lite 3.1, tous les fichiers SWF issus du même domaine de réseau peuvent accéder aux objets partagés de chacun d'eux.

## Paramètres facultatifs de SharedObject.getLocal

La classe SharedObject comprend une méthode `getLocal` avec deux paramètres en option.

- `secure` : un paramètre booléen (introduit dans Flash 8) que vous pouvez utiliser pour faire la distinction entre les objets partagés issus des domaines HTTP et HTTPS. Flash Lite 3.1 prend désormais en charge le paramètre `secure`.
- `localPath` : non pris en charge sur Flash Lite.

## API d'image

Flash Lite 3.1 prend en charge les API que les développeurs d'applications utilisent pour accroître l'expressivité en déplaçant, pivotant, mettant à l'échelle et inclinant des images dans des applications mobiles. Les packages `flash.display` et `flash.geom`, introduits pour la première fois dans Flash 8, forment la base de cette nouvelle fonctionnalité. Le package `flash.display` prend en charge la classe `BitmapData`, que vous utilisez pour créer des bitmaps de tailles arbitraires et `flash.geom` prend en charge des classes et méthodes que vous utilisez pour réaliser des manipulations au niveau des pixels, dont le déplacement, la rotation, la mise à l'échelle et l'inclinaison. Le document *Référence du langage ActionScript Adobe Flash Lite 2.x et 3.x* présente toutes les informations sur l'utilisation de ces paquets.



### Classe `flash.display.BitmapData`

Cette classe vous permet de créer des images bitmap opaques ou transparentes de tailles arbitraires et de les manipuler de différentes façons au moment de l'exécution, ce qui sépare les opérations de rendu bitmap des routines de mise à jour d'affichage interne du lecteur Flash Lite. Vous pouvez associer des objets `BitmapData` à des objets `MovieClip` à l'aide de la méthode `MovieClip.attachBitmap()`. Les méthodes `BitmapDataClass` suivantes sont prises en charge partiellement ou ne le sont pas dans Flash Lite 3.1. Toutes les autres méthodes fonctionnent comme indiqué dans la documentation Flash Lite.

### Méthodes partiellement prises en charge

Les méthodes `BitmapDataClass` suivantes sont partiellement prises en charge dans Flash Lite 3.1.

Méthode	Description
<code>copyPixels</code>	Met en place une routine rapide permettant de manipuler les pixels de différentes images sans effets d'étirement, de rotation ou de couleur. Flash Lite 3.1 ne prend pas en charge les paramètres <code>alphaBitmap</code> , <code>alphaPoint</code> ou <code>mergeAlpha</code> de cette méthode.
<code>draw</code>	Dessine une image source ou un clip sur une image de destination avec la fonctionnalité de rendu vectoriel de Flash Lite. Flash Lite 3.1 ne prend pas en charge le paramètre <code>blendmode</code> de cette méthode.

### Méthodes non prises en charge

Les méthodes `BitmapDataClass` ne sont pas prises en charge dans Flash Lite 3.1.

Méthode	Description
<code>applyFilter</code>	Prend une image source et un objet filtre et génère l'image filtrée. Flash Lite 3.1 ne prend pas les filtres en charge. Cette méthode n'est donc pas prise en charge.
<code>generateFilterRect</code>	Détermine le rectangle de destination affecté par l'appel de la méthode <code>applyFilter()</code> en fonction d'un objet <code>BitmapData</code> , d'un rectangle source et d'un objet filtre spécifiés. Flash Lite 3.1 ne prend pas les filtres en charge. Cette méthode n'est donc pas prise en charge.
<code>noise</code>	Remplit une image avec des pixels représentant un bruit aléatoire. Flash Lite 3.1 ne prend pas cette méthode en charge.
<code>paletteMap</code>	Remappe les valeurs des canaux de couleur dans une image recevant jusqu'à quatre tableaux de données de palette de couleurs, un pour chaque canal. Flash Lite 3.1 ne prend pas cette méthode en charge.
<code>perlinNoise</code>	Crée une image de bruit Perlin. Flash Lite 3.1 ne prend pas cette méthode en charge.
<code>pixelDissolve</code>	Procède à la dissolution de pixels, soit d'une image source vers une image de destination, soit en utilisant la même image. Flash Lite 3.1 ne prend pas cette méthode en charge.
<code>scroll</code>	Fait défiler une image en fonction d'un certain montant en pixels (x, y). Flash Lite 3.1 ne prend pas cette méthode en charge.
<code>threshold</code>	Teste les valeurs de pixels d'une image selon un seuil spécifié et définit les pixels qui réussissent le test sur de nouvelles valeurs de couleur. Flash Lite 3.1 ne prend pas cette méthode en charge.

**Classe flash.geom.Point**

Cette classe représente un emplacement dans un système de coordonnées à deux dimensions,  $x$  représentant l'axe horizontal et  $y$  l'axe vertical. Cette classe est entièrement prise en charge dans Flash Lite 3.1.

**Classe flash.geom.Matrix**

Cette classe représente une matrice qui détermine le mappage des points d'un espace de coordonnées à l'autre. Pour effectuer diverses transformations graphiques d'un objet, il vous suffit de définir les propriétés d'un objet Matrix et de l'appliquer à un objet MovieClip ou BitmapData. Ces fonctions de transformation incluent le déplacement, la rotation, la mise à l'échelle et l'inclinaison. Flash Lite 3.1 prend en charge toutes les méthodes de la classe Flash.geom.matrix sauf createGradientBox.

**Classe flash.geom.ColorTransform**

Cette classe permet de régler de façon mathématique l'ensemble des valeurs de couleur dans un clip. La fonction de réglage des couleurs ou de transformation de couleur peut être appliquée aux quatre canaux : rouge, vert, bleu et transparence alpha. Cette classe est entièrement prise en charge dans Flash Lite 3.1.

**Classe flash.geom.Rectangle**

Cette classe est utilisée par l'objet BitmapData pour manipuler une région spécifique d'un bitmap. La classe fournit des méthodes utilitaires pour manipuler un ou plusieurs rectangles. Cette classe est entièrement prise en charge dans Flash Lite 3.1.

**Classe flash.geom.Transform**

Cette classe rassemble des données sur les transformations de couleurs et les manipulations de coordonnées connexes à un objet MovieClip. Un objet Transform s'obtient normalement en lisant la valeur de la propriété transform dans un objet MovieClip. Cette classe est entièrement prise en charge dans Flash Lite 3.1.

**setTimeout/clearTimeout**

Les fonctions globales ActionScript 2.0 setTimeout et clearTimeout n'étaient pas prises en charge dans Flash Lite 3.0, mais le sont dans Flash Lite 3.1. La fonction setTimeout est utilisée pour exécuter une fonction spécifique après un délai spécifié (en millisecondes). Veuillez noter que l'utilisation de setTimeout/clearTimeout conjointement avec du texte d'entrée en ligne peut entraîner des problèmes pour les applications d'utilité temporaire, comme des jeux ou questionnaires en ligne. En effet, le traitement en arrière-plan de tous les ActionScript (dont le compteur) est interrompu lorsque l'utilisateur du périphérique entre du texte.

**Fonctionnalité de recherche de diffusion vidéo en continu**

Flash Lite 3.1 prend en charge la fonctionnalité de recherche dans les fichiers vidéo en flux continu (FLV). Dans Flash Lite 3.0, la relecture vidéo intègre uniquement une prise en charge minimale de la recherche. La recherche d'une position antérieure ou postérieure par rapport à la position de tête actuelle n'est prise en charge que pour les vidéos HTTP de courte durée. Flash Lite 3.1 prend en charge la capacité de recherche également dans les vidéos FLV de grande taille. La recherche d'une position antérieure est toujours prise en charge. En revanche, la recherche d'une position postérieure est soumise à certaines limitations. La recherche d'une position postérieure fonctionne uniquement pour la portion du flux qui a été téléchargée. Toute tentative de recherche vers une position au-delà de la fin du flux téléchargé ne génère aucun résultat. A la place, la vidéo poursuit sa lecture à partir de la position active de la tête. La fonctionnalité de recherche est utilisée conjointement avec la classe Netstream. Netstream.seek() est utilisée pour une recherche d'une position antérieure ou postérieure dans le flux vidéo HTTP.

### Diffusion en continu de fichiers mp3

Flash Lite 3.1 prend en charge la diffusion en continu de fichiers audio mp3 à l'aide de l'objet Sound d'ActionScript. Dans Flash Lite 3.0, les fichiers mp3 devaient être intégralement téléchargés avant de pouvoir être décodés et lus. Et si le périphérique ne parvenait pas à allouer suffisamment de mémoire pour traiter l'ensemble du fichier, des erreurs survenaient. Flash Lite 3.1 prend en charge la diffusion du son en continu progressive sur les protocoles de système de fichiers local et HTTP, ce qui signifie que les fichiers mp3 peuvent être diffusés en continu à partir d'un serveur Web ou du stockage de fichiers local du périphérique. Les données au format mp3 sont traitées à mesure qu'elles sont téléchargées, ce qui optimise la consommation de mémoire et donne la possibilité aux périphériques de lire des fichiers mp3 plus gros que ce qui était possible sur Flash Lite 3.0.

### Extensions ActionScript

La fonction des Extensions dans Flash Lite 3.1 permet à des développeurs tiers d'ajouter des fonctionnalités et des API ActionScript à Flash Lite à l'aide d'une bibliothèque statique ou de liens dynamiques. La DLL fournit la mise en œuvre pour les fonctionnalités et Flash Lite permet à la DLL d'être utilisée en appelant des API ActionScript. Il est ainsi possible pour les développeurs de Flash Lite d'interagir avec des API exposées par des dispositifs hôtes en écrivant une DLL qui réunit les capacités du périphérique et du lecteur Flash. Ces extensions peuvent être liées statiquement ou dynamiquement à la mise en œuvre de l'hôte. Par exemple, en utilisant cette fonctionnalité, les développeurs peuvent écrire des applications qui permettent d'accéder au système d'archivage du périphérique, ou d'utiliser des données GPS fournies par le périphérique, etc.

*Remarque : la fonction ActionScript Extensions n'est disponible pour les développeurs tiers que si cette fonction a été activée par le fabricant du périphérique mobile.*

### Améliorations vidéo

Les améliorations suivantes ont été apportées à la fonctionnalité vidéo dans Flash Lite 3.1.

**Lecture vidéo H.264** Tout comme la version 9.2 du lecteur Flash, Flash Lite 3.1 peut lire des fichiers SWF Flash 7, 8 et 9 qui utilisent FLV en remplaçant FLV par H.264 sur le serveur (Flash Media Server ou HTTP).

**Post-traitement vidéo** Flash Lite 3.1 prend en charge le post-traitement vidéo (en utilisant les propriétés de lissage et de dégroupage de l'objet Video) qui offre une meilleure qualité d'affichage des fichiers vidéo sur les périphériques.

## Fonctionnalités de Flash Lite 3.0

### Nouvelles fonctionnalités de Flash Lite 3.0

**Amélioration des performances** Présente les modifications visant à améliorer les performances du lecteur sur les périphériques portables et le confort d'utilisation du périphérique lorsque l'utilisateur affiche du contenu et interagit avec ce dernier. Pour plus d'informations, consultez la section « [Amélioration des performances](#) » à la page 8.

**Fonctionnalités vidéo améliorées** Prend en charge la vidéo Flash (FLV) et la vidéo de périphérique. Pour plus d'informations, consultez la section « [Fonctionnalités vidéo améliorées](#) » à la page 8.

**Navigabilité du contenu Web** Prend en charge la lecture de la plupart du contenu Flash 8 sur des périphériques mobiles, avec une certaine perte de fidélité. Pour plus d'informations, consultez la section « [Navigabilité du contenu Web](#) » à la page 8.

**Sécurité des fichiers locaux** Fournit à Flash Lite un schéma de sécurité cohérent avec le modèle de sécurité mis en œuvre dans Flash Player 8. Pour plus d'informations, consultez la section « [Sécurité des fichiers locaux](#) » à la page 9.

**Gestion du contenu** Fournit une API permettant de lire les métadonnées SWF. Les développeurs de contenu utilisent les métadonnées pour segmenter le contenu sur les périphériques et contrôler certains aspects relatifs à la gestion du contenu. Pour plus d'informations, consultez la section « [Gestion du contenu](#) » à la page 10.

## Amélioration des performances

**Accélération du rendu des graphiques** Restitue rapidement et efficacement le défilement du texte, les conversions vectorielles et les vecteurs animés avec des remplissages opaques.

**Diminution de la mémoire d'exécution** Charge passivement les classes ActionScript™ lors de l'exécution, de sorte que seul le code ActionScript requis pour le rendu de la première image est chargé au démarrage. Les autres classes ActionScript sont chargées en cas de besoin par le contenu.

## Fonctionnalités vidéo améliorées

Flash Lite 2.x ne prend en charge que la vidéo de périphérique, ce qui signifie qu'il s'appuie sur le périphérique pour décoder et rendre la vidéo. Il est limité par les formats vidéo pris en charge par chaque périphérique. Il ne peut pas synchroniser la vidéo avec le scénario, ni contrôler le volume du clip vidéo. Il ne permet pas de faire pivoter ou d'incliner la vidéo de périphérique. Enfin, il n'autorise pas la composition ni le fondu de vidéos de périphérique avec d'autres supports. (La vidéo reste toujours au-dessus du contenu Flash.)

Flash Lite 3.0 ajoute la prise en charge de Flash Video (FLV) à l'aide des versions des codecs On2 et Sorenson optimisés pour les périphériques portables. Le rendu de la vidéo FLV s'effectue directement dans le lecteur Flash Lite et non pas sur le périphérique. Ainsi, vous n'avez plus à vous demander si les périphériques cibles prennent en charge un format vidéo particulier. Flash Lite 3.0 prend en charge la lecture de Flash Video à l'aide des méthodes suivantes :

- Intégration de la vidéo dans des fichiers SWF
- Téléchargement progressif de la vidéo à partir de dossiers locaux et de serveurs Web (HTTP)
- Diffusion de la vidéo en flux continu à partir de Flash Media Server

Consultez « [Utilisation de la vidéo](#) » à la page 79 pour obtenir toutes les informations nécessaires à l'utilisation de FLV dans Flash Lite 3.0.

## Navigabilité du contenu Web

Flash Lite 3.0 prend en charge la lecture de la plupart du contenu Flash 8 sur des périphériques mobiles, avec une certaine perte de fidélité. En raison des importantes capacités (mémoire et microprocesseur) requises par le jeu de fonctions d'expressivité de Flash 8, nombre d'entre elles ne sont pas prises en charge dans Flash Lite 3.0. Ces fonctionnalités incluent la mise en mémoire cache des bitmaps, les modes de fondu, les filtres de bitmaps, l'amélioration des traits et des dégradés, ainsi que la mise à l'échelle en 9 découpes. Lorsque Flash Lite détecte l'une de ces fonctionnalités dans un fichier SWF, il rend ces objets sans leur ajouter cette expressivité supplémentaire pour maintenir la convivialité du contenu mobile.

Le tableau ci-dessous décrit le contenu Flash 8 et les fonctionnalités qui sont prises en charge, partiellement prises en charge ou pas du tout prises en charge.

Fonctionnalité	Prise en charge	Détails
Balises HTML de base Flash 8	Oui	Le lecteur Flash Lite reconnaît et lit désormais la plupart du contenu Flash 8, avec une certaine perte de fidélité.
Améliorations de la sécurité	Oui	Le modèle de sécurité de Flash Lite est désormais cohérent avec celui du lecteur de bureau. Pour plus d'informations, consultez la section « <a href="#">Sécurité des fichiers locaux</a> » à la page 9.
Flash Video	Oui	Le lecteur Flash Lite prend désormais en charge le format FLV (Flash Video) grâce aux codecs On2 et Sorenson optimisés pour les périphériques mobiles. Prend également en charge l'implémentation des clients RTMP (mode unidirectionnel, pas de mode temps réel bidirectionnel). Pour plus d'informations, consultez la section « <a href="#">Fonctionnalités vidéo améliorées</a> » à la page 8.
Balise HTML de base	Oui	Permet à Flash Lite de se comporter d'une manière similaire au Flash Player sur le bureau.
Texte FlashType	Prise en charge partielle	La prise en charge totale n'est pas indispensable pour afficher le contenu. Le texte est rendu, mais sans l'intégralité de l'implémentation FlashType.
Mise en mémoire cache, effets et améliorations des bitmaps	Non	Non implémentées car la plupart requièrent la prise en charge de la virgule flottante ou sollicitent de manière excessive les ressources du processeur et la mémoire.
Dégradés focalisés	Non	Non implémentés car ils sollicitent énormément le processeur et ne sont pas indispensables pour afficher le contenu.

## Sécurité des fichiers locaux

Dans Flash Lite 2.x, les fichiers SWF locaux peuvent manipuler et charger des données provenant d'autres fichiers SWF locaux ou distants. Ceci crée un problème de sécurité car un fichier SWF local non approuvé, téléchargé par exemple en tant que pièce jointe, peut accéder à des données privées stockées sur le périphérique et les renvoyer à l'auteur du contenu via une connexion HTTP.

Flash Lite 3.0 intègre un nouveau modèle de sécurité basé sur celui de Flash Player 8. Tous les fichiers SWF sont placés dans un « sandbox » lors de leur chargement dans le lecteur Flash Lite. Chaque fichier SWF est affecté à un sandbox différent selon son origine.

Tous les fichiers SWF du réseau sont placés dans un sandbox distant correspondant à leur domaine d'origine. Ces fichiers ne peuvent lire que les données de leur domaine d'origine. Ils n'ont pas accès aux données d'autres domaines, sauf si ces derniers autorisent explicitement le fichier SWF qui émet la demande. Par exemple, un fichier SWF de a.com peut lire (à l'aide de `XML.load`, par exemple) les données du serveur b.com si le fichier de stratégie de ce dernier autorise l'accès à partir de a.com (ou de tous les domaines). Un fichier SWF de a.com peut exécuter le script d'un fichier SWF de b.com (appeler une méthode `ActionScript™` dans le fichier SWF de b.com, par exemple) si le fichier SWF de b.com appelle `System.security.allowDomain` pour autoriser l'accès à partir de a.com.

Les fichiers SWF locaux (issus du système de fichiers local) sont placés dans l'un des trois sandbox ci-dessous.

**Local avec système de fichiers** Les fichiers SWF peuvent lire (à l'aide de `XML.load`, par exemple) le contenu des fichiers situés dans les systèmes de fichiers locaux, mais ils ne peuvent pas communiquer avec le réseau.

**Local approuvé** Les fichiers SWF peuvent lire le contenu des fichiers locaux, interagir avec les serveurs et exécuter le script de n'importe quel autre fichier SWF. (Dans Flash Lite 2.1, tous les fichiers locaux sont placés dans ce sandbox.)

**Sandbox local avec interconnexion** Les fichiers SWF sont autorisés à communiquer avec d'autres SWF locaux interconnectés et à envoyer des données à des serveurs (à l'aide de `XML.send`, par exemple).

**Remarque :** Ce nouveau schéma de sécurité modifie le fonctionnement de pratiquement tout le code `ActionScript` qui charge des données ou exécute le script d'autres fichiers, notamment des fonctions courantes comme `getURL`, `loadMovie`, `loadMovieNum`, `loadVars`, `loadSound`, `XMLSocket.send` et `XMLSocket.load`. Pour obtenir la liste complète des fonctionnalités API concernées par la sécurité, consultez le document *Flash Player 8 Security-Related APIs* disponible au format PDF sur le site *Web Flash Player Developer Center*.

Pour plus d'informations sur le fonctionnement du modèle de sécurité de Flash 8 Player, consultez l'aide du produit. Lorsque vous consultez cette aide, gardez à l'esprit que les modèles de sécurité de Flash Lite 3.0 et de Flash Player 8 présentent les différences suivantes :

- La gestion des approbations est assurée par l'application hôte dans Flash Lite.
- Aucun fichier de configuration des approbations n'est requis dans Flash Lite.
- Les développeurs de fichiers SWF, les fabricants et les opérateurs de télécommunication doivent prendre toutes les décisions de sécurité dans Flash Lite. Il n'existe aucun gestionnaire de paramètres ni de fonction de gestion des approbations configurable par l'utilisateur. Toutefois, l'application hôte peut fournir un mécanisme permettant aux utilisateurs de prendre certaines décisions relatives à la sécurité. Par exemple, l'hôte peut autoriser les utilisateurs à désigner un fichier SWF comme approuvé.
- Les opérations interdites échouent silencieusement. Aucune boîte de dialogue ne s'affiche dans Flash Lite.

## Gestion du contenu

Flash Lite 3.0 fournit une API qui autorise le logiciel hôte d'un périphérique à lire les métadonnées SWF, qui sont des informations standard intégrées aux fichiers SWF. Si l'hôte choisit d'implémenter cette fonctionnalité, les développeurs de contenu Flash lite peuvent utiliser les métadonnées afin d'organiser et de classer le contenu du périphérique.

Les catégories d'informations suivantes peuvent être incluses en tant que métadonnées. Il est possible que certaines d'entre elles soient implémentées différemment (voire pas du tout) par le fabricant des périphériques pour lesquels vous développez du contenu.

Titre	Description
Auteur du contenu	Nom de l'auteur du contenu.
Nom de la société	Nom de la société de l'auteur du contenu.
Titre	Titre du contenu.
Description	Description du contenu.
Informations de copyright	Détails sur le copyright.
Version du contenu	Numéro de version du contenu (PAS la version de Flash).
Type(s) de contenu	Catégories de types de contenus du fichier. Consultez le tableau des types de contenus ci-après pour plus d'informations sur les types de contenus.
Verrou de transmission	Spécifie si le fichier de contenu peut être transféré à un autre périphérique.
Résolutions d'affichage prises en charge	Liste des résolutions prises en charge par le contenu.
Icône	Type et emplacement de l'icône du contenu.

Titre	Description
Profil du lecteur Flash	Version de Flash/Flash Lite pour laquelle le contenu est publié.
Taille de stockage	Taille de stockage (taille de <code>SharedObject</code> ) requise par le contenu. Si la taille n'est pas spécifiée, le contenu adopte la taille par défaut définie par la plate-forme (en général, 4 Ko).
Alpha d'arrière-plan	Valeur alpha et mode de fondu préférés pour la fonction de transparence de l'arrière-plan.
Bouclage	Comportement du bouclage : le contenu se répète indéfiniment ou s'arrête après la dernière image.
Qualité par défaut	Niveau de qualité avec lequel le contenu doit être lu.

Les types de contenus suivants permettent de segmenter le contenu des périphériques et de contrôler certains aspects relatifs à la gestion du contenu.

Type de contenu	Description
Economiseur d'écran	Animation Flash qui s'exécute lorsqu'un délai spécifié par un périphérique arrive à expiration.
Sonnerie	Animation lue lorsque le téléphone reçoit un appel entrant.
Arrière-plan	Papier-peint animé qui s'affiche derrière l'écran d'inaction/d'accueil.
Jeu	Jeu interactif.
Application	Application autonome.
Présentation	Présentation convertie en clip Flash.
Icône	Icône Flash animée.
Clip	Clip Flash animé.
Habillage	Habillage ou thème du téléphone ou application unique.
Démarrage	Animation de démarrage.
Fermeture	Animation de fermeture.
Autre	Contenu Flash qui n'entre dans aucune des catégories ci-dessus.

## Fonctionnalités de Flash Lite 2.x

### A propos des fonctionnalités de Flash Lite 2.x

Les versions 2.0 et 2.1 de Flash Lite (dénommées collectivement 2.x) s'appuient sur Flash Player 7 et prennent en charge la plupart des fonctionnalités disponibles dans cette version de Flash Player, notamment le traitement XML et ActionScript 2.0. Flash Lite 2.x dispose également de fonctionnalités conçues spécifiquement pour les applications mobiles et non disponibles dans Flash Player 7.

## Nouvelles fonctionnalités de Flash Lite 2,1

Flash Lite 2.1 prend désormais en charge la saisie de texte en ligne. Dans les versions précédentes de Flash Lite, les champs texte de saisie utilisaient la boîte de dialogue de saisie de texte du périphérique. Dans Flash Lite 2.1, les champs texte de saisie peuvent être modifiés directement.

**Remarque :** Les périphériques qui prennent en charge des langues complexes (par exemple, l'hébreu, l'arabe, le farsi et certaines langues asiatiques) ne prennent pas en charge la saisie de texte en ligne. Le fonctionnement des champs de saisie de ces périphériques est identique sous Flash Lite 2.1 et Flash Lite 2.0.

Flash Lite 2.1 autorise les communications par socket à l'aide de la classe XMLSocket sur les périphériques qui la prennent en charge. (Pour plus d'informations, consultez le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.0.*) Pour plus d'informations sur les champs texte de saisie, consultez la section « [Utilisation du texte et des polices](#) » à la page 53.

Flash Lite 2.1 fournit également des outils pour développer des applications Flash Lite qui s'exécutent sur des périphériques à l'aide de la plate-forme BREW® (Binary Runtime Environment for Wireless®) du constructeur QUALCOMM Incorporated. Pour plus d'informations sur ces outils, consultez la section « [Développement d'applications Flash Lite pour BREW](#) » à la page 95.

## Nouvelles fonctionnalités de Flash Lite 2,0

Flash Lite 2.0 inclut les nouvelles fonctionnalités suivantes :

- Prise en charge d'ActionScript 2.0, ce qui permet d'appliquer des techniques de programmation avancées, incluant les classes, les interfaces et le typage strict des données ;
- Lecture de vidéos de périphérique
- Stockage de données rémanent et local (objets partagés de Flash Lite) ;
- Prise en charge du chargement des formats d'image et audio propres aux périphériques ;
- Informations sur les caractéristiques du nouveau système ;
- Prise en charge de touches de périphérique supplémentaires, ce qui inclut la prise en charge du clavier QWERTY et jusqu'à 11 touches programmables ;
- Formatage en texte enrichi (prise en charge partielle) ;
- Possibilité de contrôler la durée du rétroéclairage et de définir la couleur du rectangle de focus ;
- Synchronisation des sons de périphérique
- Prise en charge du traitement XML.

Les fonctionnalités suivantes de Flash Player 7 ne sont pas disponibles dans Flash Lite 2.0 :

- Plusieurs classes ActionScript disponibles dans Flash Player 7 ne sont pas prises en charge par Flash Lite 2.0 ou prises en charge uniquement de façon partielle. Pour plus d'informations sur le code ActionScript disponible, consultez le guide *Présentation du code ActionScript Flash Lite 2.0.*
- Communication par sockets à l'aide de la classe XMLSocket (disponible dans Flash Lite 2.1) ;
- Prise en charge des communications avec Flash Media Server (disponible dans Flash Lite 3.0) ;
- Objets partagés distants (les objets locaux partagés sont partiellement pris en charge) ;
- Prise en charge native de la lecture des fichiers Flash Video (FLV) (disponible dans Flash Lite 3.0) ;
- Prise en charge du protocole d'application de Flash (le protocole de communication binaire utilisé par Flash Remoting) ;



- Mise en forme avec feuilles de style CSS et à l'aide de champs texte ;
- Masquage avec les polices de périphérique ;
- Lissage des objets bitmap lors des opérations de rendu haute qualité.

## Code ActionScript de Flash Lite 2.0

Flash Lite 2.0 permet d'exploiter ActionScript, un langage de création de scripts pour les applications Flash Lite 2.0 et Flash Lite 2.1. Ce langage partage la plupart du code ActionScript utilisé par Flash Player 7. Flash Lite 2.0 dispose d'ajouts et d'extensions qui permettent, entre autres, d'obtenir des informations sur le périphérique, d'effectuer des appels téléphoniques ou de contrôler la durée du rétroéclairage.

Vous pouvez appliquer la syntaxe ActionScript 2.0 ou ActionScript 1.0 lorsque vous développez des applications pour Flash Lite 2.0. ActionScript 2.0 prend en charge la programmation des classes, des interfaces et le typage strict des données. La syntaxe ActionScript 2.0 permet au compilateur ActionScript de fournir des informations de débogage plus précises et facilite la conception de programmes.

Pour plus d'informations sur l'apprentissage du code ActionScript dans Flash Lite 2.0, consultez les documents et les sections ci-dessous :

- *Présentation du code ActionScript Flash Lite 2.x et 3.0*
- *Guide de référence du langage ActionScript Flash Lite 2.x et 3.0*
- Section « Apprentissage d'ActionScript 2.0 » du guide *Utilisation de Flash*.

## Lecture de vidéos de périphérique

Flash Lite 2.0 permet de lire des vidéos dans tous les formats pris en charge de façon native par le périphérique cible. Par exemple, certains périphériques enregistrent et lisent les vidéos au format 3GP. D'autres prennent en charge les formats AVI ou MPEG. Au cours de la lecture, Flash Lite transmet les données vidéo d'origine au périphérique, qui décode et rend ces données directement à l'écran. Vous pouvez incorporer des vidéos dans votre application de l'une des façons suivantes :

- Regrouper les données vidéo source dans le fichier SWF ;
- Charger un fichier vidéo externe à partir de la carte mémoire du périphérique ou sur le réseau.

L'objet Video d'ActionScript permet de contrôler la lecture des vidéos dans Flash Lite 2.0. Disponible depuis Flash Player 6, l'objet Video de Flash Lite 2.0 dispose de méthodes supplémentaires de contrôle de la vidéo, telles que les méthodes `Video.play()` et `Video.pause()`. Vous pouvez également consulter le tableau `System.capabilities.videoMIMETypes` pour déterminer les formats vidéo pris en charge par le périphérique.

Consultez « [Utilisation de la vidéo](#) » à la page 79 pour obtenir de plus amples informations sur l'objet Vidéo et l'utilisation de vidéos dans Flash Lite.

## Chargement des formats d'image et audio propres aux périphériques

Flash Lite 2.0 permet de charger les fichiers image ou audio dont le format est pris en charge par le périphérique. Pour charger des images externes, appliquez la fonction globale `loadMovie()` ou la méthode `MovieClip.loadMovie()`. Par exemple, si le périphérique prend en charge le format PNG, le code suivant permet de charger un fichier PNG provenant d'un serveur Web dans l'occurrence de clip, `image_mc` :

```
image_mc.loadMovie("http://www.adobe.com/images/mobile.png");
```

La méthode `Sound.loadSound()` permet de charger des sons externes. Dans Flash Lite 2.0, cette méthode permet de charger tout son dont le format est pris en charge par le périphérique (par exemple, MIDI ou SMAF). Les sons de périphérique externes doivent être intégralement chargés en mémoire avant d'être lus.

Comme dans Flash Lite 1.x, Flash Lite 2.0 permet de lire les sons de périphérique regroupés dans un fichier SWF. Pour plus d'informations, consultez la section « [Utilisation des regroupements de sons de périphérique](#) » à la page 68.

Consultez la rubrique suivante pour obtenir de plus amples informations sur le chargement d'images et de sons externes :

### Voir aussi

« [Lecture de sons de périphérique externes](#) » à la page 74

## Objets partagés de Flash Lite

Les objets partagés de Flash Lite permettent d'enregistrer des données de façon permanente sur le périphérique de l'utilisateur. Par exemple, utilisez un objet partagé pour échanger des informations entre plusieurs sessions d'une application, telles que les préférences utilisateur et les scores de jeu. La classe `SharedObject` permet de lire et d'écrire des objets Flash Lite partagés. Pour plus d'informations sur l'utilisation d'objets partagés de Flash Lite, consultez la classe `SharedObject` dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.0*.

**Remarque :** L'implémentation des objets partagés par Flash Lite 2.0 n'autorise pas le partage de données entre plusieurs fichiers SWF. En outre, aucune des versions de Flash Lite ne prend en charge les objets distants partagés avec Flash Media Server.

## Synchronisation des sons de périphérique

Dans les versions précédentes de Flash Lite, vous ne pouviez synchroniser que les sons Flash natifs avec le scénario de l'animation. Il était alors impossible de synchroniser les sons de périphérique, qui sont lus directement par le périphérique, et non pas de façon native par Flash Lite. Dans Flash Lite 2.0, vous pouvez synchroniser les sons de périphérique avec le scénario, à l'aide de la nouvelle propriété `_forceframerate`. Lorsque cette propriété est définie sur `true`, Flash Lite omet des images de l'animation pour préserver la cadence d'images du fichier SWF, si nécessaire. Pour plus d'informations, consultez la section « [A propos de la synchronisation des sons de périphérique avec l'animation](#) » à la page 74 et la rubrique consacrée à la propriété `forceframerate` dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.0*.

## Fonctionnalités texte

Les fonctionnalités suivantes portent sur la gestion du texte et sont nouvelles dans Flash Lite 2.0 :

- Le texte est systématiquement enregistré au format Unicode dans Flash Lite 2.0.
- Flash Lite 2.0 prend partiellement en charge la mise en forme du code HTML et la classe `TextFormat` d'ActionScript.

Pour plus d'informations sur l'utilisation des champs texte dans Flash Lite 2.x, consultez la section « [Utilisation du texte et des polices](#) » à la page 53.

## Renforcement de la prise en charge de touches

La prise en charge des touches de périphérique a été développée sous Flash Lite 2.0, ce qui débouche sur la prise en charge des claviers QWERTY et jusqu'à 12 touches programmables (ce qui inclut les touches programmables standard, Gauche et Droite).

# Chapitre 2 : Prise en main de Flash Lite

## Application Flash Lite Hello World

Ce didacticiel simple permet de vous présenter les fonctionnalités de test et de programmation des périphériques portables dans Adobe Flash CS4 Professional ainsi que le processus général de création d'un contenu par le biais d'Adobe Flash Lite. Dans cette section, vous allez créer une application Flash Lite simple pour la tester dans l'émulateur Adobe Device Central CS4. Pour obtenir un exemple d'application plus détaillé, consultez le « [Didacticiel : Création d'une application Flash Lite](#) » à la page 19.

Ce didacticiel porte sur le développement de contenu pour le lecteur autonome Flash Lite. La procédure porte sur un périphérique générique.

Vous devez d'abord choisir les périphériques et le type de contenu Flash Lite que vous ciblez.

### Configuration et création d'une application Flash Lite simple

- 1 Démarrez Flash.
- 2 Dans l'écran Flash principal, sélectionnez Créer > Fichier Flash (mobile). Flash ouvre Adobe Device Central et affiche l'onglet Nouveau document.
- 3 Sous l'onglet Nouveau document, sélectionnez successivement Flash Lite 2.0 dans la zone Version du lecteur, Adobe ActionScript™ 2.0 dans la zone Version ActionScript et Lecteur autonome dans la zone Type de contenu.
- 4 Cliquez sur Taille personnalisée pour tous les périphériques sélectionnés, en bas de l'écran. Ceci vous permet de créer du contenu pour le lecteur Flash Lite autonome.
- 5 Cliquez sur Créer. Vous revenez à Flash qui lance la création d'un document en utilisant les paramètres de publication prédéfinis et (lorsque vous spécifiez un périphérique) la taille adéquate pour le périphérique sélectionné.
- 6 Dans le panneau Outils, sélectionnez l'outil Texte et faites-le glisser sur la scène pour créer une zone de texte. Tapez **Hello, world !** (ou tout autre texte) dans la zone de texte.
- 7 Sélectionnez Contrôle > Tester l'animation pour exporter votre application vers Adobe Device Central et l'afficher dans son émulateur.

***Remarque :** Au cours de la phase de test effectué dans Device Central, il est possible de modifier le périphérique et le type de contenu pour voir l'application sur une autre plate-forme. Pour cela, double-cliquez sur un périphérique dans le panneau de bibliothèque et sélectionnez un nouveau type de contenu dans la zone Type de contenu. Lorsque vous revenez à Flash, les paramètres que vous avez utilisés la dernière fois dans l'émulateur sont conservés.*

- 8 Pour revenir à Flash, sélectionnez Fichier > Retourner à Flash.

## Emulateur Adobe Device Central et base de données de périphériques

L'émulateur Adobe Device Central permet de tester votre contenu tel qu'il s'exécutera et s'affichera sur un périphérique réel. Vous pouvez sélectionner un type de contenu ou un périphérique de test différent, puis afficher les informations détaillées relatives à votre application.

Device Central contient une base de données étendue de périphériques portables à partir desquels effectuer votre sélection pour créer votre propre environnement de test. Consultez la documentation de Device Central pour toute information complémentaire.

## Flux de travail lors de la création d'applications Flash Lite

La création de contenu Flash Lite est une procédure itérative impliquant les étapes suivantes :

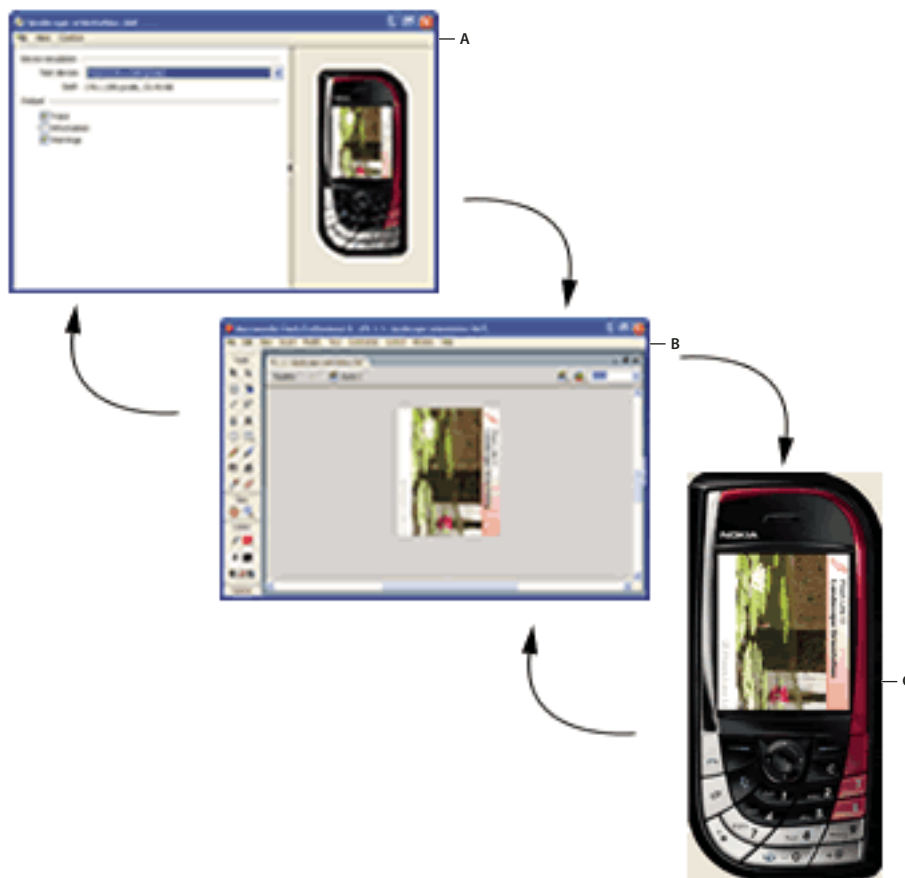
**Identification des périphériques et du type de contenu Flash Lite visés** De nombreux périphériques présentent des caractéristiques différentes, par exemple en termes de taille d'écran, de prise en charge des formats audio et de profondeur de couleur d'écran. Ces facteurs peuvent influencer sur la conception ou l'implémentation de votre application.

En outre, de nombreux périphériques prennent en charge différents types de contenus Flash Lite, tels que les économiseurs d'écran, les applications autonomes ou les sonneries animées. Le type de contenu pour lequel vous programmez détermine également les fonctionnalités disponibles pour votre application. Pour plus d'informations, consultez la section « [A propos des types de contenus Flash Lite](#) » à la page 17.

**Création de l'application dans Flash et test dans Adobe Device Central.** Adobe Flash Professional comprend un émulateur dans Adobe Device Central CS4 qui vous permet de tester votre application sans la transférer à un périphérique. L'émulateur permet de parfaire la conception de votre application et de résoudre d'éventuels problèmes avant de la tester sur un périphérique portable réel.

**Test de l'application sur les périphériques cibles** Cette opération est importante car l'émulateur Adobe Device Central ne simule pas tous les aspects du périphérique cible, tels que la vitesse de son processeur, sa profondeur d'échantillonnage ou sa latence réseau. Par exemple, une animation qui s'exécute normalement sur l'émulateur risque d'être plus lente sur le périphérique du fait d'un processeur plus lent, ou un dégradé de couleurs peut paraître régulier sur l'émulateur et présenter des bandes sur le périphérique. Après avoir testé votre application sur un périphérique, si nécessaire, vous avez toujours la possibilité d'en affiner la conception dans l'outil de programmation Flash.

La capture d'écran ci-dessous illustre le développement itératif et la procédure de test décrits ci-dessus :



A. Modification du fichier FLA dans Adobe Flash B. Test dans l'émulateur Adobe Device Central C. Test sur un périphérique

## A propos des types de contenus Flash Lite

Avant de commencer à développer une application Flash Lite, vous devez connaître les points suivants :

- Le ou les périphériques sur lesquels le contenu sera exécuté (*périphériques cibles*). Le lecteur Flash Lite est installé sur de nombreux périphériques. Pour obtenir la liste des périphériques disposant de Flash Lite, consultez la page affichant les périphériques pris en charge sur le site Web d'Adobe à l'adresse [www.adobe.com/go/mobile\\_supported\\_devices/](http://www.adobe.com/go/mobile_supported_devices/).
- Les types de contenus Flash Lite pris en charge par les périphériques cibles. Toutes les installations Flash Lite prennent en charge un ou plusieurs modes d'application (*types de contenus*). Par exemple, certains périphériques ont recours à Flash Lite pour activer les fonctionnalités d'économiseurs d'écran et de sonneries animées Flash. D'autres périphériques exploitent Flash Lite pour afficher du contenu Flash intégré dans des pages Web mobiles. Certains types de contenus ne prennent pas en charge toutes les fonctionnalités Flash Lite.

Tous les types de contenus Flash Lite, associés à un périphérique particulier, définissent un ensemble de fonctionnalités Flash Lite spécifiques disponibles pour votre application. Par exemple, une application Flash fonctionnant comme un économiseur d'écran n'est généralement pas autorisée à établir des connexions réseau ou à télécharger des données.

Les fonctionnalités de test de Flash Lite exploitées dans Adobe Device Central permettent de tester différents périphériques et types de contenus Flash Lite. Vous pouvez ainsi déterminer si votre application fait appel à des fonctionnalités non disponibles pour le type de contenu que vous développez. Pour plus d'informations sur le choix des périphériques cibles et des types de contenus, consultez la section « [Test du contenu Flash Lite](#) » à la page 117 dans le manuel *Développement d'applications Flash Lite 2.x et 3.x*.

## Chapitre 3 : Didacticiel : Création d'une application Flash Lite

Le présent didacticiel porte sur le développement d'une application Adobe Flash Lite de promotion pour un restaurant fictif, le Café Townsend. Les utilisateurs peuvent afficher une liste des spécialités servies, regarder un clip sur le chef et appeler le restaurant pour réserver une table. Ils peuvent également demander leur table préférée lors de la réservation, grâce à un objet partagé qui stocke leurs préférences entre les différentes sessions de l'application.

### Affichage de l'application terminée

Il existe deux versions de l'application terminée : une version qui utilise un fichier vidéo de périphérique 3GP et une version qui utilise un fichier FLV. Vous pouvez afficher chaque version dans l'émulateur Adobe Device Central. Vous pouvez aussi transférer les applications terminées sur le lecteur Flash Lite installé sur un périphérique mobile.

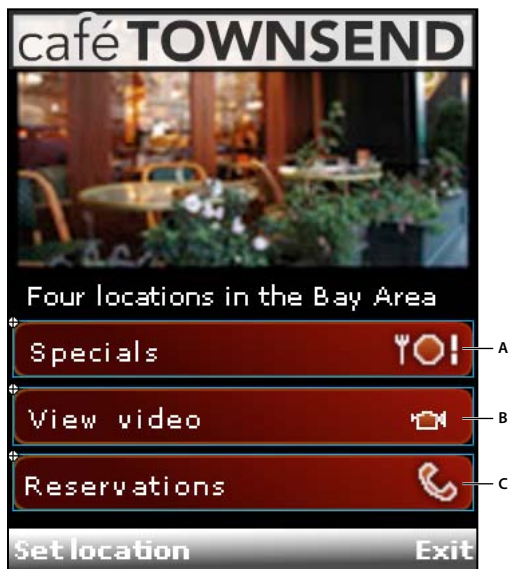
Device Central ne prend pas en charge toutes les commandes d'Adobe ActionScript. Par exemple, Device Central ne prend pas en charge `fscommand2("Quit")` ou `getURL()`. Device Central envoie des messages à la fenêtre de sortie pour vérifier que les commandes ont été appelées. Pour visualiser les messages, sélectionnez Affichage > Sortie Flash (Device Central CS4) ou Affichage > Sortie Flash > Afficher (Device Central CS3). Pour de plus amples informations, consultez « [Fonctionnalités Flash Lite non prises en charge par l'émulateur](#) » à la page 123.

- 1 Téléchargez puis décompressez les fichiers du tutoriel. Consultez la page Web à l'adresse [http://www.adobe.com/go/learn\\_flt\\_samples\\_and\\_tutorials\\_fr](http://www.adobe.com/go/learn_flt_samples_and_tutorials_fr). Téléchargez le fichier ZIP pour votre version de Flash Lite.
- 2 Dans Flash, sélectionnez Fichier > Ouvrir et naviguez vers le fichier appelé `cafe_tutorial_complete fla` ou `cafe_tutorial_FLV_complete fa` dans le dossier Tutorial.
- 3 Sélectionnez Contrôle > Tester l'animation pour vérifier l'application dans Device Central.
- 4 Pour interagir avec l'application, procédez comme suit :
  - Dans l'écran d'accueil, cliquez sur la touche Bas pour sélectionner l'élément de menu Specials. Cliquez sur la touche de sélection pour afficher l'écran Specials.
  - Dans l'écran Specials, cliquez sur la touche programmable Droite (Next) de l'émulateur pour afficher l'image et la description de chaque plat. Cliquez sur la touche programmable Gauche (intitulée Home) pour retourner dans l'écran d'accueil.
  - Sélectionnez l'élément de menu View Video pour regarder le clip. Cliquez sur la touche programmable Gauche (intitulée Home) pour retourner dans l'écran d'accueil. Vous pouvez également cliquer sur la touche droite (Pause) pour pauser le clip.
  - Dans l'écran d'accueil, cliquez sur la touche programmable Gauche (Set Location) pour accéder à l'écran de sélection du restaurant. Sélectionnez le restaurant à contacter de préférence lors des réservations, puis cliquez sur la touche programmable Gauche (Save) pour enregistrer cet emplacement et revenir à l'écran d'accueil (ou cliquez sur Annuler).
  - Dans l'écran d'accueil, sélectionnez l'élément de menu Reservations pour passer un appel téléphonique au restaurant. Dans la fenêtre de sortie, confirmez que le numéro de téléphone correspond à l'emplacement que vous avez indiqué sur l'écran Set Location.

# Création de l'application

## Création du menu de l'écran d'accueil

- 1 Dans Flash, sélectionnez Fichier > Ouvrir et accédez au fichier `cafe_tutorial_start fla` file dans le dossier Tutorials.
- 2 Sélectionnez Fichier > Enregistrer sous et enregistrez le fichier sous le nom `cafe_tutorial fla`.
- 3 Dans la fenêtre Scénario (Fenêtre > Scénario), sélectionnez l'image 1 dans le calque du menu principal.
- 4 Ouvrez le panneau Bibliothèque (Fenêtre > Bibliothèque) et faites glisser une occurrence du bouton **Specials** vers la scène. Procédez comme suit :
  - a Placez le bouton **Specials** au-dessous du champ texte présentant le restaurant.
  - b Assurez-vous que le bouton **Specials** est sélectionné puis, dans l'inspecteur des propriétés, tapez **specials\_btn** dans la zone de texte Nom de l'occurrence.
- 5 Faites glisser une occurrence du bouton **Video** du panneau Bibliothèque vers la scène. Procédez comme suit :
  - Placez le bouton **Video** au-dessous du bouton **Specials**.
  - Assurez-vous que le bouton **Video** est sélectionné dans l'inspecteur des propriétés, puis tapez **video\_btn** dans la zone de texte Nom de l'occurrence.
- 6 Faites glisser une occurrence du bouton **Reservation** du panneau Bibliothèque vers la scène. Procédez comme suit :
  - Placez le bouton **Reservation** au-dessous du bouton **Video**.
  - Assurez-vous que le bouton **Reservation** est sélectionné puis, dans l'inspecteur des propriétés, tapez **reservation\_btn** dans la zone de texte Nom de l'occurrence.
- 7 La scène devrait alors présenter l'aspect suivant :



A. specials\_btn B. video\_btn C. reservations\_btn

- 8 Dans le scénario, sélectionnez l'image 1 dans le calque intitulé ActionScript. Ouvrez le panneau Actions (Fenêtre > Actions), puis tapez le code indiqué ci-dessous :



```
stop();
_focusrect = false;
fscommand2("SetSoftKeys", "Set Location", "Exit");
fscommand2("Fullscreen", "true");
```

Ce code produit les résultats suivants :

- Arrête la tête de lecture sur cette image.
- Désactive l'affichage du rectangle de focus jaune dessiné par défaut par Flash Lite autour du bouton ou du champ texte de saisie ayant le focus (consultez la section « [A propos du rectangle de focus](#) » à la page 39).
- Enregistre les touches programmables destinées à l'application.
- Affiche l'application en mode plein écran.

Lorsque Flash Lite est en mode plein écran, les étiquettes spécifiées dans la commande `SetSoftKeys` ne sont pas visibles. Par conséquent, vous devez ajouter des étiquettes personnalisées de touche programmable sur la scène.

- 9 Ajoutez le code suivant dans l'image 1 pour traiter les événements renvoyés par les boutons de menu et pour le focus de sélection :

```
// Set initial focus when the application
// starts and also upon returning to the main
// screen from another screen.
if (selectedItem == null) {
    Selection.setFocus (specials_btn);
} else {
    Selection.setFocus (selectedItem);
}
// Assign onPress event handlers to each menu button,
// and set selectedItem variable to selected button
// object:
specials_btn.onPress = function () {
    gotoAndStop ("specials");
    selectedItem = this;
};
video_btn.onPress = function () {
    gotoAndStop ("video");
    selectedItem = this;
};
reservations_btn.onPress = function () {
    if (location_so.data.phoneNumber == undefined) {
        // User hasn't specified location so
        // go to "set location" screen:
        gotoAndStop ("locations");
    }
    else {
        // Call number in shared object:
        var phoneNum = location_so.data.phoneNumber;
        getURL ("tel:" + phoneNum);
    }
    selectedItem = this;
};
```

Les gestionnaires d'événements `onPress` attribués aux boutons `specials_btn` et `video_btn` envoient la tête de lecture aux images étiquetées respectivement « specials » et « video ».

Lorsque l'utilisateur sélectionne l'option Reservations, le gestionnaire `onPress` compose le numéro de téléphone spécifié dans l'objet partagé `location_so`. (Plus loin dans cette procédure, vous allez écrire le code de création de l'objet partagé.) Si l'utilisateur n'a spécifié aucun emplacement à appeler pour les réservations, l'application place la tête de lecture sur l'image « locations », ce qui permet à l'utilisateur de spécifier ses préférences.

**10** Ajoutez le code suivant dans l'image 1 afin de créer un écouteur pour les touches programmables gauche et droite :

```
Key.removeListener(myListener);
var myListener:Object = new Object();
myListener.onKeyDown = function() {
    var keyCode = Key.getCode();
    if (keyCode == ExtendedKey.SOFT1) {
        // Handle left soft key event:
        gotoAndStop("locations");
    } else if (keyCode == ExtendedKey.SOFT2) {
        // Handle right soft key event:
        fscommand2("Quit");
    }
};
Key.addListener(myListener);
```

Ce code utilise un objet écouteur de touche pour gérer les événements des touches programmables gauche et droite. Lorsque l'utilisateur appuie sur la touche programmable Gauche, la tête de lecture est envoyée à l'image « locations ». La touche programmable Droite ferme l'application.

Pour plus d'informations sur l'utilisation des écouteurs d'événements, consultez la section « [Gestion des événements de pression de touche à l'aide d'un écouteur de touche](#) » à la page 49.

**11** Ajoutez le code pour initialiser l'objet partagé qui enregistre le restaurant préféré de l'utilisateur pour les réservations :

```
// Define Shared Object listener function:
function so_listener (the_so:SharedObject) {
    if (the_so.getSize () == 0) {
        // The shared object doesn't exist, so the user
        // hasn't set a preference yet.
    }
    SharedObject.removeListener ("location");
}
// Create shared object:
location_so = SharedObject.getLocal ("location");
// Add SharedObject listener object:
SharedObject.addListener ("location", this, "so_listener");
```

**12** Sélectionnez Fichier > Enregistrer.

**13** Pour tester votre travail dans Device Central, sélectionnez Contrôle > Tester l'animation.

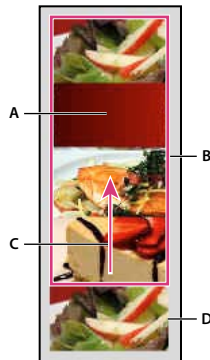
A ce stade, vous pouvez sélectionner un élément du menu en donnant le focus au bouton correspondant, puis en appuyant sur la touche de sélection de l'émulateur (ou sur la touche Entrée du clavier de l'ordinateur). Ouvrez la fenêtre Sortie dans Device Central pour afficher les informations concernant les commandes non prises en charge.

## Création de l'écran Specials

Créez l'animation qui assure la transition entre les images de chaque plat. Ajoutez des éléments d'interface utilisateur et du code `ActionScript` pour permettre à l'utilisateur de naviguer entre les images. Ajoutez du code `ActionScript` pour afficher le nom et la description de chaque plat sous l'image.

## Création de l'animation d'images

Pour créer l'animation, utilisez un clip prédéfini contenant les images de tous les plats, regroupées dans une colonne verticale. Utilisez un calque de masque pour n'afficher qu'une seule image. Une animation interpolée déplace les images dans le masque. La dernière image du clip est une duplication de la première image, ce qui permet à la séquence d'animation de reprendre son état initial une fois que l'utilisateur a consulté la dernière image.

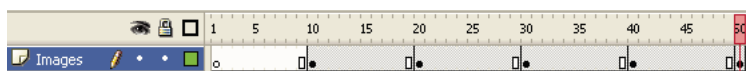


A. Objet calque de masque B. Clip d'images masqué C. Sens de l'interpolation D. Image dupliquée

- 1 Ouvrez le fichier `cafe_tutorial fla` que vous avez terminé dans la section « [Création du menu de l'écran d'accueil](#) » à la page 20.
- 2 Dans le scénario, sélectionnez l'image-clé de l'image 10 dans le calque intitulé `Specials photos`.
- 3 Ouvrez le panneau Bibliothèque et faites glisser le symbole du clip `Specials` sur la scène.
- 4 Une fois le clip `Specials` sélectionnée, définissez les coordonnées  $x$  et  $y$  du clip sur **0** dans l'inspecteur des propriétés.

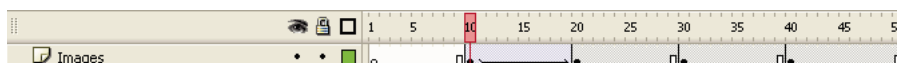
Cette action aligne le coin supérieur gauche du clip `Specials` sur le coin supérieur gauche de la scène.

- 5 Dans le masque de photos `Specials`, sélectionnez les images 20, 30, 40 et 50 puis appuyez sur F6 pour insérer les image-clés. Sélectionnez l'image 51 puis appuyez sur F7 pour insérer une image-clé vide. Le scénario doit alors être identique à l'image suivante :



- 6 Sélectionnez l'image-clé de l'image 20, puis le clip `Specials`, et définissez sa coordonnée  $y$  sur **-100** dans l'inspecteur des propriétés.
- 7 Sélectionnez l'image-clé de l'image 30 dans le scénario, puis le clip `Specials`, et définissez sa coordonnée  $y$  sur **-200** dans l'inspecteur des propriétés.
- 8 Sélectionnez l'image-clé de l'image 40, puis le clip `Specials`, et définissez sa coordonnée  $y$  sur **-300** dans l'inspecteur des propriétés.
- 9 Sélectionnez l'image-clé de l'image 50, puis le clip `Specials`, et définissez sa coordonnée  $y$  sur **-400** dans l'inspecteur des propriétés.
- 10 Sélectionnez l'image 10 en tant qu'image-clé, puis sélectionnez l'option `Mouvement` dans le menu `Interpolation` de l'inspecteur `Propriétés`.

Cette action interpole la position du clip d'images entre les images-clés des images 10 et 20.



- 11 Pour créer des transitions entre les autres images, répétez l'étape 11 pour les images-clés 20, 30 et 40.
- 12 Pour créer le calque de masque, sélectionnez le calque de photos **Specials** dans le scénario, puis Insérer > Scénario > Calque (ou cliquez sur Insérer un calque dans le scénario).
- 13 Cliquez deux fois sur le nom du nouveau masque et renommez-le **Mask**.
- 14 Sélectionnez l'image 10 du masque **Mask** et sélectionnez Insérer > Scénario > Image-clé.
- 15 Sélectionnez l'outil Rectangle dans le panneau Outils et créer un rectangle au-dessus de la première image (la plus au-dessus) dans le clip Images.

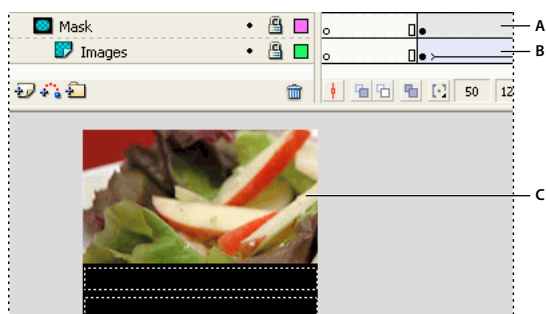
Peu importe la couleur de remplissage que vous utilisez pour le rectangle, mais il doit être totalement opaque.



Rectangle de masquage

- 16 Pour vous assurer que le rectangle recouvre toute la zone de l'image, double-cliquez dessus pour le sélectionner, puis utilisez l'inspecteur des propriétés pour définir ses coordonnées  $x$  et  $y$  sur **0**, sa largeur sur **176** et sa hauteur sur **100**.
- 17 Cliquez avec le bouton droit de la souris (Windows®) ou en appuyant sur la touche Contrôle (Macintosh®) sur le calque **Specials** dans le scénario, puis sélectionnez **Masque** dans le menu contextuel.

Le calque est converti en calque de masque, signalé par l'icône correspondante. Le calque situé immédiatement au-dessous est lié au calque de masque et son contenu affiché à travers la zone remplie sur le masque. Pour plus d'informations sur l'utilisation des calques de masque dans Flash, consultez les sections correspondantes dans le guide d'utilisation de Flash.



A. Calque de masque B. Calque masqué C. Zone masquée sur la scène

- 18 Enregistrez vos modifications (Fichier > Enregistrer).

Si vous testez l'application maintenant, l'animation des plats ne fonctionne pas. Dans « [Ajout d'éléments de navigation et de texte dans l'écran Specials](#) » à la page 25, vous ajoutez du code ActionScript qui arrête l'animation à chaque image-clé. Vous ajoutez aussi des éléments d'interface d'utilisateur qui permet à l'utilisateur de naviguer entre les images.

### Ajout d'éléments de navigation et de texte dans l'écran Specials

- 1 Ouvrez le fichier cafe\_tutorial fla dans Flash.
- 2 Dans le scénario, sélectionnez l'image 10 dans le calque Text.
- 3 Dans la palette Outils, sélectionnez l'outil Texte et tracez un champ texte sous la première image masquée de plats.  
Ce champ texte affiche le nom du plat illustré à l'écran.



Champ texte devant afficher le nom du plat

- 4 En maintenant le champ texte sélectionné sur la scène, apportez les modifications suivantes dans l'inspecteur Propriétés :
  - Sélectionnez Texte dynamique dans le menu déroulant Type de texte.
  - Saisissez **title\_txt** dans la zone de texte Nom de l'occurrence.
  - Sélectionnez l'option de style de texte Italique.
  - Définissez la taille de la police sur 10.
  - Définissez la couleur de la police sur blanche.
  - Sélectionnez Utiliser les polices de périphérique dans le menu déroulant Méthode de rendu des polices.
- 5 Créez un autre champ texte sous le premier pour afficher une courte description du plat consulté par l'utilisateur.
- 6 A l'aide de l'outil Sélection, redimensionnez le nouveau champ texte afin que sa taille soit environ trois fois plus grande que celle du champ texte situé au-dessus.



Champ texte devant afficher la description du plat

7 En maintenant le champ de texte sélectionné sur la scène, faites comme suit dans l'inspecteur Propriétés :

- Sélectionnez Texte dynamique dans le menu déroulant Type de texte.
- Tapez **description\_txt** dans la zone de texte Nom de l'occurrence.
- Sélectionnez Multiligne dans le menu contextuel Type de ligne.
- Définissez la taille de la police sur 10.
- Assurez-vous qu'Italique n'est pas sélectionné.
- Définissez la couleur de la police sur blanche.
- Sélectionnez Utiliser les polices de périphérique dans le menu déroulant Méthode de rendu des polices.

8 Dans le scénario, sélectionnez l'image 10 du calque ActionScript en tant qu'image-clé.

9 Ouvrez le panneau Actions et ajoutez le code suivant :

```
stop();
fscommand2("SetSoftKeys", "Home", "Next");
title_txt.text = "Summer Salad";
description_txt.text = "Butter lettuce with apples, blood orange segments, gorgonzola, and
raspberry vinaigrette.";
```

Ce code donne la description du nom du premier plat du jour dans les deux champs texte dynamique. Il arrête également la tête de lecture sur l'image actuelle et enregistre les touches programmables.

10 Dans le calque ActionScript, sélectionnez l'image 20 en tant qu'image-clé et saisissez le code suivant dans le panneau Actions :

```
stop();
title_txt.text = "Chinese Noodle Salad";
description_txt.text = "Rice noodles with garlic sauce, shitake mushrooms, scallions, and
bok choy.";
```

11 Dans le calque ActionScript, sélectionnez l'image 30 en tant qu'image-clé et saisissez le code suivant dans le panneau Actions :

```
stop();
title_txt.text = "Seared Salmon";
description_txt.text = "Filet of wild salmon with caramelized onions, new potatoes, and
caper and tomato salsa.";
```

- 12** Dans le calque ActionScript, sélectionnez l'image 40 en tant qu'image-clé et saisissez le code suivant dans le panneau Actions :

```
stop();
title_txt.text = "New York Cheesecake";
description_txt.text = "Creamy traditional cheesecake served with chocolate sauce and
strawberries.";
```

- 13** Dans le calque ActionScript, sélectionnez l'image 50 en tant qu'image-clé et saisissez le code suivant dans le panneau Actions :

```
gotoAndStop("specials");
```

Ce code renvoie la tête de lecture au début de la séquence d'animation. Les première et dernière images de la séquence d'animation sont identiques, ce qui crée l'illusion d'une animation continue.

- 14** Dans le scénario, sélectionnez l'image 10 dans le calque intitulé ActionScript. Ouvrez le panneau Actions et entrez le code suivant :

```
Key.removeListener (myListener);
var myListener:Object = new Object ();
myListener.onKeyDown = function () {
    var keyCode = Key.getCode ();
    if (keyCode == ExtendedKey.SOFT1) {
        // Handle left soft key event
        gotoAndPlay ("home");
    }
    else if (keyCode == ExtendedKey.SOFT2) {
        // Handle right soft key event
        play ();
        description_txt.text = "";
        title_txt.text = "";
    }
};
Key.addListener (myListener);
```

La touche programmable Gauche place la tête de lecture dans l'écran principal de l'application tandis que la touche programmable Droite permet de passer à l'image suivante de l'animation.

Pour plus d'informations sur l'utilisation des écouteurs d'événements, consultez la section « [Gestion des événements de pression de touche à l'aide d'un écouteur de touche](#) » à la page 49.

- 15** Sélectionnez Fichier > Enregistrer.

- 16** Sélectionnez Contrôle > Tester l'animation pour vérifier l'application dans Device Central.

Sélectionnez le bouton Specials. L'animation et le texte de Specials devraient fonctionner correctement.

## Création de l'écran Video

Dans cette section, vous allez ajouter de la vidéo à l'application. Vous allez ajouter également du code ActionScript qui permet à l'utilisateur de contrôler la lecture (lecture, pause) avec la touche droite.

Flash Lite prend en charge deux types de vidéo : la vidéo de périphérique et les fichiers FLV (Flash Video). Flash Lite 2.x prend en charge les vidéos de périphérique. Flash Lite 3.x prend en charge les vidéos de périphérique et les fichiers FLV. Pour finir ce tutoriel, vous devez choisir d'utiliser soit la vidéo de périphérique, soit un fichier FLV.

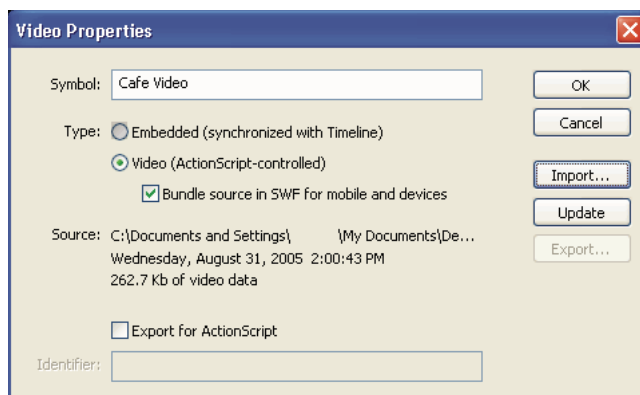
**Remarque :** Pour plus d'informations sur la prise en charge des vidéos, consultez la section « [Utilisation de la vidéo](#) » à la page 79.

### Créez l'écran vidéo avec la vidéo de périphérique (Flash Lite 2.x ou Flash Lite 3.x)

- 1 In Flash, open the cafe\_tutorial fla file you completed in « [Création de l'écran Specials](#) » à la page 22.
- 2 Open the Library options menu located in the upper-right corner of the Library panel and select New Video.
- 3 In the Video Properties dialog box, do the following:
  - a Enter a name for the video symbol in the Symbol box (**Cafe video**, for example).
  - b Select Bundle source in SWF for mobile and devices. Click Import.
  - c Browse to the cafe\_townsend\_chef.3gp file in the Tutorials folder. Select the file and click Open.

If you don't see the video file (or if you can see it but can't select it), select All Files (\*.\*) from the Files Of Type pop-up menu (Windows), or All Files from the Enable pop-up menu (Macintosh).

- 4 Make sure the Video Properties dialog box appears as follows, and click OK:



A new video symbol appears in the Library panel.

- 5 Select frame 51 in the Video layer of the Timeline. Drag the Cafe video symbol from the Library panel to the Stage.
- 6 In the Property inspector, enter **cafeVideo** in the Instance Name box. Set the x position to **0**, the y position to **30**, the Width to **176**, and the Height to **144**.





- 7 In the Timeline, select Frame 51 of the ActionScript layer. Enter, or copy and paste, the following code in to the Actions panel:

```
// Stop timeline, register soft keys, and start video.
stop ();
fscommand2 ("SetSoftKeys", "Home", "Pause");
cafeVideo.play ();
var playing:Boolean = true;
// Soft key event handler code:
Key.removeListener (myListener);
var myListener:Object = new Object ();
myListener.onKeyDown = function () {
    var keyCode = Key.getCode ();
    if (keyCode == ExtendedKey.SOFT1) {
        gotoAndPlay ("home");
    }
    else if (keyCode == ExtendedKey.SOFT2) {
        if (playing) {
            // If video is playing, pause it,
            // set status variable (playing) to false,
            // and set right soft key label to 'Play'.
            cafeVideo.pause ();
            playing = false;
            rightSoftKeyLabel.text = "Play";
        }
        else {
            // If video is paused, resume its playback,
            // set status variable (playing) to true,
            // and set right soft key label to 'Pause'.
            cafeVideo.resume ();
            playing = true;
            righttSoftKeyLabel.text = "Pause";
        }
    }
};
// Register listener object:
Key.addListener (myListener);
// Video status handler code.
// Device Central does not support the onStatus handler.
cafeVideo.onStatus = function (infoObject:Object) {
    var code = infoObject.code;
    if (code == "completed") {
        // If video has finished playing, set playing=false,
        // and set right soft key label to "Replay":
        playing = false;
        rightSoftKeyLabel.text = "Replay";
    }
};
```

- 8 Choose File > Save.

- 9 Choose Test > Control Movie to test the application in Device Central.

Select View Video on the homescreen to view the video. Press the Right soft key to pause the video. Press the Right soft key again to resume playback. On a device, you can press the Right soft key to play the video again after playback completes. However, this feature (the video.onStatus handler) is not supported in Device Central.

## Créez l'écran vidéo avec un fichier FLV (Flash Lite 3.x)

- 1 In Flash, open the `cafe_tutorial fla` file you completed in « [Création de l'écran Specials](#) » à la page 22.
- 2 Locate the `cafe_townsend_chef.flv` file in the Tutorials folder. Copy this file to the same folder as the `cafe_tutorial fla` file.
- 3 Open the Library options menu located in the upper-right corner of the Library panel and select New Video.
- 4 In the Video Properties dialog box, do the following:
  - a Enter a name for the video symbol in the Symbol box (**Cafe video**, for example).
  - b Select Video (ActionScript-Controlled). Click OK.
- 5 Select Frame 51 in the Video layer of the Timeline. Drag the Cafe video symbol from the Library panel to the Stage.
- 6 In the Property inspector, enter **cafeVideo** in the Instance Name box. Set the *x* position to **0**, the *y* position to **45**, the Width to **176**, and the Height to **144**.



- 7 In the Timeline, select Frame 51 of the ActionScript layer. Enter, or copy and paste, the following code in to the Actions panel:

```
// Frame actions
stop();
fscommand2 ("SetSoftKeys", "Home", "Pause");
var videoIsPlaying:Boolean;
// Use the NetConnection and NetStream classes to play FLV files
var nc:NetConnection = new NetConnection();
var ns:NetStream;
/* When a change in connection status occurs,
Flash Lite triggers the onStatus handler and passes it an
object with information about the change.
Use the information object to run code when
connection events occur.
*/
nc.onStatus = function(info) {
    // Run the following code if the connection is successful.
    if(info.code == "NetConnection.Connect.Success"){
        // Create a NetStream object and pass it the NetConnection object
        ns = new NetStream(nc);
        // Attach the NetStream object to the video object on Stage.
        cafeVideo.attachVideo(ns);
        // Play the file "cafe_townsend_chef.flv".
        ns.play("cafe_townsend_chef.flv");
        videoIsPlaying = true;
    }
}
// Create a connection.
```

```
// To connect to a local file or to a file on a web server, pass null.
nc.connect(null);
/* When a change in stream status occurs,
Flash Lite triggers the onStatus handler and passes it an
object with information about the change.
Use the information object to run code when
stream changes occur.
*/
ns.onStatus = function(info){
    if(info.code == "NetStream.Play.Stop"){
        videoIsPlaying = false;
        rightSoftKeyLabel.text = "Play";
    }
};
// Capture and handle soft key events.
Key.removeListener (myListener);
var myListener:Object = new Object ();
myListener.onKeyDown = function () {
    var keyCode = Key.getCode ();
    // Run this code if the user presses the left soft key.
    if (keyCode == ExtendedKey.SOFT1) {
        // Close the video.
        ns.close();
        // Go to the home screen.
        gotoAndPlay ("home");
    }
    // Run this code if the user presses the right soft key
    else if (keyCode == ExtendedKey.SOFT2) {
        // If the video is playing, run this code
        if (videoIsPlaying){
            // Pause or restart the video (this method does both).
            ns.pause();
            // Switch the label text.
            if (rightSoftKeyLabel.text == "Play"){
                rightSoftKeyLabel.text = "Pause";
            }
            else if (rightSoftKeyLabel.text == "Pause"){
                rightSoftKeyLabel.text = "Play";
            }
        }
        // If the video has finished playing, run this code.
    } else {
        // Replay the video.
        ns.play("cafe_townsend_chef.flv");
        rightSoftKeyLabel.text = "Pause";
        videoIsPlaying = true;
    }
}
};
Key.addListener (myListener);
```

8 Choose File > Save.

9 Choose Test > Control Movie to test the application in Device Central.

Select View Video on the homescreen to view the video. Press the Right soft key to pause the video. Press the Right soft key again to resume playback. When the video has finished playing, press the Right soft key to play it again.

## Création de l'écran Set Location

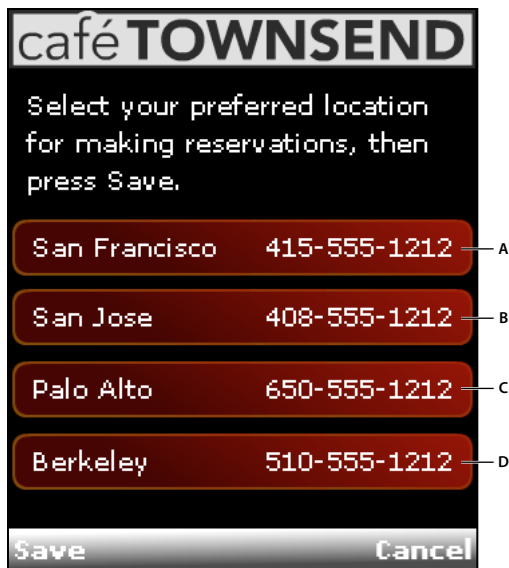
Dans cette section, créez un écran permettant à l'utilisateur de sélectionner l'un des quatre restaurants pour y effectuer des réservations.

**Remarque :** Pour afficher le numéro composé, sélectionnez *Affichage > Sortie Flash* dans *Device Central CS4* ou *Affichage > Sortie Flash > Afficher* dans *Device Central CS3*.

La première fois que l'utilisateur démarre l'application et sélectionne Reservations, l'application le conduit à l'écran Set location. Par la suite, lorsqu'il sélectionne Reservations, l'application compose le numéro du restaurant sélectionné. L'application a recours à un objet partagé pour enregistrer le restaurant sélectionné entre les sessions.

- 1 Dans Flash, ouvrez le fichier `cafe_tutorial.fla` complété dans la section « [Création de l'écran Video](#) » à la page 27.
- 2 Dans le scénario, sélectionnez l'image 66 en tant qu'image-clé dans le calque Options Menu.
- 3 Faites glisser le bouton `location_SF` de la bibliothèque vers la scène. Dans l'inspecteur des propriétés, tapez `sf_btn` dans la zone de texte Nom de l'occurrence.
- 4 Faites glisser le bouton `location_SJ` de la bibliothèque vers la scène. Dans l'inspecteur des propriétés, tapez `sj_btn` dans la zone de texte Nom de l'occurrence.
- 5 Faites glisser le bouton `location_PA` de la bibliothèque vers la scène. Dans l'inspecteur des propriétés, tapez `pa_btn` dans la zone de texte Nom de l'occurrence.
- 6 Faites glisser le bouton `location_BK` de la bibliothèque vers la scène. Dans l'inspecteur des propriétés, tapez `bk_btn` dans la zone de texte Nom de l'occurrence.

La scène de votre application devrait ressembler à la capture d'écran ci-dessous :



A. `sf_btn` B. `sj_btn` C. `pa_btn` D. `bk_btn`

- 7 Sélectionnez l'image 61 dans le calque ActionScript. Dans le panneau Actions, tapez le code suivant :

```

/* Frame Actions */
stop ();
fscommand2 ("SetSoftKeys", "Save", "Cancel");
setLocation ();

/* Handle soft key presses */
Key.removeListener (myListener);
var myListener:Object = new Object ();
myListener.onKeyDown = function () {
    var keyCode = Key.getCode ();
    if (keyCode == ExtendedKey.SOFT1 ) {
        // Save location
        saveNewLocation ();
        gotoAndPlay ("home");
    }
    else if (keyCode == ExtendedKey.SOFT2) {
        // Cancel operation, go back to home screen
        gotoAndPlay ("home");
    }
};
Key.addListener (myListener);

/*
The saveNewLocation() function is called when the user presses "Save" (left soft key).
It updates the "location" shared object with the phone # associated with that location.
*/
function saveNewLocation () {
    // Determine which button (location) the user selected:
    var selectedButton = Selection.getFocus ();
    switch (selectedButton) {
        case "_level0.sf_btn" :
            // User selected San Francisco
            location_so.data.phoneNumber = "415-555-1212";
            break;
        case "_level0.sj_btn" :
            // User selected San Jose
            location_so.data.phoneNumber = "408-555-1212";
            break;
        case "_level0.bk_btn" :
            // User selected Berkeley
            location_so.data.phoneNumber = "510-555-1212";
            break;
        case "_level0.pa_btn" :
            // User selected Palo Alto
            location_so.data.phoneNumber = "650-555-1212";
            break;
    }
}

/*
The setLocation() function sets the section to
the currently selected location,
based on the phone number saved in the shared object.
*/
function setLocation () {

```

```
// Retrieve phone number stored in shared object:
var loc:String = location_so.data.phoneNumber;
// Extract area code from phone number:
var areaCode:String = loc.substring (0, 3);
// Based on area code, set selection focus to corresponding button/menu item:
switch (areaCode) {
    case "415" :
        Selection.setFocus (_level0.sf_btn);
        break;
    case "408" :
        Selection.setFocus (_level0.sj_btn);
        break;
    case "510" :
        Selection.setFocus (_level0.bk_btn);
        break;
    case "650" :
        Selection.setFocus (_level0.pa_btn);
        break;
}
```

**8** Sélectionnez Fichier > Enregistrer.

**9** Sélectionnez Contrôle > Tester l'animation pour tester l'application dans Device Central.

Ouvrez la fenêtre Sortie de Device Central pour afficher les commentaires concernant le restaurant appelé.

# Chapitre 4 : Création d'interactivité et d'éléments de navigation

Pour interagir avec votre application Adobe Flash Lite, l'utilisateur doit pouvoir déterminer l'objet ayant le focus, parcourir les objets et lancer une action en sélectionnant un objet ou une autre touche. Bien que ces principes de base soient identiques à ceux des applications bureautiques, certaines fonctionnalités varient en fonction du périphérique mobile.

## Interaction de l'utilisateur et touches prises en charge

### A propos de l'interaction de l'utilisateur dans Flash Lite

Flash Lite prend en charge la navigation et l'interaction de l'utilisateur par le biais du clavier du périphérique, ou bien d'une interface de type stylet ou tactile sur les périphériques qui en sont dotés. Les options disponibles pour votre application dépendent du périphérique cible et du type de contenu. Pour plus d'informations sur les types de contenus, consultez la section « [A propos des types de contenus Flash Lite](#) » à la page 17.

La méthode la plus simple de mise en place de touches interactives dans une application Flash Lite consiste à appliquer la navigation par défaut. Cette dernière exploite le pavé quadridirectionnel comme les touches fléchées ou la touche de tabulation et la combinaison Maj+tabulation dans une application bureautique. L'utilisateur déplace le focus de l'objet voulu, puis appuie sur la touche de sélection. L'application inclut le code de gestionnaire d'événements pour répondre à ces événements de bouton. La navigation par défaut de Flash Lite permet d'utiliser des boutons, des champs texte de saisie et, en option, des clips. Cette fonctionnalité est généralement préférable pour les interactions utilisateur simples, telles que les menus. Pour plus d'informations sur la navigation par défaut, consultez la section « [Utilisation de la navigation par défaut dans Flash Lite](#) » à la page 37.

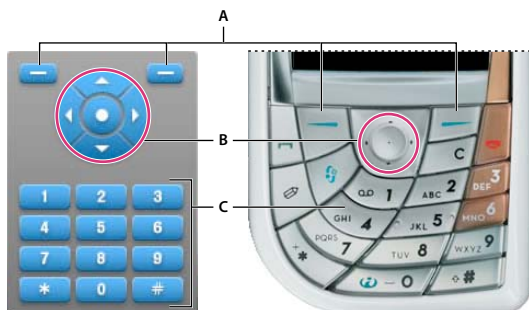
Les applications peuvent également répondre à des événements de touche arbitraires que Flash Lite génère lorsque l'utilisateur appuie sur une touche. La navigation en fonction des événements permet de créer des applications Flash Lite telles que des jeux qui ont un modèle complexe d'interaction avec l'utilisateur. Pour plus d'informations sur les événements, consultez la section « [A propos des événements de touche et de bouton](#) » à la page 43.

### Touches prises en charge par Flash Lite

Outre les touches alphanumériques disponibles sur les téléphones standard, la plupart des périphériques mobiles sont équipés d'un clavier de navigation permettant aux utilisateurs de parcourir et sélectionner des éléments sur l'écran du périphérique, ainsi que d'au moins deux touches programmables. Les *touches programmables* d'un périphérique sont des touches multifonctionnelles utilisant l'écran pour identifier leur cible à tout moment.

La plupart des pavés de navigation comportent quatre touches de navigation (Haut, Bas, Gauche et Droite) et une touche de sélection (généralement située au centre du pavé). Ces touches peuvent être exploitées différemment selon l'application active. Par exemple, un jeu permet à l'utilisateur de déplacer un personnage à l'écran à l'aide des touches de navigation, puis d'utiliser la touche de sélection pour exécuter une autre action, telle que faire sauter le personnage.

Les illustrations suivantes présentent les touches les plus courantes d'un clavier générique et d'un véritable périphérique :



A. Touches programmables Gauche et Droite B. Pavé de navigation C. Touches numériques, \* et #

Certains périphériques et types de contenus Flash Lite ne prennent pas en charge toutes ces touches. Par exemple, les périphériques qui prennent en charge la navigation bidirectionnelle ne prennent pas en charge les touches de navigation gauche et droite (consultez la section « [Modes de navigation par défaut](#) » à la page 37). En outre, seuls certains périphériques permettent d'accéder aux touches programmables.

Flash Lite prend en charge les touches suivantes sur les périphériques mobiles :

Description	Touches	Disponibilité
Numeric, *, #	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #	Tous les périphériques
Clavier à cinq directions	Sélection, Haut et Bas	Tous les périphériques
	Gauche et Droite	Périphériques qui prennent en charge la navigation dans les 4 directions uniquement (consultez la section « <a href="#">Modes de navigation par défaut</a> » à la page 37)
Touches programmables	Gauche et Droite	Périphériques prenant en charge la commande SetSoftKeys
	Touches SOFT3 à SOFT12	Périphériques ayant plus de deux touches programmables
Touches du clavier	!, ", #, \$, %, &, ', (, ), *, +, ,, -, ., /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, :, ;, <, =, >, ?, @, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, [, \, ], ^, _ ` ' a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, {,  , }, ~, Retour arrière	Périphériques disposant d'un clavier alphanumérique

Les propriétés System.Capabilities suivantes permettent de déterminer les options de navigation et de sélection disponibles sur un périphérique :

- hasMappableSoftKeys
- softKeyCount
- has4WayKeyAS
- hasQWERTYKeyboard
- hasStylus
- hasMouse

Pour plus d'informations sur la classe System.Capabilities, consultez le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*.



# Navigation par défaut

## Utilisation de la navigation par défaut dans Flash Lite

Dans les applications de bureau Flash, les touches tabulation et Maj+tabulation permettent aux utilisateurs de sélectionner les objets affichés à l'écran. Les touches fléchées ont la même fonction dans d'autres applications. Dans Flash Lite, les touches de navigation du pavé de navigation du périphérique ont la même fonction que la flèche ou les touches de tabulation et Maj+tabulation d'une application bureautique Flash. Lorsque l'objet désiré a le focus, l'utilisateur peut appuyer sur la touche sélectionnée pour déclencher une action dans l'application. Définissez des gestionnaires d'événements permettant de répondre lorsqu'un bouton ou un clip est sélectionné ; pour plus d'informations, consultez la section « [Gestion des événements de bouton](#) » à la page 43.

La navigation par défaut de Flash Lite utilise des boutons et des champs texte de saisie. Des clips sont également inclus si leur propriété `tabEnabled` est définie sur `true` ou si des gestionnaires d'événements leur sont associés et que leur propriété `tabEnabled` n'est pas définie sur `false`.

Lorsqu'un champ texte de saisie a le focus et que l'utilisateur appuie sur la touche de sélection, Flash Lite ouvre la boîte de dialogue de saisie de texte du périphérique dans laquelle l'utilisateur peut saisir du texte.

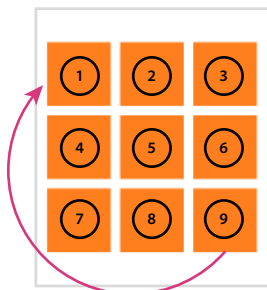
Pour visualiser un exemple de navigation par défaut, consultez la section « [Création d'un menu simple à l'aide des boutons et de la navigation par défaut](#) » à la page 46.

## Modes de navigation par défaut

Flash Lite prend en charge trois modes de navigation par défaut : navigation bidirectionnelle, navigation quadridirectionnelle et navigation quadridirectionnelle en boucle. Différents périphériques et types de contenus Flash Lite prennent en charge différents modes de navigation. Pour plus d'informations sur le choix du mode de navigation pour un type de périphérique ou de contenu spécifique, consultez la section « [Lecture d'une vidéo de périphérique avec l'émulateur](#) » à la page 124.

La *Navigation bidirectionnelle* de Flash Lite est proche de la navigation par tabulation dans les applications bureautiques Flash, où les touches Tab et Maj+Tab permettent de parcourir les objets à l'écran. La touche de navigation vers le bas du périphérique correspond à la touche de tabulation, tandis que la touche de navigation vers le haut correspond à la touche Maj+tabulation.

L'ordre de tabulation de la navigation bidirectionnelle se fait généralement de gauche à droite et de haut en bas. Par exemple, l'image suivante affiche une grille regroupant les neuf objets Button dans une application Flash Lite. Les numéros situés au-dessus de chaque bouton indiquent l'ordre dans lequel ils reçoivent le focus du clavier lorsque l'utilisateur appuie plusieurs fois sur la touche de navigation vers le bas. Lorsque le bouton situé dans le coin inférieur droit reçoit le focus du clavier, le focus revient au bouton supérieur gauche lorsque l'utilisateur appuie de nouveau sur la touche de navigation vers le bas.



Exemple d'ordre de tabulation lors de la navigation bidirectionnelle

Vous pouvez personnaliser l'ordre de tabulation de la navigation bidirectionnelle à l'aide de la propriété `tabIndex` des objets Button, MovieClip et TextField. Pour plus d'informations, consultez la section « [Contrôle de l'ordre de tabulation lors de la navigation bidirectionnelle](#) » à la page 42.

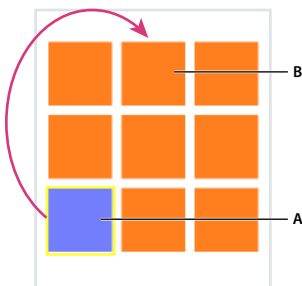
Pour obtenir un exemple de navigation bidirectionnelle, consultez la page regroupant des didacticiels et des exemples pour Flash Lite à l'adresse [www.adobe.com/go/learn\\_flt\\_samples\\_and\\_tutorials\\_fr](http://www.adobe.com/go/learn_flt_samples_and_tutorials_fr). Repérez et téléchargez le fichier .zip correspondant à votre version d'ActionScript, puis décompressez-le. Recherchez ensuite le dossier Samples et ouvrez-le pour afficher le fichier nommé 2-way fla.

La *navigation quadridirectionnelle* de Flash Lite est semblable à l'utilisation des touches fléchées du clavier d'un ordinateur de bureau pour parcourir les objets qui sont affichés à l'écran. Les touches de navigation vers le haut, le bas, la gauche et la droite du périphérique correspondent aux quatre touches fléchées du clavier d'un ordinateur. Le fait d'appuyer sur une touche déplace le focus du clavier vers l'objet situé dans cette direction, s'il y en a un. En l'absence d'objet dans cette direction, le focus du clavier reste sur l'objet actif.

**Remarque :** La propriété `tabIndex` n'est pas prise en charge sur les périphériques prenant en charge la navigation quadridirectionnelle, contrairement à `tabEnabled` et `tabChildren`, ce qui diffère du fonctionnement de ces propriétés dans les applications bureautiques Flash.

Pour obtenir un exemple de navigation quadridirectionnelle, consultez la page regroupant des didacticiels et des exemples pour Flash Lite à l'adresse [www.adobe.com/go/learn\\_flt\\_samples\\_and\\_tutorials\\_fr](http://www.adobe.com/go/learn_flt_samples_and_tutorials_fr). Repérez et téléchargez le fichier .zip correspondant à votre version d'ActionScript, puis décompressez-le. Recherchez ensuite le dossier Samples et ouvrez-le pour afficher le fichier nommé 4-way fla.

La *navigation quadridirectionnelle en boucle* associe la navigation quadridirectionnelle standard à la navigation bidirectionnelle. Tout comme pour la navigation quadridirectionnelle standard décrite précédemment, les utilisateurs déplacent le focus du clavier à l'aide des touches de navigation quadridirectionnelles. La différence est que, comme pour la navigation bidirectionnelle, le focus du clavier « retourne » à l'objet situé sur le côté opposé de l'écran. Par exemple, dans l'illustration ci-dessous, le bouton disposant du focus clavier est situé dans le coin inférieur gauche de l'écran. Lorsque l'utilisateur appuie sur la touche de navigation vers le bas, le bouton situé au milieu de la rangée supérieure de boutons reçoit le focus.



A. Bouton avec le focus B. Le bouton suivant reçoit le focus après que l'utilisateur a appuyé sur la touche de navigation

Vous pouvez tester le comportement des modes bi- et quadridirectionnels dans l'émulateur Adobe Device Central en utilisant les exemples 2-way fla et 4-way fla disponibles à l'adresse [www.adobe.com/go/learn\\_flt\\_samples\\_and\\_tutorials\\_fr](http://www.adobe.com/go/learn_flt_samples_and_tutorials_fr). Cette page Web regroupe des didacticiels et des exemples : repérez, téléchargez et décompressez le fichier .zip correspondant à votre version de Flash Lite, puis affichez le dossier Samples qu'il contient et sélectionnez les exemples qui vous intéressent. Chaque fichier d'exemple se compose de la même grille à neuf boutons, comme indiqué plus haut. La seule différence entre les exemples de fichier est que chaque fichier FLA est configuré pour cibler à la fois un périphérique et un type de contenu Flash Lite prenant en charge le mode de navigation voulu.

Pour utiliser chaque fichier d'exemple, ouvrez-le dans Flash et testez-le dans l'émulateur Adobe Device Central (sélectionnez Contrôle > Tester l'animation). Les touches fléchées du clavier de l'émulateur (ou bien celles de votre clavier) permettent de voir dans quelle mesure chaque mode de navigation affecte la navigation de l'utilisateur.



Tableau de boutons d'un fichier d'exemple

## A propos du rectangle de focus

Par défaut, Flash Lite dessine un rectangle jaune autour du bouton ou du champ texte de saisie ayant le focus. Des clips sont également inclus si leur propriété `tabEnabled` est définie sur `true` ou si des gestionnaires d'événements leur sont associés et que leur propriété `tabEnabled` n'est pas définie sur `false`.

Ce rectangle de focus permet à l'utilisateur de connaître l'objet de l'écran qui répondra s'il appuie sur la touche de sélection du périphérique. Par exemple, dans l'illustration suivante, le rectangle de focus encadre le bouton ayant le focus de clavier :



Bouton ayant le focus courant et rectangle de focus par défaut

Dans le cas des boutons et des clips, le cadre de délimitation du rectangle de focus est déterminé par la *zone active* de l'objet, la zone invisible qui (dans les applications Flash pour ordinateurs de bureau) définit la partie du bouton ou du clip réagissant aux clics de la souris. Dans le cas des champs texte de saisie, le cadre de délimitation du rectangle de focus est déterminé par les dimensions du champ texte.

Vous pouvez personnaliser la couleur du rectangle de focus ou désactiver ce dernier. Pour plus d'informations, consultez la section « [Personnalisation du rectangle de focus](#) » à la page 40.

## Directives relatives à l'utilisation de la navigation par défaut

Vous trouverez ci-dessous des conseils et des points à retenir lors de l'application de la navigation par défaut dans vos applications Flash Lite.

- Si vous désactivez le rectangle de focus par défaut en définissant `_focusRect` sur `false`, vous devez fournir un autre indicateur de focus pour vos boutons, champs texte de saisie et clips activés par la touche de tabulation. Pour chaque bouton, vous pouvez ajouter un élément visuel lorsque celui-ci est « survolé », à savoir la partie du scénario de l'objet button qui s'affiche lorsque le bouton a le focus. Pour obtenir une démonstration de cette technique, consultez l'exemple d'application dans la section « [Création d'un menu simple à l'aide des boutons et de la navigation par défaut](#) » à la page 46. Pour les champs texte de saisie, l'objet Selection permet de déterminer si ce champ a le focus et d'afficher l'indicateur voulu. Pour obtenir une démonstration, consultez l'exemple d'application dans la section « [Contrôle du focus avec le code ActionScript](#) » à la page 41.
- Il est recommandé d'avoir au moins deux objets (champs texte de saisie, boutons, clips activés par la touche de tabulation ou toute combinaison de ces objets) à l'écran en même temps. Si l'écran ne contient qu'un seul bouton, champ texte de saisie ou clip activé par la touche de tabulation, l'utilisateur ne peut pas modifier le focus ; il risque alors de se sentir bloqué par l'interface utilisateur.

Si un écran de votre application ne contient qu'un seul bouton d'interaction utilisateur, préférez la détection d'un événement de pression de touche à l'utilisation d'événements de bouton. Pour plus d'informations, consultez la section « [A propos des événements de touche et de bouton](#) » à la page 43.

- Selon les besoins, pensez à appliquer la méthode `Selection.setFocus()` pour définir le focus initial sur un objet spécifique à l'écran. Ceci permet de guider l'utilisateur au sein de l'interface et de réduire au minimum le nombre de touches à utiliser. Par exemple, supposons que l'un des écrans de votre application contienne un champ texte de saisie. Normalement, pour entrer une valeur dans le champ texte, l'utilisateur doit d'abord appuyer sur une touche de navigation pour donner le focus au champ texte, puis il appuie sur la touche sélectionnée pour ouvrir la boîte de dialogue de saisie du texte. Le code ActionScript suivant permet de placer le focus automatiquement sur le champ texte de saisie :

```
Selection.setFocus(inputTxt);
```

Pour plus d'informations sur le contrôle du focus avec le code ActionScript, consultez la section « [Contrôle du focus avec le code ActionScript](#) » à la page 41.

- L'alignement des objets à l'écran est important lors de la navigation par défaut. Lorsque les objets affichés sont décalés les uns par rapport aux autres, la séquence d'attribution du focus risque de ne pas être évidente du point de vue de l'utilisateur. Vous pouvez contourner ce problème en créant des objets de taille homogène et en les alignant à la verticale et à l'horizontale dans la mesure du possible. Avec la navigation bidirectionnelle, vous pouvez également contrôler la séquence à l'aide de la propriété `tabIndex` ; pour plus d'informations, consultez la section « [Contrôle de l'ordre de tabulation lors de la navigation bidirectionnelle](#) » à la page 42.

## Personnalisation du rectangle de focus

Le rectangle de focus est surligné par défaut en jaune, pour indiquer la zone de texte ou le bouton sélectionnés. Des clips sont également inclus si leur propriété `tabEnabled` est définie sur `true` ou si des gestionnaires d'événements leur sont associés et que leur propriété `tabEnabled` n'est pas définie sur `false`. Pour plus d'informations, consultez la section « [A propos du rectangle de focus](#) » à la page 39.

Vous pouvez désactiver le comportement par défaut du rectangle de focus en définissant la propriété globale `_focusRect` sur `false`. Vous pouvez aussi désactiver le rectangle de focus pour des clips et des boutons spécifiques (voir `focusrect` (propriété `Button._focusrect`) et `focusrect` (propriété `MovieClip._focusrect`)) dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*.

D'autre part, vous pouvez changer la couleur du rectangle de focus pour appliquer la couleur de votre choix. Pour ce faire, exécutez la commande `SetFocusRectColor` qui prend des valeurs RVB en tant que paramètres. Par exemple, le code suivant surligne le rectangle de focus en rouge :

```
fscommand2 ("SetFocusRectColor", 255, 0, 0);
```

## Contrôle du focus avec le code ActionScript

L'objet `Selection` d'ActionScript permet d'obtenir et définir le focus du clavier ou d'être prévenu lorsqu'un objet reçoit ou perd le focus du clavier. Ceci est utile, par exemple, pour attribuer automatiquement le focus à un bouton lors du premier chargement de l'application. Vous pouvez également être prévenu lorsque l'un des objets affichés à l'écran a reçu (ou perdu) le focus du clavier afin de mettre l'écran à jour en conséquence.

Par exemple, le code suivant applique la méthode `Selection.setFocus()` pour définir le focus sur l'occurrence de bouton `login_btn` :

```
Selection.setFocus(login_btn);
```

L'écouteur d'événements `Selection.onSetFocus` permet de déterminer si le focus du clavier a changé. Vous pouvez utiliser cet écouteur d'événements, par exemple, pour créer un gestionnaire de focus personnalisé pour les champs texte de saisie, au lieu d'utiliser le rectangle de focus par défaut. La procédure suivante indique comment créer un gestionnaire de focus personnalisé pour changer la couleur de la bordure de l'objet `TextField` ayant le focus. Pour obtenir un exemple d'application finalisée (`custom_focus_manager fla`), consultez la page qui propose des didacticiels et des exemples pour Flash Lite à l'adresse [www.adobe.com/go/learn\\_flt\\_samples\\_and\\_tutorials\\_fr](http://www.adobe.com/go/learn_flt_samples_and_tutorials_fr). Repérez et téléchargez le fichier .zip correspondant à votre version d'ActionScript, puis décompressez-le. Recherchez ensuite le dossier `Samples` et ouvrez-le pour afficher l'exemple qui vous intéresse.

## Création d'un gestionnaire personnalisé de focus sur le champ texte de saisie

- 1 Dans Flash, créez un document mobile et enregistrez-le sous `custom_focus_manager fla`.
- 2 Avec l'outil Texte, créez un champ texte sur la scène.
- 3 Le champ texte étant toujours sélectionné, dans l'inspecteur des propriétés, sélectionnez Texte de saisie dans le menu contextuel Type de texte, puis tapez **inputTxt\_1** dans la zone de texte Nom de l'occurrence et sélectionnez l'option Afficher la bordure autour du texte.
- 4 De la même manière, créez un autre champ texte de saisie sous le premier avec le nom d'occurrence **inputTxt\_2**, puis sélectionnez l'option Afficher la bordure autour du texte pour ce deuxième champ texte.
- 5 Dans le scénario, sélectionnez l'image 1 dans le calque intitulé ActionScript.
- 6 Ouvrez le panneau Actions (Fenêtre > Actions) et entrez (ou copiez et collez) le code suivant :

```
// Disable focus rect globally:
_focusrect = false;
// Create Selection listener object:
var focusListener:Object = new Object ();
// Define onSetFocus method:
focusListener.onSetFocus = function (oldFocus, newFocus) {
    // Enable/disable selection indicator:
    if (newFocus instanceof TextField) {
        // Set border color of text field with new focus to red:
        newFocus.borderColor = 0xFF0000;
    }
    if (oldFocus != undefined && oldFocus instanceof TextField) {
        // Set border color of text field with old focus to black:
        oldFocus.borderColor = 0x000000;
    }
};
// Add listener to Selection object:
Selection.addListener (focusListener);
// Set initial focus when application loads:
Selection.setFocus (inputTxt_1);
// Enable full-screen mode:
fscommand2 ("FullScreen", true);
```

- 7 Enregistrez vos modifications et testez l'application dans l'émulateur (Contrôle > Tester l'animation).
- 8 Appuyez sur les touches Bas et Haut de l'émulateur pour faire basculer le focus d'un champ texte à l'autre. Le champ texte ayant le focus doit avoir une bordure rouge, tandis que l'autre a alors une bordure noire. Appuyez sur la touche de sélection lorsqu'un champ texte a le focus pour afficher la boîte de dialogue de saisie.

## Contrôle de l'ordre de tabulation lors de la navigation bidirectionnelle

La navigation bidirectionnelle de Flash Lite est analogue à la navigation par tabulation de Flash, ce qui inclut la prise en charge de la propriété `tabIndex` permettant de définir avec précision l'ordre des boutons, des clips et des champs texte de saisie. Sur les périphériques qui prennent en charge la navigation quadridirectionnelle, la propriété `tabIndex` n'est pas prise en charge. Par conséquent, il n'est pas possible de définir l'ordre de tabulation avec la propriété `tabIndex` pour la navigation quadridirectionnelle.

Pour contrôler l'ordre de tabulation lors de la navigation bidirectionnelle, affectez à la propriété `tabIndex` de chaque objet un nombre spécifiant la position de ces objets dans le cadre de la navigation par défaut. Par exemple, supposons qu'une application contienne le bouton `my_button`, le clip `my_movieclip` et le champ texte de saisie `my_inputTxt`. Le code suivant définit l'ordre de tabulation de façon à donner le focus au bouton, puis au clip et enfin au champ de saisie.

```
my_button.tabIndex = 1;
my_movieclip.tabEnabled = true;
my_movieclip.tabIndex = 2;
my_inputTxt.tabIndex = 3;
```

# Événements de touche et de bouton

## A propos des événements de touche et de bouton

Les gestionnaires et les écouteurs d'événements déterminent la façon dont l'application réagit aux occurrences générées par l'utilisateur ou le système. Par exemple, lorsqu'un bouton a le focus et que l'utilisateur appuie sur la touche de sélection, un événement `onPress` est généré. Outre la navigation par défaut et les interactions avec les événements, une application Flash Lite permet également d'écouter des événements de pression de touche et d'y répondre.

Certains périphériques et types de contenus ne prennent pas en charge toutes les touches de périphérique. Par exemple, sur un périphérique prenant en charge la navigation bidirectionnelle (consultez la section « [Modes de navigation par défaut](#) » à la page 37), Flash Lite ne génère pas d'événements de pression pour les touches Gauche et Droite. Pour consulter la liste des touches et la description de leur disponibilité, consultez la section « [Touches prises en charge par Flash Lite](#) » à la page 35.

## Gestion des événements de bouton

L'emploi des boutons permet de renforcer l'interactivité de vos applications Flash Lite. Flash Lite prend en charge les mêmes événements de bouton que Flash Player sur les ordinateurs de bureau, bien que certains événements (par exemple, `onDragOut`) soient uniquement disponibles sur les périphériques recourant à une souris ou à un stylet en tant qu'interface. Sur les périphériques avec clavier uniquement, un bouton doit avoir le focus du clavier pour générer des événements.

Flash Lite prend en charge les événements ActionScript de bouton suivants :

Événement de bouton	Description
<code>onDragOut</code>	Pris en charge uniquement sur les périphériques disposant d'une souris ou d'un stylet. Cet événement est appelé lorsque l'utilisateur appuie sur le bouton de la souris et lorsque le pointeur est déplacé en dehors de la zone occupée par le bouton.
<code>onDragOver</code>	Pris en charge uniquement sur les périphériques disposant d'une souris ou d'un stylet. Cet événement est appelé lorsque l'utilisateur appuie sur le bouton de la souris et fait glisser le pointeur en dehors, puis sur le bouton.
<code>onKeyDown</code>	Cet événement est appelé lorsque le bouton a le focus et l'utilisateur appuie sur une touche.
<code>onKeyUp</code>	Cet événement est appelé lorsque le bouton a le focus et l'utilisateur relâche une touche.
<code>onKillFocus</code>	Cet événement est appelé lorsque le focus est retiré du bouton.
<code>onPress</code>	Cet événement est appelé lorsque l'utilisateur appuie sur la touche de sélection du périphérique dont le bouton a le focus. Il est aussi appelé lorsque l'utilisateur appuie sur le bouton de la souris lorsque le bouton a un focus.
<code>onRelease</code>	Cet événement est appelé lorsque l'utilisateur relâche la touche de sélection du périphérique dont le bouton a le focus. Il est aussi appelé lorsque l'utilisateur appuie sur le bouton de la souris lorsque le bouton a un focus.
<code>onReleaseOutside</code>	Cet événement est appelé lorsque l'utilisateur relâche le bouton de la souris alors que le pointeur est en dehors du bouton et après avoir appuyé sur ce même bouton alors que le pointeur était dans la surface occupée par le bouton.
<code>onRollOut</code>	Cet événement est appelé lorsqu'un bouton perd le focus.
<code>onRollOver</code>	Cet événement est appelé lorsqu'un bouton reçoit le focus.
<code>onSetFocus</code>	Cet événement est appelé lorsqu'un bouton reçoit le focus de saisie.

La procédure ci-après explique comment créer une application simple capable de gérer des événements de bouton. Pour obtenir un exemple d'utilisation des boutons en vue de créer un menu, consultez la section « [Création d'un menu simple à l'aide des boutons et de la navigation par défaut](#) » à la page 46.

## Création d'un gestionnaire d'événements de bouton

- 1 Dans Flash, créez un document mobile et enregistrez-le sous `custom_focus_manager fla`.
- 2 Sélectionnez Fenêtre > Bibliothèques communes > Boutons pour ouvrir une bibliothèque externe de symboles de bouton prédéfinis.
- 3 Dans le panneau Bibliothèque, double-cliquez sur le dossier « classic buttons » pour l'ouvrir, puis ouvrez le dossier « Circle Buttons ».
- 4 Faites glisser une occurrence du symbole de bouton Menu sur la scène.
- 5 Dans l'inspecteur des propriétés, dans la zone de texte Nom d'occurrence, tapez **btn\_1**.
- 6 Faites glisser une autre occurrence du même bouton sur la scène et placez-la directement en dessous du premier bouton.
- 7 Dans l'inspecteur des propriétés, dans la zone de texte Nom d'occurrence, tapez **btn\_2**.
- 8 Dans le scénario, sélectionnez l'image 1 dans le calque intitulé `ActionScript`.
- 9 Ouvrez le panneau Actions (Fenêtre > Actions), puis tapez le code indiqué ci-dessous :

```
// Disable the focus rectangle because buttons have an over state
_focusRect = false;
```

```
// Event handlers for btn_1
btn_1.onPress = function() {
    trace("You pressed Button 1");
}
btn_1.onRelease = function() {
    trace("You released Button 1");
}
btn_1.onRollOver = function() {
    trace("Button 1 has focus");
}
btn_1.onRollOut = function() {
    trace("Button 1 lost focus");
}
```

```
// Event handlers for btn_2
btn_2.onPress = function() {
    trace("You pressed Button 2");
}
btn_2.onRelease = function() {
    trace("You released Button 2");
}
btn_2.onRollOver = function() {
    trace("Button 2 has focus");
}
btn_2.onRollOut = function() {
    trace("Button 2 lost focus");
}
```

- 10 Testez l'application dans l'émulateur (Contrôle > Tester l'animation).



Observez les messages dans le panneau Sortie lorsque vous appuyez sur les touches Haut et Bas du clavier de l'émulateur.



D'autres types d'objets prennent en charge différents événements ; par exemple, l'objet `TextField` inclut un événement `onChanged` qui est appelé lors de la modification du contenu d'un champ texte. Vous pouvez écrire le code du gestionnaire de ces événements en reprenant le format des gestionnaires d'événements du bouton dans cette procédure. Pour plus d'informations sur les événements pris en charge pour les champs de texte et les clips, consultez les sections `TextField` et `MovieClip` dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*.

## Création d'un menu simple à l'aide des boutons et de la navigation par défaut

Pour créer un menu simple à l'aide des boutons et de la navigation par défaut, vous devez concevoir ce menu puis rédiger le code lui permettant de gérer les événements. Pour créer le menu, vous allez utiliser trois symboles de bouton, un pour chaque option du menu. Vous allez ensuite écrire le code de gestion des événements, qui va afficher un message lorsque l'utilisateur survole un élément de menu (autrement dit, lorsque l'utilisateur donne le focus au bouton correspondant) et lorsqu'il sélectionne l'élément de menu en appuyant sur la touche de sélection du périphérique. Pour plus d'informations sur la gestion des événements de bouton dans Flash Lite, consultez la section « [Gestion des événements de bouton](#) » à la page 43.

Vous allez commencer avec un document Flash partiel. Vous pouvez modifier ces paramètres pour cibler un périphérique ou un type de contenu différents (consultez la section « [Utilisation de l'émulateur](#) » à la page 118).

- 1 Téléchargez et ouvrez le fichier `simple_menu_start.fla` qui se trouve à l'adresse [www.adobe.com/go/learn\\_flt\\_samples\\_and\\_tutorials\\_fr](http://www.adobe.com/go/learn_flt_samples_and_tutorials_fr). Cette page Web regroupe des didacticiels et des exemples : repérez, téléchargez et décompressez le fichier .zip correspondant à votre version de Flash Lite, puis affichez le dossier Samples qu'il contient et sélectionnez l'exemple qui vous intéresse.
- 2 Ouvrez le panneau Bibliothèque (Fenêtre > Bibliothèque).  
La Bibliothèque contient trois symboles de bouton nommés News Button, Weather Button et Sports Button.
- 3 Dans le scénario (Fenêtre > Scénario), sélectionnez le calque intitulé Menu Buttons.
- 4 Faites glisser une occurrence du symbole News Button (Bouton Informations) du panneau Bibliothèque sur la scène.
- 5 Dans l'inspecteur des propriétés, dans la zone de texte Nom d'occurrence, tapez **btn\_news**.
- 6 Répétez les étapes 4 et 5 pour les boutons Sports et Weather, en les appelant respectivement **btn\_sports** et **btn\_weather**.
- 7 Alignez les trois boutons verticalement, comme indiqué dans l'exemple suivant :



- 8 Dans le panneau Outils, sélectionnez l'outil Texte et tracez un champ texte le long de la partie inférieure de la scène.  
Ce champ texte affiche un court message lorsque l'utilisateur survole les éléments du menu.
- 9 Avec le nouveau champ texte toujours sélectionné, effectuez les opérations suivantes dans l'inspecteur des propriétés :
  - a Sélectionnez Texte dynamique dans le menu déroulant Type de texte.
  - b Tapez **txt\_status** dans la zone de texte Nom d'occurrence.

La scène doit alors présenter l'aspect suivant :



**10** Dans le scénario, sélectionnez l'image 1 dans le calque intitulé ActionScript.

**11** Ouvrez le panneau Actions (Fenêtre > Actions), puis tapez le code indiqué ci-dessous :

```
// Disable the focus rectangle because buttons have an over state
_focusRect = false;

btn_news.onRollOver = function() {
    txt_status.text = "Press to select News";
}
btn_news.onPress = function() {
    txt_status.text = "You selected News";
}
btn_sports.onRollOver = function() {
    txt_status.text = "Press to select Sports";
}
btn_sports.onPress = function() {
    txt_status.text = "You selected Sports";
}
btn_weather.onRollOver = function() {
    txt_status.text = "Press to select Weather";
}
btn_weather.onPress = function() {
    txt_status.text = "You selected Weather";
}
```

**12** Sélectionnez Contrôle > Tester l'animation pour prévisualiser l'application dans l'émulateur.

Cliquez sur la touche Bas de l'émulateur (ou appuyez sur la flèche Bas du clavier de votre ordinateur) pour passer d'une option de menu à une autre. Pour sélectionner un élément de menu, cliquez sur la touche de sélection de l'émulateur (ou appuyez sur la touche Entrée du clavier de votre ordinateur).



## Gestion des événements de touche

Flash Lite génère des événements de touche lorsque l'utilisateur appuie sur l'une des touches du périphérique. Vous pouvez écrire le code du gestionnaire de touches afin de répondre à ces événements. Pour obtenir la liste des touches prises en charge par Flash Lite, consultez la section « [Touches prises en charge par Flash Lite](#) » à la page 35.

Le tableau suivant regroupe les touches de périphérique les plus communes et les codes de touches correspondants dans ActionScript, ainsi que les constantes de code de ces touches :

Touche du périphérique	Code de touche ActionScript/constante de code de touche
Touche de sélection	Key.ENTER
Touche de navigation Haut	Key.UP
Touche de navigation Bas	Key.DOWN
Touche de navigation Gauche	Key.LEFT
Touche de navigation Droite	Key.RIGHT
Touche programmable Gauche	ExtendedKey.SOFT1 (ou soft1)
Touche programmable Droite	ExtendedKey.SOFT2 (ou soft2)
0	48
1	49
2	50

Touche du périphérique	Code de touche ActionScript/constante de code de touche
3	51
4	52
5	53
6	54
7	55
8	56
9	57
*	56
#	51

## Ecriture d'un gestionnaire d'événements

Les écouteurs d'événement permettent à un objet, appelé *objet écouteur*, de recevoir des événements diffusés par un autre objet, appelé *objet diffuseur*. L'objet émetteur enregistre l'objet écouteur de façon à ce que ce dernier puisse recevoir les événements qu'il émet. Pour plus d'informations, consultez la section « Utilisation d'écouteurs d'événements » du manuel *Formation à ActionScript Flash Lite 2.x*.

Le meilleur moyen de traiter les événements de pression de touches consiste à créer un objet écouteur de touche définissant la fonction `onKeyDown` ou `onKeyUp`, puis à enregistrer cet objet avec la méthode `Key.addListener()`. L'exemple de code suivant définit un écouteur de touche qui est activé lorsque l'utilisateur appuie sur la touche de navigation Droite du périphérique :

```
var myListener:Object = new Object();
myListener.onKeyDown = function() {
    if (Key.getCode() == Key.RIGHT) {
        trace("You pressed the right arrow key");
    }
}
Key.addListener(myListener);
```

## Gestion des événements de pression de touche à l'aide d'un écouteur de touche

- 1 Dans Flash, créez un document mobile et enregistrez-le sous `keylistener fla`.
- 2 Sélectionnez le calque intitulé Contenu dans le scénario.
- 3 A l'aide de l'outil Ovale, créez un ovale ou un cercle sur la scène puis convertissez-le en clip.
- 4 Ce nouveau clip étant sélectionné, dans l'inspecteur des propriétés, tapez **circle** dans la zone de texte Nom de l'occurrence.
- 5 Dans le scénario, sélectionnez la première image dans le calque 1.
- 6 Ouvrez le panneau Actions (Fenêtre > Actions), puis tapez le code suivant :

```

var myListener:Object = new Object();
myListener.onKeyDown = function() {
    if (Key.getCode() == Key.LEFT) {
        circle._x -= 10;
    } else if (Key.getCode() == Key.RIGHT) {
        circle._x += 10;
    } else if (Key.getCode() == Key.UP) {
        circle._y -= 10;
    } else if (Key.getCode() == Key.DOWN) {
        circle._y += 10;
    }
};
Key.addListener(myListener);

```

## 7 Testez l'application en sélectionnant Contrôle > Tester l'animation.

Appuyez sur les quatre touches de navigation du clavier de l'émulateur (ou sur les touches correspondantes de votre clavier) pour déplacer le cercle sur la scène.

## Utilisation des touches programmables

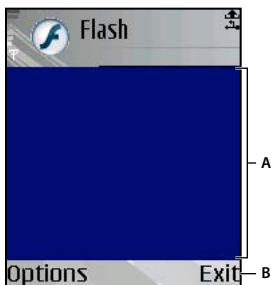
Pour utiliser les touches programmables de l'application Flash Lite, vous devez d'abord appeler la commande `SetSoftKeys`. Ensuite, Flash Lite génère un événement `ExtendedKey.SOFT1` lorsque l'utilisateur appuie sur la touche programmable Gauche et un événement `ExtendedKey.SOFT2` lorsqu'il appuie sur la touche programmable Droite. Vous pouvez rédiger un code de gestionnaire d'événements ActionScript qui réagit à ces événements et répond par l'action désirée.

La commande `SetSoftKeys` tient compte de deux paramètres spécifiant les étiquettes des touches programmables Gauche et Droite qui apparaissent lorsque votre application ne s'exécute *pas* en mode plein écran. Pour les applications fonctionnant en mode plein écran, les étiquettes spécifiées ne sont pas visibles ; vous devez donc créer vos propres étiquettes et les placer sur la scène, à l'endroit où se trouvent les touches programmables.

Considérons par exemple l'appel de la commande `SetSoftKeys` suivant :

```
fscommand2("SetSoftKeys", "Options", "Exit");
```

L'exemple ci-dessous montre le résultat de cette commande dans une application fonctionnant sur un véritable périphérique en mode normal (et non en plein écran) :



A. Zone d'écran disponible dans les applications ne fonctionnant pas en mode plein écran B. Touches programmables affichées par périphérique

Si vous activez le mode plein écran, autrement dit si vous appelez `fscommand("fullscreen", true)`, les étiquettes spécifiées comme paramètres pour la commande `SetSoftKeys` ne sont pas visibles. Par conséquent, dans les applications en mode plein écran, vous devez créer vos propres étiquettes de touches programmables, comme indiqué ci-dessous :



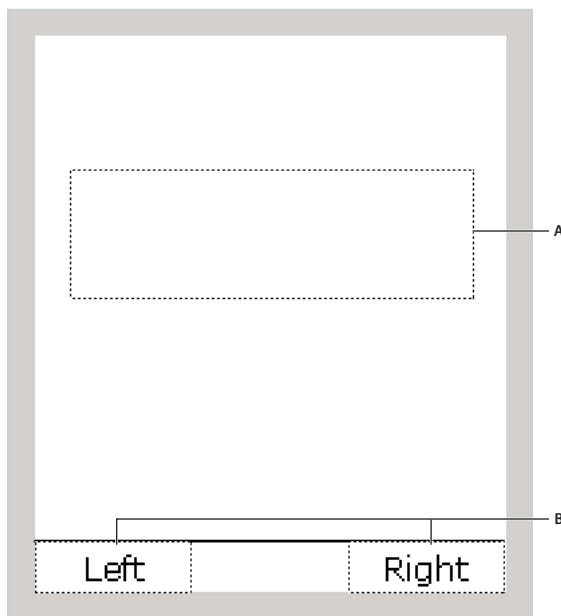
Etiquettes de touches programmables personnalisées

Pour plus d'informations sur la commande `SetSoftKeys`, consultez la section `Fonction fscommand2` dans le *Guide de référence du langage Flash Lite 2.x ActionScript*.

## Utilisation des touches programmables dans une application

- 1 Dans Flash, créez un document mobile et enregistrez-le sous `softkey fla`.
- 2 A l'aide de l'outil Texte, créez un champ de texte statique nommé Gauche et placez-le dans le coin inférieur gauche de la scène, au-dessus de la touche programmable Gauche du périphérique.
- 3 Créez un autre champ texte statique nommé Droite et placez-le dans le coin inférieur droit de la scène, au-dessus de la touche programmable Droite du périphérique.
- 4 En utilisant l'outil Texte, créez un champ texte dynamique et placez-le au milieu de la scène.

La scène de votre document doit avoir l'aspect suivant :



A. Champ texte dynamique B. Etiquettes de touches programmables

5 Le champ texte dynamique étant toujours sélectionné, dans l'inspecteur des propriétés, tapez **status** dans la zone de texte Nom de l'occurrence.

6 Ouvrez le panneau Actions (Fenêtre > Actions) et sélectionnez l'image 1 du calque 1 dans le scénario.

7 Dans le panneau Actions, tapez le code suivant :

```
fscommand2("SetSoftKeys", "Left", "Right");  
fscommand2("FullScreen", true);
```

8 Créez et enregistrez un objet permettant de répondre aux événements de pression de touche (consultez la section « [Gestion des événements de pression de touche à l'aide d'un écouteur de touche](#) » à la page 49) en entrant le code suivant dans le panneau Actions :

```
var myListener:Object = new Object();  
myListener.onKeyDown = function() {  
    if (Key.getCode() == ExtendedKey.SOFT1) {  
        // Handle left soft keypress event.  
        status.text = "You pressed the Left soft key.";  
    } else if (Key.getCode() == ExtendedKey.SOFT2) {  
        // Handle right soft keypress event.  
        status.text = "You pressed the Right soft key.";  
    }  
};  
Key.addListener(myListener);
```

9 Sélectionnez Contrôle > Tester l'animation pour tester l'application dans l'émulateur.

Pour tester l'application, cliquez sur les touches programmables Gauche et Droite de l'émulateur avec votre souris, ou bien appuyez sur les touches Page haut et Page bas de votre clavier.





# Chapitre 5 : Utilisation du texte et des polices

Vous pouvez ajouter des champs texte statique et dynamique et ajouter des champs de texte de saisie à vos applications Macromedia Flash Lite 2.x à partir d'applications Adobe et d'Adobe Flash Lite 3.x.

## Texte

### A propos du texte dans Flash Lite

Flash Lite 2.x et 3.x prennent en charge les fonctions de texte suivantes :

- champs texte de saisie, statique et dynamique

Lors de l'exécution, le contenu des champs texte statique ne peut pas changer contrairement à celui des champs texte dynamique ou de saisie. Les champs texte de saisie permettent aux utilisateurs d'entrer du texte. Flash Lite 2.x et 3.x prennent en charge la saisie de texte en ligne sur la plupart des périphériques. Pour les périphériques qui prennent en charge des langues complexes et dans Flash Lite 2.0, les champs texte de saisie appliquent le mécanisme générique de saisie de texte du périphérique. Pour plus d'informations sur l'utilisation des champs texte de saisie, consultez la section « [Utilisation des champs texte de saisie](#) » à la page 54.

- Polices intégrées et de périphérique

Vous pouvez procéder au rendu des champs texte à l'aide de polices vectorielles intégrées dans le fichier SWF ou utiliser les polices disponibles sur le périphérique. Pour plus d'informations sur les méthodes de rendu des polices, consultez la section « [Méthodes de rendu des polices dans Flash Lite](#) » à la page 63.

- Codage du texte au format Unicode

Flash Lite permet d'afficher du texte dans toutes les langues à condition que les polices contenant les glyphes requis soient disponibles. Pour plus d'informations sur la programmation multilingue dans Flash, consultez la section « *Création de texte multilingue* » du guide d'*utilisation de Flash*.

- Prise en charge partielle du formatage HTML et des propriétés de classe TextFormat
- Défilement du texte

Flash Lite ne prend pas en charge toutes les fonctionnalités texte de la version de bureau de Flash Player. Flash Lite est soumis aux restrictions suivantes :

- L'anti-alias (ou lissage) personnalisé, une technologie perfectionnée de rendu de police disponible à partir de Macromedia Flash Player 8 d'Adobe, n'est pas pris en charge.
- Le formatage du texte est pris en charge, mais seules les options couleur, type, taille, gras et italique sont disponibles. De plus, le formatage ne s'affiche que si la police du périphérique inclut l'option sélectionnée. Par exemple, un champ formaté en italiques s'affiche en texte normal lorsque le périphérique ne dispose pas d'une version en italiques.
- Le texte de périphérique ne peut pas être masqué, utilisé en tant que masque ou rendu avec de la transparence.

- L'option de mise en forme Rendre le texte au format HTML est prise en charge de façon partielle pour les champs texte de saisie et dynamiques. Le texte s'affiche sans les balises HTML, la mise en forme n'est appliquée que pour les balises suivantes : p, br, sbr, font (avec les attributs face, color et size), b et i.
- Flash Lite ne prend pas en charge les feuilles de style en cascade (CSS).
- Les composants Flash, ce qui inclut Label, TextArea et TextInput, ne sont pas pris en charge.
- Les objets TextField et TextFormat sont pris en charge de façon partielle et des limitations supplémentaires s'appliquent à l'arabe, à l'hébreu et au thaï. Pour plus d'informations, consultez le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*.
- Les objets XML et XMLNode sont pris en charge.

Les versions 2.1 et ultérieures de Flash Lite prennent en charge la saisie de texte en ligne, les moteurs de saisie intuitifs et XMLSocket :

- La fonctionnalité de saisie de texte en ligne permet à l'utilisateur d'entrer du texte directement dans les champs texte.
- La prise en charge de la saisie intuitive autorise des fonctionnalités telles que la saisie semi-automatique de mots et les listes de choix. Les versions 2.1 et ultérieures de Flash Lite prennent en charge les principaux moteurs de saisie intuitive (tels que T9, eZiTap/eZiText et iTap) sur différentes plates-formes, à condition de les mettre en œuvre de la même manière que les API standard des éditeurs de ce type de moteur.
- La prise en charge de XMLSocket étend la prise en charge du bureau Flash à Flash Lite et permet aux développeurs de créer des connexions de données continues et à faible latence pour des applications telles que des jeux et des modules de discussion en ligne.

## Création et mise en forme du texte

Le texte est créé et formaté dans Flash Lite de la même façon que dans une application de bureau Flash.

Pour plus d'informations sur l'utilisation du texte dans Flash, consultez les sections suivantes dans le guide Utilisation de Flash :

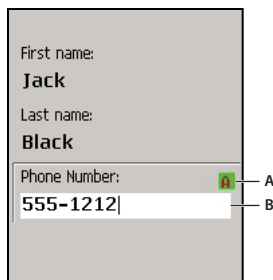
- « Création de texte »
- « Définition des attributs de texte »
- « Manipulation du texte »
- « Contrôle du texte avec ActionScript »

Pour obtenir la liste des fonctions texte qui ne sont pas prises en charge dans Flash Lite, consultez la section « [A propos du texte dans Flash Lite](#) » à la page 53 ».

## Saisie de texte

### Utilisation des champs texte de saisie

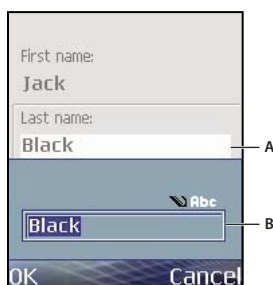
Les versions 2.1 et ultérieures de Flash Lite prennent en charge la saisie de texte en ligne. Cette fonctionnalité permet aux utilisateurs de modifier directement des champs de texte dans l'application Flash Lite, sans avoir à recourir à des zones de saisie distinctes comme c'était le cas dans les versions précédentes. Par exemple, l'illustration suivante indique comment un champ texte de saisie en ligne doit s'afficher sur un appareil Nokia série 60 :



A. Indicateur du mode de saisie B. Champ texte en cours de modification en ligne

Les versions 2.1 et ultérieures de Flash Lite prennent en charge la saisie de texte en ligne pour les langues latines et asiatiques, mais pas pour les langues complexes, notamment celles qui se lisent de droite à gauche, telles que l'arabe, l'hébreu, l'ourdou, le farsi et le yiddish, ainsi que certaines langues asiatiques.

Sur les périphériques qui exécutent Flash Lite 2.0 et Flash Lite 1.1, et dans toutes les versions de l'émulateur Adobe Device Central, les utilisateurs modifient le contenu des champs de saisie à l'aide d'une boîte de dialogue modale qui apparaît au-dessus du contenu Flash Lite. Par exemple, l'illustration suivante donne un exemple de boîte de dialogue de saisie de texte sur un appareil Symbian™ série 60 qui exécute Flash Lite 2.0 :



A. Boîte de dialogue de saisie de texte B. Champ texte en cours de modification

En général, les applications Flash Lite 2.0 et Flash Lite 1.1 existantes fonctionnent telles quelles dans les versions 2.1 et ultérieures de Flash Lite. Dans ce cas, la boîte de dialogue modale est remplacée par de simples champs texte de saisie en ligne. Cependant, le contenu existant doit être reprogrammé pour tirer parti des fonctions propres aux versions 2.1 et ultérieures de Flash Lite, telles que la possibilité de définir des sélections de texte ou la commande `activateTextField`.

Pour plus d'informations sur la boîte de dialogue de saisie de texte, consultez la section « [A propos de la boîte de dialogue de saisie du périphérique \(Flash Lite 2.0\)](#) » à la page 58.

## Utilisation de la saisie de texte en ligne (Flash Lite version 2.1 et ultérieure)

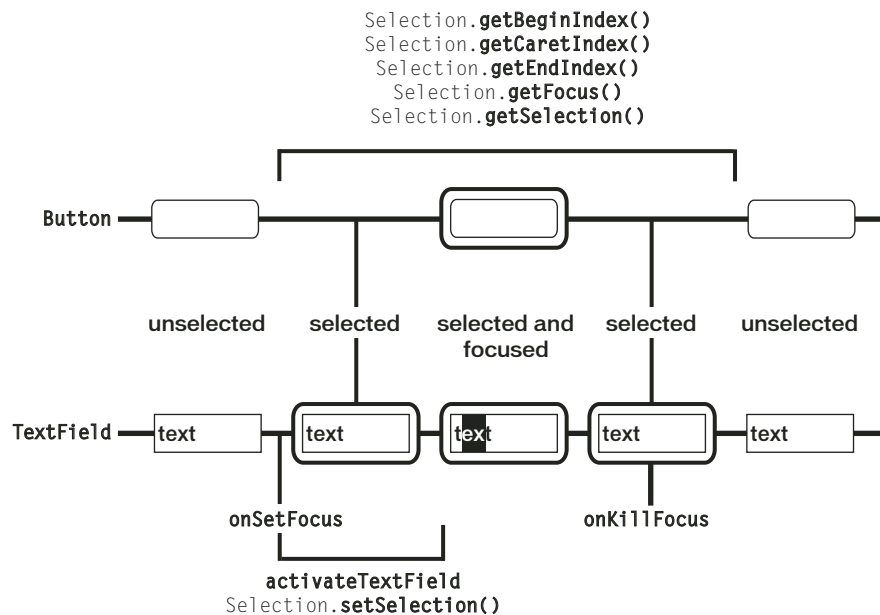
Les versions 2.1 et ultérieures de Flash Lite prennent en charge la saisie de texte en ligne, qui permet aux utilisateurs d'entrer et de modifier du texte directement dans un champ texte de saisie. Pendant la saisie de texte dans une application Flash Lite, l'utilisateur interagit avec l'éditeur natif IME (Input Method Editor) du périphérique. En effet, le lecteur Flash Lite ne traite pas directement les entrées de l'utilisateur.

**Remarque :** Dans Adobe Device Central, l'émulateur Flash Lite n'affiche pas le texte en ligne. Pour plus d'informations, consultez la section « [Test de texte en ligne dans Flash Lite 2.1 et version ultérieure](#) » à la page 118.

Lorsqu'un champ texte de saisie est actif, le lecteur Flash Lite 2.1 et 3.x s'exécute en mode *restreint*. Dans ce mode, la méthode native de saisie du périphérique traite tous les événements de pression de touche et aucun code ActionScript n'est exécuté. D'autre part, tous les événements d'arrêt d'animation, du son ou de la vidéo sont ignorés. Lorsqu'un champ texte est désactivé, Flash Lite reprend son mode de fonctionnement normal.

Les champs texte doivent être *activés* avant de traiter les saisies de l'utilisateur. L'utilisateur peut activer un champ texte de saisie en lui donnant le focus de sélection, puis en appuyant sur la touche de sélection du périphérique ou en appuyant sur une touche programmable qui donne le focus au champ texte. « [Activation de champs texte de saisie avec ActionScript](#) » à la page 56.)

L'illustration suivante présente les différents états d'un champ texte de saisie : non sélectionné, sélectionné (mais non activé) et activé :



## Champs texte de saisie en ligne et navigation

Les champs texte activés contiennent un pointeur qui représente le point d'insertion. L'utilisateur peut modifier l'emplacement du pointeur dans le champ texte à l'aide des touches de navigation du périphérique.

Par défaut, Flash Lite dessine un rectangle de focus autour du champ texte qui a le focus. Ce rectangle peut parfois masquer le pointeur clignotant ou l'indicateur d'insertion du champ texte activé. Par conséquent, il est recommandé de désactiver le rectangle de focus (en définissant setting `_focusRect` sur `false`) et d'utiliser un indicateur de focus personnalisé. Pour obtenir un exemple d'application recourant à un indicateur de focus personnalisé, consultez la page qui regroupe des didacticiels et des exemples pour Flash Lite à l'adresse [www.adobe.com/go/learn\\_flt\\_samples\\_and\\_tutorials\\_fr](http://www.adobe.com/go/learn_flt_samples_and_tutorials_fr). Repérez et téléchargez le fichier ZIP correspondant à votre version d'ActionScript, puis décompressez-le. Recherchez ensuite le dossier Samples et ouvrez-le pour afficher l'exemple qui vous intéresse.

## Activation de champs texte de saisie avec ActionScript

L'utilisateur peut activer un champ texte de saisie ayant le focus du clavier en appuyant sur la touche de sélection du périphérique. L'application Flash Lite peut également activer automatiquement un champ texte de saisie lorsque ce dernier reçoit le focus par le biais de la commande `activateTextField`. Cette commande active le champ texte sélectionné. Si aucun champ n'est sélectionné lorsque la commande s'exécute, aucune action ne se produit.

La commande `activateTextField` est généralement appelée à partir d'un gestionnaire `Selection.onSetFocus` ou `TextField.onSetFocus`. Par exemple, supposons que votre application contienne au moins deux champs texte de saisie sur la scène. Le code suivant activerait automatiquement le champ texte qui reçoit le focus :

```
var focusListener:Object = new Object ();
focusListener.onSetFocus = function (oldFocus, newFocus) {
    // Call activateTextField:
    fscommand ("activateTextField", "");
};
TextField1.addListener (focusListener);
```

Vous pouvez aussi utiliser le gestionnaire `TextField.prototype.onSetFocus` pour activer tous les champs texte lorsqu'ils reçoivent le focus.

Il est également possible d'utiliser une touche autre que la touche de sélection du périphérique pour exécuter la commande `activateTextField`. Le code suivant active un champ texte pour toutes les touches numériques, ce qui facilite, par exemple, la saisie de la lettre « a ». Par exemple, si l'application inclut `activateTextField` dans le gestionnaire `TextField.onSetFocus`, l'utilisateur doit appuyer sur la touche de sélection, puis sur la touche 2. Le code ci-dessous permet à l'utilisateur d'appuyer deux fois sur 2, ce qui est plus intuitif.

```
var keyListener:Object = new Object();
keyListener.onKeyDown = function() {
    if (Key.getCode() >= 48 && Key.getCode() <= 57 ){
        fscommand("activateTextField", "");
    }
};
Selection.addListener (keyListener);
```

Pour obtenir un exemple d'application finale utilisant cette technique, consultez l'exemple sur la saisie de texte en ligne à l'adresse [www.adobe.com/go/learn\\_flt\\_samples\\_and\\_tutorials\\_fr](http://www.adobe.com/go/learn_flt_samples_and_tutorials_fr). Repérez et téléchargez le fichier ZIP correspondant à votre version d'ActionScript, puis décompressez-le. Recherchez ensuite le dossier Samples et ouvrez-le pour afficher l'exemple qui vous intéresse.

## Indicateur du mode de saisie

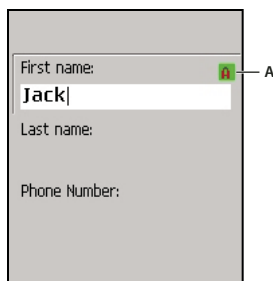
Afin de faciliter les opérations de saisie les plus communes, la plupart des périphériques disposent de plusieurs modes de saisie, tels que la saisie intuitive ou manuelle de texte (*triple appui* ou *appuis multiples*) ou les modes de saisie numérique.

Lorsque Flash Lite s'exécute en mode plein écran, le périphérique affiche la lettre A pour le mode de saisie de texte et (#) pour le mode de saisie numérique. Si Flash Lite ne s'exécute pas en mode plein écran, le périphérique affiche parfois un indicateur de saisie dans une barre d'état ou tout autre emplacement à l'écran. Par exemple, l'image suivante affiche l'indicateur du mode de saisie en mode autre que le mode plein écran avec la version autonome de Flash Lite 2.1 sur la série 60 :



A. Indicateur de mode de saisie, mode autre que le mode plein écran B. Champ texte en cours de modification

Lorsque le lecteur s'exécute en mode plein écran, le périphérique peut afficher un indicateur de mode de saisie à l'emplacement de son choix sur l'écran. Par exemple, l'illustration suivante affiche l'indicateur du mode de saisie en mode plein écran avec la version autonome de Flash Lite 2.1 sur la série 60 :



A. Indicateur de mode de saisie, mode plein écran

L'indicateur du mode de saisie pour les applications plein écran illustré plus haut sert uniquement d'exemple pour le lecteur autonome de la série 60. La forme exacte de l'indicateur, lorsque ce dernier est présent, est déterminée par le périphérique.

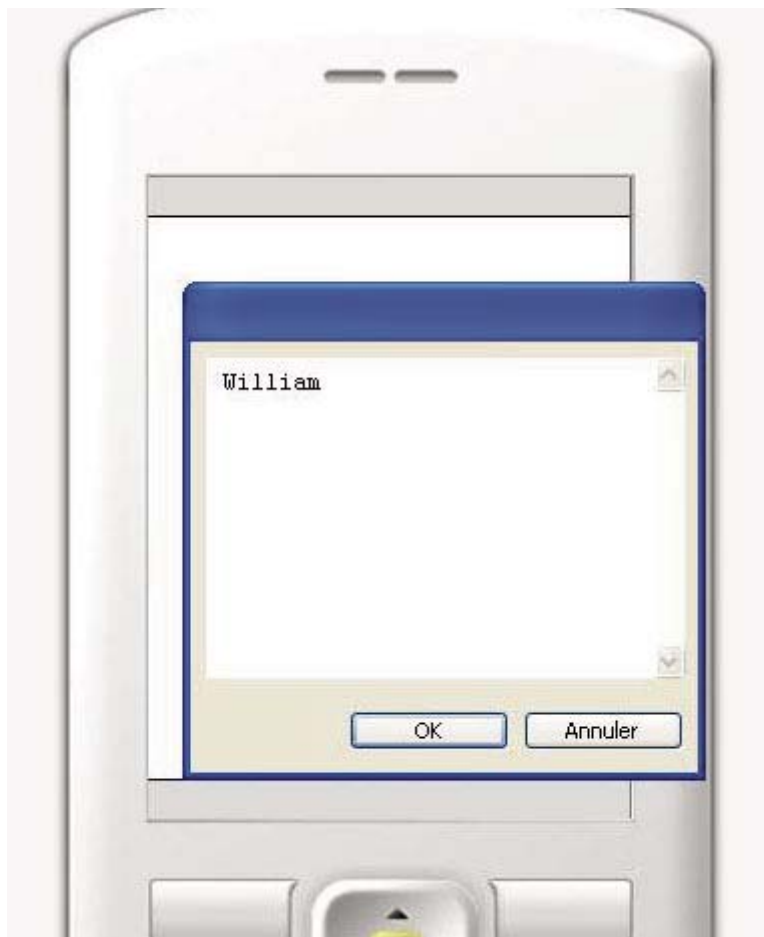
## A propos de la boîte de dialogue de saisie du périphérique (Flash Lite 2.0)

Pour ouvrir la boîte de dialogue de saisie du périphérique, l'utilisateur doit d'abord donner le focus à un champ texte de saisie, puis appuyer sur la touche de sélection.

La boîte de dialogue de saisie de texte est modale, c'est-à-dire que l'utilisateur ne peut pas interagir avec le contenu Flash lorsqu'elle a le focus. Flash Lite met également la tête de lecture en pause dans l'application Flash lorsque la boîte de dialogue a le focus.

Si l'utilisateur sélectionne OK (touche programmable Gauche), la boîte de dialogue de saisie se ferme et Flash Lite affecte automatiquement le texte au champ de saisie. Si l'utilisateur clique sur Annuler (touche programmable Droite), aucun texte n'est affecté au champ de saisie.

L'émulateur Adobe Device Central reproduit les fonctionnalités de la boîte de dialogue de saisie de texte lorsque vous testez votre application dans l'outil de programmation de Flash. L'image suivante affiche la boîte de dialogue de saisie de texte de l'émulateur :

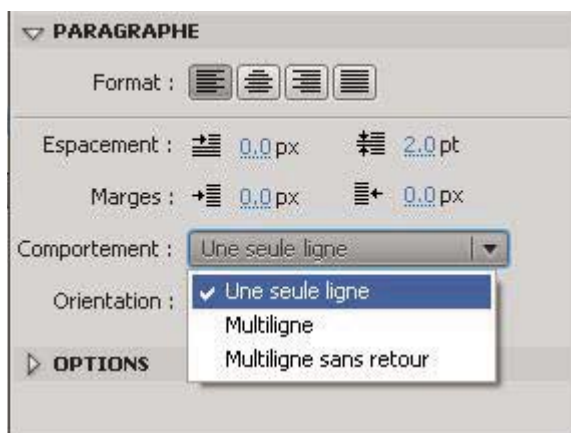


*Boîte de dialogue de saisie de texte fonctionnant dans l'émulateur*

Pour obtenir un exemple d'utilisation d'un champ texte de saisie dans une application, consultez la section « [Création d'un exemple d'application de champ texte](#) » à la page 61.

## Spécification des types de champs texte de saisie

Flash Lite prend en charge les champs texte de saisie sur une seule ligne, sur plusieurs lignes et de mot de passe. Pour spécifier un type de champ texte de saisie, utilisez le menu contextuel Type de ligne dans l'inspecteur Propriétés, comme indiqué ci-dessous :



Le type de ligne spécifié pour un champ texte de saisie détermine le comportement de la boîte de dialogue de saisie du périphérique lorsque l'utilisateur modifie le contenu du champ texte.

Par exemple, lorsque l'utilisateur modifie un champ texte de saisie sur une seule ligne, la boîte de dialogue de saisie du périphérique affiche une zone de texte de saisie sur une seule ligne. La zone de saisie défile horizontalement si l'utilisateur saisit plus de caractères qu'il ne peut en être affichés.

## Limitation des caractères saisis

La commande `SetInputTextType` permet de limiter les caractères que l'utilisateur peut entrer dans la boîte de dialogue de saisie. Ainsi, supposons qu'une application contienne un champ texte de saisie pour que les utilisateurs entrent une valeur numérique (par ex. leur âge) et que le nom de variable associé au champ soit `ageVar`. Afin de garantir que l'utilisateur ne saisisse que des valeurs numériques dans la boîte de dialogue de saisie, vous pouvez ajouter le code suivant à votre application :

```
fscommand2("SetInputTextType", "ageVar", "Numeric");
```

Si l'utilisateur ouvre la boîte de dialogue de saisie, il ne peut taper que des valeurs numériques dans les champs texte.

Pour plus d'informations, consultez la section `SetInputTextType` de l'entrée Fonction `fscommand2` du *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*.



## Boîtes de dialogue de saisie du texte (Flash Lite 2.0)

Dans Flash Lite 2.0, l'utilisateur saisit le texte grâce à une boîte de dialogue modale distincte (contrôlée par l'application hôte et non par Flash Lite) sans interagir directement avec le champ texte de saisie à l'écran. Au cours de cette interaction, le lecteur Flash Lite reste en suspens jusqu'à ce que l'utilisateur quitte la boîte de dialogue.

(Dans les versions 2.1 ou ultérieures de Flash Lite, l'utilisateur peut entrer du texte directement dans le champ texte de saisie qui s'affiche à l'écran.)

L'illustration suivante montre la boîte de dialogue de saisie de texte d'un périphérique pour un champ texte de saisie sur une seule ligne dans une application Flash Lite 2.0 :



Lorsque l'utilisateur modifie un champ texte de saisie sur plusieurs lignes, la boîte de dialogue de saisie du périphérique s'agrandit pour afficher la totalité du texte saisi, comme indiqué ci-dessous :



Lorsque l'utilisateur modifie le contenu d'un champ de saisie de mot de passe, la boîte de dialogue de saisie de texte du périphérique affiche tous les caractères saisis. Lorsque l'utilisateur clique sur OK, le mot de passe est entièrement remplacé par des astérisques, comme l'illustre l'image ci-dessous :

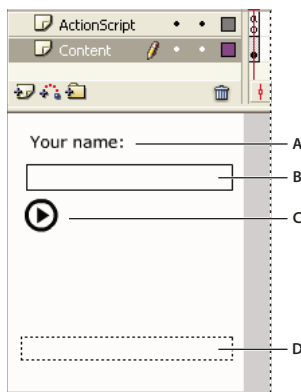


## Création d'un exemple d'application de champ texte

Cette application simple applique un formatage au texte saisi par l'utilisateur, puis affiche le résultat dans un champ texte dynamique compatible avec le code HTML. L'application a également recours à la commande `SetFocusRectColor` pour faire passer la couleur du rectangle de focus du jaune (paramètre par défaut) au noir.

Pour obtenir un exemple d'application finale (textfield\_example fla) utilisant cette technique, consultez la page [www.adobe.com/go/learn\\_flt\\_samples\\_and\\_tutorials\\_fr](http://www.adobe.com/go/learn_flt_samples_and_tutorials_fr). Repérez et téléchargez le fichier ZIP correspondant à votre version d'ActionScript, puis décompressez-le. Recherchez ensuite le dossier Samples et ouvrez-le pour afficher l'exemple qui vous intéresse.

- 1 Dans Flash, créez un document mobile et enregistrez-le sous textfield fla.
- 2 Avec l'outil Texte du panneau Outils, créez un champ texte sur une ligne en haut de la scène.
- 3 Assurez-vous que ce champ est sélectionné puis, dans l'inspecteur des propriétés, sélectionnez successivement Texte de saisie dans le menu contextuel Type de texte, Utiliser les polices de périphérique dans le menu contextuel Méthode de rendu des polices, puis tapez **inputTxt** dans la zone de texte Nom d'occurrence.
- 4 Créez un autre champ texte sous le premier et assurez-vous qu'il est beaucoup plus grand que le premier, comme indiqué ci-dessous :



- 5 Assurez-vous que le deuxième champ texte est sélectionné puis, dans l'inspecteur des propriétés, sélectionnez successivement Texte dynamique dans le menu contextuel Type de texte, Multiligne dans le menu contextuel Type de ligne et l'option Rendre le texte au format HTML ; dans le menu contextuel Méthode de rendu des polices, sélectionnez Utiliser les polices de périphérique, puis tapez **messageTxt** dans la zone de texte Nom d'occurrence.
- 6 Dans le scénario, sélectionnez l'image 1 dans le calque 1.
- 7 Ouvrez le panneau Actions (Fenêtre > Actions), puis tapez le code indiqué ci-dessous :

```
Selection.setFocus(inputTxt);
fscommand2("SetFocusRectColor", 0, 0, 0);
inputTxt.onChanged = function() {
    messageTxt.htmlText = "You entered: <i>" + this.text + "</i>";
}
```

La méthode `Selection.setFocus()` définit le focus initial sur le champ texte de saisie (`inputTxt`). Ensuite, l'appel de la fonction `fscommand2()` spécifie une couleur personnalisée du rectangle de focus. Enfin, le gestionnaire d'événements du champ texte de saisie `onChanged` (appelé lorsque le contenu du champ texte de saisie change) met en forme et affiche le texte que l'utilisateur a entré dans le champ texte `messageTxt`.

- 8 Enregistrez vos modifications et démarrez l'application dans l'émulateur (Contrôle > Tester l'animation).
- 9 Pour utiliser l'application, appuyez sur la touche de sélection de l'émulateur afin d'ouvrir la boîte de dialogue de saisie de texte, puis tapez le texte avec le clavier de l'ordinateur. Cliquez ensuite sur OK pour fermer la boîte de dialogue. Le texte que vous entrez s'affiche dans le champ texte `messageTxt` en italique.

# Rendu des polices

## Méthodes de rendu des polices dans Flash Lite

Flash Lite permet de rendre des polices de champ texte de l'une des façons suivantes :

**Utiliser les polices disponibles sur le périphérique** Vous pouvez appliquer à un champ texte une police que vous savez disponible sur le périphérique, ou bien spécifier l'une des trois polices de périphérique génériques (`_sans`, `_serif`, ou `_typewriter`) qui sont disponibles dans le menu contextuel Police. Si vous sélectionnez une police de périphérique générique, Flash Lite tente d'adapter la police générique sélectionnée à une police du périphérique (par exemple, `_sans` est affectée à une police sans serif, le cas échéant).

**Rendre la police sous forme de bitmap** Flash Lite rend du texte bitmap en alignant les polices vectorielles sur les limites des pixels, ce qui facilite la lisibilité des petits caractères (10 points ou moins). Cette option exige d'inclure les polices vectorielles dans le fichier SWF publié pour la police sélectionnée. (Consultez la section « [Intégration des polices vectorielles dans les fichiers SWF](#) » à la page 65.)

**Rendre la police sous forme de vecteurs anti-aliasés** Flash Lite rend du texte anti-aliasé en utilisant des représentations vectorielles des polices vectorielles intégrées au fichier SWF publié. (Consultez la section « [Intégration des polices vectorielles dans les fichiers SWF](#) » à la page 65.)

Le menu contextuel Méthode de rendu des polices dans l'inspecteur Propriétés permet de sélectionner une méthode de rendu des polices pour un champ texte. Ce menu contextuel contient cinq options de rendu ; cependant, trois seulement sont disponibles pour les développeurs Flash Lite. Les deux autres méthodes (Anti-alias pour la lisibilité et Anti-alias personnalisé) sont disponibles uniquement pour les applications ciblant Flash Player 8 ou les versions ultérieures sur les ordinateurs de bureau.

Vous trouverez ci-dessous des consignes permettant de déterminer quand utiliser du texte anti-aliasé, bitmap ou de périphérique :

- Si vous utilisez des polices intégrées avec des champs texte de saisie ou dynamique, intégrez uniquement les vecteurs des caractères à afficher. Ceci permet de réduire la taille du fichier. Par exemple, si vous utilisez un champ texte de saisie pour consigner l'âge d'un utilisateur (un nombre), incluez uniquement les vecteurs des caractères numériques (0 à 9). Dans ce cas, envisagez de restreindre la saisie des caractères à des nombres.
- L'émulateur Adobe Device Central ne permet pas d'émuler les polices de périphérique, à moins que celles-ci ne soient installées sur l'ordinateur servant au développement du contenu. Par conséquent, la mise en forme et l'aspect du champ texte du périphérique risque de différer sur l'émulateur et le périphérique.
- Si vous appliquez l'un des types de police génériques (`_sans`, `_serif` ou `_typewriter`), Flash Lite tente de trouver une police similaire sur le périphérique pour afficher le texte. Cependant, dans la mesure où les périphériques mobiles disposent de moins de polices et de styles que les ordinateurs de bureau, une police telle que `_sans` risque de ne pas être associée à une police sans serif. Vous devez tester l'application sur les périphériques cible pour déterminer les paramètres applicables.

Le texte avec anti-aliasing de Flash Lite est, avant tout, une forme vectorielle complexe. Comme toute autre forme vectorielle, son rendement nécessite davantage de puissance de traitement. Dans la mesure où la vitesse de traitement de la plupart des périphériques est relativement lente, l'animation de textes longs avec anti-aliasing risque de réduire les performances. Pour améliorer les performances, tentez de réduire temporairement la qualité de rendu du lecteur Flash Lite au cours de l'animation, puis rétablissez la qualité optimale à l'issue de l'animation.

## Sélection d'une méthode de rendu des polices pour un champ texte

- 1 Sélectionnez un champ texte sur la scène.

- 2 Dans l'inspecteur Propriétés, sélectionnez l'une des options suivantes dans le menu déroulant Méthode de rendu des polices.



- Sélectionnez Utiliser les polices de périphérique pour que Flash Lite utilise une police disponible sur le périphérique. Aucune donnée de police n'est intégrée au fichier SWF publié.
- Sélectionnez Texte bitmap (sans anti-alias) pour que Flash Lite aligne les polices vectorielles sur les limites des pixels, ce qui permet un affichage clair et net des petits caractères. Cette option nécessite que Flash intègre les polices vectorielles au fichier SWF publié. (Consultez la section « [Intégration des polices vectorielles dans les fichiers SWF](#) » à la page 65.)
- Sélectionnez Anti-alias pour l'animation afin que Flash Lite procède au lissage de la police du champ texte en fonction du paramètre de qualité de rendu actuel (consultez la section « [Qualité de rendu et texte anti-aliasé de Flash Lite](#) » à la page 64). Cette option nécessite que Flash intègre les polices vectorielles au fichier SWF publié.

## Qualité de rendu et texte anti-aliasé de Flash Lite

Utilisez la propriété `MovieClip._quality` pour contrôler la méthode de rendu utilisée par Flash Lite pour les graphiques vectoriels et les graphiques en mode point. Les valeurs possibles sont « LOW », « MEDIUM », « HIGH », et « BEST ». La valeur par défaut est « HIGH ». Vous pouvez indiquer cette propriété pour chaque objet de clip. Pour indiquer globalement cette propriété, placez le code `_quality = "lowMediumHighOrBest"` sur l'image 1.

Flash Lite rend le texte anti-aliasé en utilisant des représentations vectorielles des polices vectorielles. Si vous désirez anticrêneler le texte pour qu'il apparaisse aussi lisse que possible, conservez la qualité de rendu sur « HIGH ». La figure suivante montre les qualités de rendu haute, moyenne et faible pour un champ texte anti-aliasé (Arial, 24 points) :

*H i g h  
q u a l i t y  
s e t t i n g*

Pour optimiser la cadence et les effets animés, par exemple dans le cas d'une animation ou d'une séquence d'interpolation gourmande en ressources, vous pouvez définir de façon temporaire la qualité de rendu sur une valeur plus faible et revenir au paramètre précédent une fois que l'animation est terminée.

Pour régler la qualité du texte de saisie, appelez la commande `fscommand2("SetInputTextType")`. Pour de plus amples informations, consultez la section se rapportant à cette commande du *Guide de référence du langage ActionScript Flash Lite 2.x and 3.x*.

## Intégration des polices vectorielles dans les fichiers SWF

Pour rendre une police de champ texte, Flash Lite peut utiliser les polices disponibles sur le périphérique ou les polices vectorielles intégrées au fichier SWF publié (consultez la section « [Méthodes de rendu des polices dans Flash Lite](#) » à la page 63). L'intégration des polices vectorielles dans le fichier SWF permet de garantir un affichage identique de la police du champ texte sur toutes les plates-formes cibles, mais elle implique une taille de fichier plus volumineuse. Flash Lite oblige les polices vectorielles à rendre soit du texte bitmap (sans anti-alias) soit du texte anti-aliasé.

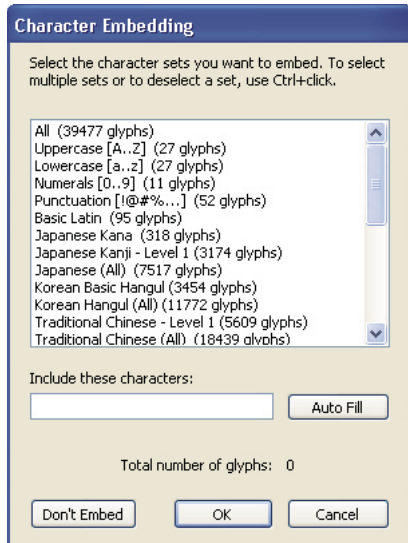
Pour les champs texte statique qui utilisent les méthodes de rendu des polices anti-alias ou bitmap, Flash intègre automatiquement les polices vectorielles requises pour afficher le contenu du champ texte. Par exemple, si un champ texte statique contient le mot *Envoyer*, Flash intègre automatiquement les polices vectorielles requises pour afficher ces sept caractères (E, n, v, o, y, e et r). Le contenu d'un champ texte statique ne pouvant pas être modifié, le fichier SWF n'a pas à inclure d'autres polices vectorielles que celles de ces caractères.

Pour les champs texte dynamique et de saisie qui utilisent les méthodes de rendu des polices anti-alias ou bitmap, vous devez spécifier les caractères pour lesquelles les polices vectorielles doivent être intégrées au fichier SWF publié. Le contenu de ces types de champs texte peut varier lors de la lecture ; par conséquent, Flash ne peut pas prévoir les polices vectorielles qui devront être disponibles. Vous pouvez inclure des polices vectorielles pour tous les caractères de la police sélectionnée, une plage de caractères ou bien des caractères spécifiques. Vous pouvez utiliser la boîte de dialogue Intégration de caractères pour spécifier les caractères à intégrer au fichier SWF publié.

## Intégration des polices vectorielles dans un champ texte dynamique ou de saisie

- 1 Sélectionnez le champ texte dynamique ou de saisie sur la scène.
- 2 Dans l'inspecteur des propriétés, sélectionnez Bitmap (sans anti-aliasing) ou Anti-alias pour l'animation dans le menu contextuel Méthode de rendu des polices.

- 3 Cliquez sur le bouton Intégrer qui se trouve en regard du menu de la méthode de rendu des polices pour ouvrir la boîte de dialogue Intégration de caractères.



- 4 Sélectionnez les caractères à intégrer dans la liste, tapez-les dans la zone de texte ou bien cliquez sur Remplissage automatique pour inclure les caractères présents dans le champ texte sélectionné.
- 5 Cliquez sur OK.

## Défilement du texte

### Création de texte défilant

Flash Lite prend en charge les propriétés `TextField.scroll` et `TextField.maxscroll`, ce qui permet de créer des champs texte défilants. La propriété `scroll` désigne la première ligne visible d'un bloc de texte ; vous pouvez lire et définir sa valeur. Par exemple, le code suivant fait défiler de cinq lignes vers le bas le champ texte dont le nom de variable est `story_text` :

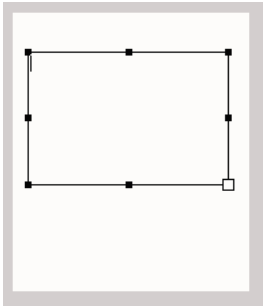
```
story_text.scroll += 5;
```

La propriété `maxscroll` spécifie la première ligne visible d'un bloc de texte lorsque la dernière ligne de ce bloc est visible ; cette propriété est en lecture seule. Vous pouvez comparer la propriété `maxscroll` d'un champ texte à sa propriété `scroll` pour déterminer le niveau de défilement atteint par l'utilisateur dans un champ texte. Cela peut s'avérer utile si vous souhaitez créer une barre de défilement fournissant un compte-rendu sur la position de défilement actuelle de l'utilisateur par rapport à la position de défilement maximale.

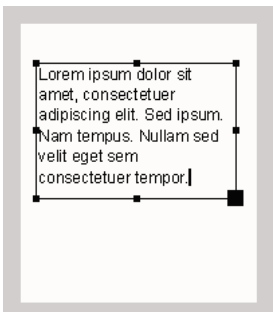
### Création d'un champ texte défilant et contrôle à l'aide du code ActionScript

- 1 Dans Flash, créez un document mobile.

- 2 En utilisant l'outil Texte, faites glisser un champ texte d'environ la taille indiquée dans l'illustration suivante sur la scène :



- 3 Dans l'inspecteur Propriétés, sélectionnez Multiligne dans le menu contextuel Type de ligne.  
4 Dans l'inspecteur Propriétés, choisissez Texte dynamique dans le menu contextuel Type de texte.  
5 Sélectionnez Utiliser les polices de périphérique dans le menu déroulant Méthode de rendu des polices dans l'inspecteur Propriétés.  
6 Sélectionnez Texte > Défilant pour faire défiler le champ texte.  
7 Dans l'inspecteur des propriétés, tapez **story** dans la zone de texte Nom d'occurrence.  
8 Double-cliquez dans le champ texte et tapez un texte suffisamment long pour qu'une ou plusieurs lignes dépassent de sa bordure inférieure.



- 9 Dans le scénario, sélectionnez la première image dans le calque 1 et ouvrez le panneau Actions  
10 (Fenêtre > Actions). Entrez le code suivant dans le panneau Actions :

```
var keyListener:Object = new Object();
keyListener.onKeyDown = function() {
    var keyCode = Key.getCode();
    if (keyCode == Key.DOWN) {
        story.scroll++;
    }
    else if (keyCode == Key.UP) {
        story.scroll--;
    }
};
Key.addListener(keyListener);
```

- 11 Sélectionnez Contrôle > Tester l'animation pour vérifier l'application dans l'émulateur Adobe Device Central.  
Appuyez sur les touches de navigation Haut et Bas de l'émulateur (ou sur les flèches Haut et Bas du clavier de votre ordinateur) pour faire défiler le texte vers le haut ou le bas.

# Chapitre 6 : Utilisation du son

Flash Lite prend en charge deux types de sons : son Flash standard (*natif*) et son de périphérique. Les sons natifs sont lus directement par le lecteur Flash Lite, tout comme dans la version de bureau d'Adobe Flash Player. Les sons de périphérique, par opposition, sont joués directement par le périphérique, et non par le lecteur Flash Lite. Le format MIDI ou MFi fait partie des sons de périphérique, mais les formats de son pris en charge varient de périphérique à périphérique.

## Son de périphérique

### A propos des sons de périphérique

Les *sons de périphérique* sont joués directement par le périphérique et non par le lecteur Flash Lite. Différents périphériques prennent en charge des formats de son différents, ce qui inclut MIDI, MFi ou MP3. Pour incorporer les sons de périphérique dans votre contenu Flash Lite, vous pouvez les inclure à votre fichier SWF publié ou charger des fichiers audio externes par l'intermédiaire du réseau ou du système de fichiers local du périphérique. Hormis quelques exceptions, utilisez l'objet Sound d'ActionScript pour contrôler les sons de périphérique, tout comme dans Flash Player, la version de bureau.

### Utilisation des regroupements de sons de périphérique

Pour regrouper des sons de périphérique dans votre application, importez au préalable un fichier audio de proxy dont le format est reconnu par l'outil de programmation de Flash, tel qu'un fichier MP3, WAV ou AIFF. Liez ensuite le fichier audio de proxy au fichier son de périphérique que vous souhaitez regrouper dans votre application. Pendant le processus de publication du fichier SWF, l'outil de programmation de Flash remplace le fichier audio de proxy par le son externe lié. Au cours de la diffusion, Flash Lite transmet les données audio au périphérique qui se charge alors du décodage et de la lecture.

Vous pouvez également regrouper différents sons de périphérique, de formats différents, dans un fichier FLS Flash unique. Cette opération s'avère utile si vous créez le même contenu pour plusieurs périphériques prenant en charge différents formats audio de périphérique. Pour plus d'informations, consultez la section « [Création d'un regroupement audio](#) » à la page 72.

### Importation et lecture d'un son de périphérique

Pour lire un son de périphérique, vous pouvez l'associer au scénario ou utiliser l'objet Sound pour lire ce son avec ActionScript.

- 1 Dans Flash, créez un document mobile et enregistrez-le sous **device\_sound fla**.
- 2 Sélectionnez Fichier > Importer > Importer dans la bibliothèque. Dans la page regroupant didacticiels et exemples à l'adresse [www.adobe.com/go/learn\\_ftl\\_samples\\_and\\_tutorials](http://www.adobe.com/go/learn_ftl_samples_and_tutorials), repérez, téléchargez et décompressez le fichier .zip correspondant à votre version de Flash Lite, puis naviguez jusqu'au dossier Samples.
- 3 Sélectionnez le fichier proxy.wav et cliquez sur OK.



**Remarque :** Vous pouvez utiliser n'importe quel fichier audio reconnu par l'outil de programmation Flash en tant que fichier audio de proxy. Le fichier *proxy.wav* est fourni pour vous faciliter la tâche.

- 1 Pour lier le fichier audio de proxy au fichier du son de périphérique, procédez comme suit :
  - a Dans le panneau Bibliothèque, cliquez avec le bouton droit de la souris (Windows) ou cliquez en maintenant la touche Contrôle enfoncée (Macintosh) sur le symbole de fichier audio de proxy et sélectionnez Propriétés dans le menu contextuel pour ouvrir la boîte de dialogue Paramètres audio.
  - b Dans la boîte de dialogue Paramètres audio, cliquez sur l'icône de dossier située à droite de la zone de texte Son de périphérique pour afficher la boîte de dialogue Sélectionner le son de périphérique.
  - c Consultez la page Web à l'adresse [http://www.adobe.com/go/learn\\_ftl\\_samples\\_and\\_tutorials\\_fr](http://www.adobe.com/go/learn_ftl_samples_and_tutorials_fr). Dans la page regroupant didacticiels et exemples, repérez, téléchargez et décompressez le fichier ZIP correspondant à votre version de Flash Lite, puis naviguez jusqu'au dossier Samples et sélectionnez le fichier nommé *hi\_hat.mid*.

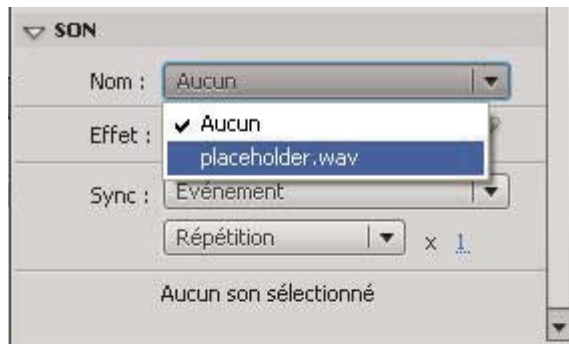


- d (Facultatif) Pour contrôler le son du périphérique avec ActionScript, cliquez sur Avancé pour afficher les paramètres audio avancés, sélectionnez Exporter pour ActionScript, et tapez **device\_sound** dans la zone de texte Identificateur.

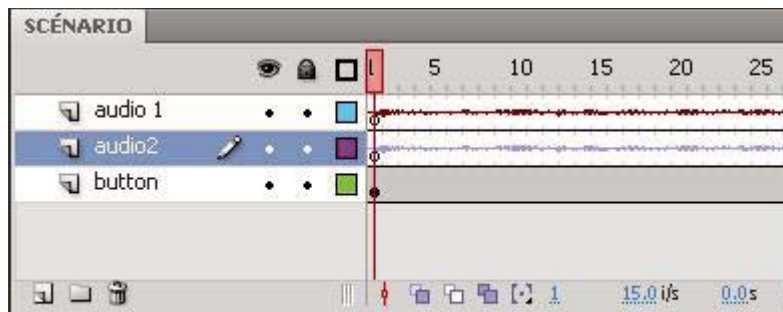
The screenshot shows the 'Paramètres audio' (Audio Settings) dialog box. It has two main sections: 'Liaison' (Link) and 'Partage' (Sharing). In the 'Liaison' section, there are two checked checkboxes: 'Exporter pour ActionScript' and 'Exporter dans l'image 1'. Below these, the 'Identifiant' (Identifier) field contains the text 'device\_sound'. There are also fields for 'Classe' (Class) and 'Classe de base' (Base class), both of which are currently empty. To the right of these fields are small checkmark and edit icons. The 'Partage' section is below the 'Liaison' section and contains two unchecked checkboxes: 'Exporter pour le partage à l'exécution' and 'Importer pour le partage à l'exécution'. Below these is a 'URL' field, which is also empty.

- e Cliquez sur OK pour fermer la boîte de dialogue Paramètres audio.
- Pour diffuser le son de périphérique, vous pouvez lier le fichier audio de proxy au scénario ou utiliser l'objet sound d'ActionScript. Pour utiliser l'objet sound d'ActionScript, passez à l'étape 6.
- 2 Pour associer le son de périphérique au scénario, procédez comme suit :
- a Sélectionnez l'image-clé sur l'image 1 du calque Content.
  - b Dans l'inspecteur Propriétés, sélectionnez proxy.wav dans le menu déroulant Son.

Cette action associe le son de proxy à l'image-clé.



L'outil de programmation Flash affiche la forme d'onde du fichier audio de proxy dans le scénario. Les formes d'onde des sons liés à des sons de périphérique externes sont de couleur verte ; les formes d'onde des sons qui ne sont pas liés à des sons de périphérique externes sont de couleur bleue, comme illustré ci-dessous.



*A. Son lié à un son de périphérique externe B. Son non lié à un son de périphérique externe*

- 3 Pour lire le son avec ActionScript, procédez comme suit :
- 4 Sélectionnez le calque Actions dans le scénario.
- 5 Ouvrez le panneau Actions (Fenêtre > Actions), puis tapez le code suivant :
 

```
var deviceSound:Sound = new Sound();
deviceSound.attachSound("device_sound");
deviceSound.start();
```
- 6 Sélectionnez Contrôle > Tester l'animation pour lancer l'émulateur Adobe Device Central et tester votre fichier SWF.

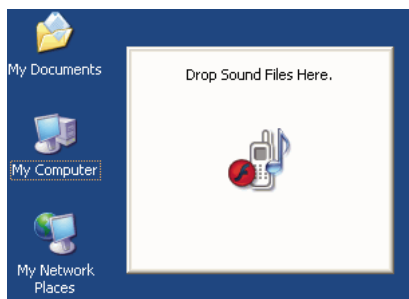
## Création d'un regroupement audio

Flash Lite permet d'encapsuler plusieurs sons de périphérique de différents formats dans un même *regroupement audio*. Par exemple, une application Flash simple peut contenir le même son représenté aux formats MIDI et MFi. Lorsque l'application est lue sur un périphérique qui prend en charge les fichiers audio MIDI, Flash Lite sélectionne les données audio MIDI dans le regroupement et les transmet au périphérique qui les lit. De même, si l'application est lue sur un périphérique qui prend uniquement en charge le format MFi, Flash Lite transmet les données audio MFi au périphérique.

L'utilitaire Système de regroupement audio de Flash Lite permet de créer un fichier de regroupement audio (FLS). Liez ensuite le fichier FLS à un fichier audio de proxy dans votre document Flash Lite, comme s'il s'agissait d'un son de périphérique unique. Pour plus d'informations sur l'ajout de sons de périphérique à vos applications Flash Lite, consultez la section « [Utilisation des regroupements de sons de périphérique](#) » à la page 68.

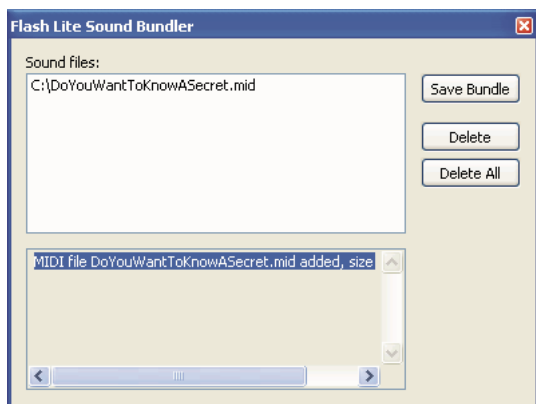
**Remarque :** Lors de la rédaction de ce document, l'utilitaire Système de regroupement audio n'est pris en charge que sur les systèmes Windows.

- 1 Recherchez et téléchargez le système de regroupement audio Flash Lite (FlashLiteBundler.exe) à l'adresse [www.adobe.com/go/developer\\_flashlite](http://www.adobe.com/go/developer_flashlite).
- 2 Le Système de regroupement audio s'affiche sous la forme d'une fenêtre flottante.



- 3 Faites glisser le premier fichier audio à regrouper du bureau vers la fenêtre Système de regroupement audio.

Une autre fenêtre s'affiche pour présenter le contenu du regroupement audio. La partie inférieure de la fenêtre contient des informations relatives aux sons du regroupement audio, y compris le format audio, la taille des données audio et le nom du fichier.



A. Informations relatives aux fichiers du regroupement audio B. Liste des fichiers du regroupement audio

- 4 Faites glisser les autres fichiers audio à regrouper dans la fenêtre.  
Vous ne pouvez pas regrouper plusieurs fichiers dans un format audio donné. Par exemple, vous ne pouvez pas regrouper deux fichiers MIDI dans un même fichier FLS.
- 5 Pour supprimer un fichier du groupe audio, sélectionnez-le dans la liste et cliquez sur Supprimer.  
Pour supprimer tous les fichiers du groupe audio, cliquez sur Tout supprimer.
- 6 Pour enregistrer le regroupement audio, cliquez sur Enregistrer le groupe, puis sélectionnez le nom et l'emplacement du fichier FLS.
- 7 Pour quitter Système de regroupement audio, cliquez sur le bouton de fermeture (X) dans la fenêtre Système de regroupement audio.

L'étape suivante consiste à ajouter le fichier de regroupement audio (FLS) à votre document Flash. La procédure est la même que l'ajout de sons de périphérique standard aux documents Flash, excepté qu'au lieu de spécifier un fichier de son de périphérique simple pour remplacer le fichier audio de proxy, vous devez spécifier le fichier FLS que vous avez créé. Pour plus d'informations, consultez la section « [A propos des sons de périphérique](#) » à la page 68.

## Lecture de sons de périphérique externes

Vous pouvez non seulement lire les sons de périphérique qui sont regroupés dans le fichier SWF publié (consultez la section « [Utilisation des regroupements de sons de périphérique](#) » à la page 68), mais aussi charger et lire des fichiers audio externes. Pour lire des sons de périphérique externes, appliquez la méthode `loadSound()` de l'objet `Sound`. Comme pour les sons de périphérique regroupés, le lecteur Flash Lite transmet les fichiers audio chargés en externe sur le périphérique pour les décoder et les lire.

Les informations suivantes, concernant la lecture de sons de périphérique externes dans Flash Lite, sont importantes :

- Contrairement à la version de bureau de Flash Player, les sons chargés en externe dans Flash Lite sont toujours traités en tant que sons d'événement. Cela signifie que les sons de périphérique externes ne sont pas diffusés en flux continu (lecture pendant le téléchargement). Les données audio doivent être téléchargées dans leur intégralité avant de lire le son. Pour les mêmes raisons, vous devez appeler la méthode `start()` de l'objet `Sound` pour lire le son à l'issue de son chargement (consultez l'exemple de code suivant).
- L'implémentation de la méthode `loadSound()` par Flash Lite ne prend pas en charge le deuxième paramètre de la méthode (*isStreaming*). Flash Lite ignore ce paramètre s'il est présent.
- Flash Lite ne lit pas les fichiers MP3 chargés en externe de façon native. Si votre application charge un fichier MP3 externe, Flash Lite transmet les données audio au périphérique pour décoder et lire le fichier, comme pour tout autre fichier audio téléchargé en externe.

Le code suivant indique comment charger et lire un fichier audio externe :

```
// Create the sound object.
var mySound:Sound = new Sound();
// Define onLoad handler for sound,
// which starts the sound once it has fully loaded.
mySound.onLoad = function(success){
    if(success == true) {
        mySound.start();
    }
}
// Load the sound.
mySound.loadSound("http://www.adobe.com/audio.midi");
```

## A propos de la synchronisation des sons de périphérique avec l'animation

Les sons de périphérique de Flash Lite sont toujours traités en tant que sons d'événement. Cela signifie que vous ne pouvez pas synchroniser les sons de périphérique avec l'animation dans le scénario de la même manière que pour les sons Flash natifs. Cependant, vous pouvez utiliser les sons de périphérique pour obtenir une approximation des sons véritablement synchronisés en définissant la propriété `_forceframerate` du lecteur Flash Lite sur `true`. Lorsque cette propriété est définie sur `true`, Flash Lite omet des images de l'animation pour préserver la cadence d'images du fichier SWF. Tant que les données du son de périphérique sont programmées pour la durée correcte, et tant que le périphérique lit les données de son à la cadence prévue, l'animation et le son restent proches d'une synchronisation totale.

Prenons l'exemple d'un son de périphérique de 5 secondes que vous souhaitez lire de façon synchronisée avec l'animation figurant dans le scénario. Supposons également que la cadence d'images de l'application soit définie sur 15 ips. Lorsque vous lancez la lecture du son, en l'associant à une image dans le scénario ou en appelant `Sound.start()`, vous définissez également `_forceframerate = "true"`. Par la suite, pour chaque seconde de lecture du périphérique audio, Flash Lite s'assure que la tête de lecture a avancé de 15 images dans le scénario. Si, pour quelque raison que ce soit, le lecteur ne peut pas rendre toutes les images de l'animation pendant cette période, il omet des images pour préserver la cadence.

Pour plus d'informations sur la propriété `_forceframerate`, consultez la rubrique consacrée à la propriété `_forceframerate` dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*.

## Identification des formats de son de périphérique pris en charge

La propriété `System.capabilities.audioMIMETypes` contient un tableau des types audio MIME pris en charge par le périphérique. Ces informations permettent de déterminer les types de son de périphérique pouvant être lus par votre application. Les index du tableau reprennent les types MIME pris en charge, ce qui permet de vérifier facilement si un périphérique prend en charge un format audio spécifique.

Par exemple, le code suivant vérifie d'abord que le périphérique prend en charge la lecture des données audio MIDI avant de charger un fichier MIDI externe :

```
if (System.capabilities.audioMIMETypes["audio/midi"]) {  
    my_sound.loadSound("soundtrack.mid");  
}
```

## Son natif

### A propos des sons Flash natifs

Outre les sons de périphérique, Flash Lite prend en charge les sons Flash standard ou *natifs*. De manière générale, un son natif est un son reconnu par l'outil de programmation de Flash, quel que soit son format. Les sons natifs de Flash Lite peuvent être lus en tant que sons d'événement ou sons synchronisés (consultez la section « [A propos des sons de périphérique](#) » à la page 68).

Le flux de travail général d'utilisation des sons natifs dans vos applications Flash Lite reprend celui des applications Flash de bureau, avec les exceptions suivantes :

- Flash Lite ne prend pas en charge de façon *native* la lecture de fichiers MP3 externes. Cependant, si le périphérique peut lire des fichiers MP3, vous pouvez charger ces fichiers dans votre application. Pour plus d'informations, consultez la section « [Lecture de sons de périphérique externes](#) » à la page 74.
- Flash Lite ne prend pas en charge l'option de compression audio Discours.

Pour plus d'informations sur l'utilisation du son natif dans Flash, consultez la section « Utilisation du son » dans le guide *Utilisation de Flash*.

## Utilisation de la fonctionnalité d'échantillonnage sur 8 kHz

Par défaut, l'outil de programmation Flash exporte les données audio natives en appliquant des taux d'échantillonnage de 5, 11, 22 ou 44 kilohertz (kHz). Cependant, de nombreux périphériques ne prennent pas en charge la lecture de données audio à ces taux. Par exemple, les périphériques de la gamme Series 60 de Nokia prennent uniquement en charge les taux de 8 et 16 kHz. Dans les versions précédentes de Flash Lite, le lecteur a échantillonné de nouveau les données audio sur 8 kHz lors de l'exécution, avant d'appeler les API du son natif pour le lire. Ce nouvel échantillonnage prend davantage de temps et consomme davantage de mémoire.

Dans Adobe Flash CS4 Professional, vous pouvez échantillonner de nouveau les données audio natives sur 8 kHz à l'aide de l'outil de programmation pendant le processus de publication du fichier SWF. Vous pouvez appliquer ce paramètre à l'ensemble des sons, d'événement ou en flux continu, de votre application ou le limiter aux sons figurant dans le panneau Bibliothèque de votre document.

La première procédure, ci-dessous, explique comment activer l'option d'échantillonnage sur 8 kHz pour des sons spécifiques, dans la boîte de dialogue Paramètres audio. La deuxième procédure indique comment utiliser la boîte de dialogue Paramètres de publication pour définir cette option de façon globale pour tous les sons d'événement et en flux continu.

**Remarque :** Pour les sons en flux continu, Flash applique systématiquement les options globales de compression audio qui ont été spécifiées à l'aide de la boîte de dialogue Paramètres de publication. Les options de compression que vous sélectionnez dans la boîte de dialogue Paramètres audio s'appliquent uniquement aux sons d'événement.

## Activation de la fonctionnalité d'échantillonnage sur 8 kHz pour un son spécifique

- 1 Dans Flash, cliquez avec le bouton droit de la souris (Windows) ou cliquez en maintenant la touche Contrôle enfoncée (Macintosh) sur un symbole de son dans le panneau Bibliothèque et sélectionnez Propriétés dans le menu contextuel.
- 2 Dans la boîte de dialogue Paramètres audio qui s'affiche, sélectionnez MP3 dans la liste déroulante Compression.  
La fonctionnalité d'échantillonnage sur 8 kHz est disponible uniquement pour les données audio compressées au format MP3.
- 3 Assurez-vous que l'option Utiliser la qualité mp3 importée est désactivée.
- 4 Sélectionnez l'option Utiliser le taux d'échantillonnage 8 kHz.





A. Utilisation de l'option de taux d'échantillonnage 8 kHz

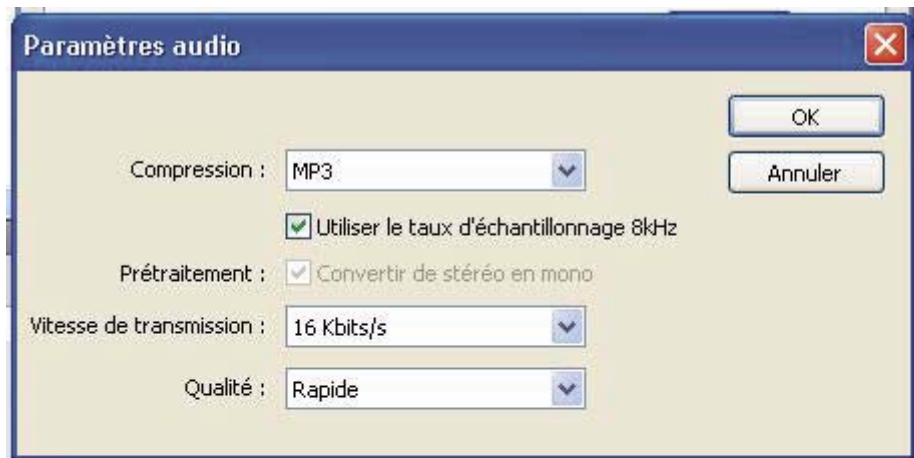
- 5 Cliquez sur OK.

## Activation de la fonctionnalité d'échantillonnage sur 8 kHz pour tous les sons natifs

- 1 Dans Flash, sélectionnez Fichier > Paramètres de publication.
- 2 Dans la boîte de dialogue Paramètres de publication, cliquez sur l'onglet Flash.
- 3 Pour activer l'échantillonnage sur 8 kHz pour l'ensemble des sons en flux continu de votre application, procédez comme suit :
  - a Cliquez sur le bouton Définir de l'option Flux continu.
  - b Dans la boîte de dialogue Paramètres audio qui s'affiche, sélectionnez MP3 dans la liste déroulante Compression.

La fonctionnalité d'échantillonnage sur 8 kHz est disponible uniquement pour les données audio compressées au format MP3.

- c Sélectionnez l'option Utiliser le taux d'échantillonnage 8 kHz.



- d Cliquez sur OK pour fermer la boîte de dialogue Paramètres de publication.
- 4 Pour activer l'échantillonnage sur 8 kHz pour l'ensemble des sons d'événement, procédez comme suit :
- a Cliquez sur le bouton Définir près de Son d'événement.
  - b Dans la boîte de dialogue Paramètres audio qui s'affiche, sélectionnez MP3 dans la liste déroulante Compression.

**Remarque :** La fonctionnalité d'échantillonnage sur 8 kHz est disponible uniquement pour les données audio compressées au format MP3.

- c Sélectionnez l'option Utiliser le taux d'échantillonnage 8 kHz.
  - d Cliquez sur OK.
- 5 Cliquez sur OK pour fermer la boîte de dialogue Paramètres de publication.

# Chapitre 7 : Utilisation de la vidéo

Vous pouvez inclure de la vidéo dans les applications Flash Lite 2.x et Flash Lite 3.x. Dans les applications Flash Lite 2.x, le périphérique mobile décode et donne un rendu de la vidéo. Ce type de vidéo s'appelle une *vidéo de périphérique*. Les applications Flash Lite 3.x prennent en charge deux types de vidéo : la vidéo de périphérique et les fichiers FLV (Flash Video). Lorsque vous lisez un fichier FLV dans une application, Flash Lite 3.x décode et donne un rendu de la vidéo.

## Travailler avec les fichiers FLV

### À propos de la prise en charge de fichiers FLV dans Flash Lite 3.0 et version ultérieure

Flash Lite 3.0 a ajouté la prise en charge de fichiers FLV (Flash Video) à l'aide des versions des codecs On2 et Sorenson optimisés pour les périphériques mobiles. Les fichiers FLV sont rendus par Flash Lite 3.x et non pas par le périphérique. Si un périphérique prend en charge Flash Lite 3.x, vous pouvez ajouter des fichiers FLV à votre application.

**Remarque :** *Les fabricants de périphériques peuvent décider de ne pas prendre en charge les fichiers FLV s'ils ont déjà mis en œuvre un mécanisme de lecture vidéo efficace. Utilisez alors la vidéo de périphérique dans vos applications. Vérifiez l'onglet Profils de périphériques dans Adobe Device Central pour vérifier si les fichiers FLV (Flash Video) sont pris en charge par un périphérique spécifique.*

Vous pouvez incorporer des fichiers FLV dans une application de l'une des façons suivantes :

- Intégrez des fichiers FLV dans l'application.
- Lisez des fichiers FLV externes à partir d'un disque dur local ou d'un serveur Web via HTTP.
- Lisez des fichiers FLV externes du Flash Media Server via RTMP.

Pour convertir des vidéos (tels que des fichiers vidéo QuickTime ou Windows Media) en fichiers FLV, utilisez l'une des méthodes suivantes :

- Utilisez un encodeur (tel qu'Adobe Flash Video Encoder ou Sorenson™ Squeeze).
- Importez le fichier dans l'outil de programmation de Flash et exportez-le en fichier FLV.
- Utilisez le module d'exportation FLV pour exporter des fichiers FLV à partir des applications de montage vidéo prises en charge.

Les fonctionnalités de vidéo suivantes ne sont pas disponibles dans Flash Lite 3.x :

- La classe Camera ou l'enregistrement de vidéos.
- Connexions RTMPT et RTMPS. Ces tentatives de connexion utilisent une connexion RTMP par défaut.
- La communication bidirectionnelle entre des clients. Les objets partagés distants ne sont pas pris en charge. Les clients ne peuvent pas partager des objets partagés distants.
- Vidéo avec canal alpha.
- Le composant FLVPlayback. Consultez la section « [A propos des composants de Flash Lite](#) » à la page 1 pour obtenir des informations sur les composants.

**Voir aussi**

« [Intégrer des fichiers FLV dans une application](#) » à la page 80

« [Lire des fichiers FLV externes dans une application](#) » à la page 80

**Intégrer des fichiers FLV dans une application**

Pour intégrer un fichier FLV, importez-le dans la bibliothèque d'un document. Lorsque vous publiez l'application, l'outil de programmation intègre le fichier FLV dans le fichier SWF publié.

Les fichiers FLV intégrés sont attachés au scénario. Par exemple, si un document Flash est réglé pour lire à 10 images par secondes, un fichier FLV de 10 secondes occupe le scénario de l'image 1 à l'image 100. Pour contrôler la lecture d'un fichier FLV intégré, utilisez du code Adobe ActionScript pour lancer, arrêter et déplacer la tête de lecture sur le scénario. Vous pouvez aussi donner un nom d'occurrence à l'objet vidéo intégré et le contrôler avec les propriétés de l'objet Video d'ActionScript.

Lorsque vous intégrez des fichiers FLV, vous pouvez les synchroniser avec d'autres éléments de votre document. Par exemple, vous pourriez ajouter des éléments interactifs liés à d'autres documents sur certaines images. L'intégration de fichiers FLV est recommandée si les fichiers sont courts et n'ont pas de piste audio. L'intégration des fichiers FLV peut augmenter significativement la taille du fichier SWF publié.

**Remarque :** Dans les applications Adobe Flash Lite 2.0, vous pouvez intégrer une vidéo de périphérie dans une application. Cette méthode s'appelle le « groupage de vidéo ». Consultez « [Importation d'une vidéo de périphérie dans une application](#) » à la page 83.

- 1 Dans Flash, sélectionnez Fichier > Importer > Importer de la vidéo.  
L'Assistant Importer de la vidéo s'affiche.
- 2 Sélectionnez l'option relative à l'importation à partir de l'ordinateur et cliquez sur Parcourir.
- 3 Naviguez vers le dossier qui contient le fichier FLV et sélectionnez ce dernier.  
Si le fichier vidéo n'est pas listé (ou si vous pouvez le voir, mais pas le sélectionner), sélectionnez Tous les fichiers (\*.\*) dans le menu déroulant (de Windows) Types de fichier, ou Tous les fichiers à partir du menu déroulant Activer (Macintosh).
- 4 Cliquez sur Ouvrir. Cliquez sur Suivant dans l'assistant Importation de vidéo.
- 5 Sur l'écran de déploiement, sélectionnez Intégration de la vidéo dans des fichiers SWF et lecture dans le scénario puis cliquez sur Suivant.
- 6 Sur l'écran d'intégration, sélectionnez Vidéo intégrée comme type de symbole.
- 7 Cliquez sur suivant pour vous rendre à l'écran Terminer l'importation de vidéo puis cliquez sur Terminer pour importer la vidéo.  
Un nouveau symbole de vidéo lié au fichier FLV apparaît dans le panneau de la Bibliothèque et sur la scène.
- 8 Sélectionnez l'objet vidéo sur la scène et donnez-lui un nom d'occurrence dans l'inspecteur Propriétés.  
Vous pouvez utiliser le nom de l'occurrence pour contrôler l'objet vidéo avec les propriétés de la classe Video d'ActionScript.

**Lire des fichiers FLV externes dans une application**

Un fichier externe FLV n'est pas intégré dans un fichier SWF. Un fichier externe FLV se trouve sur la carte mémoire d'un périphérique, sur un serveur Web distant ou sur un serveur Adobe Flash Media Server. Vous pouvez utiliser ActionScript pour lire et contrôler les fichiers FLV externes au moment de l'exécution dans Flash Lite 3.x.

Lire des fichiers FLV externes offre plusieurs avantages par rapport à l'intégration de vidéos dans un document Flash. Vous pouvez utiliser de long clips dans ralentir la lecture car, les fichiers FLV externes nécessitent moins de mémoire que les fichiers vidéo intégrés. Les fichiers FLV externes peuvent avoir une vitesse de défilement différente du document Flash dans lequel ils sont lus. Par exemple, vous pouvez régler la vitesse de défilement du document Flash sur 30 images par secondes et celle de la vidéo sur 21 images par seconde. Ce contrôle garantit une lecture vidéo fluide. Ce contrôle permet également de lire des fichiers FLV à diverses vitesses de défilement sans avoir à modifier le contenu média enrichi existant.

Affichage des fichiers FLV externes dans les objets Video. Vous pouvez manipuler les objets Video de façon similaire à la manipulation des objets MovieClip. Vous pouvez faire pivoter et changer les dimensions de la vidéo. Vous pouvez régler la valeur de transparence alpha. Vous pouvez procéder à une composition ou au fondu des vidéos avec d'autres contenus média enrichis. Vous pouvez aussi utiliser la classe Sound pour contrôler le son de la vidéo.

- 1 Dans la bibliothèque de Flash, sélectionnez Nouvelle vidéo pour créer un symbole vidéo. Assurez-vous que le type de vidéo est « Vidéo (contrôlée par ActionScript) » puis cliquez sur OK.
- 2 Faites glisser le symbole vidéo sur la scène et donnez-lui un nom d'occurrence. Cette exemple utilise le nom d'occurrence **vo**.
- 3 Sélectionnez la première image dans le scénario puis ouvrez le panneau Actions.
- 4 Saisissez le code suivant :

```
// Create a NetConnection object
var nc:NetConnection = new NetConnection();
// Declare a NetStream object
var ns:NetStream;
/* When a change in connection status occurs,
the onStatus handler is called and passed an
information object with information about the change.
Use the information object to run code when
connection events occur.
*/
nc.onStatus = function(info) {
    // Output all connection status information
    trace(info.code);
    // Run the following code if the connection is successful.
    if(info.code == "NetConnection.Connect.Success"){
        // Create a NetStream object and pass it the NetConnection object
        ns = new NetStream(nc);
        // Attach the NetStream object to the video object on Stage.
        vo.attachVideo(ns);
        // Play the file "sample.flv". Do not include ".flv" when
        // playing the file on Flash Media Server.
        ns.play("sample");
    }
}
/* Call NetConnection.connect() to create a connection.
To connect to a local file or to a file on a web server, pass null.
To connect to Flash Media Server, pass the address to
an application on the server.
*/
nc.connect("rtmp://fmsexamples.adobe.com/vod");
```

**Remarque :** Pour charger des fichiers FLV à partir d'un serveur Web, enregistrez l'extension du nom de fichier et le type de MIME avec le serveur Web. Le type MIME des fichiers FLV est video/x-flv. Il est possible que le type de MIME soit déjà enregistré. Référez-vous à la documentation du serveur Web pour de plus amples informations.

## Contrôler les fichiers FLV avec ActionScript.

Utilisez ActionScript pour vous connecter, afficher, et contrôler des fichiers FLV externes. Utilisez la classe `NetConnection` pour établir une connexion. Utilisez la classe `Video` pour afficher la vidéo. Utilisez la classe `NetStream` pour contrôler la lecture.

Pour contrôler les fichiers FLV, appelez les méthodes de la classe `NetStream`, comme par exemple la méthode `NetStream.pause()`. Pour contrôler une vidéo de périphérique, appelez les méthodes de la classe `Video`. Par exemple, pour pauser la vidéo de périphérique, appelez la méthode `Video.pause()`.

Pour obtenir les descriptions détaillées de ces classes ainsi que des exemples de code illustrant leur utilisation, consultez le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*. En outre, Flash Lite 3.x prend en charge toutes les méthodes, propriétés et événements documentés dans le *Guide de référence du langage ActionScript 2.0*, excepté pour la propriété `MovieClipLoader.checkPolicyFile`.

## Travailler avec les vidéos de périphérique

### À propos des vidéos de périphérique

Flash Lite 2.0 permet de lire des *vidéos de périphérique*, c'est-à-dire tous les formats vidéo et d'encodage pris en charge par le périphérique cible. Les périphériques prennent en charge différents formats et codecs vidéo. Certains des formats vidéo de périphérique les plus fréquents sont 3GP, 3G2 (ou 3GPP2) et MPEG-4. Flash Lite peut lire n'importe quel format vidéo pris en charge par le périphérique cible.

Flash Lite n'effectue aucun décodage ou rendu de vidéo de périphérique. Il s'appuie sur le périphérique pour décoder et rendre les vidéos de périphérique. Pour cette raison, les vidéos de périphériques ont les limitations suivantes :

- Vous ne pouvez pas faire pivoter ou incliner les vidéos de périphérique. Certains périphériques prennent en charge la mise à l'échelle.
- Vous ne pouvez pas synchroniser les vidéos de périphérique avec le scénario.
- Vous ne pouvez pas procéder à une composition ou au fondu des vidéos de périphérique avec d'autres supports. Le périphérique affiche la vidéo au-dessus de tout autre contenu média enrichi.
- Vous ne pouvez pas contrôler le volume sonore d'un clip.

Pour déployer une vidéo de périphérique, groupez-la dans une application ou chargez-la à partir d'un fichier externe sur le périphérique ou à partir d'un emplacement sur un réseau.

### Voir aussi

« [Identification des formats de vidéo de périphérique pris en charge](#) » à la page 82

« [Importation d'une vidéo de périphérique dans une application](#) » à la page 83

« [Lecture d'une vidéo de périphérique externe dans une application](#) » à la page 91

## Identification des formats de vidéo de périphérique pris en charge

Différents périphériques prennent en charge différents formats et codages vidéo. Pour déterminer les formats pris en charge par un périphérique, effectuez l'une des opérations suivantes :

- Utilisez l'onglet Profils de périphérique d'Adobe Device Central. Pour de plus amples informations, consultez le fichier d'aide de Device Central.

- Reportez-vous aux spécifications du fabricant du périphérique.
- Utilisez la propriété `System.capabilities.videoMIMETypes` d'ActionScript.

La propriété `System.capabilities.audioMIMETypes` contient un tableau des types vidéo MIME pris en charge par le périphérique. Chaque élément du tableau a le format `video/video-type`. Par exemple, le code suivant affiche dans la fenêtre de sortie tous les types de vidéo MIME pris en charge par le périphérique :

```
var mimeTypes = System.capabilities.videoMIMETypes;
trace(mimeTypes.toString());
```

**Remarque :** Pour afficher la fenêtre de sortie dans Adobe Device Central, sélectionnez *Affichage > Sortie Flash > Afficher*.

Le code suivant vérifie si un périphérique prend en charge la vidéo 3GPP avant de lire un fichier vidéo de ce type :

```
if (System.capabilities.videoMIMETypes["video/3gpp"]) {
    my_video.play("movie.3gp");
}
```

## Importation d'une vidéo de périphérique dans une application

Pour grouper une vidéo de périphérique, importez-la dans la bibliothèque d'un document Flash. Lorsque vous publiez l'application, l'outil de programmation incorpore le fichier FLV au fichier SWF publié.

Pour importer un fichier vidéo dans un document Flash, utilisez l'assistant d'importation vidéo ou le panneau Bibliothèque. Les deux techniques ajoutent un symbole de vidéo à la bibliothèque du document.

### Utilisez l'assistant d'importation de vidéo

- 1 Dans Flash, sélectionnez *Fichier > Importer > Importer de la vidéo*.
- 2 Sélectionnez l'option relative à l'importation à partir de l'ordinateur et cliquez sur *Parcourir*.
- 3 Ouvrez le dossier contenant le fichier de vidéo de périphérique et sélectionnez-le.  
Si vous ne voyez pas le fichier vidéo voulu (ou si vous le voyez, mais ne pouvez pas le sélectionner), sélectionner Tous les fichiers (\*.\*) à partir du menu déroulant Types de fichier (sous Windows), ou Tous les fichiers à partir du menu déroulant Activer (sous Macintosh).
- 4 Cliquez sur *Ouvrir*.
- 5 Cliquez sur *Suivant* dans l'assistant d'importation vidéo.
- 6 Pour les applications Flash Lite 2.0, la seule option de déploiement est comme vidéo de périphérique mobile groupée dans SWF. Cliquez sur *Terminer*.  
Un symbole vidéo s'affiche dans le panneau Bibliothèque. Ce symbole est associé au fichier de vidéo de périphérique.
- 7 Faites glisser l'objet de vidéo de la Bibliothèque vers la scène et donnez-lui un nom d'occurrence dans l'inspecteur de propriétés.

### Utilisez le panneau Bibliothèque pour importer une vidéo de périphérique

- 1 Dans Flash, ouvrez le panneau Bibliothèque (*Fenêtre > Bibliothèque*).
- 2 Ouvrez le menu déroulant des options et choisissez *Nouvelle vidéo*.  
La boîte de dialogue Propriétés de la vidéo s'affiche.
- 3 Dans cette boîte de dialogue, sélectionnez l'option de regroupement de la vidéo source dans le fichier SWF, puis cliquez sur *Importer*.

- 4 Ouvrez le dossier contenant le fichier de vidéo de périphérique et sélectionnez-le.

Si vous ne voyez pas le fichier vidéo voulu (ou si vous le voyez, mais ne pouvez pas le sélectionner), sélectionner Tous les fichiers (\*.\*) à partir du menu déroulant Types de fichier (sous Windows), ou Tous les fichiers à partir du menu déroulant Activer (sous Macintosh).

- 5 Cliquez sur Ouvrir.

- 6 Dans la boîte de dialogue Propriétés de la vidéo, sélectionnez l'option Exporter pour ActionScript et saisissez une chaîne dans la boîte Identificateur.

- 7 Cliquez sur OK pour fermer la boîte de dialogue Propriétés de la vidéo.

Un symbole de vidéo apparaît dans le panneau Bibliothèque. Ce symbole est associé avec le fichier vidéo de périphérique.

## Lecture d'une vidéo de périphérique groupée dans une application

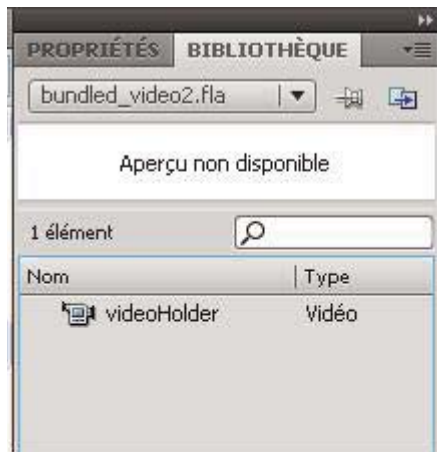
Pour créer cette application, faites glisser une vidéo sur la scène et ajoutez du code ActionScript aux boutons pour lire et arrêter la vidéo.

- 1 Dans Flash, sélectionnez Fichier > Nouveau > Fichier Flash (Mobile) puis cliquez sur OK.
- 2 Dans Adobe Device Central, cliquez sur Créer.
- 3 Dans Flash sélectionnez Fichier > Enregistrer puis enregistrez le fichier sous le nom bundled\_video fla.
- 4 Importez le fichier vidéo de périphérique ocean.3gp à partir de Flash\_Lite\_2x\Samples\Video\folder. Pour de plus amples informations sur l'importation de vidéos de périphérique, consultez la section « [Importation d'une vidéo de périphérique dans une application](#) » à la page 83.

**Remarque :** Sur la page regroupant didacticiels et exemples à l'adresse ([http://www.adobe.com/go/learn\\_flt\\_samples\\_and\\_tutorials\\_fr](http://www.adobe.com/go/learn_flt_samples_and_tutorials_fr)), téléchargez et décompressez le fichier ZIP correspondant à votre version de Flash Lite.

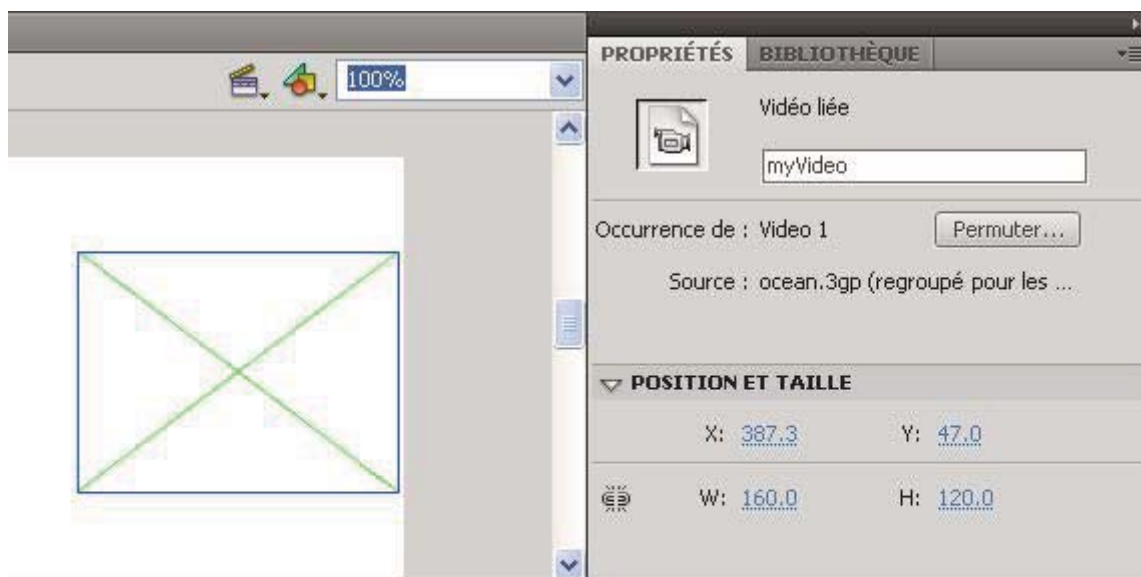


Un nouveau symbole vidéo s'affiche dans le panneau Bibliothèque.



- 5 Faites glisser une occurrence du symbole de vidéo sur la scène.
- 6 Sélectionnez l'objet vidéo sur la scène puis, dans l'inspecteur des propriétés, saisissez **myVideo** dans la zone de texte Nom de l'occurrence.

Utilisez ce nom pour vous reporter à l'objet vidéo dans le code ActionScript.



- 7 Réglez la largeur sur 176 et la hauteur sur 144 dans l'inspecteur de propriétés.  
Ces dimensions correspondent à la source de la vidéo. Selon le périphérique, les vidéos Flash Lite de périphérique ne se redimensionnent pas toujours pour occuper tout le cadre de délimitation.
- 8 Pour ajouter des boutons de contrôle de la vidéo, ouvrez la bibliothèque de boutons prédéfinis (Fenêtre > Bibliothèques communes > Boutons).
- 9 Dans la bibliothèque de boutons, cliquez deux fois sur le dossier Circle Buttons pour l'ouvrir.
- 10 Faites glisser sur la scène une occurrence du symbole de bouton de lecture.
- 11 Faites glisser sur la scène une occurrence du symbole de bouton d'arrêt.
- 12 Sélectionnez le bouton de lecture sur la scène et saisissez (ou copiez et collez) le code suivant dans le panneau Actions :
 

```
on (press) {
    myVideo.play();
}
```
- 13 Sélectionnez le bouton d'arrêt sur la scène et saisissez le code suivant dans le panneau Actions :
 

```
on (press) {
    myVideo.stop();
}
```
- 14 Sélectionnez Contrôle > Tester l'animation pour vérifier le film dans Device Central.

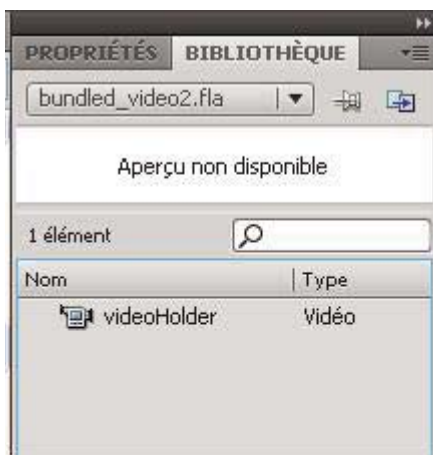
## Lecture d'une vidéo de périphérique groupée dans une application au moment de l'exécution

Pour créer cette application, faites glisser un marqueur de vidéo sur la scène. Ajoutez du code ActionScript aux boutons pour lire et arrêter la vidéo. Au moment de l'exécution, le bouton de lecture récupère la vidéo dynamiquement dans le panneau Bibliothèque. Cette application n'utilise qu'une seule vidéo. Une application peut avoir plusieurs fichiers vidéo dans le panneau Bibliothèque et les placer dans le marqueur de vidéo lors de l'exécution.

- 1 Dans Flash, sélectionnez Fichier > Nouveau > Fichier Flash (Mobile) puis cliquez sur OK.
- 2 Dans Adobe Device Central, cliquez sur Créer.
- 3 Dans Flash sélectionnez Fichier > Enregistrer puis enregistrez le fichier sous le nom `bundled_video fla`.
- 4 Importez le fichier vidéo de périphérique `ocean.3gp` à partir de `Flash_Lite_2x\Samples\Video\folder`. Pour de plus amples informations sur l'importation de vidéos de périphérique, consultez la section « [Importation d'une vidéo de périphérique dans une application](#) » à la page 83.

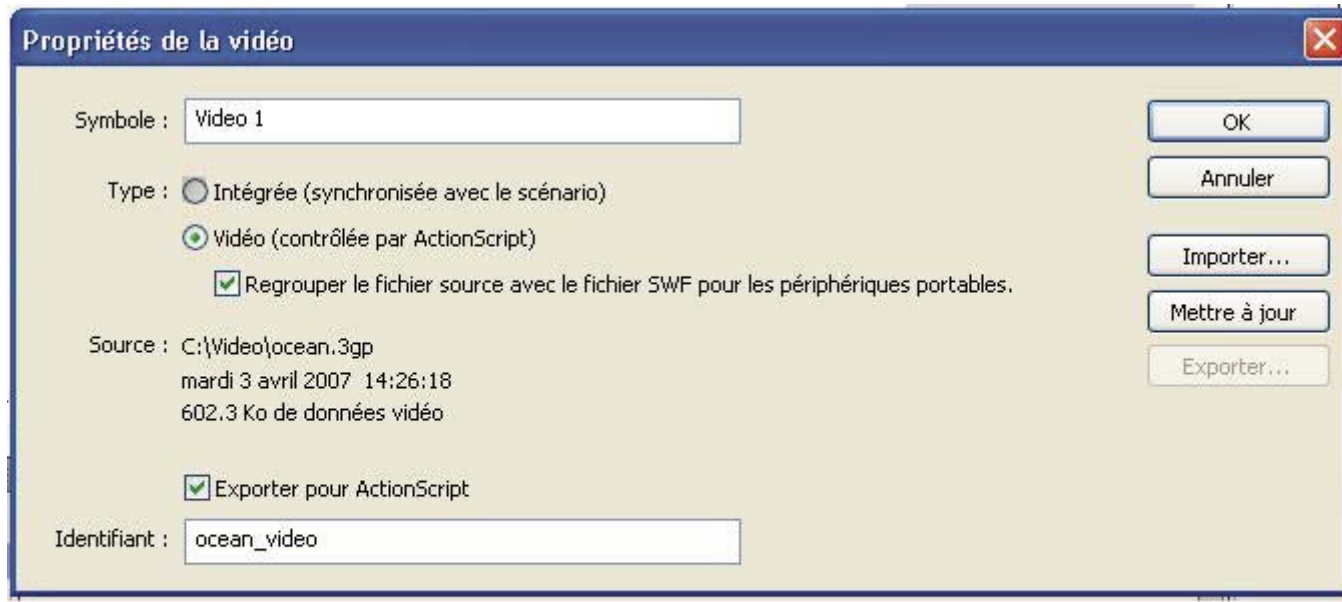
**Remarque :** Sur la page regroupant didacticiels et exemples à l'adresse ([http://www.adobe.com/go/learn\\_flt\\_samples\\_and\\_tutorials\\_fr](http://www.adobe.com/go/learn_flt_samples_and_tutorials_fr)), téléchargez et décompressez le fichier ZIP correspondant à votre version de Flash Lite.

Un nouveau symbole vidéo s'affiche dans le panneau Bibliothèque.



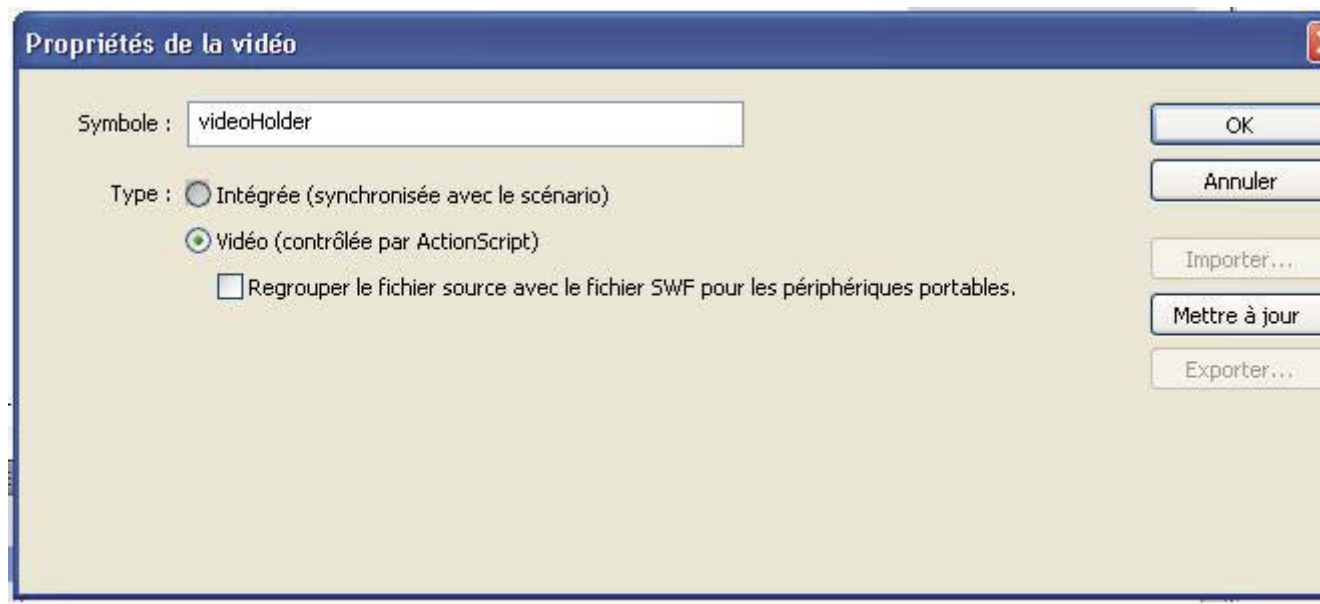
- 5 Cliquez avec le bouton droit de la souris (Windows) ou cliquez en maintenant la touche Contrôle enfoncée (Macintosh) sur le symbole vidéo `ocean.3gp` dans la bibliothèque, puis sélectionnez Propriétés dans le menu contextuel. La boîte de dialogue Propriétés de la vidéo s'affiche.

- 6 Dans cette boîte de dialogue, sélectionnez Exporter pour ActionScript, puis saisissez **ocean\_video** dans la zone de texte Identifiant, comme illustré dans l'image suivante :



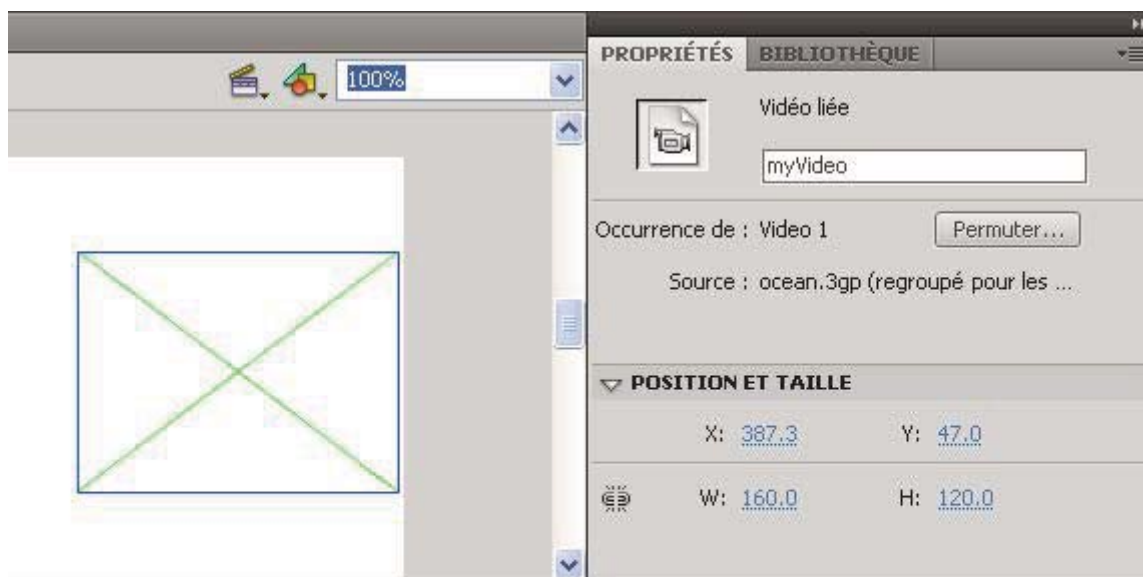
- 7 Cliquez sur OK pour fermer la boîte de dialogue Propriétés de la vidéo.
- 8 Pour créer l'emplacement réservé du clip vidéo, effectuez les opérations suivantes :
- a Dans le panneau Bibliothèque, cliquez sur le bouton du menu des options situé dans la barre de titre et sélectionnez Nouvelle vidéo. La boîte de dialogue Propriétés de la vidéo s'affiche.

- b Dans la zone de texte Symbole, saisissez **videoHolder**.



- c Cliquez sur OK pour fermer la boîte de dialogue Propriétés de la vidéo.
- 9 Faites glisser une occurrence du symbole de vidéo sur la scène.
- 10 Sélectionnez l'objet vidéo sur la scène puis, dans l'inspecteur des propriétés, saisissez **myVideo** dans la zone de texte Nom de l'occurrence.

Utilisez ce nom pour vous reporter à l'objet vidéo dans le code ActionScript.



**11** Réglez la largeur sur 176 et la hauteur sur 144 dans l'inspecteur de propriétés.

Ces dimensions correspondent à la source de la vidéo. Selon le périphérique, les vidéos Flash Lite de périphérique ne se redimensionnent pas toujours pour occuper tout le cadre de délimitation.

**12** Pour ajouter des boutons de contrôle de la vidéo, ouvrez la bibliothèque de boutons prédéfinis (Fenêtre > Bibliothèques communes > Boutons).

**13** Dans la bibliothèque de boutons, cliquez deux fois sur le dossier Circle Buttons pour l'ouvrir.

**14** Faites glisser sur la scène une occurrence du symbole de bouton de lecture.

**15** Faites glisser sur la scène une occurrence du symbole de bouton d'arrêt.

**16** Sélectionnez le bouton de lecture sur la scène et saisissez (ou copiez et collez) le code suivant dans le panneau Actions :

```
on (press) {
    myVideo.play("symbol://ocean_video");
}
```

**17** Sélectionnez le bouton d'arrêt sur la scène et saisissez le code suivant dans le panneau Actions :

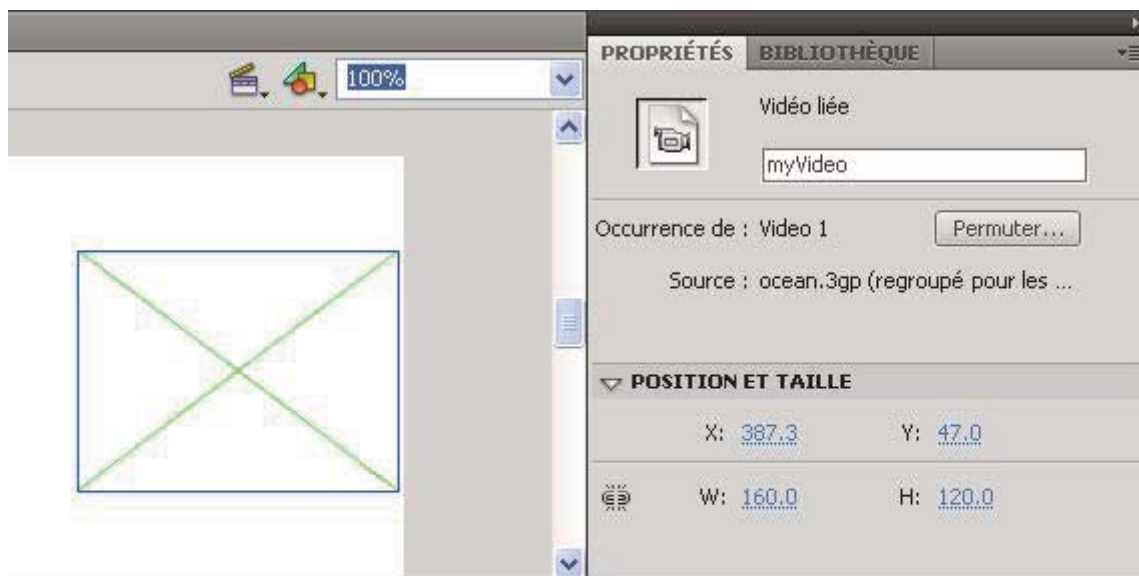
```
on (press) {
    myVideo.stop();
}
```

**18** Sélectionnez Contrôle > Tester l'animation pour vérifier le film dans Device Central.

## Lecture d'une vidéo de périphérique externe dans une application

Les fichiers vidéo du périphérique externe se trouvent sur la carte mémoire d'un périphérique ou sur un serveur Web. Comme avec tous les périphériques vidéo, le périphérique est responsable pour le décodage et l'affichage des fichiers.

- 1 Dans Flash, sélectionnez Fichier > Nouveau > Fichier Flash (Mobile) puis cliquez sur OK.
- 2 Dans Adobe Device Central, cliquez sur Créer.
- 3 Dans Flash sélectionnez Fichier > Enregistrer puis enregistrez le fichier sous le nom external\_device\_video fla.
- 4 Enregistrez le film ocean.3gp dans le même dossier que le fichier external\_device\_video fla.  
Le fichier se trouve dans Flash\_Lite\_2x\Samples\Video\ folder dans le paquet Samples and Tutorials que vous pouvez télécharger à l'adresse [www.adobe.com/go/learn\\_flt\\_samples\\_and\\_tutorials](http://www.adobe.com/go/learn_flt_samples_and_tutorials).
- 5 Faites comme suit pour créer le symbole de vidéo pour afficher le fichier externe :
  - a Dans le panneau Bibliothèque, cliquez sur le bouton du menu des options situé dans la barre de titre et sélectionnez Nouvelle vidéo. La boîte de dialogue Propriétés de la vidéo s'affiche.
  - b Dans la boîte Symbole, saisissez **Video Display**.  
Ce nom est utilisé dans le panneau Bibliothèque.
  - c Sélectionnez Vidéo (contrôlée par ActionScript) puis cliquez sur OK.  
Assurez-vous que l'option Bundle source in SWF for mobile and devices n'est pas cochée.
- 6 Faites glisser une occurrence du symbole de vidéo sur la scène.
- 7 Sélectionnez l'objet vidéo sur la scène puis, dans l'inspecteur des propriétés, saisissez **myVideo** dans la zone de texte Nom de l'occurrence.  
Utilisez ce nom pour vous reporter à l'objet vidéo dans le code ActionScript.



- 8 Réglez la largeur de la vidéo sur 176 et la hauteur sur 144 dans l'inspecteur de propriétés.

Ces dimensions correspondent à la source de la vidéo. Selon le périphérique, les vidéos Flash Lite de périphérique ne se redimensionnent pas toujours pour occuper tout le cadre de délimitation.

- 9 Pour ajouter des boutons de contrôle de la vidéo, ouvrez la bibliothèque de boutons prédéfinis (Fenêtre > Bibliothèques communes > Boutons).

- 10 Dans la bibliothèque de boutons, cliquez deux fois sur le dossier Circle Buttons pour l'ouvrir.

- 11 Faites glisser sur la scène une occurrence du symbole de bouton de lecture.

- 12 Faites glisser sur la scène une occurrence du symbole de bouton d'arrêt.

- 13 Sélectionnez le bouton de lecture sur la scène et saisissez (ou copiez et collez) le code suivant dans le panneau Actions :

```
on (press) {  
    myVideo.play("ocean.3gp");  
}
```

Pour lire un fichier vidéo externe, transmettez l'emplacement ou l'URL, absolu ou relatif, du fichier vidéo à la méthode `video.play()`. Dans l'exemple suivant, les fichiers SWF et 3GP figurent dans le même dossier sur le périphérique. Vous pouvez également spécifier un chemin de dossier relatif au fichier SWF, comme dans l'exemple suivant :

```
myVideo.play("folder1/folder2/ocean.3gp");
```

Selon le périphérique, vous pouvez également utiliser le protocole `file://` pour lire un fichier vidéo à un emplacement spécifique, comme suit :

```
myVideo.play("file://c:/folder1/folder2/ocean.3gp");
```

**Remarque :** Tous les périphériques ne prennent pas en charge le protocole `file://`. Vous devez tester votre application sur l'ensemble des périphériques cibles si vous appliquez ce protocole.

Certains périphériques prennent en charge le chargement de fichiers vidéo à partir d'une adresse réseau via RTSP (Protocole de lecture en transit en temps réel). Le flux HTTP n'est pas pris en charge. L'exemple suivant charge et lit un fichier 3GPP.

```
myVideo.play("rtsp://www.example.com/video/ocean.3gp");
```

- 14 Sélectionnez le bouton d'arrêt sur la scène et saisissez le code suivant dans le panneau Actions :

```
on (press) {  
    myVideo.stop();  
}
```

- 15 Sélectionnez Contrôle > Tester l'animation pour vérifier le film dans Device Central.

## Affichage et modification des propriétés des symboles d'une vidéo de périphérique

La boîte de dialogue Propriétés de la vidéo permet d'afficher et de modifier les informations relatives aux symboles vidéo dans le panneau Bibliothèque.



### Ouverture de la boîte de dialogue Propriétés de la vidéo.

Procédez au choix comme suit :

- Cliquez avec le bouton droit de la souris (Windows) ou cliquez en maintenant la touche Contrôle enfoncée (Macintosh) sur un symbole vidéo dans le panneau Bibliothèque et sélectionnez Propriétés dans le menu contextuel.
- Sélectionnez un symbole vidéo dans le panneau Bibliothèque, puis sélectionnez Propriétés dans le menu des options, accessible à partir de la barre de titre.

### Importation d'une vidéo de périphérique dans un symbole vidéo

- 1 Sélectionnez un symbole vidéo dans le panneau Bibliothèque et ouvrez la boîte de dialogue Propriétés de la vidéo.
- 2 Dans cette boîte de dialogue, cochez l'option de regroupement de la source vidéo dans le fichier SWF.
- 3 Cliquez sur Importer, puis recherchez et sélectionnez le fichier vidéo de périphérique à importer.  
Si vous ne voyez pas le fichier vidéo voulu (ou si vous pouvez le voir, mais pas le sélectionner), sélectionnez Tous les fichiers (\*.\*) à partir du menu déroulant Fichiers de type (sous Windows), ou Tous les fichiers à partir du menu déroulant Activer (sous Macintosh).
- 4 Cliquez sur Ouvrir pour fermer la boîte de dialogue de sélection.
- 5 Cliquez sur OK pour fermer la boîte de dialogue Propriétés de la vidéo.

**Remarque :** Le bouton OK est désactivé jusqu'à ce que vous importiez une vidéo de périphérique ou désélectionniez l'option de regroupement de la source vidéo dans le fichier SWF.

### Association d'un identifiant à un symbole vidéo

- 1 Sélectionnez le symbole vidéo et ouvrez la boîte de dialogue Propriétés de la vidéo.
- 2 Si ce n'est pas déjà fait, importez une vidéo de périphérique dans le symbole.
- 3 Sélectionnez Exporter pour ActionScript.
- 4 Dans la zone de texte Identifiant, entrez l'identifiant du symbole vidéo.  
Comme tous les identifiants d'ActionScript, le premier caractère doit être une lettre, un signe souligné (\_) ou dollar (\$). Les caractères suivants peuvent être des nombres, des lettres, des signes souligné ou dollar.
- 5 Cliquez sur OK pour fermer la boîte de dialogue Propriétés de la vidéo.

## Contrôle d'une vidéo de périphérique avec le code ActionScript

Utilisez l'objet Video d'ActionScript pour contrôler la lecture de la vidéo de périphérique. Les méthodes suivantes de l'objet Video sont disponibles dans Flash Lite 2.0 et les versions ultérieures :

- Video.play()
- Video.stop()
- Video.pause()
- Video.resume()
- Video.close()
- Video.attachVideo() (Flash Lite 3.0 et versions ultérieures)

**Remarque :** Ces méthodes de l'objet Video sont proches de celles de l'objet NetStream dans la version de bureau de Flash Player. Ces méthodes de l'objet Video ne sont pas disponibles pour la version de bureau de Flash Player.

Flash Lite 2.0 et les versions ultérieures ne prennent pas en charge les méthodes et propriétés suivantes de l'objet Video :

- `Video.clear()`
- `Video.deblocking`
- `Video.height`
- `Video.smoothing`
- `Video.width`
- `Video._visible`

Pour plus d'informations sur l'utilisation de l'objet Video dans Flash Lite, consultez l'entrée de l'objet Video dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*.

# Chapitre 8 : Développement d'applications Flash Lite pour BREW

Vous pouvez développer des applications Adobe Flash Lite 2.x exécutables sur des périphériques utilisant la plate-forme BREW (Binary Runtime Environment for Wireless) de QUALCOMM Incorporated. Il s'adresse principalement aux développeurs d'applications qui ont une connaissance approfondie de Flash Lite, mais qui maîtrisent encore mal la plate-forme BREW. Ce chapitre sera également utile à ceux qui connaissent bien les autres technologies de développement d'applications et qui souhaitent découvrir l'utilisation de Flash Lite pour développer des applications BREW.

## Généralités sur BREW

### A propos de BREW

La plate-forme pour périphériques portables QUALCOMM BREW s'installe et est prise en charge sur bon nombre d'appareils fabriqués par divers constructeurs dans le monde entier. Adobe Flash CS4 Professional comprend un éventail d'outils et de ressources que les développeurs utilisent pour créer du contenu et des applications Flash Lite téléchargeables et exploitables par les utilisateurs sur des périphériques BREW.

Les développeurs utilisent l'outil de programmation de Flash pour créer des applications et du contenu Flash Lite pour les périphériques compatibles BREW. Ces développeurs soumettent ensuite leurs applications aux laboratoires NSTL (National Software Testing Laboratories) pour qu'elles subissent des tests TBT (True BREW Testing). Après avoir passé les tests TBT avec succès, les applications sont chargées dans le système BDS (BREW Delivery System) et, lorsqu'un opérateur de télécommunications en assure la distribution, elles sont mises à la disposition des utilisateurs sur le serveur de téléchargement d'applications (ADS, Application Download Server) de l'opérateur. Les utilisateurs de périphériques peuvent alors les télécharger et les acheter en profitant de fonctions indispensables, présentes sur la plate-forme BREW, pour assurer la diffusion radio et la facturation par l'intermédiaire du système BDS.

Les développeurs qui souhaitent essayer BREW sans s'acquitter des frais inhérents à un développement BREW complet peuvent choisir de ne pas s'enregistrer en tant que développeurs BREW authentifiés et ne télécharger que les composants SDK (Kit de développement logiciel) gratuits. Ces derniers sont suffisants pour produire et tester une application à l'aide du simulateur BREW Simulator, mais pas pour la charger et la tester sur un périphérique, ni pour simuler les caractéristiques d'un périphérique spécifique avec des packs de périphériques personnalisés.

L'environnement BREW est destiné à fonctionner comme un « écosystème » afin de faciliter le développement et accroître l'offre de contenus pour périphérique sans fil. Son objectif est d'aider les développeurs à créer, à distribuer et à retirer un profit des applications sans fil. Cet écosystème se compose des principaux groupes de participants suivants :

*Les développeurs* utilisent les outils du kit de développement logiciel BREW pour développer du contenu en combinant le langage C++ avec des API ou avec une extension compatible BREW. Le téléchargement du kit de développement BREW est gratuit, mais vous devez vous enregistrer auprès de QUALCOMM pour accéder à la suite d'outils et aux outils du kit.

*Le laboratoire NSTL (National Software Testing Laboratories)* teste les applications envoyées par les développeurs. Si le test est concluant, il propose de les inclure dans le système BDS. Si les applications ne réussissent pas les tests, les développeurs peuvent les corriger et les soumettre à nouveau.

QUALCOMM gère le système BDS (BREW Delivery System). Ce système, de type Web, diffuse le contenu du catalogue de QUALCOMM, avec les prix et des informations de facturation. Les tarifs peuvent impliquer plusieurs modèles, y compris une mise aux enchères de l'opérateur.

Les *opérateurs* utilisent le serveur ADS pour proposer aux clients d'acheter, au prix de détail, le contenu qu'ils ont sélectionné via un dispositif de diffusion radio (OTA).

Les *fabricants de périphériques* utilisent le client BREW, qui présente un ensemble commun d'interfaces de programmation d'applications (API) pour le développement normalisé d'applications sans fil. Ce client comprend également un gestionnaire d'application que les utilisateurs peuvent acheter et exploiter pour gérer les applications BREW.

## Outils de développement Flash Lite et BREW

Pour développer et tester des applications Flash Lite pour l'environnement BREW, vous devez acquérir et installer des outils spécifiques, aussi bien matériels que logiciels.

### Matériel

**Ordinateur Windows** Pour plus d'informations sur le kit de développement logiciel BREW (BREW SDK) et la suite d'outils associés, consultez la section « System Requirements » du guide *Starting with BREW* (disponible sur le site Web de BREW). Pour plus d'informations sur la configuration système requise pour les logiciels Adobe, consultez la page Web [www.adobe.com/go/sysreqs\\_fr](http://www.adobe.com/go/sysreqs_fr).

**Périphériques de type BREW avec jeu de bits de contrôle** Pour plus d'informations sur la façon d'acquérir, d'activer et de tester les périphériques BREW, consultez la section « Handset Acquisition and Readiness » du guide *Getting Started with BREW* (disponible sur le site Web de BREW). Notez que les périphériques qui prennent en charge BREW 2.x doivent être envoyés à QUALCOMM pour être activés pour le test ; les périphériques BREW 3x sont préactivés.

**Câbles de données** Vous devez disposer de câbles de données pour charger les applications Flash sur les périphériques BREW que vous comptez prendre en charge. La plupart des périphériques sont livrés avec des câbles de données. Dans le cas contraire, consultez la section « Acquire BREW Handsets and Data Cables » du guide *Getting Started with BREW* (disponible sur le site Web de BREW).

### Logiciels

**Kit de développement BREW et suite d'outils BREW** Pour plus d'informations sur l'installation du kit de développement logiciel BREW (BREW SDK) et des outils correspondants, consultez le chapitre 2, « Installing the BREW SDK », du guide *Getting Started with BREW* (disponible sur le site Web de BREW). Pour télécharger et installer la suite d'outils BREW, vous devez vous enregistrer en tant que développeur BREW, mais pas pour le kit de développement proprement dit. Pour obtenir des instructions sur le mode d'enregistrement, consultez le site Web de BREW.

**Pilotes USB pour les périphériques cible** Procurez-vous et installez les pilotes USB pour les périphériques que vous visez. Pour plus d'informations sur l'acquisition de ces pilotes, contactez le fabricant du périphérique.

**Extension Flash Lite pour les périphériques compatibles BREW** Ces périphériques sont disponibles sur le site Web de BREW à des fins de test uniquement. Cette extension est automatiquement téléchargée sur les périphériques des utilisateurs lorsqu'ils sélectionnent et téléchargent une application Flash Lite pour BREW.

Ces composants sont décrits en détail dans la section « [Configuration de votre système pour BREW](#) » à la page 99.

## Fonctionnalités Flash Lite prises en charge dans BREW

Flash Lite pour BREW prend en charge un sous-ensemble de fonctionnalités disponibles à la fois dans Flash Lite 2.0 et Flash Lite 2.1. Les fonctionnalités prises en charge par Flash Lite pour BREW sont les suivantes. Pour plus d'informations sur ces caractéristiques, consultez la section « [Fonctionnalités de Flash Lite 3.0](#) » à la page 7.

- Repose sur Macromedia Flash Player 7 d'Adobe, prend en charge ActionScript 2.0
- Gestion des données XML
- Gestion des données permanentes
- Périphérique, polices vectorielles
- Son de périphérique pour des événements
- Vidéo en ligne (limitée aux formats vidéo pris en charge par le périphérique)
- Diffusion vidéo en continu
- Sockets XML
- Accès réseau et diffusion HTTP en continu
- Prise en charge de texte en ligne (limitée à 64 000 caractères, avec des restrictions supplémentaires imposées par les implémentations des périphériques individuels. Certains périphériques ne prennent en charge que 1 000 caractères.)
- Multimédia dynamique
- Prise en charge du clavier et du clavier numérique

## Fonctionnalités Flash Lite non prises en charge dans BREW

Les fabricants de périphériques BREW peuvent limiter l'usage de certaines fonctionnalités Flash Lite, qui ne peuvent alors pas être utilisées sur le périphérique. Des informations concernant certains périphériques spécifiques sont disponibles dans l'espace réservé aux ressources des développeurs (Developer Resources) sur le site Web de BREW.

Outre les fonctions non prises en charge sur un périphérique donné, la liste suivante répertorie celles que l'implémentation Flash Lite BREW actuelle ne prend pas en charge (et qui ne sont donc disponibles sur aucun périphérique) :

**Décodage des sons Flash Lite** Tous les formats audio sont transmis au périphérique pour une lecture sans modification. Par exemple, si le son est au format MP3 et que le périphérique prend en charge ce format, la lecture est possible. Flash Lite ne décode aucun son en interne dans BREW, comme il le fait sur d'autres plates-formes.

**Diffusion du son en continu** Tous les sons doivent être entièrement chargés avant leur lecture. L'actuelle implémentation de BREW n'assure pas la lecture progressive des sons, car ils sont chargés sur un réseau.

**Commandes ActionScript** Les fonctions ActionScript `fscommand2()` suivantes ne sont disponibles sur aucun périphérique BREW :

<code>ExtendBacklightDuration()</code>	<code>GetNetworkConnectStatus()</code>
<code>GetBatteryLevel()</code>	<code>GetNetworkName()</code>
<code>GetMaxBatteryLevel()</code>	<code>GetNetworkRequestStatus</code>
<code>GetMaxSignalLevel()</code>	<code>GetPowerSource()</code>
<code>GetNetworkConnectionName()</code>	<code>GetSignalLevel()</code>

La fonction ActionScript `fscommand()` suivante n'est disponible sur aucun périphérique BREW :

fsccommand (Launch)	
---------------------	--

**Prise en charge de `getURL()`** Selon l'implémentation, certains combinés BREW prennent en charge une partie des fonctions `getURL()`. En règle générale, ni MMS (Multimedia Messaging Service) ni HTTP n'est pris en charge (aucune API ne permet à Flash Lite de lancer un navigateur).

**Remarque :** Les requêtes `mailto` ne sont pas prises en charge directement, mais vous pouvez utiliser le protocole SMS (Short Message Service), qui limite la taille maximale des messages à 160 caractères.

**Contenu papier peint** Le contenu papier peint Flash Lite n'est pas pris en charge dans l'implémentation Flash Lite pour BREW.

**Sonneries animées** Les sonneries animées ne sont pas prises en charge dans l'implémentation Flash Lite pour BREW.

## Versions du kit de développement BREW prises en charge

Flash Lite 2.1 pour périphériques BREW prend actuellement en charge les versions BREW 2.1.3 et ultérieures. Vous trouverez la version de BREW fonctionnant sur un périphérique particulier dans la liste des caractéristiques techniques des périphériques qui est publiée sur le site Web de BREW. Les deux périphériques qui prennent actuellement en charge Flash Lite pour BREW (le Samsung SCH-A950 et le LG VX9800) gèrent des versions différentes de BREW (2.x et 3.x respectivement). Les kits de développement logiciel BREW étant compatibles avec les versions antérieures, vous pouvez télécharger la version du kit et des outils la plus récente (3.x lors de la rédaction de ce document) plutôt que la version prise en charge par le périphérique visé. Par exemple, si vous envisagez le Samsung SCH-A950, périphérique 2.x, vous pouvez télécharger et utiliser la version 3.x du kit de développement et de la suite d'outils BREW.

Les différences entre BREW 2.x et 3.x sont minimales en termes d'interface utilisateur et Flash Lite se comporte de la même façon sur les deux plates-formes. Les différences notables concernent le stockage des fichiers sur le périphérique et le chargement des applications dans la suite d'outils BREW. Pour plus d'informations sur la structure du système de fichiers, consultez la section « Structures des fichiers de périphériques pour les différentes versions de BREW » à la page 106.

## Périphériques pris en charge

Les périphériques qui prennent actuellement en charge la plate-forme BREW sont les suivants :

Périphérique	Versions prises en charge
Samsung SCH-A950	BREW version 2.1.3
LG VX9800	BREW version 3.1.2

L'équipe d'assistance clientèle d'Adobe délivrera des certifications à d'autres périphériques lorsqu'ils seront disponibles. Pour obtenir la liste la plus récente des périphériques pris en charge, consultez [www.adobe.com/go/mobile\\_supported\\_devices/](http://www.adobe.com/go/mobile_supported_devices/).

## Ressources supplémentaires

Pour plus d'informations sur BREW, consultez le site Web de la plate-forme BREW. La section Developer Home du site propose des informations sur le kit de développement, les outils et les utilitaires BREW, et les caractéristiques des périphériques spécifiques prenant en charge BREW.

**Remarque :** Pour télécharger et installer la suite d'outils BREW et pour que vos périphériques puissent subir les tests BREW, vous devez être enregistré en tant que développeur BREW. Vous trouverez sur le site davantage d'informations sur les modalités d'enregistrement.

## Configuration de votre système pour BREW

### Liste des tâches de configuration du système pour BREW

Pour configurer votre système afin de produire des fichiers Flash Lite pour BREW, vous devez localiser et installer le logiciel approprié à partir de plusieurs sources. Vous devez également vous procurer le matériel approprié (périphériques de type BREW, câbles de données, etc.). Vous trouverez la plupart des informations requises pour installer et configurer ces composants dans les autres documents Flash Lite ou BREW.

Pour configurer votre système pour qu'il prenne en charge la programmation Flash Lite pour BREW, effectuez les tâches suivantes :

- 1 Inscription en tant que développeur BREW
- 2 Installation du kit de développement et de la suite d'outils BREW
- 3 Installation de Flash Lite 2.1 d'Adobe pour périphériques BREW
- 4 Installation des pilotes USB pour les périphériques cibles.

#### Inscription en tant que développeur BREW

Pour vous inscrire comme développeur BREW, accédez au site principal BREW de QUALCOMM et suivez les liens de la page d'accueil Developer (Développeur) afin de devenir un développeur BREW authentifié.

#### Installation du kit de développement et de la suite d'outils BREW

Les instructions suivantes présentent la procédure d'installation du kit de développement BREW sur votre ordinateur. Elles partent du principe que vous êtes déjà enregistré en tant que développeur BREW. Pour plus d'informations sur l'installation du kit de développement logiciel BREW (BREW SDK) ou des outils associés, consultez la section « Installing the BREW SDK » dans *Starting with BREW*. Accédez au site Web de BREW, puis sélectionnez Developer Home > BREW Documentation > Application Development Documentation.

- 1 Dans l'espace réservé aux développeurs, BREW Developer Home, situé sur le site Web de la plate-forme BREW, cliquez sur Download BREW Tools pour afficher la page BREW Development and Commercialization Tools.
- 2 Naviguez jusqu'à la section BREW Development et cliquez sur le lien approprié pour télécharger la dernière version du kit de développement BREW (BREW SDK 3.1 lors de la rédaction de ce document).

Suivez les instructions qui s'affichent pour télécharger et installer le kit de développement.

- 3 Lorsque le kit de développement est installé, installez les outils.

Naviguez jusqu'à la section BREW Commercialization de la page BREW Development and Commercialization Tools et cliquez sur le lien BREW Testing and Commercialization Utilities. Suivez les instructions qui s'affichent pour installer la suite d'outils sur votre ordinateur.

Votre menu Démarrer doit maintenant contenir des entrées pour le kit de développement BREW et la suite d'outils BREW.

### Installation Flash Lite 2.1 d'Adobe pour périphériques BREW

Flash Lite 2.1 pour périphériques BREW comprend trois composants :

- Flash Lite Publisher pour BREW (disponible sur [www.adobe.com/go/support\\_flashlite](http://www.adobe.com/go/support_flashlite))
- Flash Lite 2.1 pour le simulateur BREW (disponible sur [www.adobe.com/go/support\\_flashlite](http://www.adobe.com/go/support_flashlite))
- Extension Flash Lite pour les périphériques compatibles BREW (disponible sur le site Web de BREW à des fins de test uniquement. Cette extension est automatiquement téléchargée sur les périphériques des utilisateurs lorsqu'ils sélectionnent et téléchargent une application Flash Lite pour BREW.)

### Installation des pilotes USB pour les périphériques

Chaque périphérique visé pour le développement BREW comprend son propre ensemble de pilotes USB, que vous devez installer sur votre ordinateur. Les nouveaux périphériques doivent être accompagnés d'un CD-ROM contenant ces pilotes. Lorsque cela est possible, utilisez ces versions. Pour installer les pilotes USB d'un périphérique, suivez la procédure standard d'installation de pilote de périphérique de Windows.

## Ressources supplémentaires

Pour plus d'informations sur le développement et le test d'applications Flash Lite, consultez les autres sections de ce document, vous y trouverez des informations complètes. L'aide en ligne de Flash CS4 Professional présentent également des informations détaillées sur le développement de Flash Lite.

Pour plus d'informations sur BREW, consultez la meilleure source en la matière, le site Web de QUALCOMM BREW.

Vous pouvez également consulter les fichiers d'aide du kit de développement pour plus d'informations sur les processus de développement et les outils BREW.

Lors de la rédaction de ce document, l'ensemble de la documentation de BREW comprend deux documents aux noms similaires, tous deux très utiles pour découvrir les bases de BREW. Ces documents sont les suivants :

- *Getting Started with BREW* (résumé de deux pages sur le matériel et les logiciels nécessaires pour développer des applications BREW).
- *Starting with BREW* (document plus détaillé contenant toutes les informations nécessaires pour commencer à développer des applications BREW).

## Programmation de fichiers Flash Lite pour BREW

### Organisation de la tâche de création de fichiers Flash Lite pour BREW

D'une manière générale, la programmation pour BREW et la programmation pour Flash Lite sont similaires, à l'exception du fait que vous adaptez vos applications aux caractéristiques spécifiques du combiné de type BREW visé. Cette section explique comment utiliser Flash pour créer un contenu Flash Lite pouvant être ensuite publié pour BREW à l'aide de Flash Lite Publisher pour BREW (décrit dans la section « [Publication de fichiers Flash Lite pour BREW](#) » à la page 103).

La création de contenu Flash Lite pour la plate-forme BREW se présente sous la forme d'une procédure itérative impliquant les étapes suivantes :



### Identification des périphériques et du type de contenu Flash Lite visés

Au moment de la parution de ce document, les deux périphériques suivants prennent en charge BREW : le Samsung SCH-A950, qui comprend la plate-forme BREW version 2.1.3 et le LG VX9800, qui comprend la plate-forme BREW version 3.1.2. Les caractéristiques de ces deux périphériques étant très différentes (par exemple, le LG dispose d'un clavier QWERTY, mais pas le Samsung), il est important d'adapter vos plans de développement aux capacités spécifiques du périphérique visé.

De même, vous pouvez créer deux types de contenus (application et écran de veille) qui demandent chacun une conception différente. Pour plus d'informations sur la façon dont les écrans de veille diffèrent des applications, visitez le site Web de BREW en effectuant la navigation suivante : Developer FAQs > BREW Tools > MIF Settings > Screensaver.

**Remarque :** Pour l'instant, seul le Samsung SCH-A950 prend en charge le type de contenu Ecran de veille.

### Création et test de l'application dans Flash

Adobe Flash CS4 Professional comprend un émulateur dans Adobe Device Central CS4 qui vous permet de tester votre application sans la transférer à un périphérique. L'émulateur de Device Central permet de parfaire la conception générale de votre application et de résoudre d'éventuels problèmes avant de la tester sur un périphérique portable.

**Remarque :** Les tests effectués dans Flash à l'aide de l'émulateur diffèrent de ceux effectués à l'aide du simulateur inclus dans le kit de développement BREW. Cette section présente les tests effectués au moyen de l'émulateur Flash ; ceux réalisés par le biais du simulateur BREW sont décrits dans la section « [Publication de fichiers Flash Lite pour BREW](#) » à la page 103. L'émulateur Device Central ne fournissant pas pour l'instant de packs pour périphériques BREW, il ne peut être utilisé que pour tester le fonctionnement de base. Pour tester le fonctionnement de votre application dans l'environnement BREW, utilisez le simulateur BREW.

### Identification du périphérique et du type de contenu visés

L'espace réservé aux développeurs sur le site QUALCOMM regroupe des informations sur les caractéristiques techniques spécifiques de tous les périphériques prenant actuellement en charge une version de BREW. D'autres informations sont également disponibles sur les périphériques qui ne sont pas encore commercialisés. Vous pouvez les utiliser pour identifier les périphériques visés par votre application BREW. Plusieurs exemples d'application englobant différents packs pour périphériques sont également inclus dans le kit de développement logiciel BREW. Utilisez-les pour en savoir plus sur le fonctionnement de périphériques spécifiques.

### Création de l'application dans Flash

La procédure de création d'applications Flash Lite pour BREW est semblable à celle d'une application Flash Lite générique. A l'exception des fonctions non prises en charge par les périphériques BREW (énumérées à la section « [Fonctionnalités Flash Lite non prises en charge dans BREW](#) » à la page 97) et des notes de programmation spécifiques de BREW (ci-dessous), le développement d'applications pour les périphériques BREW peut s'effectuer selon la procédure de développement détaillée dans le reste de la documentation Flash.

Les informations présentées dans les sections suivantes ne concernent que la programmation Flash Lite pour BREW.

#### Démarrage d'un navigateur

Les combinés BREW visés utilisent le navigateur OpenWave, qui ne fournit pas à l'application BREW l'accroche appropriée pour l'invoquer. De ce fait, si vous spécifiez une URL HTTP dans les fonctions `getUrl()`, `loadVars(send)` ou `XML.send()`, aucun navigateur ne démarre. Pour envoyer des informations à un serveur et poursuivre la lecture de votre fichier SWF sans ouvrir de nouvelle fenêtre ni remplacer le contenu d'une fenêtre ou d'une image, utilisez la fonction `loadVars.sendAndLoad()`.

### Comportement par défaut des touches programmables

Pour obtenir plus d'informations et d'exemples sur la programmation du comportement des touches programmables dans Flash Lite, consultez la section « [Utilisation des touches programmables](#) » à la page 50. Le tableau ci-dessous présente le comportement par défaut des touches programmables dans Flash Lite 2.1 pour BREW :

Action du contenu	Mode d'écran	Action de la touche gauche	Action de la touche droite
Par défaut (le contenu n'appelle pas la commande <code>SetSoftKeys</code> )	Mode plein écran	Affiche le menu d'options	Ferme le lecteur
Mode non plein écran (le contenu n'appelle pas la commande <code>SetSoftKeys</code> )	Mode non plein écran	Affiche le menu d'options	Ferme le lecteur
Le contenu désactive les touches programmables	sans objet	Aucune	Aucune
Le contenu remplace une ou deux touches programmables (appelle la commande <code>SetSoftKeys</code> )	sans objet	Affiche le menu d'options s'il n'y a pas de remplacement. S'il y a remplacement, le comportement est spécifié par le contenu.	Ferme le lecteur s'il n'y a pas de remplacement. S'il y a remplacement, le comportement est spécifié par le contenu.

## Test de l'application dans Flash

Comme pour la création d'application, l'utilisation de l'émulateur Adobe Device Central pour tester votre application BREW revient pratiquement à tester une application Flash Lite générique. N'oubliez pas que les tests effectués dans Flash (à l'aide de l'émulateur Adobe Device Central) diffèrent de ceux effectués au moyen du simulateur inclus dans le kit de développement BREW. Cette section présente les tests effectués au moyen de l'émulateur Device Central ; ceux réalisés par le biais du simulateur BREW sont décrits dans la section « [Publication de fichiers Flash Lite pour BREW](#) » à la page 103.

L'émulateur Device Central ne fournissant pas pour l'instant de packs pour périphériques BREW, il ne peut être utilisé que pour tester le fonctionnement de base. Pour tester le fonctionnement de votre application dans l'environnement BREW, utilisez le simulateur BREW.

- 1 Ouvrez votre application dans Flash.
- 2 Sélectionnez Fichier > Paramètres de publication. Dans l'onglet Flash, sélectionnez Flash Lite 2.1 dans la liste Version, puis ActionScript 2.0 dans la liste ActionScript. Cliquez sur OK pour enregistrer vos paramètres.
- 3 Lancez l'émulateur Adobe Device Central (Contrôle > Tester l'animation), puis cliquez sur l'onglet Profils de périphérique.
- 4 Dans le panneau de navigation, effectuez l'une des opérations suivantes :
  - a Si vous n'avez pas téléchargé de pack de périphérique pour le périphérique visé, sélectionnez Générique > Flash Lite 2.0 > Téléphone générique.
  - b Si vous avez téléchargé un pack de périphérique pour le périphérique visé (disponible sur le site [www.adobe.com/products/flash/download/device\\_profiles](http://www.adobe.com/products/flash/download/device_profiles)), cliquez sur le nom du périphérique pour le développer, sélectionnez la taille appropriée, puis faites glisser le nom du périphérique visé du panneau Périphériques disponibles jusqu'au panneau Ensemb. de périph.

5 Sélectionnez Fichier > Retourner à Flash puis, dans le menu Flash, sélectionnez Contrôle > Tester l'animation. Si l'assistant de publication BREW s'ouvre, cliquez sur Annuler pour fermer la boîte de dialogue. Flash exporte l'application vers Device Central et l'ouvre dans l'émulateur.

6 Servez-vous de l'émulateur pour tester votre application.

Pour plus d'informations sur l'utilisation de l'émulateur, consultez la section « [Utilisation de l'émulateur](#) » à la page 118.

Lorsque le test est terminé dans l'émulateur, enregistrez vos fichiers et passez à la section « [Publication de fichiers Flash Lite pour BREW](#) » à la page 103 pour savoir comment publier votre application pour la plate-forme BREW.

## Ressources supplémentaires

Pour plus d'informations sur le développement et le test d'applications Flash Lite, consultez les sources suivantes qui sont disponibles dans l'aide de Flash :

- *Prise en main de Flash Lite 2.x et 3.0* (PDF)
- Test du contenu Flash Lite (rubriques d'aide)

# Publication de fichiers Flash Lite pour BREW

## A propos de la publication de fichiers Flash Lite pour BREW

Pour pouvoir s'exécuter sur des périphériques BREW, les applications Flash Lite doivent respecter la norme BREW. L'assistant de publication BREW pour Flash Lite aide les développeurs à créer des applications conformes à ce standard en automatisant une partie des étapes nécessaires. Outre la création des fichiers pour l'application elle-même, l'assistant utilise également le simulateur du kit de développement BREW pour créer des fichiers qui vous permettront de tester votre application BREW.

Avant de commencer à utiliser l'assistant de publication BREW, des informations de base sur les types de fichiers d'applications BREW, sur la structure des fichiers et sur la configuration requise sont nécessaires.

## Types de fichier

Le principal objectif de Flash Lite Publisher pour BREW est de publier un fichier SWF standard, puis d'utiliser un post-processeur pour placer le fichier SWF dans un package de sorte qu'il puisse s'exécuter sur un périphérique BREW. Les fichiers nécessaires pour utiliser du contenu Flash Lite sur un périphérique BREW sont regroupés dans le tableau suivant :

Fichier	Contenu du fichier
Fichier MIF	Informations de configuration
Fichier MOD	Fichier d'application pour périphérique BREW
Fichier SIG	Fichier spécifique au périphérique provenant de QUALCOMM

Les fichiers nécessaires pour utiliser du contenu Flash Lite avec le simulateur BREW sont les suivants :

Fichier	Contenu du fichier
Fichier MIF	Informations de configuration
Fichier DLL	Fichier d'application pour le simulateur BREW

Tous ces fichiers doivent présenter le même nom de fichier (plus l'extension appropriée : .mif, .mod, .sig ou .dll). Le nom du fichier doit également être le même que le nom du dossier de l'application. De même, pour fonctionner, ces fichiers doivent être situés dans les dossiers appropriés sur le périphérique. La structure de dossiers varie selon la version de BREW utilisée par le périphérique visé. Pour plus d'informations, consultez la section « [Structures des fichiers de périphériques pour les différentes versions de BREW](#) » à la page 106.

## Fichiers MIF

Les fichiers MIF sont des fichiers de configuration qui fournissent les informations requises par les applets BREW. L'assistant de publication BREW génère pour vous les fichiers MIF dans le cadre du processus de publication. Toutefois, il revient au développeur de contenu individuel de bien comprendre les exigences et le format MIF, et de s'assurer que les paramètres du fichier MIF sont appropriés avant de soumettre les applications aux véritables tests TBT (True BREW Testing).

Le code suivant affiche les valeurs MIF par défaut qui sont produites respectivement par l'assistant de publication BREW pour les applications et les écrans de veille. Si certaines de ces valeurs sont incorrectes pour votre application, téléchargez l'outil MIF Editor depuis le site Web de BREW et modifiez les fichiers avant de soumettre vos applications aux tests.

```
<Applet>
- List of Applets defined in this Module: [Class ID in post processor]
- Applet Information:
* Name: [Applet Name in post processor]
* Class ID: [Class ID in post processor]
* Applet Type: [blank]
- Graphics
* Large: [Large Icon in post processor]
* Medium: [Medium Icon in post processor]
* Small: [Small Icon in post processor]
<Extension>
- Exported Classes: [blank]
- Exported MIME Types: [blank]
<Dependencies>
- External Classes Used by this Module: [blank]
- External Classes Directory (Checkbox): [checked]
- Used: [Class IDs generated by post processor]
<Privileges>
File [checked]
Network [checked]
Web Access [checked]
TAPI [checked]
<Access Control>
- ACL type [blank]
- Rights [blank]
- Groups [blank]
<License>
No License (Checkbox): [checked]
<General>
- Author Name: [Author in post processor]
```

```
- Copyright String: [Copyright in post processor]
- Module Version: [Version in post processor]
- EFS Restrictions
* Max Files: [blank]
* Max Space: (bytes) [blank]
- String Encoding Format: [ISOLATIN1]
Default MIF output for Screensavers
<Applet>
- List of Applets defined in this Module: [Class ID in post processor]
- Applet Information:
* Name: [Applet Name in post processor]
* Class ID: [Class ID in post processor]
* Applet Type: [blank]
- Graphics
* Large: [Large Icon in post processor]
* Medium: [Medium Icon in post processor]
* Small: [Small Icon in post processor]
- Notifications, Flags, Settings...
* Flags: Screensaver [checked]
<Extension>
- Exported Classes: [blank]
- Exported MIME Types: [brew/ssaver Class ID]
<Dependencies>
- External Classes Used by this Module: [blank]
- External Classes Directory (Checkbox): [checked]
- Used: [Class IDs generated by post processor]
<Privileges>
All unchecked
<Access Control>
- ACL type [blank]
- Rights [blank]
- Groups [blank]
<License>
No License (Checkbox): [checked]
<General>
- Author Name: [Author in post processor]
- Copyright String: [Copyright in post processor]
- Module Version: [Version in post processor]
- EFS Restrictions
* Max Files: [blank]
* Max Space: (bytes) [blank]
- String Encoding Format: [ISOLATIN1]
```

## fichiers SIG

Les fichiers SIG sont des fichiers de signatures propres aux périphériques. Ils sont requis par chaque périphérique utilisé pour le développement de votre application. Ces fichiers peuvent être récupérés auprès de QUALCOMM et sont valables 90 jours. Passé ce délai, le fichier de signatures arrive à expiration et QUALCOMM doit en publier un nouveau.

Un seul fichier SIG est requis par périphérique. Toutefois, comme chaque application individuelle requiert son propre fichier SIG correspondant, créez une copie du fichier SIG original et renommez-la en fonction du nom de l'application que vous avez créée. N'oubliez pas que le fichier SIG doit résider dans le même dossier que le fichier MOD, le fichier d'application pour le périphérique BREW.

Pour accéder à l'utilitaire de génération de signatures sur le site Web de BREW, vous devez être un développeur BREW authentifié.

***Remarque :** Les fichiers SIG sont requis pour tester les applications sur les périphériques, mais pas pour la réalisation de tests à l'aide du simulateur BREW.*

## Obtention d'un fichier SIG

- 1 Visitez le site Web de BREW et affichez la page qui contient le lien vers le générateur de signatures de tests de QUALCOMM.
- 2 Sélectionnez BREW Testing Generator.
- 3 Entrez votre mot de passe et votre nom de développeur BREW.
- 4 Dans le champ ESN, tapez **0x<ESN Number>**, où ESN Number correspond au numéro indiqué sous la batterie du périphérique.
- 5 Cliquez sur Generate et attendez que le lien Download Signature apparaisse.
- 6 Téléchargez la signature et enregistrez le fichier ; il sera nécessaire pour tous les tests effectués sur le périphérique.

## fichiers BAR

Vous pouvez ajouter, supprimer et déplacer des fichiers sur un périphérique de type BREW à l'aide d'un câble de données et d'une application Windows nommée AppLoader (cette application est fournie par QUALCOMM avec la suite d'outils BREW). Toutefois, vous ne pouvez pas copier les fichiers présentant les extensions .mod, .mif ou .bar depuis le périphérique BREW vers un autre périphérique ou un autre support de stockage. Cette restriction est imposée par QUALCOMM afin que les fichiers achetés via le système BDS (BREW Delivery System) ne puissent pas être distribués à des utilisateurs non autorisés.

Pour que les développeurs de contenu Flash puissent tirer parti de cette fonctionnalité de sécurité, le lecteur Flash Lite qui s'exécute sur les périphériques BREW cherche des fichiers BAR lorsqu'il tente de référencer un fichier sur le périphérique local. Cette restriction s'applique à toutes les fonctions ActionScript ou de bibliothèques partagées qui utilisent un chemin d'accès pour référencer un fichier (par exemple `loadMovie()`, `loadSound()`, `loadVariables()`, `XML.load()`, etc). Pour utiliser cette fonctionnalité, les développeurs peuvent ajouter l'extension .bar aux fichiers qu'ils créent : le lecteur pourra toujours les localiser et les lire, mais personne ne pourra les copier depuis le périphérique BREW sur lequel ils ont été chargés à l'origine. Flash Lite Player vérifie d'abord l'existence du fichier en utilisant le chemin local fourni (par exemple, `loadMovie(exemple.swf)`). S'il ne trouve pas le fichier à cet emplacement, il vérifie le même chemin en ajoutant l'extension .bar (par exemple, `loadMovie(exemple.swf.bar)`). Le fichier est alors chargé.

## Structures des fichiers de périphériques pour les différentes versions de BREW

Les structures de fichiers des périphériques qui prennent en charge BREW 2.x et BREW 3.x sont différentes. Il est important de bien comprendre ces différences avant d'utiliser l'assistant de publication BREW pour spécifier la structure des fichiers de votre application BREW.

Le système de fichiers des périphériques BREW 2.x est structuré comme suit :

- Tous les dossiers d'applications sont stockés dans le dossier racine, appelé brew/.
- Les fichiers d'application (y compris les fichiers SIG) sont stockés dans le dossier de l'application (les sous-dossiers ne sont pas autorisés).
- Les fichiers MIF sont stockés dans le dossier racine, brew/.

- Les fichiers partagés, tels que les images et les fichiers BAR, sont stockés dans le dossier brew/shared. Vous pouvez également stocker ces fichiers partagés dans le dossier qui contient les fichiers de l'application principale.

Exemple :

brew/*appname*/*appname.mod*

brew/*appname*/*appname.sig*

brew/*nomappli.mif*

brew/shared/*fichier.xxx*, où xxx est l'extension des fichiers de ressources, y compris des graphiques, des vidéos, des sons et des fichiers BAR.

ou

brew/*nomappli*/*fichier.xxx*, où xxx est l'extension des fichiers de ressources, y compris des graphiques, des vidéos, des sons et des fichiers BAR.

Le système de fichiers des périphériques BREW 3.x est structuré comme suit :

- Tous les répertoires d'application sont stockés dans le dossier brew/mod.
- Les fichiers d'application (y compris les fichiers SIG) sont stockés dans le dossier de l'application (les sous-dossiers ne sont pas autorisés).
- Les fichiers MIF sont stockés dans le dossier brew/mif.
- Vous pouvez stocker les fichiers partagés, par exemple les images et les fichiers BAR, dans le dossier brew/shared. Vous pouvez également stocker ces fichiers partagés dans le dossier qui contient les fichiers de l'application principale.

Exemple :

brew/**mod**/*appname*/*appname.mod*

brew/**mod**/*appname*/*appname.sig*

brew/**mif**/*nomappli.mif*

brew/**partagé**/*fichier.xxx*, où xxx est l'extension des fichiers de ressources, y compris des graphiques, des vidéos, des sons et des fichiers BAR.

ou

brew/**mod**/*nomappli*/*fichier.xxx*, où xxx est l'extension des fichiers de ressources, y compris des graphiques, des vidéos, des sons et des fichiers BAR.

## Organisation des tâches de publication et de test des fichiers Flash Lite pour BREW

L'organisation des tâches de publication et de test des fichiers Flash Lite pour BREW est la suivante :

- Dans Flash, ouvrez l'assistant de publication BREW et suivez les instructions apparaissant à l'écran pour définir les paramètres appropriés de votre application BREW et publier les fichiers.
- Dans le kit de développement BREW, servez-vous du simulateur BREW pour tester l'application.

## Utilisation de l'assistant de publication BREW pour publier les fichiers d'application

- 1 Dans Flash, sélectionnez Fichier > Paramètres de publication.  
La boîte de dialogue Paramètres de publication s'affiche.
- 2 Dans l'onglet Formats, sélectionnez Flash (.swf), si cette option n'est pas déjà sélectionnée.  
Vous pouvez désactiver tous les autres types de fichiers car seul le fichier SWF est nécessaire pour générer l'applet BREW.
- 3 Dans la liste Version de l'onglet Flash, sélectionnez une version de lecteur prise en charge (Flash Player 7 ou une version antérieure, ou n'importe quelle version de Flash Lite), puis la version de votre choix dans la liste des versions ActionScript (toutes les versions ActionScript sont prises en charge).
- 4 Dans la liste Post-processeur placée au bas de l'onglet Flash, sélectionnez Flash Lite for BREW.

**Remarque :** Si cette liste n'apparaît pas dans l'onglet Flash, consultez les instructions de la section « [Configuration de votre système pour BREW](#) » à la page 99 pour vous assurer que vous avez correctement installé Flash Lite 2.1 pour BREW. Cette liste n'apparaît qu'après l'installation du logiciel.

- 1 Cliquez sur Paramètres, à droite de la liste des post-processeurs pour afficher l'assistant de publication BREW.
- 2 (Obligatoire) Utilisez les champs de saisie de la fenêtre Identify Applet pour spécifier un ID de classe unique et le nom de votre applet BREW.

Le tableau suivant rassemble des informations détaillées sur ces champs. Après avoir indiqué l'ID de classe et le nom de l'applet, cliquez sur Suivant pour continuer.

Nom du champ	Contenu du champ
Class ID	Ce champ est prérenseigné par défaut à l'aide d'une valeur générée de façon aléatoire. Remplacez cette valeur par votre ID de classe certifié si vous en avez un. Un ID de classe est un code d'identification unique de 32 bits (valeur hexadécimale à 8 chiffres), utilisé par le mécanisme de création de l'interface de BREW. Dans le cas d'un test sur un système local, vous pouvez utiliser la valeur générée par défaut de façon aléatoire (sous réserve qu'elle soit unique), mais QUALCOMM doit publier un ID unique pour les applications destinées à la distribution. Pour plus d'informations sur l'obtention de fichiers d'ID de classe, consultez le site Web de BREW.
Applet Name	<p>Ce champ est prérenseigné à l'aide de la portion racine du nom du fichier FLA actuellement actif. Par exemple, si le fichier FLA actif est appelé BREW fla, le nom de l'applet est BREW. Le nom saisi dans ce champ apparaît dans le gestionnaire d'application de BREW Application Manager sur le périphérique visé.</p> <p>Respectez les consignes suivantes pour choisir le nom des applets :</p> <ul style="list-style-type: none"> <li>• Les noms des applets doivent être en minuscules et commencer par un caractère alphabétique.</li> <li>• Les nombres sont autorisés, sauf pour le premier caractère.</li> <li>• Le caractère de soulignement est le seul caractère spécial autorisé et ne doit pas correspondre au premier caractère du nom.</li> <li>• La longueur maximale du nom dépend du périphérique.</li> </ul> <p>Pour plus d'informations, consultez les caractéristiques techniques des périphériques sur le site Web de BREW.</p>

- 3 (Facultatif) Utilisez la fenêtre Include Applet Information pour saisir les informations que vous souhaitez publier avec votre applet BREW.



Les informations saisies ici sont stockées dans le fichier MIF de l'applet. Le tableau suivant rassemble des informations détaillées sur chaque champ. Lorsque la saisie de vos informations est terminée, cliquez sur Suivant pour continuer.

Nom du champ	Contenu du champ
Author	Entrez le nom de l'auteur de l'applet (23 caractères maximum).
Version	Entrez la version de l'applet (23 caractères maximum).
Copyright	Entrez les informations de copyright (23 caractères maximum).
Domain URL	Entrez le nom de domaine pleinement qualifié de votre applet, en terminant par une barre oblique (exemple : <a href="http://www.exemple.com/">http://www.exemple.com/</a> ). Le domaine de l'applet contrôle les informations auxquelles l'applet a accès au moment de l'exécution. Dans les versions précédentes de Flash Player, les fichiers SWF provenant de domaines similaires (par exemple, <a href="http://www.exempleapps.com">www.exempleapps.com</a> et <a href="http://www.exemplejeux.com">www.exemplejeux.com</a> ) pouvaient librement communiquer entre eux et avec d'autres documents. Dans Flash Player 7 et les versions ultérieures, le domaine des données accessibles doit correspondre exactement au domaine du fournisseur de données pour qu'ils puissent communiquer. Pour plus d'informations sur les domaines et la sécurité, consultez les sections de la documentation Flash abordant la sécurité, la sécurité inter-domaines et l'octroi d'accès aux données entre fichiers SWF inter-domaines.
Type d'application (application ou économiseur d'écran)	Indique si l'applet BREW en cours de création est une application ou un écran de veille. Pour plus d'informations sur la façon dont les écrans de veille diffèrent des applications, visitez le site Web de BREW en effectuant la navigation suivante : Developer FAQs > BREW Tools > MIF Settings > Screensaver.

- 4 (Facultatif pour le test de votre application, mais obligatoire pour les applications devant subir les tests TBT, True BREW Testing.) Utilisez la fenêtre Applet Icons pour entrer ou repérer le nom et l'emplacement d'une ou plusieurs icônes pour votre applet.

Les types de fichiers pris en charge sont : JPEG, BMP, PNG et BCI (BREW Compressed Image). Ces icônes s'affichent avec le nom de l'applet dans le gestionnaire d'application de BREW (Application Manager) sur le périphérique visé. Lorsque vous avez spécifié toutes vos icônes, cliquez sur Suivant pour continuer.

Nom du champ	Contenu du champ
Small icon	Taille de l'image pour une petite icône (affichée sur le périphérique). Taille maximale : 16 x 16.
Medium icon	Taille de l'image pour une icône moyenne (affichée sur le périphérique). Taille maximale : 26 x 26.
Large icon	Taille de l'image pour une grande icône (affichée sur le serveur de distribution et certains périphériques). Taille maximale : 65 x 42.

- 5 Utilisez la fenêtre Define Output Settings pour indiquer les emplacements dans lesquels doivent être publiés les fichiers pour le simulateur BREW et le périphérique visé.

Ces champs sont renseignés avec des valeurs par défaut issues de l'emplacement de votre fichier SWF, combiné au nom de l'applet et au nom des dossiers (simulateur et périphérique) qui indiquent les fichiers à utiliser pour les tests et les fichiers à charger sur le périphérique. Vous pouvez accepter les valeurs par défaut ou spécifier des emplacements de sorties différents. Lorsque vous avez terminé, cliquez sur Suivant pour continuer.

Nom du champ	Contenu du champ
For Simulator <ul style="list-style-type: none"> <li>MIF Folder</li> <li>Applet Folder</li> </ul>	Contient les fichiers dont vous avez besoin pour tester votre applet à l'aide du simulateur du kit de développement BREW. Pour plus d'informations sur les emplacements de fichiers prévus et la structure des dossiers, consultez la documentation du simulateur BREW sur le site Web de BREW.
For Device	Contient les fichiers que vous chargez sur le périphérique BREW. Pour plus d'informations sur les différences de structures de dossiers selon la version de BREW prise en charge par le périphérique visé, consultez la section « <a href="#">Téléchargement des fichiers sur les périphériques BREW</a> » à la page 112.

- 6 Utilisez l'écran Summary of Your Selections pour vérifier les paramètres de l'applet que vous avez sélectionnée.

Pour modifier ces paramètres, servez-vous du bouton Back pour revenir aux fenêtres précédentes. Lorsque vos paramètres sont définis, cochez ou décochez la case située au bas de l'écran pour indiquer si l'assistant doit s'afficher de nouveau lorsque vous publiez, puis cliquez sur Finish pour enregistrer vos paramètres.

Nom du champ	Contenu du champ
Summary Of Your Selections	Présente la liste de tous vos paramètres à passer en revue.
Do Not Show This Wizard On Publish	Permet d'indiquer si l'assistant doit de nouveau être affiché lorsque vous publiez les fichiers de l'applet ou si le processus de publication doit commencer sans que l'assistant ne réapparaisse. Notez que, quel que soit ce paramètre, vous pouvez toujours accéder à l'assistant en affichant la boîte de dialogue Paramètres de publication et en cliquant sur le bouton Paramètres, à droite de la liste des post-processeurs.

- 7 Cliquez sur OK pour enregistrer vos paramètres, puis sélectionnez une nouvelle fois Fichier > Paramètres de publication. La boîte de dialogue Paramètres de publication s'affiche.

**Remarque :** Cette étape est nécessaire uniquement lorsque vous sélectionnez pour la première fois Flash Lite pour BREW dans la liste des post-processeurs d'un fichier donné. Par la suite, vous pouvez publier sans d'abord cliquer sur OK.

- 8 Cliquez sur Publier pour publier vos fichiers aux emplacements définis dans l'assistant.

## Test de l'application à l'aide du simulateur BREW

Avant de charger votre application sur le périphérique, vous devez la tester à l'aide du simulateur du kit de développement BREW. Pour plus d'informations sur cet outil, consultez la documentation du simulateur BREW disponible sur le site Web de BREW.

- 1 Naviguez jusqu'au dossier Simulator de l'applet que vous créez à l'aide de l'assistant de publication BREW.
- 2 Ouvrez ce dossier et copiez l'ensemble de son contenu dans le dossier d'exemples du kit de développement BREW.

**Remarque :** Si vous spécifiez le dossier d'exemples du kit de développement BREW comme emplacement des fichiers de sortie du simulateur à l'étape de publication des fichiers par l'assistant de publication BREW, les fichiers sont déjà placés dans le dossier approprié et la copie n'est pas nécessaire.

- 1 Sélectionnez Démarrer > Programmes > BREW SDK v<votre version>, puis sélectionnez Simulator for BREW pour ouvrir l'application.
- 2 Dans le simulateur, sélectionnez File > Load Device, puis le pack de périphérique correspondant au combiné visé.
- 3 Dans l'onglet des propriétés du simulateur, définissez le dossier de votre application en tant qu'Applet Directory.
- 4 Servez-vous des outils et des fonctions du simulateur pour tester votre application.

## Utilisation des packs de périphérique

En tant que développeur BREW authentifié, vous pouvez télécharger des packs propres aux périphériques que vous prévoyez d'utiliser à partir de l'espace réservé aux développeurs, Developer Resources, situé sur le site Web BREW. Il est possible que le pack de périphérique du simulateur ne corresponde pas exactement à votre véritable périphérique. Par exemple, le pack de périphérique du Samsung SCH-A950 présente une touche BACK, nommée CLR sur le véritable combiné. De même, le pack de périphérique du LG VX9800 ne prend pas en charge la saisie au moyen des touches alphabétiques du clavier, contrairement au véritable périphérique. Pour plus d'informations sur les packs de périphériques spécifiques, les mises à jour ou tout problème éventuel, contactez QUALCOMM.

Si vous n'êtes pas enregistré en tant que développeur BREW authentifié, vous pouvez utiliser les packs de périphérique par défaut téléchargés en même temps que le kit de développement. Toutefois, ces packs par défaut ayant moins de mémoire que les packs spécifiques aux périphériques, il vous faudra peut-être sélectionner le pack présentant le plus de mémoire (Device Pack 3 au moment de la rédaction de ce document), ou augmenter la mémoire jusqu'à environ 10 000 000 octets pour éviter les erreurs et les problèmes de performance.

### Augmentation de la mémoire pour une session du simulateur

- 1 Ouvrez le simulateur et sélectionnez l'onglet Device Details.
- 2 Localisez la section Memory.
- 3 Augmentez la valeur de Heap Size (en octets). Cette valeur est appelée RAM Size dans certaines versions du simulateur.

### Maintien de l'augmentation de mémoire entre deux sessions du simulateur

- 1 Ouvrez le fichier DSD du pack de périphérique à utiliser dans un éditeur de texte (devicepack<x>.dsd, où x représente le numéro du pack).
- 2 Repérez le paramètre approprié et augmentez la valeur de Text :

```
<String Id="24" Name="IDS_DD_HW_RAM_SIZE">
  <Text>1048576</Text>
```

- 3 Enregistrez et fermez le fichier.

Vous pouvez également rencontrer des difficultés dans l'utilisation du simulateur si l'espace alloué au dossier d'exemples et à ses sous-dossiers est insuffisant pour le bon fonctionnement de votre application. Vous pouvez dans ce cas supprimer manuellement les fichiers inutiles dans ces dossiers ou augmenter la quantité d'espace qui leur est allouée.

### Maintien de l'augmentation de la taille des fichiers d'exemples entre deux sessions du simulateur

- 1 Ouvrez le fichier de configuration dans un éditeur de texte et modifiez la valeur appropriée :
  - Pour le Samsung SCH-A950 : Le fichier de configuration est SCH-A950.qsc. La valeur à modifier est indiquée en gras :

```
FS_LIMITS_PER_MODULE 65535 15204352
```

- Pour le LG9800 : Le fichier de configuration est LG-VX9800.dsd. La valeur à modifier est indiquée en gras :

```
<String Id="20" Name="IDS_DD_HW_EFS_SIZE">
  <Text>47000000</Text>
  <Comment>50MB</Comment>
</String>
```

- 2 Enregistrez et fermez le fichier.

## Ressources supplémentaires

Pour plus d'informations sur le simulateur BREW, consultez les sources suivantes :

Nom et type du document	Emplacement
<i>Starting with BREW</i> , « About the BREW Tools Suite » (PDF)	Accédez au site Web de BREW, puis sélectionnez Developer Home > Application Development Documentation.
BREW Simulator (Aide)	Ouvrez le simulateur et cliquez sur Help dans la barre de menus pour afficher les fichiers d'aide.

## Téléchargement des fichiers sur les périphériques BREW

### A propos du téléchargement d'applications Flash Lite pour BREW

Après avoir créé une application Flash Lite et utilisé l'assistant de publication BREW pour Flash Lite afin de créer des fichiers compatibles BREW, chargez votre application sur un périphérique BREW pris en charge et testez-la.

Dans l'idéal, vous avez déjà visé ce périphérique dès la conception initiale et testé votre application par le biais de l'émulateur Adobe Device Central et du simulateur BREW. Il est cependant très important de tester votre application minutieusement sur le périphérique visé afin d'être certain qu'il fonctionnera comme prévu.

Pour pouvoir charger votre application, vous devez disposer des éléments suivants :

- Un périphérique prenant en charge Flash Lite pour BREW (lors de la rédaction de ce document, un Samsung SCH-A950 ou un LG VX9800).
- Un câble série ou USB permettant de charger votre application de l'ordinateur vers le périphérique (généralement disponible auprès du fabricant du périphérique).
- L'outil AppLoader (disponible pour être téléchargé avec la suite d'outils BREW par les développeurs authentifiés depuis le site Web de BREW).
- Les fichiers d'application, avec structure et formats de fichiers requis (créés à l'aide de l'assistant de publication BREW pour Flash Lite).

Voici les fichiers d'application requis :

- *appname*.SIG (disponible pour le téléchargement des développeurs authentifiés depuis le générateur de signatures de tests de QUALCOMM)
- Le fichier *nomappli*.MOD (créé par Flash Lite Publisher pour BREW).
- Le fichier *nomappli*.MIF (créé par Flash Lite Publisher pour BREW).
- Tous les sons, vidéos, images et fichiers BAR requis par votre application.

Vous trouverez des informations complètes sur la structure et les formats de fichiers requis pour le périphérique visé dans la section « [Structures des fichiers de périphériques pour les différentes versions de BREW](#) » à la page 106.

### Organisation des tâches de chargement des applications

Cette procédure part du principe que vous avez déjà installé les pilotes USB appropriés à votre périphérique sur votre ordinateur. Pour plus d'informations, consultez la section « [Configuration de votre système pour BREW](#) » à la page 99.

Pour charger et tester des applications Flash Lite sur des périphériques BREW, procédez comme suit :

- 1 Connectez le combiné à l'ordinateur à l'aide d'un câble.
- 2 Ouvrez AppLoader sur l'ordinateur :
  - a Copiez les fichiers de l'application dans les emplacements appropriés pour le périphérique. Pour plus d'informations, consultez la section « [Structures des fichiers de périphériques pour les différentes versions de BREW](#) » à la page 106.
  - b Au besoin, copiez l'extension Flash Lite et les fichiers du lecteur sur le périphérique (la première fois uniquement).
  - c Redémarrez le périphérique.
- 3 Testez l'application sur le périphérique.

La suite de cette section reprend en détail chacune de ces tâches.

## Téléchargement des fichiers de l'extension Flash Lite sur le périphérique (première utilisation uniquement)

Avant de tester vos applications Flash Lite pour BREW sur un périphérique donné, copiez l'extension Flash Lite Extension sur ce dernier. Cette opération n'a lieu qu'une seule fois, lors de l'acquisition du périphérique pour le test. Les utilisateurs de votre application recevront ces fichiers dans le cadre du téléchargement de votre application complète depuis le système BDS ou le serveur ADS, mais les développeurs doivent copier ces fichiers manuellement. La liste des fichiers est présentée ci-après. Suivez les instructions de la suite de cette section pour charger les fichiers sur votre périphérique à l'aide de l'outil AppLoader. Notez que les emplacements de destination des copies des fichiers diffèrent pour les périphériques BREW 2.x et 3.x. Pour plus d'informations, consultez la section « [Structures des fichiers de périphériques pour les différentes versions de BREW](#) » à la page 106.

Fichier d'extension	Source
flashlite_2_1.sig	Créés automatiquement par des développeurs au moyen d'un outil disponible à partir du site Web de BREW.
flashlite_2_1.mif	Disponible en téléchargement depuis le site Web de BREW.
flashlite_2_1.mod	Disponible en téléchargement depuis le site Web de BREW.
brewsoplayer.sig	Créés automatiquement par des développeurs au moyen d'un outil disponible à partir du site Web de BREW.
brewsoplayer.mif	Disponible en téléchargement depuis le site Web de BREW.
brewsoplayer.mod	Disponible en téléchargement depuis le site Web de BREW.

## Conseils et meilleures pratiques relatives à l'outil AppLoader

Voici une liste de conseils d'utilisation de l'outil AppLoader :

- Lorsque vous créez un dossier sur le périphérique, n'utilisez pas de barre oblique inversée (\).
- N'exécutez aucune application sur le périphérique pendant la copie des fichiers.
- Redémarrez le périphérique après avoir utilisé AppLoader pour apporter des modifications.
- Evitez de remplacer les fichiers dans AppLoader ; supprimez plutôt les anciennes versions de ces fichiers sur le périphérique et remplacez-les par de nouvelles copies.

- Ne chargez pas de fichiers ou de dossiers dont les noms se composeraient uniquement de chiffres. Vous ne pourriez alors plus supprimer ces fichiers et dossiers sur certains périphériques.
- Ne commencez pas de nom de fichier ou de dossier par les caractères « shared ». « Shared » est un mot réservé dans BREW et vous ne pourrez pas charger les fichiers ou les dossiers commençant par ces caractères.

## Chargement d'applications sur un périphérique BREW 2.x

Les instructions suivantes décrivent le chargement d'une application Flash Lite pour BREW sur un Samsung SCH-A950, ce qui inclut la plate-forme BREW version 2.x.

Cette procédure part du principe que vous avez déjà installé les pilotes USB appropriés à votre périphérique sur votre ordinateur. Pour plus d'informations, consultez la section « [Configuration de votre système pour BREW](#) » à la page 99.

Cette procédure suppose également que vous avez déjà chargé l'extension Flash Lite et les fichiers du lecteur Flash Lite requis sur le périphérique. Cette opération n'a lieu qu'une seule fois, lors de votre première utilisation du périphérique. Pour plus d'informations, consultez la section « [Téléchargement des fichiers de l'extension Flash Lite sur le périphérique \(première utilisation uniquement\)](#) » à la page 113.

- 1 Utilisez le câble de données fourni par le fabricant du combiné pour relier ce dernier à un port COM de l'ordinateur sur lequel sont stockés vos fichiers d'application BREW.
- 2 Naviguez jusqu'au dossier contenant les fichiers d'application et vérifiez que la structure des répertoires est appropriée pour le périphérique visé.  
  
Pour toute information concernant la structure des périphériques BREW 2.x, consultez la section « [Structures des fichiers de périphériques pour les différentes versions de BREW](#) » à la page 106.
- 3 Sélectionnez Démarrer > Programmes > BREW Tools Suite <dernière version> > BREW Apploader pour démarrer l'outil AppLoader. L'outil AppLoader ouvre une fenêtre de connexion.
- 4 Sélectionnez le numéro du port COM auquel votre périphérique est connecté (si vous ne savez pas quel port sélectionner, prenez connaissance de la remarque suivante), puis la version BREW 2.x pour votre périphérique et cliquez sur OK pour vous connecter à ce dernier.

**Remarque :** Pour connaître le port COM auquel votre périphérique est connecté, examinez les propriétés du port correspondant au périphérique installé dans le Gestionnaire de périphériques de Windows. Si vous connectez toujours le même modèle de périphérique au même port COM, ce numéro ne change jamais. Etant donné qu'il change lorsque vous utilisez un modèle de périphérique différent, pensez à toujours vérifier le numéro de port COM lorsque vous connectez un nouveau modèle de périphérique.

Lorsque vous êtes connecté, l'outil AppLoader affiche le système de fichiers BREW du périphérique.

- 1 Pour charger vos fichiers d'application (fichiers MOD, SIG et ressources) sur le combiné, utilisez l'une des méthodes suivantes :
  - Sélectionnez File > Directory > New Directory pour créer un dossier portant le même nom que votre application, puis copiez vos fichiers d'application dans ce dossier. Pour plus d'informations, consultez la section « [Structures des fichiers de périphériques pour les différentes versions de BREW](#) » à la page 106.
  - Faites glisser votre dossier d'application depuis l'explorateur de Windows jusqu'à la fenêtre de l'outil AppLoader.
- 2 Redémarrez le périphérique.  
  
Le périphérique doit être redémarré avant d'exécuter votre application. Pour redémarrer le périphérique, effectuez l'une des opérations suivantes :
  - Sélectionnez Device > Reset dans le menu de l'outil AppLoader de BREW.
  - Maintenez la touche End du combiné enfoncée.

- 3 Naviguez jusqu'à Get it Now > Get Going, puis sélectionnez le nom de votre application pour la lancer dans l'environnement BREW.

## Chargement d'applications sur un périphérique BREW 3.x

Les instructions suivantes décrivent le chargement d'une application Flash Lite pour BREW sur un LG VX9800, ce qui inclut la plate-forme BREW version 3.x.

Cette procédure part du principe que vous avez déjà installé les pilotes USB appropriés à votre périphérique sur votre ordinateur. Pour plus d'informations, consultez la section « [Configuration de votre système pour BREW](#) » à la page 99.

Cette procédure suppose également que vous avez déjà chargé l'extension Flash Lite et les fichiers du lecteur Flash Lite requis sur le périphérique. Cette opération n'a lieu qu'une seule fois, lors de votre première utilisation du périphérique. Pour plus d'informations, consultez la section « [Téléchargement des fichiers de l'extension Flash Lite sur le périphérique \(première utilisation uniquement\)](#) » à la page 113.

- 1 Utilisez le câble fourni par le fabricant du combiné pour relier ce dernier à un port COM de l'ordinateur sur lequel sont stockés vos fichiers d'application BREW.
- 2 Naviguez jusqu'au dossier contenant les fichiers d'application et vérifiez que la structure des répertoires est appropriée pour le périphérique visé.

Pour toute information concernant la structure des périphériques BREW 3.x, consultez la section « [Structures des fichiers de périphériques pour les différentes versions de BREW](#) » à la page 106.

- 3 Sélectionnez Démarrer > Programmes > BREW Tools Suite <dernière version> > BREW Apploader pour démarrer l'outil AppLoader.
- 4 L'outil AppLoader ouvre une fenêtre de connexion.

Sélectionnez le numéro du port COM auquel votre périphérique est connecté, puis la version BREW 3.x pour votre périphérique et cliquez sur OK pour vous connecter à ce dernier.

**Remarque :** Pour connaître le port COM auquel votre périphérique est connecté, examinez les propriétés du port correspondant au périphérique installé dans le Gestionnaire de périphériques de Windows. Si vous connectez toujours le même modèle de périphérique au même port COM, ce numéro ne change jamais. Etant donné qu'il change lorsque vous utilisez un modèle de périphérique différent, pensez à toujours vérifier le numéro de port COM lorsque vous connectez un nouveau modèle de périphérique.

Lorsque vous êtes connecté, l'outil AppLoader affiche le système de fichiers BREW du périphérique.

- 1 Pour charger vos fichiers d'application (fichiers MOD, SIG et ressources) sur le combiné, utilisez l'une des méthodes suivantes :
  - Sélectionnez File > New > Directory > New Directory pour créer un dossier portant le même nom que votre application, puis copiez vos fichiers d'application dans ce dossier. Pour plus d'informations, consultez la section « [Structures des fichiers de périphériques pour les différentes versions de BREW](#) » à la page 106.
  - Faites glisser votre dossier d'application depuis l'explorateur de Windows jusqu'à la fenêtre de l'outil AppLoader.
- 2 Pour redémarrer le périphérique, effectuez l'une des opérations suivantes :
  - Sélectionnez Device > Reset dans le menu de l'outil AppLoader de BREW.
  - Maintenez la touche End du combiné enfoncée.
- 3 Naviguez jusqu'à Get it Now > Get Going, puis sélectionnez le nom de votre application pour la lancer dans l'environnement BREW.

Sur certains périphériques, votre application ne s'affichera pas dans le menu Get Going standard. Pour afficher votre application sur ces périphériques, procédez comme suit :

- a Dans l'écran d'accueil Verizon, appuyez sur Select/OK pour accéder au menu.
- b Appuyez sur la touche zéro (0) pour accéder au menu de services.
- c Entrez le mot de passe par défaut, c'est-à-dire six zéros (000000).
- d Appuyez sur la touche neuf (9) pour afficher Get it Now.
- e Appuyez sur la touche g pour accéder à la fenêtre dans laquelle s'affichent les applications Flash Lite.

## Test des applications sur le périphérique

Avant de lancer et de tester votre application sur le périphérique, vous devez vous procurer une signature de test auprès de QUALCOMM. Les développeurs authentifiés peuvent utiliser un outil de type Web appelé Testsig sur le réseau Extranet des développeurs BREW pour générer des signatures de test. Pour obtenir des informations particulières sur la création et l'utilisation de votre Testsig, visitez le site Web de BREW. Vous trouverez des informations détaillées sur le test de vos applets BREW afin d'être certain qu'elles seront finalement compatibles avec les normes du test TBT (True BREW Testing) sur le réseau Extranet des développeurs BREW.

## Ressources supplémentaires

Pour plus d'informations sur l'outil AppLoader, consultez les sources suivantes :

Nom et type du document	Emplacement
<i>Starting with BREW</i> , « About the BREW Tools Suite » (PDF)	Accédez au site Web de BREW, puis sélectionnez Developer Home > Application Development Documentation
BREW)_Apploader.chm (Aide)	C:\Program Files\BREW Tools Suite <dernière version>\BREWApploader (par défaut)
Générateur BREW Testing	Accédez au site Web de BREW, puis naviguez jusqu'à la page de téléchargement du kit de développement logiciel (SDK)



# Chapitre 9 : Test du contenu Flash Lite

Adobe Flash CS4 Professional comprend un émulateur Adobe Flash Lite qui est intégré à Adobe Device Central CS4 : ce composant permet de tester directement dans l'outil de programmation l'affichage et l'exécution de votre application. Lorsque l'application s'exécute dans l'émulateur vous convient, vous pouvez la tester sur un périphérique réel.

## Présentation du test

### Fonctionnalités de test Flash Lite

Les fonctionnalités de test Flash Lite de Flash CS4 Professional font partie d'Adobe Device Central et comprennent une vaste base de données de profils de périphériques ainsi qu'un émulateur de périphérique. Device Central fonctionne également avec de nombreux autres produits Adobe, tels que Adobe Creative Suite® et Adobe Dreamweaver®.

L'émulateur Adobe Device Central permet de prévisualiser votre contenu Flash Lite dans l'outil de programmation de Flash. L'émulateur est configuré pour reproduire le comportement et l'aspect du périphérique cible. L'interface Device Central permet de sélectionner et de gérer les périphériques cibles et de spécifier le type de contenu Flash Lite cible de votre application, par exemple une sonnerie personnalisée, un navigateur ou une application autonome. Dans Device Central, chaque combinaison de périphérique test et de type de contenu Flash Lite définit une configuration de périphérique présentant les fonctionnalités disponibles pour votre application, notamment les formats audio pris en charge, la possibilité d'établir des connexions réseau ainsi que d'autres fonctionnalités. Pour plus d'informations sur les types de contenus Flash Lite, consultez la section « [Types de contenus pris en charge](#) » à la page 119.

Pour plus d'informations, y compris des précisions sur les interactions avec l'émulateur, consultez l'Aide de Device Central.

### Fonctionnalités de test non prises en charge par l'émulateur

L'émulateur Adobe Device Central ne prend pas en charge toutes les fonctionnalités disponibles dans la fenêtre de test standard (ordinateur de bureau). Vous trouverez ci-dessous la liste des fonctionnalités qui ne sont pas disponibles dans l'émulateur Adobe Device Central, mais qui le sont dans la fenêtre de test de la version de bureau de Flash :

- Les fonctionnalités Lister les variables (Débogage > Lister les variables) et Lister les objets (Débogage > Lister les objets) ;
- Les fonctionnalités Testeur de bande passante, Graphique de lecture en continu et Graphique image par image ;
- La commande de menu Affichage > Simuler le téléchargement ;
- Le débogueur ActionScript ;
- La commande de menu Affichage > Afficher les zones redessinées ;
- La barre d'outils Contrôleur (Fenêtre > Barre d'outils > Contrôleur).

## Test de texte en ligne dans Flash Lite 2.1 et version ultérieure

Vous ne pouvez pas tester la fonctionnalité de saisie de texte en ligne avec l'émulateur Adobe Device Central, vous devez donc la tester directement sur le périphérique. Lors de tests dans l'émulateur, vous devez modifier le contenu des champs texte de saisie à l'aide d'une boîte de dialogue modale qui s'affiche sur le contenu Flash Lite (cela signifie que l'émulateur fonctionne de la même façon que pour Flash Lite 1.x et Flash Lite 2.0). Pour plus d'informations sur le fonctionnement de la boîte de dialogue modale, consultez la section « [Utilisation des champs texte de saisie](#) » à la page 54.

## Utilisation de l'émulateur

### Appel d'Adobe Device Central

Pour démarrer l'émulateur, sélectionnez **Contrôle > Tester l'animation** dans l'outil de programmation Flash, de la même façon que pour obtenir un aperçu de votre contenu Flash sur un ordinateur de bureau. (Cependant, Adobe Device Central présente un aspect et des fonctionnalités qui diffèrent par rapport à la fenêtre de test du contenu de la version de bureau de Flash.) Sélectionnez **Contrôle > Tester l'animation** ou appuyez sur **Ctrl+Entrée** pour exécuter Device Central. Une barre de progression s'affiche pour indiquer que Flash exporte votre fichier SWF vers Device Central. Lorsque l'exportation est terminée, Device Central s'exécute, active l'émulateur et charge votre fichier SWF :

Dans Device Central, la liste **Ensemb. de périph.** regroupe tous les appareils enregistrés par Flash avec votre application. Par défaut, le premier périphérique du groupe est sélectionné pour l'émulation. Le type de contenu sélectionné est celui qui a été enregistré avec l'application au moment de sa création (pour en savoir plus sur les types de contenus Flash Lite, consultez la section « [Types de contenus pris en charge](#) » à la page 119).

Vous pouvez tester l'application en fonction de différents types de contenus et de différents périphériques. Le changement de type de contenu et l'ajout ou la suppression de périphériques dans la liste modifie automatiquement les paramètres définis dans Flash.

Pour tester le fonctionnement du contenu avec un autre périphérique, double-cliquez sur le périphérique de votre choix dans le haut ou le bas de la liste. Une icône rotative apparaît en regard du nouveau périphérique à tester, et l'interface de l'émulateur change pour représenter l'exécution de votre application dans l'appareil sélectionné.

### Définition des options de débogage de l'émulateur

Pendant l'exécution du contenu, l'émulateur Adobe Device Central peut transmettre des messages de débogage au panneau **Sortie de Flash** et afficher ces mêmes messages dans une représentation locale de ce panneau.

Les informations transmises sont les suivantes :

**Messages de suivi** générés par l'appel d'une fonction `trace()` dans votre application Flash Lite. Pour plus d'informations sur l'utilisation de `trace()`, consultez la section `trace()` dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*.

**Messages d'informations** contenant des informations générales relatives au périphérique de test sélectionné, à la taille du fichier SWF ou autres. Ces messages apparaissent dans le panneau d'alerte de l'émulateur.

**Messages d'avertissement** contenant des informations relatives aux problèmes de votre contenu Flash Lite susceptibles d'affecter la lecture.

Pour filtrer le type d'informations transmises par l'émulateur, procédez comme indiqué ci-dessous.

- 1 Sélectionnez **Contrôle > Tester l'animation**. Flash exporte votre application vers Adobe Device Central et l'affiche dans l'émulateur.

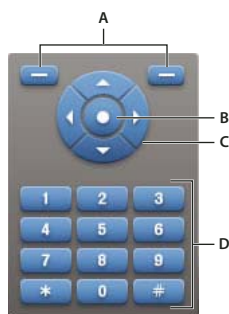
2 Sélectionnez Affichage > Sortie Flash > Afficher dans Device Central.

- Sélectionnez ou désélectionnez l'option Suivi.
- Sélectionnez ou désélectionnez l'option Informations.
- Sélectionnez ou désélectionnez l'option Avertissements.

## Interactions avec l'émulateur

Vous pouvez interagir avec le clavier de l'émulateur en utilisant la souris de votre ordinateur ou des raccourcis clavier. Vous pouvez interagir avec les touches suivantes du clavier de l'émulateur :

- Les touches numériques (0 à 9) et les touches astérisque (\*) et dièse (#)
- Le pavé de navigation (gauche, droite, bas, haut et sélection)
- Touches programmables Gauche et Droite



A. Touches programmables B. Touche de sélection C. Pavé de navigation D. Autres touches prises en charge

Vous pouvez cliquer directement sur le clavier de l'émulateur avec la souris ou bien utiliser les raccourcis clavier équivalents suivants :

- Les touches fléchées du clavier (gauche, droite, haut, bas) correspondent aux touches du pavé de navigation de l'émulateur.
- Les touches Entrée et Retour correspondent à la touche de sélection de l'émulateur.
- Les touches Page précédente et Page suivante correspondent respectivement aux touches programmables Gauche et Droite de l'émulateur.
- Les touches numériques de votre clavier correspondent aux touches numériques équivalentes du clavier de l'émulateur.

Pour toute information complémentaire sur les interactions possibles avec l'émulateur pour tester votre application, consultez l'aide en ligne d'Adobe Device Central.

## Types de contenus pris en charge

Flash Lite est installé sur de nombreux périphériques. Toutes les installations prennent en charge un ou plusieurs modes d'application, ou *types de contenus*. Par exemple, certains périphériques utilisent Flash Lite pour activer les fonctionnalités d'économiseurs d'écran et de sonneries animées. D'autres périphériques exploitent Flash Lite pour afficher du contenu Flash intégré dans des pages Web mobiles.

Le tableau suivant répertorie et décrit tous les types de contenus Flash Lite disponibles à l'heure actuelle. Pour obtenir des informations supplémentaires et actualisées sur la disponibilité des types de contenus Flash Lite, consultez la page Flash Enabled Mobile Device (Périphériques mobiles équipés de Flash) à l'adresse [www.adobe.com/go/mobile\\_supported\\_devices/](http://www.adobe.com/go/mobile_supported_devices/).

**Remarque :** Lors de la rédaction du présent manuel, Flash Lite 2.0 ne prend en charge que les contenus conçus pour le lecteur autonome.

Type de contenu Flash Lite	Description	Disponibilité
Carnet d'adresses	Utilise Flash Lite pour permettre aux utilisateurs d'associer un fichier SWF à une entrée dans l'application du carnet d'adresses de leur périphérique.	DoCoMo et VodafoneKK (Japon uniquement)
Alarme	Utilise Flash Lite pour permettre à l'utilisateur de sélectionner un fichier SWF devant être lu pour l'alarme de son périphérique.	KDDI et Vodafone (Japon uniquement)
Navigateur	Utilise Flash Lite pour rendre le contenu Flash intégré aux pages Web mobiles et affiché dans le navigateur Web du périphérique.	DoCoMo, KDDI et VodafoneKK (Japon uniquement)
Historique des appels	Utilise Flash Lite pour afficher une image ou une animation associée à chaque entrée du carnet d'adresses de l'utilisateur, ainsi que le nom et le numéro de téléphone correspondants.	KDDI (téléphones Casio uniquement)
Ecran d'appel	Utilise Flash Lite pour afficher une animation lorsque l'utilisateur reçoit un appel ou en passe un.	DoCoMo et KDDI (Japon uniquement)
Chaku Flash	Utilise Flash Lite pour permettre aux utilisateurs de sélectionner un fichier SWF devant être lu comme sonnerie pour les appels entrants.	KDDI (Japon uniquement)
Boîte de données	Utilise Flash Lite pour rendre du contenu Flash dans l'application Boîte de données du périphérique, ce qui permet à l'utilisateur de gérer et de prévisualiser des fichiers multimédia sur le périphérique.	DoCoMo, KDDI et VodafoneKK (Japon uniquement)
Dossier de données	Utilise Flash Lite pour rendre du contenu Flash dans l'application Dossier de données du périphérique, ce qui permet à l'utilisateur de gérer et de prévisualiser des fichiers multimédia sur le périphérique.	KDDI (Japon uniquement)
Menu Icône	Utilise Flash Lite pour permettre à l'utilisateur de sélectionner des menus d'icône personnalisés pour l'application de lancement du périphérique (utilisée par le type de contenu UILauncher).	KDDI (téléphones Casio uniquement)

Type de contenu Flash Lite	Description	Disponibilité
Visionneuse	Utilise l'application Visionneuse pour permettre à l'utilisateur de gérer et de prévisualiser des fichiers multimédias sur son périphérique, y compris des fichiers SWF.	DoCoMo (Japon uniquement)
Appel entrant	Utilise Flash Lite pour afficher une animation lorsque l'utilisateur reçoit un appel.	DoCoMo, KDDI et VodafoneKK (Japon uniquement)
Messagerie	Utilise Flash Lite pour afficher une animation lorsque l'utilisateur envoie ou reçoit un message électronique.	VodafoneKK (Japon uniquement)
Multimédia	Utilise Flash Lite pour prévisualiser des fichiers SWF (ainsi que d'autres formats multimédias).	KDDI (Japon uniquement)
Mes images	Utilise l'application Mes images pour permettre à l'utilisateur de gérer et de prévisualiser des fichiers SWF sur son périphérique, ainsi que d'autres formats d'image.	DoCoMo (Japon uniquement)
OpenEMIRO	Affiche du contenu Flash Lite lorsque le périphérique sort du mode d'attente. Ce type est identique au type de contenu Ecran de sortie de veille figurant sur d'autres périphériques.	KDDI (périphériques Casio uniquement)
Ecran de veille	Utilise Flash Lite pour afficher l'écran de veille du périphérique.	KDDI et Vodafone (Japon uniquement)
Lecteur SMIL	Utilise Flash Lite pour prévisualiser des fichiers SWF (ainsi que d'autres formats multimédias).	KDDI (Japon uniquement)
Lecteur autonome	Met Flash Lite à la disposition de l'utilisateur en tant qu'application autonome pour lui permettre de lancer et d'afficher des fichiers SWF arbitraires situés sur son périphérique ou reçus dans la boîte de réception de sa messagerie.	Disponible dans le monde entier pour les périphériques Symbian Series 60 et UIQ
Ecran d'attente	Utilise Flash Lite pour afficher l'écran d'attente du périphérique (ou papier peint).	DoCoMo et KDDI (Japon uniquement)
Ecran externe	Utilise Flash Lite pour afficher du contenu sur l'écran externe ou secondaire disponible sur certains téléphones à rabat.	KDDI (Japon uniquement)
UILauncher	Utilise Flash Lite pour le lanceur d'applications du périphérique.	Utilise Flash Lite pour afficher l'application de lancement du périphérique (autrement dit, l'application permettant à l'utilisateur de lancer d'autres applications)
Ecran de sortie de veille	Utilise Flash Lite pour afficher une animation au moment du démarrage du téléphone.	DoCoMo (Japon uniquement)

## Informations spécifiques à Flash Lite dans l'émulateur

Dans l'émulateur, des panneaux affichent des informations spécifiques à l'application Flash Lite. Ces panneaux sont alignés sur le côté droit de la fenêtre et vous pouvez les développer ou les réduire tout comme dans Flash.

Pour toute information complémentaire sur ces panneaux et leur utilisation, consultez l'aide en ligne d'Adobe Device Central.

## Taille d'écran et taille de scène disponible

Toutes les combinaisons de périphérique cible et de type de contenu Flash Lite déterminent, entre autres, la zone d'écran disponible qu'une application Flash Lite peut occuper. La zone de la scène disponible peut être égale ou inférieure à la taille de l'écran du périphérique.

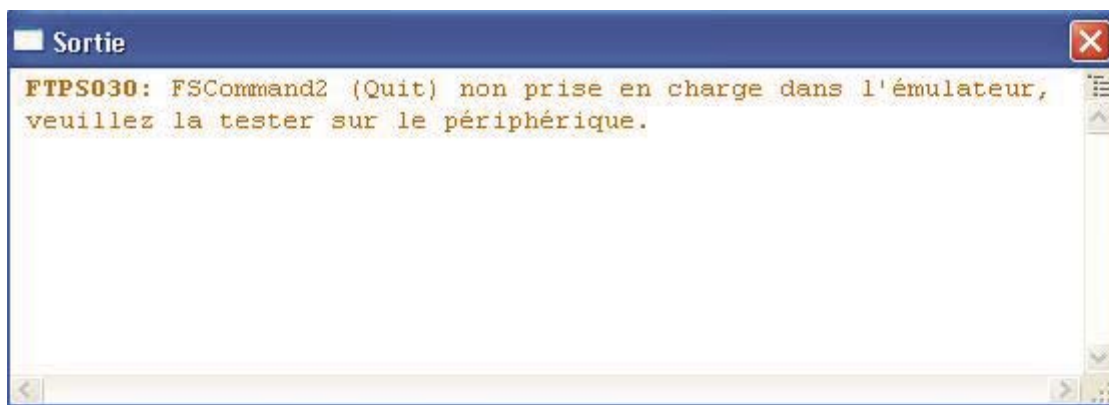
Par exemple, la zone de scène disponible pour un fichier SWF affiché en mode plein écran dans le lecteur autonome sur un périphérique de la gamme Series 60 de Nokia est égale à la taille de l'écran (176 x 208 pixels). Sur d'autres périphériques (comme ceux vendus au Japon), la zone de scène disponible pour un fichier SWF exécuté dans l'un des types de contenus spécialisés (comme Carnet d'adresses ou Economiseur d'écran) peut être inférieure à la taille totale de l'écran. Par exemple, la taille d'écran du Fujitsu 700i est de 240 x 320 pixels ; cependant, un fichier SWF exécuté dans l'application Carnet d'adresses du périphérique dispose d'une zone de scène de 96 x 72 pixels.

Si la taille de scène d'un fichier SWF est différente de la taille de la scène disponible, le lecteur Flash Lite redimensionne le contenu (proportionnellement) pour l'ajuster à l'espace disponible. Lorsque vous testez votre contenu dans l'émulateur, ce dernier vous signale également si la taille de scène de votre application est différente de celle qui est disponible.

Pour éviter tout problème de redimensionnement indésirable, Adobe recommande d'adapter les dimensions de la scène de votre document Flash à celles de la zone de scène disponible pour le type de contenu et le périphérique de test sélectionnés.

## Fonctionnalités Flash Lite non prises en charge par l'émulateur

L'émulateur ne prend pas en charge toutes les commandes ActionScript ni toutes les fonctionnalités du lecteur disponibles dans les applications Flash Lite qui s'exécutent sur un périphérique réel. Par exemple, l'émulateur ne prend pas en charge la fonctionnalité de lancement d'appels téléphoniques ou d'envoi de messages SMS. Si vous tentez d'utiliser une commande ou une fonctionnalité non prise en charge par l'émulateur, ce dernier génère un message dans le panneau Sortie, comme illustré ci-dessous :



Vous devez tester votre fichier SWF sur un périphérique réel pour vous assurer du bon fonctionnement de ces fonctionnalités.

L'émulateur Adobe Device Central ne prend pas en charge les commandes `fscommand()` et `fscommand2()` suivantes :

- `FullScreen`
- `GetFreePlayerMemory`
- `GetTotalPlayerMemory`
- `Launch`
- `Quit`
- `StartVibrate`
- `GetNetworkConnectStatus`
- `GetNetworkRequestStatus`
- `GetNetworkStatus`

## Lecture d'une vidéo de périphérique avec l'émulateur

Le lecteur Flash Lite utilise le gestionnaire vidéo par défaut du périphérique pour lire le contenu vidéo du fichier SWF et ne décode pas la vidéo de façon native. Cette pratique permet aux développeurs de contenus Flash Lite d'exploiter les formats vidéo pris en charge par le périphérique, tels que 3GPP, MPEG ou AVI. Consultez « [Utilisation de la vidéo](#) » à la page 79 pour obtenir de plus amples informations sur l'utilisation de vidéos dans Flash Lite.

L'émulateur Adobe Device Central utilise QuickTime Player pour le rendu des vidéos de périphérique pendant le test effectué avec l'outil de programmation de Flash. La dernière version de QuickTime Player (version 7 lors de la rédaction du présent document) prend en charge la lecture de différents formats vidéo de périphérique, ce qui inclut le 3GPP entre autres. Cependant, par défaut, QuickTime peut ne pas prendre en charge certains formats vidéo normalement compatibles avec un périphérique réel, ces formats ne seront donc pas lus dans l'émulateur Adobe Device Central. Il est donc essentiel de toujours tester votre contenu sur le périphérique réel.

Si la vidéo de périphérique n'est pas lue par défaut par QuickTime Player, essayez les opérations suivantes :

- Procédez à une mise à jour vers la dernière version de QuickTime Player.
- Si disponible, installez un codec (abréviation de compresseur-décompresseur) vidéo provenant d'un éditeur tiers et prenant en charge le format vidéo que vous utilisez.



# Chapitre 10 : Présentation du code ActionScript Flash Lite 2.x et 3.x

Les logiciels Macromedia Flash Lite 1.0 et Macromedia Flash Lite 1.1 d'Adobe, premières versions de Flash Lite, reposent sur le logiciel Macromedia Flash Player 4 d'Adobe. Les logiciels Macromedia Flash Lite 2.0 et Macromedia Flash Lite 2.1 d'Adobe ainsi qu'Adobe Flash Lite 3.0 et 3.1 reposent sur Macromedia Flash Player 7 d'Adobe, mais ils présentent les particularités suivantes :

- Flash Lite ne prend en charge certaines fonctionnalités que de façon partielle.
- Flash Lite dispose de fonctionnalités propres aux périphériques mobiles.

Ce document décrit les différences entre le code Adobe ActionScript pris en charge pour Flash Lite 2.0, 2.1, 3.0 et 3.1 (appelés collectivement 2.x et 3.x respectivement) et le code ActionScript pris en charge pour Flash Player 7.

## Éléments de langage et classes ActionScript non pris en charge, et pris en charge intégralement ou partiellement

### Accessibilité

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Accessibility et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
isActive() : Boolean	Oui	Non	Non	Non
updateProperties() : Void	Oui	Non	Non	Non

### arguments

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe d'arguments et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
callee:Object	Oui	Oui	Oui	Oui
caller:Object	Oui	Oui	Oui	Oui
length:Number	Oui	Oui	Oui	Oui

### Array

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Array et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
CASEINSENSITIVE:Number	Oui	Oui	Oui	Oui
concat([value:Object]) : Array	Oui	Oui	Oui	Oui
DESCENDING:Number	Oui	Oui	Oui	Oui
join([delimiter:String]) : String	Oui	Oui	Oui	Oui
length:Number	Oui	Oui	Oui	Oui
NUMERIC:Number	Oui	Oui	Oui	Oui
pop() : Object	Oui	Oui	Oui	Oui
push(value:Object) : Number	Oui	Oui	Oui	Oui
RETURNINDEXEDARRAY:Number	Oui	Oui	Oui	Oui
reverse() : Void	Oui	Oui	Oui	Oui
shift() : Object	Oui	Oui	Oui	Oui
slice([startIndex:Number], [endIndex:Number]) : Array	Oui	Oui	Oui	Oui
sort([compareFunction:Object], [options:Number]) : Array	Oui	Oui	Oui	Oui
sortOn(fieldName:Object, [options:Object]) : Array	Oui	Oui	Oui	Oui
splice(startIndex:Number, [deleteCount:Number], [value:Object]) : Array	Oui	Oui	Oui	Oui
toString() : String	Oui	Oui	Oui	Oui
UNIQUESORT:Number	Oui	Oui	Oui	Oui
unshift(value:Object) : Number	Oui	Oui	Oui	Oui

## AsBroadcaster

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe AsBroadcaster et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_listeners:Array [lecture seule]	Oui	Non	Oui	Oui
addListener(listenerObj:Object) : Boolean	Oui	Non	Oui	Oui
broadcastMessage(eventName:String) : Void	Oui	Non	Oui	Oui
initialize(obj:Object) : Void	Oui	Non	Oui	Oui
removeListener(listenerObj:Object) : Boolean	Oui	Non	Oui	Oui

## BevelFilter

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe BevelFilter et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
angle:Number	Oui	Non	Non	Non
BevelFilter([distance:Number], [angle:Number], [highlightColor:Number], [highlightAlpha:Number], [shadowColor:Number], [shadowAlpha:Number], [blurX:Number], [blurY:Number], [strength:Number], [quality:Number], [type:String], [knockout:Boolean])	Oui	Non	Non	Non
blurX:Number	Oui	Non	Non	Non
blurY:Number	Oui	Non	Non	Non
clone() : BevelFilter	Oui	Non	Non	Non
distance:Number	Oui	Non	Non	Non
highlightAlpha:Number	Oui	Non	Non	Non
highlightColor:Number	Oui	Non	Non	Non
knockout:Boolean	Oui	Non	Non	Non
quality:Number	Oui	Non	Non	Non
shadowAlpha:Number	Oui	Non	Non	Non
shadowColor:Number	Oui	Non	Non	Non
strength:Number	Oui	Non	Non	Non
type:String	Oui	Non	Non	Non

## BitmapData

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe BitmapData et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
applyFilter(sourceBitmap:BitmapData, sourceRect:Rectangle, destPoint:Point, filter:BitmapFilter) : Number	Oui	Non	Non	Non
BitmapData ((width:Number, height:Number, [transparent:Boolean], [fillColor:Number])	Oui	Non	Non	Oui
clone() : BitmapData	Oui	Non	Non	Oui
colorTransform(rect:Rectangle, colorTransform:ColorTransform) : Void	Oui	Non	Non	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
compare(otherBitmapData:BitmapData) : Object	Oui	Non	Non	Non
copyChannel(sourceBitmap:BitmapData, sourceRect:Rectangle, destPoint:Point, sourceChannel:Number, destChannel:Number) : Void	Oui	Non	Non	Oui
copyPixels(sourceBitmap:BitmapData, sourceRect:Rectangle, destPoint:Point, [alphaBitmap:BitmapData], [alphaPoint:Point], [mergeAlpha:Boolean]) : Void	Oui	Non	Non	Partielle
dispose() : Void	Oui	Non	Non	Oui
draw(source:Object, [matrix:Matrix], [colorTransform:ColorTransform], [blendMode:Object], [clipRect:Rectangle], [smooth:Boolean]) : Void	Oui	Non	Non	partielle
fillRect(rect:Rectangle, color:Number) : Void	Oui	Non	Non	Oui
floodFill(x:Number, y:Number, color:Number) : Void	Oui	Non	Non	Oui
generateFilterRect(sourceRect:Rectangle, filter:BitmapFilter) : Rectangle	Oui	Non	Non	Non
getColorBoundsRect(mask:Number, color:Number, [findColor:Boolean]) : Rectangle	Oui	Non	Non	Oui
getPixel(x:Number, y:Number) : Number	Oui	Non	Non	Oui
getPixel32(x:Number, y:Number) : Number	Oui	Non	Non	Oui
height:Number [read-only]	Oui	Non	Non	Oui
hitTest(firstPoint:Point, firstAlphaThreshold:Number, secondObject:Object, [secondBitmapPoint:Point], [secondAlphaThreshold:Number]) : Boolean	Oui	Non	Non	Oui
loadBitmap(id:String) : BitmapData	Oui	Non	Non	Oui
merge(sourceBitmap:BitmapData, sourceRect:Rectangle, destPoint:Point, redMult:Number, greenMult:Number, blueMult:Number, alphaMult:Number) : Void	Oui	Non	Non	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
noise(randomSeed:Number, [low:Number], [high:Number], [channelOptions:Number], [grayScale:Boolean]) : Void	Oui	Non	Non	Non
paletteMap(sourceBitmap:BitmapData, sourceRect:Rectangle, destPoint:Point, [redArray:Array], [greenArray:Array], [blueArray:Array], [alphaArray:Array]) : Void	Oui	Non	Non	Non
perlinNoise(baseX:Number, baseY:Number, numOctaves:Number, randomSeed:Number, stitch:Boolean, fractalNoise:Boolean, [channelOptions:Number], [grayScale:Boolean], [offsets:Object]) : Void	Oui	Non	Non	Non
pixelDissolve(sourceBitmap:BitmapData, sourceRect:Rectangle, destPoint:Point, [randomSeed:Number], [numberOfPixels:Number], [fillColor:Number]) : Number	Oui	Non	Non	Non
rectangle:Rectangle [read-only]	Oui	Non	Non	Oui
scroll(x:Number, y:Number) : Void	Oui	Non	Non	Non
setPixel(x:Number, y:Number, color:Number) : Void	Oui	Non	Non	Oui
setPixel32(x:Number, y:Number, color:Number) : Void	Oui	Non	Non	Oui
threshold(sourceBitmap:BitmapData, sourceRect:Rectangle, destPoint:Point, operation:String, threshold:Number, [color:Number], [mask:Number], [copySource:Boolean]) : Number	Oui	Non	Non	Non
transparent:Boolean [read-only]	Oui	Non	Non	Oui
width:Number [read-only]	Oui	Non	Non	Oui

## BitmapFilter

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe BitmapFilter et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
clone() : BitmapFilter	Oui	Non	Non	Non

## BlurFilter

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe BlurFilter et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
BlurFilter([blurX:Number], [blurY:Number], [quality:Number])	Oui	Non	Non	Non
blurX:Number	Oui	Non	Non	Non
blurY:Number	Oui	Non	Non	Non
clone() : BlurFilter	Oui	Non	Non	Non
quality:Number	Oui	Non	Non	Non

## Boolean

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Boolean et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
Boolean([value:Object])	Oui	Oui	Oui	Oui
toString() : String	Oui	Oui	Oui	Oui
valueOf() : Boolean	Oui	Oui	Oui	Oui

## Bouton

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Button et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_alpha:Number	Oui	Oui	Oui	Oui
_focusrect:Boolean	Oui	Oui	Oui	Oui
_height:Number	Oui	Oui	Oui	Oui
_highquality:Number	Déconseillé	Déconseillé	Déconseillé	Déconseillé
_name:String	Oui	Oui	Oui	Oui
_parent:MovieClip	Oui	Oui	Oui	Oui
_quality:String	Oui	Oui	Oui	Oui
_rotation:Number	Oui	Oui	Oui	Oui
_soundbuftime:Number	Oui	Oui	Oui	Oui
_target:String [lecture seule]	Oui	Oui	Oui	Oui
_url:String [lecture seule]	Oui	Oui	Oui	Oui
_visible:Boolean	Oui	Oui	Oui	Oui
_width:Number	Oui	Oui	Oui	Oui
_x:Number	Oui	Oui	Oui	Oui
_xmouse:Number [lecture seule]	Oui	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_xscale:Number	Oui	Oui	Oui	Oui
_y:Number	Oui	Oui	Oui	Oui
_ymouse:Number [lecture seule]	Oui	Oui	Oui	Oui
_yscale:Number	Oui	Oui	Oui	Oui
blendMode:Object	Oui	Non	Non	Non
cacheAsBitmap:Boolean	Oui	Non	Non	Non
enabled:Boolean	Oui	Oui	Oui	Oui
filters:Array	Oui	Non	Non	Non
getDepth() : Number	Oui	Oui	Oui	Oui
menu:ContextMenu	Oui	Non	Non	Non
onDragOut = function() {}	Oui	Oui	Oui	Oui
onDragOver = function() {}	Oui	Oui	Oui	Oui
onKeyDown = function() {}	Oui	Oui	Oui	Oui
onKeyUp = function() {}	Oui	Oui	Oui	Oui
onKillFocus = function(newFocus:Object) {}	Oui	Oui	Oui	Oui
onPress = function() {}	Oui	Oui	Oui	Oui
onRelease = function() {}	Oui	Oui	Oui	Oui
onReleaseOutside = function() {}	Oui	Oui	Oui	Oui
onRollOut = function() {}	Oui	Oui	Oui	Oui
onRollOver = function() {}	Oui	Oui	Oui	Oui
onSetFocus = function(oldFocus:Object) {}	Oui	Oui	Oui	Oui
scale9Grid:Rectangle	Oui	Non	Non	Non
tabEnabled:Boolean	Oui	Oui	Oui	Oui
tabIndex:Number	Oui	Oui	Oui	Oui
trackAsMenu:Boolean	Oui	Oui	Oui	Oui
useHandCursor:Boolean	Oui	Non	Non	Non

## Camera

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Camera et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
activityLevel:Number [read-only]	Oui	Non	Non	Non
bandwidth:Number [read-only]	Oui	Non	Non	Non
currentFps:Number [read-only]	Oui	Non	Non	Non
fps:Number [read-only]	Oui	Non	Non	Non
get([index:Number]) : Camera	Oui	Non	Non	Non
height:Number [read-only]	Oui	Non	Non	Non
index:Number [read-only]	Oui	Non	Non	Non
motionLevel:Number [read-only]	Oui	Non	Non	Non
motionTimeOut:Number [read-only]	Oui	Non	Non	Non
muted:Boolean [read-only]	Oui	Non	Non	Non
name:String [read-only]	Oui	Non	Non	Non
names:Array [read-only]	Oui	Non	Non	Non
onActivity = function(active:Boolean) {}	Oui	Non	Non	Non
onStatus = function(infoObject:Object) {}	Oui	Non	Non	Non
quality:Number [read-only]	Oui	Non	Non	Non
setMode([width:Number], [height:Number], [fps:Number], [favorArea:Boolean]) : Void	Oui	Non	Non	Non
setMotionLevel([motionLevel:Numbe r], [timeOut:Number]) : Void	Oui	Non	Non	Non
setQuality([bandwidth:Number], [quality:Number]) : Void	Oui	Non	Non	Non
width:Number [read-only]	Oui	Non	Non	Non

## capabilities

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe de capacités et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
audioMIMETypes:Array [lecture seule]	Non	Oui	Oui	Oui
avHardwareDisable:Boolean [lecture seule]	Oui	Oui	Oui	Oui
has4WayKeyAS:Boolean [lecture seule]	Non	Oui	Oui	Oui
hasAccessibility:Boolean [lecture seule]	Oui	Oui	Oui	Oui
hasAudio:Boolean [lecture seule]	Oui	Oui	Oui	Oui



Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
hasAudioEncoder:Boolean [lecture seule]	Oui	Oui	Oui	Oui
hasCMIDI:Boolean [lecture seule]	Non	Oui	Oui	Oui
hasCompoundSound:Boolean [lecture seule]	Non	Oui	Oui	Oui
hasDataLoading:Boolean [lecture seule]	Non	Oui	Oui	Oui
hasEmail:Boolean [lecture seule]	Non	Oui	Oui	Oui
hasEmbeddedVideo:Boolean [lecture seule]	Oui	Oui	Oui	Oui
hasIME:Boolean [read-only]	Oui	Non	Non	Non
hasMappableSoftKeys:Boolean	Non	Oui	Oui	Oui
hasMFI:Boolean [lecture seule]	Non	Oui	Oui	Oui
hasMIDI:Boolean [lecture seule]	Non	Oui	Oui	Oui
hasMMS:Boolean [lecture seule]	Non	Oui	Oui	Oui
hasMouse:Boolean [lecture seule]	Non	Oui	Oui	Oui
hasMP3:Boolean [lecture seule]	Oui	Oui	Oui	Oui
hasPrinting:Boolean [lecture seule]	Oui	Oui	Oui	Oui
hasQWERTYKeyboard:Boolean [lecture seule]	Non	Oui	Oui	Oui
hasScreenBroadcast:Boolean [lecture seule]	Oui	Oui	Oui	Oui
hasScreenPlayback:Boolean [lecture seule]	Oui	Oui	Oui	Oui
hasSharedObjects:Boolean [lecture seule]	Non	Oui	Oui	Oui
hasSMAF:Boolean [lecture seule]	Non	Oui	Oui	Oui
hasSMS:Number [lecture seule]	Non	Oui	Oui	Oui
hasStreamingAudio:Boolean [lecture seule]	Oui	Oui	Oui	Oui
hasStreamingVideo:Boolean [lecture seule]	Oui	Oui	Oui	Oui
hasStylus:Boolean [lecture seule]	Non	Oui	Oui	Oui
hasVideoEncoder:Boolean [lecture seule]	Oui	Oui	Oui	Oui
isDebugger:Boolean [lecture seule]	Oui	Oui	Oui	Oui
language:String [lecture seule]	Oui	Non	Non	Non
localFileReadDisable:Boolean [lecture seule]	Oui	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
manufacturer:String [read-only]	Oui	Non	Non	Non
MIMETypes:Array [lecture seule]	Non	Oui	Oui	Oui
os:String [lecture seule]	Oui	Non	Non	Non
pixelAspectRatio:Number [read-only]	Oui	Non	Non	Non
playerType:String [read-only]	Oui	Non	Non	Non
screenColor:String [read-only]	Oui	Non	Non	Non
screenDPI:Number [read-only]	Oui	Non	Non	Non
screenOrientation:String [lecture seule]	Non	Oui	Oui	Oui
screenResolutionX:Number [lecture seule]	Oui	Oui	Oui	Oui
screenResolutionY:Number [lecture seule]	Oui	Oui	Oui	Oui
serverString:String [read-only]	Oui	Non	Non	Non
softKeyCount:Number [lecture seule]	Non	Oui	Oui	Oui
version:String [lecture seule]	Oui	Oui	Oui	Oui
videoMIMETypes:Array [lecture seule]	Non	Oui	Oui	Oui

## Color

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Color et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
getTransform() : Object	Déconseillé	Oui	Déconseillé	Déconseillé
setRGB(offset:Number) : Void	Déconseillé	Oui	Déconseillé	Déconseillé
setTransform(transformObject:Object) : Void	Déconseillé	Oui	Déconseillé	Déconseillé

## ColorMatrixFilter

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe ColorMatrixFilter et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
clone() : ColorMatrixFilter	Oui	Non	Non	Non
ColorMatrixFilter(matrix:Array)	Oui	Non	Non	Non
matrix:Array	Oui	Non	Non	Non

## ColorTransform

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe ColorTransform et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
alphaMultiplier:Number	Oui	Non	Non	Oui
alphaOffset:Number	Oui	Non	Non	Oui
blueMultiplier:Number	Oui	Non	Non	Oui
blueOffset:Number	Oui	Non	Non	Oui
ColorTransform([redMultiplier:Number], [greenMultiplier:Number], [blueMultiplier:Number], [alphaMultiplier:Number], [redOffset:Number], [greenOffset:Number], [blueOffset:Number], [alphaOffset:Number])	Oui	Non	Non	Oui
concat(second:ColorTransform) : Void	Oui	Non	Non	Oui
greenMultiplier:Number	Oui	Non	Non	Oui
greenOffset:Number	Oui	Non	Non	Oui
redMultiplier:Number	Oui	Non	Non	Oui
redOffset:Number	Oui	Non	Non	Oui
rgb:Number	Oui	Non	Non	Oui
toString() : String	Oui	Non	Non	Oui

## ContextMenu

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe ContextMenu et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
builtInItems:Object	Oui	Non	Non	Non
ContextMenu([callbackFunction:Function])	Oui	Non	Non	Non
customItems:Array	Oui	Non	Non	Non
hideBuiltInItems() : Void	Oui	Non	Non	Non
onSelect = function(item:Object, item_menu:Object) {}	Oui	Non	Non	Non

## ContextMenuItem

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe ContextMenuItem et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
caption:String	Oui	Non	Non	Non
ContextMenu(caption:String, callbackFunction:Function, [separatorBefore:Boolean], [enabled:Boolean], [visible:Boolean])	Oui	Non	Non	Non
copy() : ContextMenuItem	Oui	Non	Non	Non
enabled:Boolean	Oui	Non	Non	Non
onSelect = function(obj:Object, menuItem:Object) {}	Oui	Non	Non	Non
separatorBefore:Boolean	Oui	Non	Non	Non
visible:Boolean	Oui	Non	Non	Non

## ConvolutionFilter

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe ConvolutionFilter et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
alpha:Number	Oui	Non	Non	Non
bias:Number	Oui	Non	Non	Non
clamp:Boolean	Oui	Non	Non	Non
clone() : ConvolutionFilter	Oui	Non	Non	Non
color:Number	Oui	Non	Non	Non
ConvolutionFilter(matrixX:Number, matrixY:Number, matrix:Array, [divisor:Number], [bias:Number], [preserveAlpha:Boolean], [clamp:Boolean], [color:Number], [alpha:Number])	Oui	Non	Non	Non
divisor:Number	Oui	Non	Non	Non
matrix:Array	Oui	Non	Non	Non
matrixX:Number	Oui	Non	Non	Non
matrixY:Number	Oui	Non	Non	Non
preserveAlpha:Boolean	Oui	Non	Non	Non

## CustomActions

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe CustomActions et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
get(name:String) : String	Oui	Non	Non	Non
install(name:String, data:String) : Boolean	Oui	Non	Non	Non
list() : Array	Oui	Non	Non	Non
uninstall(name:String) : Boolean	Oui	Non	Non	Non

## Date

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Date et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
Date([yearOrTimevalue:Number], [month:Number], [date:Number], [hour:Number], [minute:Number], [second:Number], [millisecond:Number])	Oui	Oui	Oui	Oui
getDate() : Number	Oui	Oui	Oui	Oui
getDay() : Number	Oui	Oui	Oui	Oui
getFullYear() : Number	Oui	Oui	Oui	Oui
getHours() : Number	Oui	Oui	Oui	Oui
getLocaleLongDate() : String	Non	Oui	Oui	Oui
getLocaleShortDate() : String	Non	Oui	Oui	Oui
getLocaleTime() : String	Non	Oui	Oui	Oui
getMilliseconds() : Number	Oui	Oui	Oui	Oui
getMinutes() : Number	Oui	Oui	Oui	Oui
getMonth() : Number	Oui	Oui	Oui	Oui
getSeconds() : Number	Oui	Oui	Oui	Oui
getTime() : Number	Oui	Oui	Oui	Oui
getTimezoneOffset() : Number	Oui	Oui	Oui	Oui
getUTCDate() : Number	Oui	Oui	Oui	Oui
getUTCDay() : Number	Oui	Oui	Oui	Oui
getUTCFullYear() : Number	Oui	Oui	Oui	Oui
getUTCHours() : Number	Oui	Oui	Oui	Oui
getUTCMilliseconds() : Number	Oui	Oui	Oui	Oui
getUTCMinutes() : Number	Oui	Oui	Oui	Oui
getUTCMonth() : Number	Oui	Oui	Oui	Oui
getUTCSeconds() : Number	Oui	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
getUTCYear() : Number	Oui	Oui	Oui	Oui
getFullYear() : Number	Oui	Oui	Oui	Oui
setDate(date:Number) : Number	Oui	Oui	Oui	Oui
setFullYear(year:Number, [month:Number], [date:Number]) : Number	Oui	Oui	Oui	Oui
setHours(hour:Number) : Number	Oui	Oui	Oui	Oui
setMilliseconds(millisecond:Number) : Number	Oui	Oui	Oui	Oui
setMinutes(minute:Number) : Number	Oui	Oui	Oui	Oui
setMonth(month:Number, [date:Number]) : Number	Oui	Oui	Oui	Oui
setSeconds(second:Number) : Number	Oui	Oui	Oui	Oui
setTime(millisecond:Number) : Number	Oui	Oui	Oui	Oui
setUTCDate(date:Number) : Number	Oui	Oui	Oui	Oui
setUTCFullYear(year:Number, [month:Number], [date:Number]) : Number	Oui	Oui	Oui	Oui
setUTCHours(hour:Number, [minute:Number], [second:Number], [millisecond:Number]) : Number	Oui	Oui	Oui	Oui
setUTCMilliseconds(millisecond:Num ber) : Number	Oui	Oui	Oui	Oui
setUTCMinutes(minute:Number, [second:Number], [millisecond:Number]) : Number	Oui	Oui	Oui	Oui
setUTCMonth(month:Number, [date:Number]) : Number	Oui	Oui	Oui	Oui
setUTCSeconds(second:Number, [millisecond:Number]) : Number	Oui	Oui	Oui	Oui
setYear(year:Number) : Number	Oui	Oui	Oui	Oui
toString() : String	Oui	Oui	Oui	Oui
UTC(year:Number, month:Number, [date:Number], [hour:Number], [minute:Number], [second:Number], [millisecond:Number]) : Number	Oui	Oui	Oui	Oui
valueOf() : Number	Oui	Oui	Oui	Oui

## DisplacementMapFilter

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe DisplacementMapFilter et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
alpha:Number	Oui	Non	Non	Non
clone() : DisplacementMapFilter	Oui	Non	Non	Non
color:Number	Oui	Non	Non	Non
componentX:Number	Oui	Non	Non	Non
componentY:Number	Oui	Non	Non	Non
DisplacementMapFilter(mapBitmap:BitmapData, mapPoint:Point, componentX:Number, componentY:Number, scaleX:Number, scaleY:Number, [mode:String], [color:Number], [alpha:Number])	Oui	Non	Non	Non
mapBitmap:BitmapData	Oui	Non	Non	Non
mapPoint:Point	Oui	Non	Non	Non
mode:String	Oui	Non	Non	Non
scaleX:Number	Oui	Non	Non	Non
scaleY:Number	Oui	Non	Non	Non

## DropShadowFilter

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe DropShadowFilter et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
alpha:Number	Oui	Non	Non	Non
angle:Number	Oui	Non	Non	Non
blurX:Number	Oui	Non	Non	Non
blurY:Number	Oui	Non	Non	Non
clone() : DropShadowFilter	Oui	Non	Non	Non
color:Number	Oui	Non	Non	Non
distance:Number	Oui	Non	Non	Non
DropShadowFilter([distance:Number], [angle:Number], [color:Number], [alpha:Number], [blurX:Number], [blurY:Number], [strength:Number], [quality:Number], [inner:Boolean], [knockout:Boolean], [hideObject:Boolean])	Oui	Non	Non	Non

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
hideObject:Boolean	Oui	Non	Non	Non
inner:Boolean	Oui	Non	Non	Non
knockout:Boolean	Oui	Non	Non	Non
quality:Number	Oui	Non	Non	Non
strength:Number	Oui	Non	Non	Non

## Error

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Error et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
Error([message:String])	Oui	Oui	Oui	Oui
message:String	Oui	Oui	Oui	Oui
name:String	Oui	Oui	Oui	Oui
toString() : String	Oui	Oui	Oui	Oui

## ExtendedKey

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe ExtendedKey et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
SOFT1:String	Non	Oui	Oui	Oui
SOFT10:String	Non	Oui	Oui	Oui
SOFT11:String	Non	Oui	Oui	Oui
SOFT12:String	Non	Oui	Oui	Oui
SOFT2:String	Non	Oui	Oui	Oui
SOFT3:String	Non	Oui	Oui	Oui
SOFT4:String	Non	Oui	Oui	Oui
SOFT5:String	Non	Oui	Oui	Oui
SOFT6:String	Non	Oui	Oui	Oui
SOFT7:String	Non	Oui	Oui	Oui
SOFT8:String	Non	Oui	Oui	Oui
SOFT9:String	Non	Oui	Oui	Oui

## ExternalInterface

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe ExternalInterface et ses membres.



Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addCallback(methodName:String, instance:Object, method:Function) : Boolean	Oui	Non	Oui	Oui
available:Boolean [read-only]	Oui	Non	Oui	Oui
call(methodName:String, [parameter1:Object]) : Object	Oui	Non	Oui	Oui

## FileReference

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe FileReference et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
creationDate:Date [read-only]	Oui	Non	Non	Non
creator:String [read-only]	Oui	Non	Non	Non
modificationDate:Date [read-only]	Oui	Non	Non	Non
name:String [read-only]	Oui	Non	Non	Non
postData:String	Oui	Non	Non	Non
size:Number [read-only]	Oui	Non	Non	Non
type:String [read-only]	Oui	Non	Non	Non

## FileReferenceList

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe FileReferenceList et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addListener(listener:Object) : Void	Oui	Non	Non	Non
browse([typelist:Array]) : Boolean	Oui	Non	Non	Non
fileList:Array	Oui	Non	Non	Non
FileReferenceList()	Oui	Non	Non	Non
onCancel = function(fileRefList:FileReferenceList) {	Oui	Non	Non	Non
onSelect = function(fileRefList:FileReferenceList) {	Oui	Non	Non	Non
removeListener(listener:Object) : Boolean	Oui	Non	Non	Non

## Function

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Function et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
apply(thisObject:Object, [argArray:Array]) : Void	Oui	Oui	Oui	Oui
call(thisObject:Object, [parameter1:Object]) : Object	Oui	Déconseillé	Déconseillé	Déconseillé

## GlowFilter

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe GlowFilter et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
alpha:Number	Oui	Non	Non	Non
blurX:Number	Oui	Non	Non	Non
blurY:Number	Oui	Non	Non	Non
color:Number	Oui	Non	Non	Non
inner:Boolean	Oui	Non	Non	Non
knockout:Boolean	Oui	Non	Non	Non
quality:Number	Oui	Non	Non	Non
strength:Number	Oui	Non	Non	Non

## GradientBevelFilter

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe GradientBevelFilter et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
alphas:Array	Oui	Non	Non	Non
angle:Number	Oui	Non	Non	Non
blurX:Number	Oui	Non	Non	Non
blurY:Number	Oui	Non	Non	Non
clone() : GradientBevelFilter	Oui	Non	Non	Non
colors:Array	Oui	Non	Non	Non
distance:Number	Oui	Non	Non	Non
GradientBevelFilter([distance:Number], [angle:Number], [colors:Array], [alphas:Array], [ratios:Array], [blurX:Number], [blurY:Number], [strength:Number], [quality:Number], [type:String], [knockout:Boolean])	Oui	Non	Non	Non
knockout:Boolean	Oui	Non	Non	Non

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
quality:Number	Oui	Non	Non	Non
ratios:Array	Oui	Non	Non	Non
strength:Number	Oui	Non	Non	Non
type:String	Oui	Non	Non	Non

## GradientGlowFilter

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe GradientGlowFilter et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
alphas:Array	Oui	Non	Non	Non
angle:Number	Oui	Non	Non	Non
blurX:Number	Oui	Non	Non	Non
blurY:Number	Oui	Non	Non	Non
colors:Array	Oui	Non	Non	Non
distance:Number	Oui	Non	Non	Non
GradientGlowFilter([distance:Number], [angle:Number], [colors:Array], [alphas:Array], [ratios:Array], [blurX:Number], [blurY:Number], [strength:Number], [quality:Number], [type:String], [knockout:Boolean])	Oui	Non	Non	Non
knockout:Boolean	Oui	Non	Non	Non
quality:Number	Oui	Non	Non	Non
ratios:Array	Oui	Non	Non	Non
strength:Number	Oui	Non	Non	Non
type:String	Oui	Non	Non	Non

## IME

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe IME et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addListener(listener:Object) : Void	Oui	Non	Non	Non
ALPHANUMERIC_FULL:String	Oui	Non	Non	Non
ALPHANUMERIC_HALF:String	Oui	Non	Non	Non
CHINESE:String	Oui	Non	Non	Non
doConversion() : Boolean	Oui	Non	Non	Non

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
getConversionMode() : String	Oui	Non	Non	Non
getEnabled() : Boolean	Oui	Non	Non	Non
JAPANESE_HIRAGANA:String	Oui	Non	Non	Non
JAPANESE_KATAKANA_FULL:String	Oui	Non	Non	Non
JAPANESE_KATAKANA_HALF:String	Oui	Non	Non	Non
KOREAN:String	Oui	Non	Non	Non
onIMEComposition = fonction([readingString:String]) {}	Oui	Non	Non	Non
removeListener(listener:Object) : Boolean	Oui	Non	Non	Non
setCompositionString(composition:String) : Boolean	Oui	Non	Non	Non
setConversionMode(mode:String) : Boolean	Oui	Non	Non	Non
setEnabled(enabled:Boolean) : Boolean	Oui	Non	Non	Non
UNKNOWN:String	Oui	Non	Non	Non

## Key

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Key et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_listeners:Array [lecture seule]	Oui	Oui	Oui	Oui
addListener(listener:Object) : Void	Oui	Oui	Oui	Oui
BACKSPACE:Number	Oui	Oui	Oui	Oui
CAPSLock:Number	Oui	Oui	Oui	Oui
CONTROL:Number	Oui	Oui	Oui	Oui
DELETEKEY:Number	Oui	Oui	Oui	Oui
DOWN:Number	Oui	Oui	Oui	Oui
END:Number	Oui	Oui	Oui	Oui
ENTER:Number	Oui	Oui	Oui	Oui
ESCAPE:Number	Oui	Oui	Oui	Oui
getAscii() : Number	Oui	Oui	Oui	Oui
getCode() : Number	Oui	Oui	Oui	Oui
HOME:Number	Oui	Oui	Oui	Oui
INSERT:Number	Oui	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
isAccessible() : Boolean	Oui	Non	Non	Non
isDown(code:Number) : Boolean	Oui	Oui	Oui	Oui
isToggled(code:Number) : Boolean	Oui	Non	Non	Non
LEFT:Number	Oui	Oui	Oui	Oui
onKeyDown = function() {}	Oui	Oui	Oui	Oui
onKeyUp = function() {}	Oui	Oui	Oui	Oui
PGDN:Number	Oui	Oui	Oui	Oui
PGUP:Number	Oui	Oui	Oui	Oui
removeListener(listener:Object) : Boolean	Oui	Non	Oui	Oui
RIGHT:Number	Oui	Oui	Oui	Oui
SHIFT:Number	Oui	Oui	Oui	Oui
SPACE:Number	Oui	Oui	Oui	Oui
TAB:Number	Oui	Oui	Oui	Oui
UP:Number	Oui	Oui	Oui	Oui

## LoadVars

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe LoadVars et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addRequestHeader(header:Object, headerValue:String) : Void	Oui	Oui	Oui	Oui
contentType:String	Oui	Oui	Oui	Oui
decode(queryString:String) : Void	Oui	Oui	Oui	Oui
getBytesLoaded() : Number	Oui	Oui	Oui	Oui
getBytesTotal() : Number	Oui	Oui	Oui	Oui
load(url:String) : Boolean	Oui	Oui	Oui	Oui
loaded:Boolean	Oui	Oui	Oui	Oui
LoadVars()	Oui	Oui	Oui	Oui
onData = function(src:String) {}	Oui	Oui	Oui	Oui
onHTTPStatus = function(httpStatus:Number) {}	Oui	Non	Non	Oui
onLoad = function(success:Boolean) {}	Oui	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
send(url:String, target:String, [method:String]) : Boolean	Oui	Oui	Oui	Oui
sendAndLoad(url:String, target:Object, [method:String]) : Boolean	Oui	Oui	Oui	Oui
toString() : String	Oui	Oui	Oui	Oui

## LocalConnection

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe LocalConnection et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
allowDomain = fonction([sendingDomain:String]) {}	Oui	Non	Non	Oui
allowInsecureDomain = fonction([sendingDomain:String]) {}	Oui	Non	Non	Oui
close() : Void	Oui	Non	Non	Oui
connect(connectionName:String) : Boolean	Oui	Non	Non	Oui
domain() : String	Oui	Non	Non	Oui
LocalConnection()	Oui	Non	Non	Oui
onStatus = fonction(infoObject:Object) {}	Oui	Non	Non	Oui
send(connectionName:String, methodName:String, [args:Object]) : Boolean	Oui	Non	Non	Oui

## Locale

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Locale et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addDelayedInstance(instance:Object, stringID:String) : Void	Oui	Non	Non	Non
addXMLPath(langCode:String, path:String) : Void	Oui	Non	Non	Non
autoReplace:Boolean	Oui	Non	Non	Non
checkXMLStatus() : Boolean	Oui	Non	Non	Non
getDefaultLang() : String	Oui	Non	Non	Non
initialize() : Void	Oui	Non	Non	Non
languageCodeArray:Array [read-only]	Oui	Non	Non	Non

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
loadLanguageXML(xmlLanguageCode:String, customXmlCompleteCallback:Function) : Void	Oui	Non	Non	Non
loadString(id:String) : String	Oui	Non	Non	Non
loadStringEx(stringID:String, languageCode:String) : String	Oui	Non	Non	Non
setDefaultLang(langCode:String) : Void	Oui	Non	Non	Non
setLoadCallback(loadCallback:Function) : Void	Oui	Non	Non	Non
setString(stringID:String, languageCode:String, stringValue:String) : Void	Oui	Non	Non	Non
stringIDArray:Array [read-only]	Oui	Non	Non	Non

## Math

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Math et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
abs(x:Number) : Number	Oui	Oui	Oui	Oui
acos(x:Number) : Number	Oui	Oui	Oui	Oui
asin(x:Number) : Number	Oui	Oui	Oui	Oui
atan(tangent:Number) : Number	Oui	Oui	Oui	Oui
atan2(y:Number, x:Number) : Number	Oui	Oui	Oui	Oui
ceil(x:Number) : Number	Oui	Oui	Oui	Oui
cos(x:Number) : Number	Oui	Oui	Oui	Oui
E:Number	Oui	Oui	Oui	Oui
exp(x:Number) : Number	Oui	Oui	Oui	Oui
floor(x:Number) : Number	Oui	Oui	Oui	Oui
LN10:Number	Oui	Oui	Oui	Oui
LN2:Number	Oui	Oui	Oui	Oui
log(x:Number) : Number	Oui	Oui	Oui	Oui
LOG10E:Number	Oui	Oui	Oui	Oui
LOG2E:Number	Oui	Oui	Oui	Oui
max(x:Number, y:Number) : Number	Oui	Oui	Oui	Oui
min(x:Number, y:Number) : Number	Oui	Oui	Oui	Oui
PI:Number	Oui	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
pow(x:Number, y:Number) : Number	Oui	Oui	Oui	Oui
random() : Number	Oui	Oui	Oui	Oui
round(x:Number) : Number	Oui	Oui	Oui	Oui
sin(x:Number) : Number	Oui	Oui	Oui	Oui
sqrt(x:Number) : Number	Oui	Oui	Oui	Oui
SQRT1_2:Number	Oui	Oui	Oui	Oui
SQRT2:Number	Oui	Oui	Oui	Oui
tan(x:Number) : Number	Oui	Oui	Oui	Oui

## Matrix

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Matrix et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
a:Number	Oui	Non	Non	Oui
b:Number	Oui	Non	Non	Oui
c:Number	Oui	Non	Non	Oui
clone() : Matrix	Oui	Non	Non	Oui
concat(m:Matrix) : Void	Oui	Non	Non	Oui
createBox(scaleX:Number, scaleY:Number, [rotation:Number], [tx:Number], [ty:Number]) : Void	Oui	Non	Non	Oui
createGradientBox(width:Number, height:Number, [rotation:Number], [tx:Number], [ty:Number]) : Void	Oui	Non	Non	Non
d:Number	Oui	Non	Non	Oui
deltaTransformPoint(pt:Point) : Point	Oui	Non	Non	Oui
identity() : Void	Oui	Non	Non	Oui
invert() : Void	Oui	Non	Non	Oui
Matrix([a:Number], [b:Number], [c:Number], [d:Number], [tx:Number], [ty:Number])	Oui	Non	Non	Oui
rotate(angle:Number) : Void	Oui	Non	Non	Oui
scale(sx:Number, sy:Number) : Void	Oui	Non	Non	Oui
toString() : String	Oui	Non	Non	Oui
transformPoint(pt:Point) : Point	Oui	Non	Non	Oui



Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
translate(tx:Number, ty:Number) : Void	Oui	Non	Non	Oui
tx:Number	Oui	Non	Non	Oui
ty:Number	Oui	Non	Non	Oui

## Microphone

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Microphone et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
activityLevel:Number [read-only]	Oui	Non	Non	Non
gain:Number [read-only]	Oui	Non	Non	Non
get([index:Number]) : Microphone	Oui	Non	Non	Non
index:Number [read-only]	Oui	Non	Non	Non
muted:Boolean [read-only]	Oui	Non	Non	Non
name:String [read-only]	Oui	Non	Non	Non
names:Array [read-only]	Oui	Non	Non	Non
onActivity = function(active:Boolean) {}	Oui	Non	Non	Non
onStatus = function(infoObject:Object) {}	Oui	Non	Non	Non
rate:Number [read-only]	Oui	Non	Non	Non
setGain(gain:Number) : Void	Oui	Non	Non	Non
setRate(rate:Number) : Void	Oui	Non	Non	Non
setSilenceLevel(silenceLevel:Number, [timeOut:Number]) : Void	Oui	Non	Non	Non
setUseEchoSuppression(useEchoSuppression:Boolean) : Void	Oui	Non	Non	Non
silenceLevel:Number [read-only]	Oui	Non	Non	Non
silenceTimeOut:Number [read-only]	Oui	Non	Non	Non
useEchoSuppression:Boolean [read-only]	Oui	Non	Non	Non

## Mouse

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Mouse et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addListener(listener:Object) : Void	Oui	Oui	Oui	Oui
hide() : Number	Oui	Non	Non	Non
onMouseDown = function() {}	Oui	Oui	Oui	Oui
onMouseMove = function() {}	Oui	Oui	Oui	Oui
onMouseUp = function() {}	Oui	Oui	Oui	Oui
onMouseWheel = function([delta:Number], [scrollTarget:Object]) {}	Oui	Non	Non	Non
removeListener(listener:Object) : Boolean	Oui	Oui	Oui	Oui
show() : Number	Oui	Non	Non	Non

## MovieClip

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe MovieClip et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_alpha:Number	Oui	Oui	Oui	Oui
_currentframe:Number [lecture seule]	Oui	Oui	Oui	Oui
_droptarget:String [lecture seule]	Oui	Oui	Oui	Oui
_focusrect:Boolean	Oui	Oui	Oui	Oui
_framesloaded:Number [lecture seule]	Oui	Oui	Oui	Oui
_height:Number	Oui	Oui	Oui	Oui
_highquality:Number	Déconseillé	Déconseillé	Déconseillé	Déconseillé
_lockroot:Boolean	Oui	Oui	Oui	Oui
_name:String	Oui	Oui	Oui	Oui
_parent:MovieClip	Oui	Oui	Oui	Oui
_quality:String	Oui	Oui	Oui	Oui
_rotation:Number	Oui	Oui	Oui	Oui
_soundbuftime:Number	Oui	Oui	Oui	Oui
_target:String [lecture seule]	Oui	Oui	Oui	Oui
_totalframes:Number [lecture seule]	Oui	Oui	Oui	Oui
_url:String [lecture seule]	Oui	Oui	Oui	Oui
_visible:Boolean	Oui	Oui	Oui	Oui
_width:Number	Oui	Oui	Oui	Oui
_x:Number	Oui	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_xmouse:Number [lecture seule]	Oui	Oui	Oui	Oui
_xscale:Number	Oui	Oui	Oui	Oui
_y:Number	Oui	Oui	Oui	Oui
_ymouse:Number [lecture seule]	Oui	Oui	Oui	Oui
_yscale:Number	Oui	Oui	Oui	Oui
attachAudio(id:Object) : Void	Oui	Non	Non	Non
attachBitmap(bmp:BitmapData, depth:Number, [pixelSnapping:String], [smoothing:Boolean]) : Void	Oui	Non	Non	Oui
attachMovie(id:String, name:String, depth:Number, [initObject:Object]) : MovieClip	Oui	Oui	Oui	Oui
beginBitmapFill(bmp:BitmapData, [matrix:Matrix], [repeat:Boolean], [smoothing:Boolean]) : Void	Oui	Non	Non	Oui
beginFill(rgb:Number, [alpha:Number]) : Void	Oui	Oui	Oui	Oui
beginGradientFill(fillType:String, colors:Array, alphas:Array, ratios:Array, matrix:Object, [spreadMethod:String], [interpolationMethod:String], [focalPointRatio:Number]) : Void	Oui	Non	Non	Non
blendMode:Object	Oui	Non	Non	Non
cacheAsBitmap:Boolean	Oui	Non	Non	Non
clear() : Void	Oui	Oui	Oui	Oui
createEmptyMovieClip(name:String, depth:Number) : MovieClip	Oui	Oui	Oui	Oui
createTextField(instanceName:String, depth:Number, x:Number, y:Number, width:Number, height:Number) : TextField	Oui	Oui	Oui	Oui
curveTo(controlX:Number, controlY:Number, anchorX:Number, anchorY:Number) : Void	Oui	Oui	Oui	Oui
duplicateMovieClip(name:String, depth:Number, [initObject:Object]) : MovieClip	Oui	Oui	Oui	Oui
enabled:Boolean	Oui	Oui	Oui	Oui
endFill() : Void	Oui	Oui	Oui	Oui
filters:Array	Oui	Non	Non	Non
focusEnabled:Boolean	Oui	Oui	Oui	Oui
forceSmoothing:Boolean	Oui	Non	Non	Non

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
getBounds(bounds:Object) : Object	Oui	Oui	Oui	Oui
getBytesLoaded() : Number	Oui	Oui	Oui	Oui
getBytesTotal() : Number	Oui	Oui	Oui	Oui
getDepth() : Number	Oui	Oui	Oui	Oui
getInstanceAtDepth(depth:Number) : MovieClip	Oui	Oui	Oui	Oui
getNextHighestDepth() : Number	Oui	Oui	Oui	Oui
getRect(bounds:Object) : Object	Oui	Non	Non	Non
getSWFVersion() : Number	Oui	Oui	Oui	Oui
getTextSnapshot() : TextSnapshot	Oui	Non	Non	Non
getURL(url:String, [window:String], [method:String]) : Void	Oui	Oui	Oui	Oui
globalToLocal(pt:Object) : Void	Oui	Oui	Oui	Oui
gotoAndPlay(frame:Object) : Void	Oui	Oui	Oui	Oui
gotoAndStop(frame:Object) : Void	Oui	Oui	Oui	Oui
hitArea:Object	Oui	Oui	Oui	Oui
hitTest() : Boolean	Oui	Oui	Oui	Oui
lineGradientStyle(fillType:String, colors:Array, alphas:Array, ratios:Array, matrix:Object, [spreadMethod:String], [interpolationMethod:String], [focalPointRatio:Number]) : Void	Oui	Non	Non	Non
lineStyle(thickness:Number, rgb:Number, alpha:Number, pixelHinting:Boolean, noScale:String, capsStyle:String, jointStyle:String, miterLimit:Number) : Void	Oui	Oui	Oui	Oui
lineTo(x:Number, y:Number) : Void	Oui	Oui	Oui	Oui
loadMovie(url:String, [method:String]) : Void	Oui	Oui	Oui	Oui
loadVariables(url:String, [method:String]) : Void	Oui	Oui	Oui	Oui
localToGlobal(pt:Object) : Void	Oui	Oui	Oui	Oui
menu:ContextMenu	Oui	Non	Non	Non
moveTo(x:Number, y:Number) : Void	Oui	Oui	Oui	Oui
nextFrame() : Void	Oui	Oui	Oui	Oui
onData = function() {}	Oui	Oui	Oui	Oui
onDragOut = function() {}	Oui	Oui	Oui	Oui
onDragOver = function() {}	Oui	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
onEnterFrame = function() {}	Oui	Oui	Oui	Oui
onKeyDown = function() {}	Oui	Oui	Oui	Oui
onKeyUp = function() {}	Oui	Oui	Oui	Oui
onKillFocus = function(newFocus:Object) {}	Oui	Oui	Oui	Oui
onLoad = function() {}	Oui	Oui	Oui	Oui
onMouseDown = function() {}	Oui	Oui	Oui	Oui
onMouseMove = function() {}	Oui	Oui	Oui	Oui
onMouseUp = function() {}	Oui	Oui	Oui	Oui
onPress = function() {}	Oui	Oui	Oui	Oui
onRelease = function() {}	Oui	Oui	Oui	Oui
onReleaseOutside = function() {}	Oui	Oui	Oui	Oui
onRollOut = function() {}	Oui	Oui	Oui	Oui
onRollOver = function() {}	Oui	Oui	Oui	Oui
onSetFocus = function(oldFocus:Object) {}	Oui	Oui	Oui	Oui
onUnload = function() {}	Oui	Oui	Oui	Oui
opaqueBackground:Number	Oui	Non	Non	Non
play() : Void	Oui	Oui	Oui	Oui
prevFrame() : Void	Oui	Oui	Oui	Oui
removeMovieClip() : Void	Oui	Oui	Oui	Oui
scale9Grid:Rectangle	Oui	Non	Non	Non
scrollRect:Object	Oui	Non	Non	Non
setMask(mc:Object) : Void	Oui	Oui	Oui	Oui
startDrag([lockCenter:Boolean], [left:Number], [top:Number], [right:Number], [bottom:Number]) : Void	Oui	Oui	Oui	Oui
stop() : Void	Oui	Oui	Oui	Oui
stopDrag() : Void	Oui	Oui	Oui	Oui
swapDepths(target:Object) : Void	Oui	Oui	Oui	Oui
tabChildren:Boolean	Oui	Oui	Oui	Oui
tabEnabled:Boolean	Oui	Oui	Oui	Oui
tabIndex:Number	Oui	Oui	Oui	Oui
trackAsMenu:Boolean	Oui	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
transform:Transform	Oui	Non	Non	Non
unloadMovie() : Void	Oui	Oui	Oui	Oui
useHandCursor:Boolean	Oui	Non	Non	Non

## MovieClipLoader

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe MovieClipLoader et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addListener(listener:Object) : Boolean	Oui	Oui	Oui	Oui
checkPolicyFile:Boolean	Oui	Non	Non	Non
getProgress(target:Object) : Object	Oui	Oui	Oui	Oui
loadClip(url:String, target:Object) : Boolean	Oui	Oui	Oui	Oui
MovieClipLoader()	Oui	Oui	Oui	Oui
onLoadComplete = function([target_mc:MovieClip], [httpStatus:Number]) {}	Oui	Oui	Oui	Oui
onLoadError = function(target_mc:MovieClip, errorCode:String, [httpStatus:Number]) {}	Oui	Oui	Oui	Oui
onLoadInit = function([target_mc:MovieClip]) {}	Oui	Oui	Oui	Oui
onLoadProgress = function([target_mc:MovieClip], loadedBytes:Number, totalBytes:Number) {}	Oui	Oui	Oui	Oui
onLoadStart = function([target_mc:MovieClip]) {}	Oui	Oui	Oui	Oui
removeListener(listener:Object) : Boolean	Oui	Oui	Oui	Oui
unloadClip(target:Object) : Boolean	Oui	Oui	Oui	Oui

## NetConnection

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe NetConnection et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
connect(targetURI:String) : Boolean	Oui	Non	Oui	Oui
NetConnection()	Oui	Non	Oui	Oui

## NetStream

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe NetStream et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
bufferLength:Number [read-only]	Oui	Non	Oui	Oui
bufferTime:Number [read-only]	Oui	Non	Oui	Oui
bytesLoaded:Number [read-only]	Oui	Non	Oui	Oui
bytesTotal:Number [read-only]	Oui	Non	Oui	Oui
checkPolicyFile:Boolean	Oui	Non	Non	Non
close() : Void	Oui	Non	Oui	Oui
currentFps:Number [read-only]	Oui	Non	Oui	Oui
NetStream(connection:NetConnectio n)	Oui	Non	Oui	Oui
onCuePoint = function(infoObject:Object) {}	Oui	Non	Oui	Oui
onMetaData = function(infoObject:Object) {}	Oui	Non	Oui	Oui
onStatus = function(infoObject:Object) {}	Oui	Non	Oui	Oui
pause([flag:Boolean]) : Void	Oui	Non	Oui	Oui
play(name:Object, start:Number, len:Number, reset:Object) : Void	Oui	Non	Oui	Oui
seek(offset:Number) : Void	Oui	Non	Oui	Oui
setBufferTime(bufferTime:Number) : Void	Oui	Non	Oui	Oui
time:Number [read-only]	Oui	Non	Oui	Oui

## Number

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Number et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
MAX_VALUE:Number	Oui	Oui	Oui	Oui
MIN_VALUE:Number	Oui	Oui	Oui	Oui
NaN:Number	Oui	Oui	Oui	Oui
NEGATIVE_INFINITY:Number	Oui	Oui	Oui	Oui
Number(num:Object)	Oui	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
POSITIVE_INFINITY:Number	Oui	Oui	Oui	Oui
toString(radix:Number) : String	Oui	Oui	Oui	Oui
valueOf() : Number	Oui	Oui	Oui	Oui

## Object

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Object et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
__proto__:Object	Oui	Oui	Oui	Oui
__resolve:Object	Oui	Oui	Oui	Oui
addProperty(name:String, getter:Function, setter:Function) : Boolean	Oui	Oui	Oui	Oui
constructor:Object	Oui	Oui	Oui	Oui
hasOwnProperty(name:String) : Boolean	Oui	Oui	Oui	Oui
isPropertyEnumerable(name:String) : Boolean	Oui	Oui	Oui	Oui
isPrototypeOf(theClass:Object) : Boolean	Oui	Oui	Oui	Oui
Object()	Oui	Oui	Oui	Oui
prototype:Object	Oui	Oui	Oui	Oui
registerClass(name:String, theClass:Function) : Boolean	Oui	Oui	Oui	Oui
toString() : String	Oui	Oui	Oui	Oui
unwatch(name:String) : Boolean	Oui	Oui	Oui	Oui
valueOf() : Object	Oui	Oui	Oui	Oui
watch(name:String, callback:Function, [userData:Object]) : Boolean	Oui	Oui	Oui	Oui

## Point

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Point et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
add(v:Point) : Point	Oui	Non	Non	Oui
clone() : Point	Oui	Non	Non	Oui
distance(pt1:Point, pt2:Point) : Number	Oui	Non	Non	Oui



Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
equals(toCompare:Object) : Boolean	Oui	Non	Non	Oui
interpolate(pt1:Point, pt2:Point, f:Number) : Point	Oui	Non	Non	Oui
length:Number	Oui	Non	Non	Oui
normalize(length:Number) : Void	Oui	Non	Non	Oui
offset(dx:Number, dy:Number) : Void	Oui	Non	Non	Oui
Point(x:Number, y:Number)	Oui	Non	Non	Oui
polar(len:Number, angle:Number) : Point	Oui	Non	Non	Oui
subtract(v:Point) : Point	Oui	Non	Non	Oui
toString() : String	Oui	Non	Non	Oui
x:Number	Oui	Non	Non	Oui
y:Number	Oui	Non	Non	Oui

## PrintJob

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe PrintJob et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addPage(target:Object, [printArea:Object], [options:Object], [frameNum:Number]) : Boolean	Oui	Non	Non	Non
orientation:String [read-only]	Oui	Non	Non	Non
pageHeight:Number [read-only]	Oui	Non	Non	Non
pageWidth:Number [read-only]	Oui	Non	Non	Non
paperHeight:Number [read-only]	Oui	Non	Non	Non
paperWidth:Number [read-only]	Oui	Non	Non	Non
PrintJob()	Oui	Non	Non	Non
send() : Void	Oui	Non	Non	Non
start() : Boolean	Oui	Non	Non	Non

## Rectangle

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Rectangle et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
bottom:Number	Oui	Non	Non	Oui
bottomRight:Point	Oui	Non	Non	Oui
clone() : Rectangle	Oui	Non	Non	Oui
contains(x:Number, y:Number) : Boolean	Oui	Non	Non	Oui
containsPoint(pt:Point) : Boolean	Oui	Non	Non	Oui
containsRectangle(rect:Rectangle) : Boolean	Oui	Non	Non	Oui
equals(toCompare:Object) : Boolean	Oui	Non	Non	Oui
height:Number	Oui	Non	Non	Oui
inflate(dx:Number, dy:Number) : Void	Oui	Non	Non	Oui
inflatePoint(pt:Point) : Void	Oui	Non	Non	Oui
intersection(toIntersect:Rectangle) : Rectangle	Oui	Non	Non	Oui
intersects(toIntersect:Rectangle) : Boolean	Oui	Non	Non	Oui
isEmpty() : Boolean	Oui	Non	Non	Oui
left:Number	Oui	Non	Non	Oui
offset(dx:Number, dy:Number) : Void	Oui	Non	Non	Oui
offsetPoint(pt:Point) : Void	Oui	Non	Non	Oui
Rectangle(x:Number, y:Number, width:Number, height:Number)	Oui	Non	Non	Oui
right:Number	Oui	Non	Non	Oui
setEmpty() : Void	Oui	Non	Non	Oui
size:Point	Oui	Non	Non	Oui
top:Number	Oui	Non	Non	Oui
topLeft:Point	Oui	Non	Non	Oui
toString() : String	Oui	Non	Non	Oui
union(toUnion:Rectangle) : Rectangle	Oui	Non	Non	Oui
width:Number	Oui	Non	Non	Oui
x:Number	Oui	Non	Non	Oui
y:Number	Oui	Non	Non	Oui

## sécurité

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe de sécurité et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
allowDomain(domain1:String) : Void	Oui	Oui	Oui	Oui
allowInsecureDomain(domain:String) : Void	Oui	Oui	Oui	Oui
loadPolicyFile(url:String) : Void	Oui	Oui	Oui	Oui
sandboxType:String [read-only]	Oui	Non	Non	Oui

## Selection

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Selection et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addListener(listener:Object) : Void	Oui	Oui	Oui	Oui
getBeginIndex() : Number	Oui	Non	Non	Non
getCaretIndex() : Number	Oui	Non	Non	Non
getEndIndex() : Number	Oui	Non	Non	Non
getFocus() : String	Oui	Oui	Oui	Oui
onSetFocus = function([oldfocus:Object], [newfocus:Object]) {}	Oui	Oui	Oui	Oui
removeListener(listener:Object) : Boolean	Oui	Oui	Oui	Oui
setFocus(newFocus:Object) : Boolean	Oui	Oui	Oui	Oui
setSelection(beginIndex:Number, endIndex:Number) : Void	Oui	Non	Non	Non

## SharedObject

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe SharedObject et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addListener(objectName:String, notifyFunction:Function) : Void	Non	Oui	Oui	Oui
clear() : Void	Oui	Oui	Oui	Oui
data:Object	Oui	Oui	Oui	Oui
flush([minDiskSpace:Number]) : Object	Oui	Partielle	Partielle	Partielle
getLocal(name:String, [localPath:String], [secure:Boolean]) : SharedObject	Oui	Partielle	Partielle	Partielle
getMaxSize() : Number	Non	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
getSize() : Number	Oui	Oui	Oui	Oui
onStatus = function(info:Object) {}	Oui	Oui	Oui	Oui
removeListener(objectName:String)	Non	Oui	Oui	Oui

## Sound

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Sound et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
attachSound(id:String) : Void	Oui	Oui	Oui	Oui
checkPolicyFile:Boolean	Oui	Non	Non	Non
duration:Number [lecture seule]	Oui	Partielle	Partielle	Partielle
getBytesLoaded() : Number	Oui	Oui	Oui	Oui
getBytesTotal() : Number	Oui	Oui	Oui	Oui
getPan() : Number	Oui	Partielle	Partielle	Partielle
getTransform() : Object	Oui	Partielle	Partielle	Partielle
getVolume() : Number	Oui	Oui	Oui	Oui
id3:Object [lecture seule]	Oui	Oui	Oui	Oui
loadSound(url:String, isStreaming:Boolean) : Void	Oui	Partielle	Partielle	Partielle
onID3 = function() {}	Oui	Oui	Oui	Oui
onLoad = function(success:Boolean) {}	Oui	Oui	Oui	Oui
onSoundComplete = function() {}	Oui	Oui	Oui	Oui
position:Number [lecture seule]	Oui	Partielle	Partielle	Partielle
setPan(value:Number) : Void	Oui	Partielle	Partielle	Partielle
setTransform(transformObject:Object) : Void	Oui	Partielle	Partielle	Partielle
setVolume(value:Number) : Void	Oui	Partielle	Partielle	Partielle
Sound([target:Object])	Oui	Oui	Oui	Oui
start([secondOffset:Number], [loops:Number]) : Void	Oui	Oui	Oui	Oui
stop([linkageID:String]) : Void	Oui	Oui	Oui	Oui

## Stage

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Stage et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addListener(listener:Object) : Void	Oui	Oui	Oui	Oui
align:String	Oui	Oui	Oui	Oui
displayState:String	Oui	Non	Non	Non
height:Number	Oui	Oui	Oui	Oui
onFullScreen = function(bFull:Boolean) {}	Oui	Non	Non	Non
onResize = function() {}	Oui	Oui	Oui	Oui
removeListener(listener:Object) : Boolean	Oui	Oui	Oui	Oui
scaleMode:String	Oui	Oui	Oui	Oui
showMenu:Boolean	Oui	Non	Non	Non
width:Number	Oui	Oui	Oui	Oui

## String

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe String et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
charAt(index:Number) : String	Oui	Oui	Oui	Oui
charCodeAt(index:Number) : Number	Oui	Oui	Oui	Oui
concat(value:Object) : String	Oui	Oui	Oui	Oui
fromCharCode() : String	Oui	Oui	Oui	Oui
indexOf(value:String, [startIndex:Number]) : Number	Oui	Oui	Oui	Oui
lastIndexOf(value:String, [startIndex:Number]) : Number	Oui	Oui	Oui	Oui
length:Number	Oui	Oui	Oui	Oui
slice(start:Number, end:Number) : String	Oui	Oui	Oui	Oui
split(delimiter:String, [limit:Number]) : Array	Oui	Oui	Oui	Oui
String(value:String)	Oui	Oui	Oui	Oui
substr(start:Number, length:Number) : String	Oui	Oui	Oui	Oui
substring(start:Number, end:Number) : String	Oui	Oui	Oui	Oui
toLowerCase() : String	Oui	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
toString() : String	Oui	Oui	Oui	Oui
toUpperCase() : String	Oui	Oui	Oui	Oui
valueOf() : String	Oui	Oui	Oui	Oui

## StyleSheet

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe StyleSheet et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
clear() : Void	Oui	Non	Non	Oui
getStyle(name:String) : Object	Oui	Non	Non	Oui
getStyleNames() : Array	Oui	Non	Non	Oui
load(url:String) : Boolean	Oui	Non	Non	Oui
onLoad = function(success:Boolean) {}	Oui	Non	Non	Oui
parseCSS(cssText:String) : Boolean	Oui	Non	Non	Oui
setStyle(name:String, style:Object) : Void	Oui	Non	Non	Oui
StyleSheet()	Oui	Non	Non	Oui
transform(style:Object) : TextFormat	Oui	Non	Non	Oui

## System

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe System et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
exactSettings:Boolean	Oui	Non	Non	Non
onStatus = function(infoObject:Object) {}	Oui	Oui	Oui	Oui
setClipboard(text:String) : Void	Oui	Non	Non	Non
showSettings([tabID:Number]) : Void	Oui	Non	Non	Non
useCodepage:Boolean	Oui	Oui	Oui	Oui

## TextField

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe TextField et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_alpha:Number	Oui	Oui	Oui	Oui
_height:Number	Oui	Oui	Oui	Oui
_highquality:Number	Déconseillé	Déconseillé	Déconseillé	Déconseillé
_name:String	Oui	Oui	Oui	Oui
_parent:MovieClip	Oui	Oui	Oui	Oui
_quality:String	Oui	Oui	Oui	Oui
_rotation:Number	Oui	Oui	Oui	Oui
_soundbuftime:Number	Oui	Oui	Oui	Oui
_target:String [lecture seule]	Oui	Oui	Oui	Oui
_url:String [lecture seule]	Oui	Oui	Oui	Oui
_visible:Boolean	Oui	Oui	Oui	Oui
_width:Number	Oui	Oui	Oui	Oui
_x:Number	Oui	Oui	Oui	Oui
_xmouse:Number [lecture seule]	Oui	Oui	Oui	Oui
_xscale:Number	Oui	Oui	Oui	Oui
_y:Number	Oui	Oui	Oui	Oui
_ymouse:Number [lecture seule]	Oui	Oui	Oui	Oui
_yscale:Number	Oui	Oui	Oui	Oui
addListener(listener:Object) : Boolean	Oui	Oui	Oui	Oui
antiAliasType:String	Oui	Non	Non	Non
autoSize:Object	Oui	Oui	Oui	Oui
background:Boolean	Oui	Oui	Oui	Oui
backgroundColor:Number	Oui	Oui	Oui	Oui
border:Boolean	Oui	Oui	Oui	Oui
borderColor:Number	Oui	Oui	Oui	Oui
bottomScroll:Number [lecture seule]	Oui	Oui	Oui	Oui
condenseWhite:Boolean	Oui	Oui	Oui	Oui
embedFonts:Boolean	Oui	Oui	Oui	Oui
filters:Array	Oui	Non	Non	Non
getDepth() : Number	Oui	Oui	Oui	Oui
getFontList() : Array	Oui	Non	Non	Non
getNewTextFormat() : TextFormat	Oui	Oui	Oui	Oui
getTextFormat([beginIndex:Number], [endIndex:Number]) : TextFormat	Oui	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
gridFitType:String	Oui	Non	Non	Non
hscroll:Number	Oui	Oui	Oui	Oui
html:Boolean	Oui	Oui	Oui	Oui
htmlText:String	Oui	Oui	Oui	Oui
length:Number [lecture seule]	Oui	Oui	Oui	Oui
maxChars:Number	Oui	Oui	Oui	Oui
maxhscroll:Number [lecture seule]	Oui	Oui	Oui	Oui
maxscroll:Number [lecture seule]	Oui	Oui	Oui	Oui
menu:ContextMenu	Oui	Non	Non	Non
mouseWheelEnabled:Boolean	Oui	Non	Non	Non
multiline:Boolean	Oui	Oui	Oui	Oui
onChanged = function(changedField:TextField) {}	Oui	Oui	Oui	Oui
onKillFocus = function(newFocus:Object) {}	Oui	Oui	Oui	Oui
onScroller = function(scrolledField:TextField) {}	Oui	Oui	Oui	Oui
onSetFocus = function(oldFocus:Object) {}	Oui	Oui	Oui	Oui
password:Boolean	Oui	Oui	Oui	Oui
removeListener(listener:Object) : Boolean	Oui	Oui	Oui	Oui
removeTextField() : Void	Oui	Oui	Oui	Oui
replaceSet(newText:String) : Void	Oui	Non	Non	Non
replaceText(beginIndex:Number, endIndex:Number, newText:String) : Void	Oui	Oui	Oui	Oui
restrict:String	Oui	Non	Non	Non
scroll:Number	Oui	Oui	Oui	Oui
selectable:Boolean	Oui	Oui	Oui	Oui
setNewTextFormat(tf:TextFormat) : Void	Oui	Oui	Oui	Oui
setTextFormat([beginIndex:Number], [endIndex:Number], textFormat:TextFormat) : Void	Oui	Oui	Oui	Oui
sharpness:Number	Oui	Non	Non	Non
styleSheet:StyleSheet	Oui	Non	Non	Non
tabEnabled:Boolean	Oui	Oui	Oui	Oui



Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
tabIndex:Number	Oui	Oui	Oui	Oui
text:String	Oui	Oui	Oui	Oui
textColor:Number	Oui	Oui	Oui	Oui
textHeight:Number	Oui	Oui	Oui	Oui
textWidth:Number	Oui	Oui	Oui	Oui
thickness:Number	Oui	Non	Non	Non
type:String	Oui	Oui	Oui	Oui
variable:String	Oui	Oui	Oui	Oui
wordWrap:Boolean	Oui	Oui	Oui	Oui

## TextFormat

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe TextFormat et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
align:String	Oui	Oui	Oui	Oui
blockIndent:Number	Oui	Oui	Oui	Oui
bold:Boolean	Oui	Oui	Oui	Oui
bullet:Boolean	Oui	Partielle	Partielle	Partielle
color:Number	Oui	Oui	Oui	Oui
font:String	Oui	Partielle	Partielle	Partielle
getTextExtent(text:String, [width:Number]) : Object	Oui	Oui	Oui	Oui
indent:Number	Oui	Oui	Oui	Oui
italic:Boolean	Oui	Oui	Oui	Oui
kerning:Boolean	Oui	Non	Non	Non
leading:Number	Oui	Oui	Oui	Oui
leftMargin:Number	Oui	Oui	Oui	Oui
letterSpacing:Number	Oui	Non	Non	Non
rightMargin:Number	Oui	Oui	Oui	Oui
size:Number	Oui	Oui	Oui	Oui
tabStops:Array	Oui	Partielle	Partielle	Partielle
target:String	Oui	Non	Non	Non

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
TextFormat([font:String], [size:Number], [color:Number], [bold:Boolean], [italic:Boolean], [underline:Boolean], [url:String], [target:String], [align:String], [leftMargin:Number], [rightMargin:Number], [indent:Number], [leading:Number])	Oui	Oui	Oui	Oui
underline:Boolean	Oui	Oui	Oui	Oui
url:String	Oui	Non	Non	Non

## TextRenderer

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe TextRenderer et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
displayMode:String	Oui	Non	Non	Non
maxLevel:Number	Oui	Non	Non	Non
setAdvancedAntialiasingTable(fontName:String, fontStyle:String, colorType:String, advancedAntialiasingTable:Array) : Void	Oui	Non	Non	Non

## TextSnapshot

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe TextSnapshot et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
findText(startIndex:Number, textToFind:String, caseSensitive:Boolean) : Number	Oui	Non	Non	Non
getCount() : Number	Oui	Non	Non	Non
getSelected(start:Number, [end:Number]) : Boolean	Oui	Non	Non	Non
getSelectedText([includeLineEndings:Boolean]) : String	Oui	Non	Non	Non
getText(start:Number, end:Number, [includeLineEndings:Boolean]) : String	Oui	Non	Non	Non
getTextRunInfo(beginIndex:Number, endIndex:Number) : Array	Oui	Non	Non	Non

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
hitTestTextNearPos(x:Number, y:Number, [closeDist:Number]) : Number	Oui	Non	Non	Non
setSelectColor(color:Number) : Void	Oui	Non	Non	Non
setSelected(start:Number, end:Number, select:Boolean) : Void	Oui	Non	Non	Non

## Transformer

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Transform et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
colorTransform:ColorTransform	Oui	Non	Non	Oui
concatenatedColorTransform:ColorTransform [read-only]	Oui	Non	Non	Oui
concatenatedMatrix:Matrix [read-only]	Oui	Non	Non	Oui
matrix:Matrix	Oui	Non	Non	Oui
pixelBounds:Rectangle	Oui	Non	Non	Oui
Transform(mc:MovieClip)	Oui	Non	Non	Oui

## Video

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe Video et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_alpha:Number	Oui	Non	Non	Non
_height:Number	Oui	Non	Non	Non
_name:String	Oui	Non	Non	Non
_parent:MovieClip	Oui	Non	Non	Non
_rotation:Number	Oui	Non	Non	Non
_visible:Boolean	Oui	Non	Non	Non
_width:Number	Oui	Non	Non	Non
_x:Number	Oui	Non	Non	Non
_xmouse:Number [lecture seule]	Oui	Non	Non	Non
_xscale:Number	Oui	Non	Non	Non
_y:Number	Oui	Non	Non	Non
_ymouse:Number [lecture seule]	Oui	Non	Non	Non

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
_yscale:Number	Oui	Non	Non	Non
attachVideo(source:Object) : Void	Oui	Non	Oui	Oui
clear() : Void	Oui	Non	Oui	Oui
close() : Void	Non	Oui	Oui	Oui
deblocking:Number	Oui	Non	Non	Oui
height:Number [read-only]	Oui	Non	Oui	Oui
onStatus = function(infoObject:Object) {}	Non	Oui	Oui	Oui
pause() : Void	Non	Oui	Oui	Oui
play() : Boolean	Non	Oui	Oui	Oui
resume() : Void	Non	Oui	Oui	Oui
smoothing:Boolean	Oui	Non	Non	Oui
stop() : Void	Non	Oui	Oui	Oui
width:Number [read-only]	Oui	Non	Oui	Oui

## XML

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe XML et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
addRequestHeader(header:Object, headerValue:String) : Void	Oui	Oui	Oui	Oui
contentType:String	Oui	Oui	Oui	Oui
createElement(name:String) : XMLNode	Oui	Oui	Oui	Oui
createTextNode(value:String) : XMLNode	Oui	Oui	Oui	Oui
docTypeDecl:String	Oui	Oui	Oui	Oui
getBytesLoaded() : Number	Oui	Oui	Oui	Oui
getBytesTotal() : Number	Oui	Oui	Oui	Oui
idMap:Object	Oui	Non	Non	Non
ignoreWhite:Boolean	Oui	Oui	Oui	Oui
load(url:String) : Boolean	Oui	Oui	Oui	Oui
loaded:Boolean	Oui	Oui	Oui	Oui
onData = function(src:String) {}	Oui	Oui	Oui	Oui
onHTTPStatus = function(httpStatus:Number) {}	Oui	Non	Non	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
onLoad = function(success:Boolean) {}	Oui	Non	Non	Oui
parseXML(value:String) : Void	Oui	Oui	Oui	Oui
send(url:String, [target:String], [method:String]) : Boolean	Oui	Oui	Oui	Oui
sendAndLoad(url:String, resultXML:XML) : Void	Oui	Oui	Oui	Oui
status:Number	Oui	Oui	Oui	Oui
XML(text:String)	Oui	Oui	Oui	Oui
xmlDecl:String	Oui	Oui	Oui	Oui

## XMLNode

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe XMLNode et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
appendChild(newChild:XMLNode) : Void	Oui	Oui	Oui	Oui
attributes:Object	Oui	Oui	Oui	Oui
childNodes:Array [lecture seule]	Oui	Oui	Oui	Oui
cloneNode(deep:Boolean) : XMLNode	Oui	Oui	Oui	Oui
firstChild:XMLNode [lecture seule]	Oui	Oui	Oui	Oui
getNamespaceForPrefix(prefix:String) : String	Oui	Non	Non	Non
getPrefixForNamespace(nsURI:String) : String	Oui	Non	Non	Non
hasChildNodes() : Boolean	Oui	Oui	Oui	Oui
insertBefore(newChild:XMLNode, insertPoint:XMLNode) : Void	Oui	Oui	Oui	Oui
lastChild:XMLNode [lecture seule]	Oui	Oui	Oui	Oui
localName:String [read-only]	Oui	Non	Non	Non
namespaceURI:String [read-only]	Oui	Non	Non	Non
nextSibling:XMLNode [lecture seule]	Oui	Oui	Oui	Oui
nodeName:String	Oui	Oui	Oui	Oui
nodeType:Number [lecture seule]	Oui	Oui	Oui	Oui
nodeValue:String	Oui	Oui	Oui	Oui
parentNode:XMLNode [lecture seule]	Oui	Oui	Oui	Oui
prefix:String [read-only]	Oui	Non	Non	Non

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
previousSibling:XMLNode [lecture seule]	Oui	Oui	Oui	Oui
removeNode() : Void	Oui	Oui	Oui	Oui
toString() : String	Oui	Oui	Oui	Oui
XMLNode(type:Number, value:String)	Oui	Oui	Oui	Oui

## XMLSocket

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe XMLSocket et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
close() : Void	Oui	Oui	Oui	Oui
connect(url:String, port:Number) : Boolean	Oui	Oui	Oui	Oui
onClose = function() {}	Oui	Oui	Oui	Oui
onConnect = function(success:Boolean) {}	Oui	Oui	Oui	Oui
onData = function(src:String) {}	Oui	Oui	Oui	Oui
onXML = function(src:XML) {}	Oui	Oui	Oui	Oui
send(data:Object) : Void	Oui	Oui	Oui	Oui
XMLSocket()	Oui	Oui	Oui	Oui

## XMLUI

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge la classe XMLUI et ses membres.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
accept() : Void	Oui	Non	Non	Non
cancel() : Void	Oui	Non	Non	Non
get(name:String) : String	Oui	Non	Non	Non
set(name:String, value:String) : Void	Oui	Non	Non	Non

## Propriétés et fonctions globales

Le tableau ci-dessous présente les versions de Flash Lite prenant en charge les diverses propriétés et fonctions globales.

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
add	Non	Oui	Oui	Oui
and	Non	Oui	Oui	Oui
Array([numElements:Number], [elementN:Object])	Oui	Oui	Oui	Oui
asfunction(function:String, parameter:String)	Oui	Non	Non	Non
Boolean(expression:Object)	Oui	Oui	Oui	Oui
break	Non	Oui	Oui	Oui
call(frame:Object)	Déconseillé	Déconseillé	Déconseillé	Déconseillé
case	Non	Oui	Oui	Oui
chr(number:Number)	Déconseillé	Déconseillé	Déconseillé	Déconseillé
clearInterval(intervalID:Number)	Oui	Oui	Oui	Oui
continue	Non	Oui	Oui	Oui
par défaut	Non	Oui	Oui	Oui
do while	Non	Oui	Oui	Oui
duplicateMovieClip(target:Object, newname:String, depth:Number)	Oui	Oui	Oui	Oui
else	Non	Oui	Oui	Oui
else if	Non	Oui	Oui	Oui
eq	Non	Oui	Oui	Oui
escape(expression:String)	Oui	Oui	Oui	Oui
eval(expression:Object)	Oui	Oui	Oui	Oui
0	Non	Oui	Oui	Oui
for	Non	Oui	Oui	Oui
for ... in	Non	Oui	Oui	Oui
fscommand(command:String, parameters:String)	Oui	Oui	Oui	Oui
fscommand2(command:String, parameters:String)	Non	Oui	Oui	Oui
fonction	Non	Oui	Oui	Oui
getProperty(my_mc:String, property)	Déconseillé	Oui	Oui	Oui
getTimer()	Oui	Oui	Oui	Oui
getURL(url:String, [window:String], [method:String])	Oui	Oui	Oui	Oui
getVersion()	Oui	Oui	Oui	Oui
gotoAndPlay([scene:String], frame:Object)	Oui	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
gotoAndStop([scene:String], frame:Object)	Oui	Oui	Oui	Oui
if	Non	Oui	Oui	Oui
iffFrameLoaded([scene:String], frame:Object)	Déconseillé	Déconseillé	Déconseillé	Déconseillé
int(value:Number)	Déconseillé	Déconseillé	Déconseillé	Déconseillé
isFinite(expression:Object)	Oui	Oui	Oui	Oui
isNaN(expression:Object)	Oui	Oui	Oui	Oui
length(expression:String, variable:Object)	Déconseillé	Déconseillé	Déconseillé	Déconseillé
loadMovie(url:String, target:Object, [method:String])	Oui	Oui	Oui	Oui
loadMovieNum(url:String, level:Number, [method:String])	Oui	Oui	Oui	Oui
loadVariables(url:String, target:Object, [method:String])	Oui	Oui	Oui	Oui
loadVariablesNum(url:String, level:Number, [method:String])	Oui	Oui	Oui	Oui
mbchr(number:Number)	Déconseillé	Déconseillé	Déconseillé	Déconseillé
mblength(string:String)	Déconseillé	Déconseillé	Déconseillé	Déconseillé
mbord(character:String)	Déconseillé	Déconseillé	Déconseillé	Déconseillé
mbsubstring(value:, index:Number, count:Number)String	Déconseillé	Déconseillé	Déconseillé	Déconseillé
MMExecute(command:String)	Oui	Non	Non	Non
ne	Non	Oui	Oui	Oui
new	Non	Oui	Oui	Oui
newline	Non	Oui	Oui	Oui
nextFrame()	Oui	Oui	Oui	Oui
nextScene()	Oui	Oui	Oui	Oui
pas	Non	Oui	Oui	Oui
Number(expression:Object)	Oui	Oui	Oui	Oui
Object([value:Object])	Oui	Oui	Oui	Oui
on(mouseEvent:Object)	Oui	Oui	Oui	Oui
onClipEvent(movieEvent:Object)	Oui	Oui	Oui	Oui
ord(character:String)	Déconseillé	Déconseillé	Déconseillé	Déconseillé
parseFloat(string:String)	Oui	Oui	Oui	Oui
parseInt(expression:String, [radix:Number])	Oui	Oui	Oui	Oui



Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
play()	Oui	Oui	Oui	Oui
prevFrame()	Oui	Oui	Oui	Oui
prevScene()	Oui	Oui	Oui	Oui
print(target:Object, boundingBox:String)	Oui	Non	Non	Non
printAsBitmap(target:Object, boundingBox:String)	Oui	Non	Non	Non
printAsBitmapNum(level:Number, boundingBox:String)	Oui	Non	Non	Non
printNum(level:Number, boundingBox:String)	Oui	Non	Non	Non
random(value:Number)	Déconseillé	Déconseillé	Déconseillé	Déconseillé
removeMovieClip(target:Object)	Oui	Oui	Oui	Oui
return	Non	Oui	Oui	Oui
set	Non	Oui	Oui	Oui
setInterval(functionReference:Function, interval:Number, [param:Object], objectReference:Object, methodName:String)	Oui	Oui	Oui	Oui
setProperty(target:Object, property:Object, expression:Object)	Oui	Oui	Oui	Oui
showRedrawRegions(enable:Boolean, [color:Number])	Oui	Non	Non	Non
startDrag(target:Object, [lock:Boolean], [left,top,right,bottom:Number])	Oui	Oui	Oui	Oui
stop()	Oui	Oui	Oui	Oui
stopAllSounds()	Oui	Oui	Oui	Oui
stopDrag()	Oui	Oui	Oui	Oui
String(expression:Object)	Oui	Oui	Oui	Oui
substring(:String, index:Number, count:Number)String	Déconseillé	Déconseillé	Déconseillé	Déconseillé
switch	Non	Oui	Oui	Oui
targetPath(targetObject:Object)	Oui	Oui	Oui	Oui
tellTarget(target:String, statement(s))	Déconseillé	Déconseillé	Déconseillé	Déconseillé
toggleHighQuality()	Déconseillé	Déconseillé	Déconseillé	Déconseillé
trace(expression:Object)	Oui	Oui	Oui	Oui
unescape(string:String)	Oui	Oui	Oui	Oui

Nom de la fonction	ActionScript 2 ;	Flash Lite 2.1	Flash Lite 3.0	Flash Lite 3.1
unloadMovie(target:Object)	Oui	Oui	Oui	Oui
unloadMovieNum(level:Number)	Oui	Oui	Oui	Oui
updateAfterEvent()	Oui	Non	Non	Non

## Classes non prises en charge ou partiellement prises en charge : détails

Le présent chapitre décrit les classes ActionScript 2.0 qui ne sont pas prises en charge par Macromedia Flash Lite 2.0, Flash Lite 2.1 et Flash Lite 3.x d'Adobe ou qui le sont partiellement. Il décrit également les extensions propres au code ActionScript pour Flash Lite 2.x et 3.x.

### Classe Button

Tous les symboles de bouton présents dans un fichier SWF sont des occurrences de l'objet Button. La classe Button fournit des méthodes, propriétés et gestionnaires d'événements pour l'utilisation des boutons.

Pour plus d'informations sur la classe Button, consultez les documents et chapitres suivants :

- Chapitre 10, « Gestion d'événements », dans *Formation à ActionScript 2.0 dans Adobe Flash*
- Chapitre 2, « Classes ActionScript », dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*

### Résumé des propriétés

Le tableau suivant regroupe les propriétés qui ne sont pas prises en charge, ou seulement de façon partielle, par la classe Button lorsque vous utilisez le code ActionScript pour Flash Lite 2.x. et 3.x.

Propriété	Description	Prise en charge
menu	Objet qui associe un objet ContextMenu à un bouton.	Pas de prise en charge
trackAsMenu	Valeur booléenne indiquant si d'autres boutons peuvent recevoir des événements lorsque l'utilisateur relâche le bouton de la souris. Prise en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .	Prise en charge partielle
useHandCursor	Valeur booléenne indiquant si le pointeur doit s'afficher lorsque la souris survole un bouton.	Pas de prise en charge
_xmouse	Lecture seule ; coordonnée x du pointeur, par rapport à une occurrence de bouton. Prise en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .	Prise en charge partielle
_ymouse	Lecture seule ; coordonnée y du pointeur, par rapport à une occurrence de bouton. Prise en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .	Prise en charge partielle

## Résumé des événements

Le tableau ci-dessous répertorie les gestionnaires d'événements qui sont pris en charge de façon partielle par la classe `Button` lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et 3.x.

Événement	Description
<code>onDragOut</code>	Appelé lorsque vous cliquez sur le bouton à l'aide du bouton de la souris, puis faites glisser le pointeur hors du bouton.  <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .
<code>onDragOver</code>	Appelé lorsque le pointeur glisse au-dessus du bouton.  <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .
<code>onReleaseOutside</code>	Appelé lorsque l'utilisateur relâche la souris tandis que le pointeur se trouve hors du bouton après avoir appuyé sur celui-ci lorsque le pointeur se trouvait à l'intérieur du bouton.  <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .

## Classe Date

La classe `Date` détermine la représentation des dates et des heures dans ActionScript et prend en charge leur manipulation. La classe `Date` permet également d'obtenir la date et l'heure du système d'exploitation.

Pour plus d'informations sur la classe `Date`, consultez les documents et chapitres suivants :

- Chapitre 4, « Données et types de données », dans *Formation à ActionScript 2.0 dans Adobe Flash*
- Chapitre 2, « Classes ActionScript », dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*

## Résumé de la méthode

Le tableau ci-dessous affiche la liste des méthodes (fonctions) qui ont été ajoutées à la classe `Date` lors de l'utilisation du code ActionScript pour Flash Lite 2.x et 3.x.

Méthode	Description
<code>getLocaleLongDate()</code>	Cette fonction définit un paramètre sur une chaîne représentant la date actuelle, sous forme longue, mise en forme selon les paramètres régionaux actuellement définis. Le paramètre est transmis avec le nom. La valeur renvoyée est une chaîne de longueur variable et à plusieurs caractères. Le formatage final dépend du périphérique et des paramètres régionaux sélectionnés.
<code>getLocaleShortDate()</code>	Cette fonction définit un paramètre sur une chaîne représentant la date actuelle, sous forme abrégée, mise en forme selon les paramètres régionaux actuellement définis. Le paramètre est transmis avec le nom. La valeur renvoyée est une chaîne de longueur variable et à plusieurs caractères. Le formatage final dépend du périphérique et des paramètres régionaux sélectionnés.
<code>getLocaleTime()</code>	Cette fonction définit un paramètre sur une chaîne représentant l'heure actuelle, mise en forme selon les paramètres régionaux actuellement définis. Le paramètre est transmis avec le nom. La valeur renvoyée est une chaîne de longueur variable et à plusieurs caractères. Le formatage final dépend du périphérique et des paramètres régionaux sélectionnés.

## Classe Key

La classe Key fournit des méthodes et des propriétés permettant d'obtenir des informations sur les actions effectuées sur le clavier et les frappes de l'utilisateur.

Pour plus d'informations sur la classe Key, consultez les documents et chapitres suivants :

- Chapitre 14, « Création d'interactivité avec ActionScript », dans *Formation à ActionScript 2.0 dans Adobe Flash*
- Chapitre 2, « Classes ActionScript », dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*

## Extensions des propriétés Flash Lite

Le tableau ci-dessous décrit la propriété qui a été ajoutée à la classe Key lors de l'utilisation du code ActionScript pour Flash Lite 2.x et versions ultérieures.

Propriété	Description	Prise en charge
isToggled	Renvoie true si la touche de verrouillage numérique ou des majuscules est active.	Pas de prise en charge

## Extensions des méthodes Flash Lite

Le tableau ci-dessous décrit l'extension de méthode qui a été ajoutée à la classe Key lors de l'utilisation du code ActionScript pour Flash Lite 2.x et versions ultérieures.

Méthode	Description
getCode ()	Renvoie la valeur de code de touche virtuelle de la dernière touche enfoncée.

L'implémentation de Flash Lite 2.x et 3.x de la méthode `of the getCode ()` renvoie une chaîne ou un nombre, suivant le paramètre renvoyé par la plate-forme. Les seuls codes valides sont les codes standard acceptés par cette classe et les codes « spéciaux » répertoriés comme propriétés de la classe `ExtendedKey`. Cette restriction est appliquée par le lecteur. Pour les codes de touche valides, consultez la classe Key dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*. Ce document comporte des tableaux de correspondance entre les touches et les codes de lettres, de chiffres, du pavé numérique, des touches de fonction, des touches spéciales de constante et d'autres touches.

## Classe Mouse

La classe Mouse permet de contrôler la souris dans un fichier SWF. Vous pouvez l'utiliser pour masquer ou afficher le pointeur de la souris, par exemple.

Pour plus d'informations sur la classe Mouse, consultez les références suivantes :

- Chapitre 14, « Création d'interactivité avec ActionScript », dans *Formation à ActionScript 2.0 dans Adobe Flash*
- Chapitre 2, « Classes ActionScript », dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*

## Résumé de la méthode

Le tableau ci-dessous répertorie les méthodes qui ne sont pas prises en charge, ou seulement de façon partielle, par la classe Mouse lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Méthode	Description	Prise en charge
<code>addListener()</code>	Enregistre un objet pour qu'il reçoive les notifications des écouteurs <code>onMouseDown</code> , <code>onMouseMove</code> , <code>onMouseUp</code> et <code>onMouseWheel</code> .  <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .	Prise en charge partielle
<code>hide()</code>	Masque le pointeur dans un fichier SWF.	Pas de prise en charge
<code>removeListener()</code>	Supprime un objet précédemment enregistré avec <code>addListener()</code> .  <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .	Prise en charge partielle
<code>show()</code>	Affiche le pointeur dans un fichier SWF.	Pas de prise en charge

## Résumé des événements

Le tableau ci-dessous répertorie les événements qui ne sont pas pris en charge, ou seulement de façon partielle, par la classe `MouseEvent` lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Événement	Description	Prise en charge
<code>onMouseDown</code>	Signalé lorsque vous cliquez avec la souris.  <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .	Prise en charge partielle
<code>onMouseMove</code>	Signalé lorsque la souris bouge.  <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> est défini sur <code>true</code> .	Prise en charge partielle
<code>onMouseUp</code>	Signalé lorsque le bouton de la souris est relâché.  <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .	Prise en charge partielle
<code>onMouseWheel</code>	Signalé lorsque l'utilisateur actionne la molette de la souris.	Pas de prise en charge

## classe `MovieClip`

La classe `MovieClip` permet d'utiliser les fonctions de rappel de l'écouteur qui indiquent l'état du chargement/téléchargement des fichiers SWF ou JPEG dans des clips. Pour utiliser les fonctionnalités `MovieClip`, utilisez `MovieClipLoader.loadClip()` à la place de `loadMovie()` ou `MovieClip.loadMovie()` pour charger des fichiers SWF.

Pour plus d'informations sur la classe `MovieClip`, consultez les documents et chapitres suivants :

- Chapitre 11, « Utilisation des clips », dans *Formation à ActionScript 2.0 dans Adobe Flash*
- Chapitre 2, « Classes ActionScript », dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*

## Résumé de la méthode

Le tableau ci-dessous répertorie les méthodes qui ne sont pas prises en charge, ou seulement de façon partielle, par la classe MovieClip lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Méthode	Description	Prise en charge
<code>attachAudio()</code>	Capture et lit des données audio locales à l'aide du microphone du périphérique.	Pas de prise en charge
<code>getTextSnapshot()</code>	Renvoie un objet <code>TextSnapshot</code> contenant le texte dans des champs texte statique dans le clip spécifié.	Pas de prise en charge
<code>startDrag()</code>	Rend un clip déplaçable et amorce son déplacement.  <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .	Prise en charge partielle
<code>stopDrag()</code>	Arrête une méthode <code>MovieClip.startDrag()</code> . Tout clip déplaçable avec <code>startDrag()</code> le reste jusqu'à ce qu'une méthode <code>stopDrag()</code> soit ajoutée ou qu'un autre clip soit déplaçable. Vous ne pouvez déplacer qu'un seul clip à la fois.  <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .	Prise en charge partielle

## Résumé des propriétés

Le tableau suivant regroupe les propriétés qui ne sont pas prises en charge, ou seulement de façon partielle, par la classe MovieClip lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Propriété	Description	Prise en charge
<code>_droptarget</code>	Renvoie le chemin absolu, en utilisant une notation de syntaxe à barre oblique, de l'occurrence de clip sur laquelle ce clip a été déposé. La propriété <code>_droptarget</code> renvoie toujours un chemin qui commence par une barre oblique (/). Pour comparer la propriété <code>_droptarget</code> d'une instance à une référence, utilisez la fonction <code>eval()</code> afin de convertir la valeur renvoyée d'une syntaxe à barre oblique en référence de syntaxe à point.  <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .	Prise en charge partielle
<code>menu</code>	Associe l'objet <code>ContextMenu</code> spécifié au clip.	Pas de prise en charge
<code>_quality</code>	Définit ou extrait la qualité du rendu appliqué à un fichier SWF. Les polices de périphérique sont toujours aliasées, ce qui implique qu'elles ne sont pas affectées par la propriété <code>_quality</code> .	Prise en charge partielle
<code>trackAsMenu</code>	Valeur booléenne indiquant si d'autres boutons ou clips peuvent recevoir des événements lorsque l'utilisateur relâche le bouton de la souris. La propriété <code>trackAsMenu</code> permet de créer des menus. Vous pouvez définir la propriété <code>trackAsMenu</code> sur n'importe quel bouton ou objet clip. Si la propriété <code>trackAsMenu</code> n'existe pas, le comportement par défaut est défini sur <code>false</code> .  <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .	Prise en charge partielle

Propriété	Description	Prise en charge
<code>useHandCursor</code>	Valeur booléenne qui détermine si le curseur en forme de main doit apparaître lorsque la souris survole un clip.	Pas de prise en charge
<code>_xmouse</code>	Renvoie la coordonnée x de la position de la souris. <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .	Prise en charge partielle
<code>_ymouse</code>	Renvoie la coordonnée y de la position de la souris. <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .	Prise en charge partielle

## Résumé des événements

Le tableau ci-dessous répertorie les gestionnaires d'événements qui sont pris en charge de façon partielle par la classe `MovieClip` lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Gestionnaire d'événements	Description
<code>onDragOut</code>	Appelé lorsque l'utilisateur appuie sur le bouton de la souris et si le pointeur se déplace hors de l'objet. Vous devez définir une fonction qui s'exécute lorsque le gestionnaire d'événements est appelé. Cette fonction peut être définie dans le scénario ou dans un fichier de classe qui étend la classe <code>MovieClip</code> ou est lié à l'un des symboles de la bibliothèque. <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .
<code>onDragOver</code>	Appelé lorsque l'utilisateur fait glisser le pointeur hors du clip, puis sur le clip. Vous devez définir une fonction qui s'exécute lorsque le gestionnaire d'événements est appelé. Cette fonction peut être définie dans le scénario ou dans un fichier de classe qui étend la classe <code>MovieClip</code> ou est lié à l'un des symboles de la bibliothèque. <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .
<code>onMouseDown</code>	Appelé lorsque le bouton gauche de la souris est sollicité. <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .
<code>onMouseMove</code>	Appelé lorsque la souris bouge. <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> est défini sur <code>true</code> .
<code>onMouseUp</code>	Appelé lorsque le bouton gauche de la souris est relâché. <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .
<code>onReleaseOutside</code>	Appelé lorsque l'utilisateur appuie sur le bouton de la souris tandis que le pointeur est sur le clip, puis le relâche alors que le pointeur est hors de la zone occupée par le clip. <b>Limitations :</b> pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .

## Classe NetConnection

La classe NetConnection permet de créer un objet que vous pouvez utiliser avec un objet NetStream pour appeler des commandes sur un serveur d'applications distant ou pour lire des fichiers vidéo en flux continu (FLV) soit localement soit à partir d'un serveur. La prise en charge de cette classe a été ajoutée dans Flash Lite 3.0.

Pour plus d'informations sur la classe NetConnection, consultez les documents et chapitres suivants :

- « Utilisation des images, du son et de la vidéo » dans *Formation à ActionScript 2.0 dans Flash*
- Chapitre 2, « Classes ActionScript », dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*

## Classe NetStream

La classe NetStream permet de créer un flux que vous pouvez utiliser avec un objet NetConnection pour lire des fichiers FLV à partir d'un système de fichiers local, d'une adresse HTTP ou de Flash Media Server. La prise en charge de cette classe a été ajoutée dans Flash Lite 3.0.

Pour plus d'informations sur la classe NetStream, consultez les documents et chapitres suivants :

- « Utilisation des images, du son et de la vidéo » dans *Formation à ActionScript 2.0 dans Flash*
- Chapitre 2, « Classes ActionScript », dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*

## Résumé des propriétés

Le tableau ci-dessous indique la propriété qui n'est pas prise en charge par la classe NetStream lorsque vous utilisez le code ActionScript pour Flash Lite 3.0 ou versions ultérieures.

Propriété	Description
checkPolicyFile	Indique si le lecteur doit tenter de télécharger un fichier de régulation inter-domaines à partir du serveur du fichier FLV chargé avant de commencer à charger le fichier FLV en lui-même.

## Classe Selection

La classe Selection vous permet de définir et de contrôler le champ texte dans lequel se trouve le point d'insertion (à savoir, le champ ayant le focus).

La prise en charge partielle de la classe Selection permet de prendre également en charge la saisie de texte en ligne dans Flash Lite 2.1. Dans Flash Lite, l'objet Selection n'est valide que si le périphérique prend en charge la saisie de texte en ligne. Dans le cas contraire, et s'il repose sur un processeur frontal (FEP - front-end processor) pour saisir du texte, tous les appels à l'objet Selection sont ignorés.

Pour de plus amples informations sur la classe Selection, reportez-vous au chapitre 2, « Classes ActionScript » du *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*.

## Résumé de la méthode

Le tableau ci-dessous répertorie les méthodes qui ne sont pas prises en charge par la classe Selection lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.



Méthode	Description
getBeginIndex()	Renvoie l'index au début de la plage de sélection qui a actuellement le focus.
getCaretIndex()	Renvoie l'index de la position du point d'insertion clignotant (caret).
getEndIndex	Renvoie l'index à la fin de la plage de sélection qui a actuellement le focus.

## Classe SharedObject

Un objet Flash Lite partagé, tel qu'implémenté dans la classe SharedObject d'ActionScript, permet au contenu Flash Lite d'enregistrer des données sur le périphérique lorsque l'application est fermée et de charger ces données lorsque l'application est activée de nouveau.

Pour plus d'informations sur la classe SharedObject, consultez les références suivantes :

- Chapitre 14, « Création d'interactivité avec ActionScript », dans *Formation à ActionScript 2.0 dans Adobe Flash*
- Chapitre 2, « Classes ActionScript », dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*

## Résumé de la méthode

Le tableau ci-dessous répertorie les méthodes qui sont prises en charge de façon partielle par la classe SharedObject lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Méthode	Description
flush()	<p>Ecrit immédiatement un objet rémanent dans un fichier local.</p> <p><b>Limitations :</b> l'opération d'écriture est asynchrone et le résultat n'est pas disponible immédiatement.</p>
getLocal()	<p>Renvoie une référence à un objet partagé rémanent du périphérique et disponible uniquement pour le client actuel.</p> <p><b>Limitations :</b> Dans Flash Lite, un objet partagé ne peut pas être partagé entre deux fichiers SWF.</p>

## Extensions des méthodes Flash Lite

Le tableau ci-dessous dresse la liste des méthodes qui ont été ajoutées en tant qu'extensions à la classe SharedObject lors de l'utilisation du code ActionScript pour Flash Lite 2.x et versions ultérieures.

Méthode	Description
GetMaxSize()	Flash Lite renvoie la taille maximale attribuée au stockage fixe d'un fichier SWF.

## Classe System

La classe System regroupe les propriétés liées à certaines opérations qui ont lieu sur l'ordinateur de l'utilisateur, telles que les opérations au niveau des objets partagés, des paramètres locaux pour les caméras et les microphones et de l'utilisation du Presse-papiers. Les propriétés et les méthodes supplémentaires suivantes figurent dans des classes spécifiques dans le package System : la classe capabilities, la classe security et la classe IME.

Pour plus d'informations sur la classe System, consultez les références suivantes :

- *Développement d'applications Flash Lite 2.x et 3.x*
- *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*

- Chapitre 7, « Classes », de *Formation à ActionScript 2.0 dans Adobe Flash*

## Résumé de la méthode

Le tableau ci-dessous répertorie les méthodes qui ne sont pas prises en charge par la classe System lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Méthode	Description	Prise en charge
<code>setClipboard()</code>	Remplace le contenu du Presse-papiers par une chaîne de texte spécifiée.	Pas de prise en charge
<code>showSettings()</code>	Affiche le panneau Paramètres de Flash Player spécifié.	Pas de prise en charge

## Résumé des propriétés

Le tableau ci-dessous répertorie les propriétés qui ne sont pas prises en charge par la classe System lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Propriété	Description	Prise en charge
<code>exactSettings</code>	Permet d'appliquer ou non les règles de correspondance exacte de superdomaines ou de domaines lors de l'accès à des paramètres locaux.	Pas de prise en charge

## Résumé des événements

Le tableau ci-dessous indique les gestionnaires d'événements qui sont pris en charge partiellement par la classe System lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Gestionnaire d'événements	Description	Prise en charge
<code>onStatus</code>	Fournit un gestionnaire de superévénements pour certains objets. La propriété <code>SharedObject</code> est prise en charge. Les propriétés <code>LocalConnection</code> et <code>NetStream</code> ne sont pas prises en charge.	Prise en charge partielle

## Classe capacités (System.capabilities)

La classe `System.capabilities` permet de déterminer les fonctionnalités du système et le lecteur hébergeant un fichier SWF, ceci dans le but d'adapter le contenu à différents formats. Par exemple, l'écran d'un téléphone portable (noir et blanc, 100 pixels carrés) diffère de l'écran couleurs de 1 000 pixels carrés d'un PC. Pour fournir le contenu approprié au plus grand nombre d'utilisateurs possible, vous pouvez utiliser l'objet `System.capabilities` afin de déterminer le type de périphérique dont dispose un utilisateur. Vous pouvez ensuite demander au serveur d'envoyer différents fichiers SWF en fonction des fonctionnalités propres à chaque périphérique ou demander au fichier SWF de modifier sa présentation en fonction des fonctionnalités du périphérique.

## Résumé des propriétés

Le tableau ci-dessous dresse la liste des propriétés qui ne sont pas prises en charge par la classe `System.capabilities` lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Propriété	Description
hasIME	Indique si le système est doté d'un IME (éditeur de méthode d'entrée).
manufacturer	Chaîne désignant le système d'exploitation de Flash Player, sous la forme « Adobe nomSE » (nomSE peut être « Windows », « Macintosh », « Linux » ou « Autre nom de SE »).
pixelAspectRatio	Indique les proportions en pixels de l'écran.
playerType	Indique le type de lecteur : autonome, externe, module externe ou ActiveX.
screenColor	Indique si l'écran est en couleur, en niveaux de gris ou noir et blanc.
screenDPI	Indique la résolution de l'écran, en pixels (telle que ppp).
serverString	Chaîne de code URL spécifiant les valeurs de chaque propriété <code>System.capabilities</code> .

## Extensions des propriétés Flash Lite

Le tableau ci-dessous répertorie les propriétés servant d'extensions à la classe `System.capabilities` lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Propriété	Description
hasCompoundSound	Valeur booléenne en lecture seule qui a la valeur <code>true</code> si le lecteur peut traiter les données audio composites ; elle est définie sur <code>false</code> dans les autres cas.
hasEmail	Valeur booléenne en lecture seule qui est définie sur <code>true</code> si le lecteur peut envoyer des messages électroniques à l'aide de la commande ActionScript <code>GetURL</code> ; elle est définie sur <code>false</code> dans le cas contraire.
hasMMS	Valeur booléenne en lecture seule qui est définie sur <code>true</code> si le lecteur peut envoyer des messages MMS (Multimedia message service) à l'aide de la commande ActionScript <code>GetURL</code> ; elle est définie sur <code>false</code> dans le cas contraire.
hasSMS	Valeur en lecture seule dont la variable <code>_capsms</code> indique si Flash Lite permet d'envoyer des messages SMS à l'aide de la commande ActionScript <code>GetURL</code> . Si Flash Lite peut envoyer des messages SMS, cette variable est définie et prend la valeur 1. Sinon, cette variable n'est pas définie.
hasMFI	Valeur booléenne en lecture seule qui est définie sur <code>true</code> si le lecteur peut lire des données audio au format Melody pour i-mode (MFI) ; elle est définie sur <code>false</code> dans le cas contraire.
hasMIDI	Valeur booléenne en lecture seule qui est définie sur <code>true</code> si le lecteur peut lire les données audio au format MIDI ; elle est définie sur <code>false</code> dans le cas contraire.
hasSMAF	Valeur booléenne en lecture seule qui est définie sur <code>true</code> si le lecteur peut lire les données audio au format SMAF (Synthetic Music Mobile Application Format) ; elle est définie sur <code>false</code> dans le cas contraire.
hasDataLoading	Valeur booléenne en lecture seule qui est définie sur <code>true</code> si le lecteur peut charger de façon dynamique d'autres données via des appels à <code>loadMovie()</code> , <code>loadMovieNum()</code> , <code>loadVariables()</code> , <code>loadVariablesNum()</code> , <code>XML.parseXML()</code> , <code>Sound.loadSound()</code> , <code>MovieClip.loadVariables()</code> , <code>MovieClip.loadMovie()</code> , <code>MovieClipLoader.loadClip()</code> , <code>LoadVars.load()</code> et <code>LoadVars.sendAndLoad()</code> ; elle est définie sur <code>false</code> dans le cas contraire.

Propriété	Description
has4WayKeyAS	Valeur booléenne en lecture seule qui est définie sur <code>true</code> si le lecteur peut exécuter un code ActionScript joint aux gestionnaires <code>KeyEvent</code> associés aux touches Gauche, Droite, Haut et Bas ; elle est définie sur <code>false</code> dans le cas contraire.  Si la valeur de cette variable est <code>true</code> , lorsque l'une des quatre touches de direction est sollicitée, le lecteur recherche d'abord un gestionnaire correspondant. S'il n'en trouve aucun, Flash procède à un contrôle de navigation. Cependant, si un gestionnaire d'événements est trouvé, aucune action de navigation ne se produit pour cette touche. En d'autres termes, la présence d'un gestionnaire de pression de touche pour la touche Bas désactive la fonctionnalité de navigation vers le bas.
hasMouse	Valeur booléenne en lecture seule qui est définie sur <code>true</code> si le lecteur peut envoyer des événements liés à la souris et sur <code>false</code> si la plate-forme ne prend pas en charge de souris.
hasMappableSoftKeys	Permet à l'utilisateur de définir les valeurs des touches programmables et de gérer les événements de ces touches.
hasStylus	Valeur booléenne en lecture seule qui est définie sur <code>true</code> si le lecteur peut envoyer des événements liés au stylet et sur <code>false</code> si la plate-forme ne prend pas en charge de stylet.  L'événement <code>onMouseMove</code> n'est pas pris en charge par un stylet. Cet indicateur permet au contenu de vérifier si l'événement est pris en charge.
hasCMIDI	Valeur booléenne en lecture seule qui est définie sur <code>true</code> si la plate-forme prend en charge les sons CMIDI et sur <code>false</code> dans le cas contraire.
hasXMLSocket	(Ajoutée dans Flash Lite 2.1) Indique si l'application hôte prend en charge les sockets XML.
softKeyCount	Valeur numérique indiquant le nombre de touches programmables prises en charge par la plate-forme.
hasSharedObjects	Valeur booléenne en lecture seule qui est définie sur <code>true</code> si le contenu Flash lu dans cette application peut accéder aux objets partagés de Flash Lite ; elle est définie sur <code>false</code> dans les autres cas.
hasQWERTYKeyboard	Valeur booléenne en lecture seule qui est définie sur <code>true</code> si le code ActionScript peut être associé à l'ensemble des touches d'un clavier QWERTY standard, ce qui inclut la touche Retour arrière ; elle est définie sur <code>false</code> dans le cas contraire.
audioMIMEtypes	Propriété en lecture seule qui contient un tableau des types MIME pris en charge par les codecs audio du périphérique et qui peut être utilisée par l'objet <code>Sound</code> d'ActionScript.
imageMIMEtypes	Propriété en lecture seule qui contient un tableau des types MIME pris en charge par les codecs d'image du périphérique et qui peut être utilisée par la fonction <code>loadMovie</code> d'ActionScript.
videoMIMEtypes	Propriété en lecture seule qui contient un tableau des types MIME pris en charge par les codecs vidéo du périphérique et qui peut être utilisée par l'objet <code>Video</code> d'ActionScript.
MIMEtypes	Propriété en lecture seule qui contient un tableau de l'ensemble des types MIME que les objets <code>Sound</code> et <code>Video</code> prennent en charge et qui peut être utilisée par la fonction <code>loadMovie()</code> d'ActionScript.

## Classe Sound

ActionScript pour Flash Lite 2.x et versions ultérieures prend en charge les sons de périphérique par l'intermédiaire de la classe Sound et des valeurs System.capabilities. La classe Sound est totalement prise en charge pour les sons natifs pris en charge dans Flash Player 7, mais est uniquement prise en charge de façon partielle pour les sons de périphérique.

Flash Lite 2.x a ajouté une prise en charge qui permet de synchroniser la lecture de sons de périphérique avec l'animation de rendu.

**Remarque :** Flash Lite 2.x et versions ultérieures ne prennent pas en charge l'enregistrement de sons.

### Résumé de la méthode

Le tableau ci-dessous indique les méthodes qui sont prises en charge de façon partielle par la classe Sound lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Méthode	Description	Prise en charge
<code>getPan()</code>	Renvoie le niveau panoramique défini dans le dernier appel <code>setPan()</code> sous la forme d'un entier compris entre -100 (gauche) et +100 (droite). (0 définit de la même manière les canaux gauche et droit.) Le réglage panoramique contrôle la balance gauche-droite des sons actuels et futurs d'un fichier SWF.  <b>Limitations :</b> prise en charge pour l'utilisation avec des sons Flash natifs ; non prise en charge pour l'utilisation avec des sons de périphérique.	Prise en charge partielle
<code>getTransform()</code>	Renvoie les informations de transformation du son pour l'objet Sound spécifié défini avec le dernier appel <code>Sound.setTransform()</code> .  <b>Limitations :</b> prise en charge pour l'utilisation avec des sons Flash natifs ; non prise en charge pour l'utilisation avec des sons de périphérique.	Prise en charge partielle
<code>loadSound()</code>	Charge un fichier MP3 dans un objet Sound. Vous pouvez utiliser le paramètre <code>isStreaming</code> pour indiquer si le son est un événement ou un son en flux continu.  Les sons d'événement sont totalement chargés avant d'être lus. Ils sont gérés par la classe ActionScript Sound et répondent à toutes les méthodes et propriétés de cette classe.  <b>Limitations :</b> le paramètre de diffusion en flux continu est ignoré lorsqu'il est utilisé avec Flash Lite 2.x et versions ultérieures.	Prise en charge partielle

Méthode	Description	Prise en charge
<code>setPan()</code>	Détermine le mode de lecture du son dans les canaux gauche et droit (haut-parleurs). Pour les sons mono, pan détermine le haut-parleur (gauche ou droit) de lecture du son.  <b>Limitations :</b> prise en charge pour l'utilisation avec des sons Flash natifs ; non prise en charge pour l'utilisation avec des sons de périphérique.	Prise en charge partielle
<code>setTransform()</code>	Définit les informations de transformation du son (ou balance) d'un objet Sound.  Le paramètre <code>soundTransformObject</code> est un objet que vous créez à l'aide de la méthode de constructeur de la classe <code>Object</code> générique avec les paramètres spécifiant la manière dont le son est réparti dans les canaux gauche et droit (haut-parleurs).  <b>Limitations :</b> prise en charge pour l'utilisation avec des sons Flash natifs ; non prise en charge pour l'utilisation avec des sons de périphérique.	Prise en charge partielle
<code>setVolume()</code>	Définit le volume pour l'objet Sound.  <b>Limitations :</b> prise en charge pour l'utilisation avec des sons Flash natifs ; non prise en charge pour l'utilisation avec des sons de périphérique.	Prise en charge partielle

## Résumé des propriétés

Le tableau ci-dessous répertorie les propriétés de la classe `Sound` qui sont prises en charge de façon partielle lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Propriété	Description
<code>duration</code>	Durée d'un son, en millisecondes.  <b>Limitations :</b> prise en charge pour l'utilisation avec des sons Flash natifs ; non prise en charge pour l'utilisation avec des sons de périphérique.
<code>position</code>	La durée de diffusion du son exprimée en millisecondes.  <b>Limitations :</b> prise en charge pour l'utilisation avec des sons Flash natifs ; non prise en charge pour l'utilisation avec des sons de périphérique.

## Extensions des méthodes Flash Lite

Le tableau ci-dessous répertorie les nouvelles méthodes de la classe `Sound` qui sont propres au code ActionScript pour Flash Lite 2.x et versions ultérieures.

Méthode	Description
<code>getPan()</code>	Renvoie la valeur de l'appel <code>setPan()</code> précédent. Cette méthode n'est pas prise en charge pour les sons de périphérique.
<code>getTransform()</code>	Renvoie la valeur de l'appel <code>setTransform()</code> précédent. Cette méthode n'est pas prise en charge pour les sons de périphérique.
<code>loadSound()</code>	Charge les données audio de tous les formats dans Flash Player.  Cette méthode diffère de l'implémentation de Flash Player 7 dans la mesure où les données audio chargées à l'aide de cette méthode sont toujours traitées en tant que son d'événement. Par conséquent, le deuxième paramètre de cette méthode est toujours ignoré. Dans l'exemple suivant, la valeur <code>true</code> est ignorée :  <code>my_sound.loadSound("mysnd.mp3", true);</code>

## Classe Stage

La classe Stage est une classe de niveau supérieur dont les méthodes, les propriétés et les gestionnaires sont accessibles sans l'aide d'un constructeur. Les méthodes et propriétés de cette classe permettent d'accéder aux informations et de les modifier dans les limites d'un fichier SWF.

Pour plus d'informations, consultez le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*.

## Extensions des propriétés Flash Lite

Le tableau ci-dessous indique les nouvelles propriétés de la classe Stage qui sont propres au code ActionScript pour Flash Lite 2.x et versions ultérieures.

Propriété	Description
showMenu	Affiche ou masque les éléments par défaut dans le menu contextuel de Flash Player.
getTransform()	Renvoie la valeur de l'appel <code>setTransform()</code> précédent. Cette méthode n'est pas prise en charge pour les sons de périphérique.
loadSound()	Charge les données audio de tous les formats dans Flash Player. Cette méthode diffère de la version pour Flash Player 7 dans la mesure où les données audio chargées à l'aide de cette méthode sont toujours traitées en tant que sons d'événement, de façon à ce que le deuxième paramètre de cette méthode soit toujours ignoré. Par exemple, dans l'appel suivant, la valeur <code>true</code> est ignorée :  <code>my_sound.loadSound("mysnd.mp3", true);</code>

## TextField, classe

Les champs texte sont des éléments visuels pouvant être placés sur la scène pour présenter du texte à l'utilisateur. Tout comme les champs de saisie ou les zones de texte de formulaires du code HTML, Flash permet de définir des champs texte modifiables, en lecture seule, ainsi que d'autoriser la mise en forme HTML et les champs de texte multiligne.

La classe TextField permet de créer des champs texte. Tous les champs texte de saisie et dynamique dans un fichier SWF sont des occurrences de la classe TextField. Vous pouvez donner un nom d'occurrence à un champ texte dans l'inspecteur Propriétés, puis utiliser les méthodes et les propriétés de la classe TextField pour la modifier avec ActionScript. Les noms d'occurrence de TextField s'affichent dans l'explorateur d'animations et dans la boîte de dialogue Insérer un chemin cible du panneau Actions.

Pour créer un champ texte de façon dynamique, n'utilisez pas l'opérateur `NEW`, mais `MovieClip.createTextField()`.

Les méthodes de la classe TextField permettent de définir, sélectionner et manipuler du texte dans un champ texte dynamique ou de saisie que vous créez en cours de programmation ou à l'exécution.

Toutes les propriétés de la classe TextField sont prises en charge par Flash Lite 2.x et versions ultérieures, mais les champs texte ne peuvent afficher que les polices de périphérique. Les polices de périphérique sont des polices propres à Flash qui ne sont pas intégrées dans un fichier SWF. Flash Lite utilise la police du périphérique mobile qui s'approche le plus de sa police de périphérique. Dans la mesure où les vecteurs de polices ne sont pas intégrés, le fichier SWF prend moins d'espace que des polices vectorielles intégrées. Toutefois, du fait que les polices vectorielles ne sont pas intégrées, le texte que vous créez avec ces polices n'a pas l'aspect attendu sur les périphériques ne disposant pas de la police de périphérique. Flash intègre trois polices de périphérique : `_sans` (similaire à Helvetica et Arial), `_serif` (similaire à Times Roman) et `_typewriter` (similaire à Courier).

Pour plus d'informations sur la classe TextField, consultez les références suivantes :

- Chapitre 12, « Utilisation du texte et des chaînes », dans *Formation à ActionScript 2.0 dans Adobe Flash*

- Chapitre 2, « Classes ActionScript », dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*

## Résumé des propriétés

Le tableau ci-dessous dresse la liste des propriétés de la classe TextField qui ne sont pas prises en charge lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Propriété	Description
antialiasType	Indique le type d'anti-aliasing appliqué à cette occurrence TextField.
menu	Associe un objet ContextMenu à un champ texte.
mouseWheelEnabled	Valeur booléenne qui indique si Flash Player doit automatiquement faire défiler des champs texte multilignes lorsque le pointeur de la souris clique sur un champ texte et que l'utilisateur actionne la molette de la souris.
restrict	Indique le jeu de caractères qu'un utilisateur peut entrer dans le champ texte.
sharpness	Netteté des bords du glyphe dans cette occurrence TextField.
styleSheet	Associe une feuille de style au champ texte.
thickness	Indique l'épaisseur des bords du glyphe dans cette occurrence TextField.

## Résumé de la méthode

Le tableau ci-dessous dresse la liste des méthodes de la classe TextField qui ne sont pas prises en charge lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Méthode	Description
getFontList	Renvoie les noms des polices sur le système hôte du lecteur sous forme de tableau.

## TextFormat, classe

La classe TextFormat regroupe les informations de mise en forme de caractères. La classe TextFormat permet de personnaliser la mise en forme des champs texte. Vous pouvez formater le texte des champs statiques et dynamiques. Certaines propriétés de la classe TextFormat ne sont pas disponibles pour les polices incorporées et de périphérique.

La classe TextFormat permet de mettre en forme un champ texte ou certains caractères de ce champ texte. Parmi les options de mise en forme du texte disponibles, citons l'alignement, l'indentation, les caractères en gras et/ou couleur, la taille de la police, les largeurs des marges, les italiques et l'espacement des lettres. Vous pouvez formater le texte des champs statiques et dynamiques. Certaines propriétés de la classe TextFormat ne sont pas disponibles pour les polices incorporées et de périphérique.

Vous devez utiliser le constructeur `TextFormat()` pour créer un objet TextFormat avant d'appeler ses méthodes.

**Remarque :** *Flash Lite 2.x et versions ultérieures offrent une prise en charge partielle des fonctionnalités de mise en forme disponibles dans la classe TextFormat. Les fonctionnalités de mise en forme ne sont pas disponibles lorsque vous utilisez des polices de périphérique.*

Flash Lite 2.x et versions ultérieures prennent en charge de façon partielle la classe TextFormat. Par exemple, `TextFormat.font`, `TextFormat.bold` et `TextFormat.tabstop` ne sont pas pris en charge lorsque vous utilisez des polices de périphérique.

Pour plus d'informations sur la classe TextFormat, consultez les références suivantes :

- Chapitre 12, « Utilisation du texte et des chaînes », dans *Formation à ActionScript 2.0 dans Adobe Flash*



- Chapitre 2, « Classes ActionScript », dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*

## Résumé des propriétés

Le tableau ci-dessous dresse la liste des propriétés qui sont prises en charge de façon partielle lors de l'utilisation d'ActionScript 2.0 pour créer du contenu Flash.

Propriété	Description
<code>bold</code>	Valeur booléenne indiquant si le texte est en gras.  <b>Limitations</b> : Cette valeur est limitée à la mise en forme au niveau du paragraphe ; vous ne pouvez pas mettre en gras des caractères spécifiques.
<code>bullet</code>	Valeur booléenne indiquant que le texte fait partie d'une liste à puces.  <b>Limitations</b> : réservée aux polices intégrées.
<code>color</code>	Indique la couleur du texte.  <b>Limitations</b> : à utiliser uniquement avec la mise en forme au niveau du paragraphe ; vous ne pouvez pas mettre en couleur des caractères individuels.
<code>font</code>	Nom de la police du texte dans ce format de texte, sous forme de chaîne.  <b>Limitations</b> : pour Flash Lite, cette propriété ne s'applique qu'aux polices intégrées. Cette propriété n'est pas prise en charge pour l'arabe, l'hébreu et le thaïlandais.
<code>italic</code>	Valeur booléenne indiquant si le texte dans ce format est en italique.  <b>Limitations</b> : cette valeur est limitée à la mise en forme au niveau du paragraphe ; vous ne pouvez pas mettre en italique des caractères individuels.
<code>size</code>	Taille en points du texte dans ce format de texte.  <b>Limitations</b> : cette valeur est limitée à la mise en forme au niveau du paragraphe ; vous ne pouvez pas appliquer de tailles de police différentes à des caractères individuels.
<code>tabStops</code>	Spécifie des taquets de tabulation personnalisés, sous forme d'un tableau d'entiers non négatifs.  <b>Limitations</b> : réservée aux polices intégrées.
<code>underline</code>	Valeur booléenne indiquant si le texte qui utilise ce format texte est souligné ( <code>true</code> ) ou non ( <code>false</code> ).  <b>Limitations</b> : s'applique uniquement à la mise en forme au niveau du paragraphe.

## Video, classe

Flash Lite 2.x permet de manipuler des formats vidéo propres au périphérique et prend en charge les types suivants de lecture vidéo :

- Vidéo intégrée dans un fichier SWF ;
- Vidéo disponible sous forme de fichier distinct sur le périphérique ;
- Vidéo diffusée en continu sur le réseau (en temps réel).

Flash Lite 2.x prend en charge les vidéos de périphérique. La vidéo de périphérique est stockée dans le fichier SWF publié au format vidéo natif du périphérique. Pour lire la vidéo de périphérique, Flash Lite transmet les données vidéo au périphérique, qui les décode et les lit.

**Remarque** : Le code ActionScript de Flash Lite 2.x ne prend pas en charge les classes *NetConnection* ou *NetStream*.

Flash Lite 3.0 ajoute la prise en charge de Flash Video (FLV) à l'aide des versions des codecs On2 et Sorenson optimisés pour les périphériques portables. Le rendu de la vidéo FLV s'effectue directement dans le lecteur Flash Lite et non pas sur le périphérique. Ainsi, vous n'avez plus à vous demander si les périphériques cibles prennent en charge un format vidéo particulier. Les classes `NetConnection` et `NetStream` d'ActionScript, qui n'étaient auparavant pas disponibles dans Flash Lite, vous permettent de contrôler la lecture de vidéos FLV à partir d'un disque local ou d'une adresse HTTP. Ces classes sont décrites dans le Chapitre 2, « Classes ActionScript », dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*. La diffusion en flux continu de données vidéo sur une connexion RTMP vers Flash Media Server est également prise en charge par Flash Lite 3.0 (mais les connexions RTMPT et RTMPS ne le sont pas). La classe `Camera` et l'enregistrement de vidéos ne sont pas pris en charge pour Flash Video dans Flash Lite.

Flash Lite 3.0 inclut également une nouvelle propriété dans la classe `Video`, `attachVideo`, qui spécifie l'affichage d'un flux vidéo dans le cadre de l'objet `Video` sur la scène. Vous utilisez les méthodes de la classe `NetStream` plutôt que celles de la classe `Video` pour contrôler la lecture des vidéos FLV (non-périphériques). Par exemple, pour mettre la vidéo du périphérique en pause, vous utilisez la méthode `Video.pause`, mais pour mettre la vidéo FLV en pause, vous utilisez `NetStream.pause`.

Pour plus d'informations sur la classe `Video`, consultez les références suivantes :

- Chapitre 15, « Utilisation des images, du son et de la vidéo », dans *Formation à ActionScript 2.0 dans Adobe Flash*
- Chapitre 2, « Classes ActionScript », dans le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*

## Résumé de la méthode

Le tableau ci-dessous répertorie les méthodes qui ne sont pas prises en charge par la classe `Video` lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Méthode	Description	Prise en charge
<code>clear</code>	Efface l'image actuellement affichée dans l'objet <code>Video</code> . Cette fonction est par exemple utile lorsque vous souhaitez afficher des informations en attente sans devoir cacher l'objet <code>Video</code> .	Pas de prise en charge

## Résumé des propriétés

Le tableau ci-dessous dresse la liste des propriétés de la classe `Video` qui ne sont pas prises en charge lorsque vous utilisez le code ActionScript pour Flash Lite 2.x et versions ultérieures.

Propriété	Description
<code>deblocking</code>	Indique le type de filtre de dégroupage appliqué aux vidéos décodées dans le cadre du post-traitement. Deux filtres de dégroupage sont disponibles : l'un dans le codec Sorenson, l'autre dans le codec On2 VP6.
<code>height</code>	Nombre entier spécifiant la hauteur en pixels du flux vidéo.
<code>smoothing</code>	Indique si la vidéo doit être lissée (interpolée) lors de son redimensionnement. Pour faciliter le lissage, le lecteur doit être en mode haute qualité.
<code>width</code>	Nombre entier spécifiant la largeur en pixels du flux vidéo.

## Extensions des méthodes Flash Lite

La classe `Video` de Flash Lite 2.x et versions ultérieures ajoute les nouvelles méthodes suivantes.

Méthode	Description
<code>play()</code>	Ouvre une source vidéo et amorce sa lecture.
<code>close()</code>	Arrête la lecture de la vidéo, libère la mémoire associée à cet objet Video et efface la zone réservée à la diffusion de l'écran.
<code>stop()</code>	Arrête la lecture de la vidéo et poursuit le rendu de l'image affichée. L'appel suivant à <code>Video.resume()</code> reprend la lecture à partir de la première image de la vidéo.
<code>pause()</code>	Arrête la lecture de la vidéo et poursuit le rendu de l'image affichée. L'appel suivant à <code>Video.resume()</code> reprend la lecture à partir de la position actuelle.
<code>resume()</code>	Reprend la lecture de la vidéo.

La classe Video de Flash Lite 3.0 a ajouté les nouvelles méthodes suivantes.

Méthode	Description
<code>attachVideo()</code>	Spécifie un flux vidéo (source) à afficher dans le cadre de l'objet Video figurant sur la scène.

## Éléments ActionScript non pris en charge ou pris en charge de façon partielle: Détails

Cette section décrit les fonctions et propriétés globales, les constantes, les opérateurs, les instructions, les extensions et les mots-clés qui ne sont pas pris en charge, ou seulement de façon partielle, par le code ActionScript pour les logiciels Macromedia Flash Lite 2.x ou d'Adobe, ou par Adobe Flash Lite 3.x.

### Commandes émises par l'intermédiaire de `fsCommand` et `fsCommand2`

Les fonctions globales `fsCommand()` et `fsCommand2()` permettent au fichier SWF de communiquer avec Flash Player ou le programme hébergeant Flash Player, tel qu'un navigateur Web.

Flash Lite 2.x et versions ultérieures modifient les commandes Macromedia Flash Player 7 standard et ajoutent des commandes qui sont propres aux périphériques incorporés. Flash Lite 2.x et version ultérieures prennent également en charge la fonction `fsCommand2()` qui est similaire à la fonction `fsCommand()`, à la différence que la commande s'exécute immédiatement et ne nécessite pas le traitement préalable de l'image ayant effectué l'appel.

Pour plus d'informations sur la fonction `fsCommand2()`, consultez le *Guide de référence du langage ActionScript Flash Lite 2.x et 3.x*.

### Commandes non prises en charge

Le tableau ci-dessous affiche la liste des commandes qui ne sont pas prises en charge par `fsCommand()` lors de l'utilisation d'ActionScript 2.0 pour créer du contenu Flash Lite.

Commande	Description
quit	Ferme la projection.
fullscreen	Spécifiez <code>true</code> pour exécuter Flash Player en mode plein écran. Spécifiez <code>false</code> pour rétablir le mode d'affichage normal du menu.
allowscale	Spécifiez <code>false</code> pour définir le lecteur afin que le fichier SWF soit toujours affiché dans sa taille d'origine et que son échelle ne soit jamais modifiée. Spécifiez <code>true</code> pour obliger le fichier SWF à utiliser 100 % de l'écran.
showmenu	Spécifiez <code>true</code> pour activer le jeu complet des éléments de menu contextuel. Spécifiez <code>false</code> pour masquer tous les éléments du menu contextuel, à l'exception de A propos de Flash Player et de Paramètres.
exec	Exécute une application depuis la projection.
trapallkeys	Spécifiez <code>true</code> pour envoyer tous les événements de touche, y compris les touches de raccourci, au gestionnaire <code>onClipEvent (keyDown/keyUp)</code> de Flash Player.

## Résumé des commandes

Le tableau ci-dessous affiche la liste des commandes, fonctions et mots-clés ActionScript qui ne sont pas pris en charge ou qui le sont de façon partielle par Flash Lite 2.x et version ultérieures.

Commande, fonction ou mot-clé	Description	Prise en charge
asfunction	Protocole pour les URL dans les champs texte HTML.	Pas de prise en charge
fscommand()	Fonction qui permet au fichier SWF de communiquer avec Flash Player ou le programme hébergeant Flash Player, tel qu'un navigateur Web.	Prise en charge partielle
on	Préfixe des événements à exécuter lorsque l'événement se produit.  Limitation : les événements pris en charge sont <code>rollout</code> , <code>rollover</code> et <code>keyPress</code> .	Prise en charge partielle
onClipEvent	Gestionnaire d'événements ; déclenche les actions définies pour une instance spécifique d'un clip.  <b>Limitations :</b> les événements pris en charge sont <code>press</code> , <code>load</code> , <code>unload</code> , <code>enterFrame</code> , <code>keyDown</code> , <code>keyUp</code> et <code>data</code> .  Les événements <code>mouseDown</code> , <code>mouseUp</code> et <code>mouseMove</code> sont pris en charge si <code>System.capabilities.hasMouse</code> ou <code>System.capabilities.hasStylus</code> est défini sur <code>true</code> .	Prise en charge partielle
onUpdate	Le gestionnaire d'événements <code>onUpdate</code> est défini pour obtenir un aperçu en direct à l'aide d'un composant.	Pas de prise en charge
print()	La fonction <code>print()</code> cible le clip en fonction des limites spécifiées dans le paramètre.	Pas de prise en charge
printAsBitmap()	Fonction qui imprime le clip cible en tant que bitmap.	Pas de prise en charge
printAsBitmapNum()	Fonction qui imprime un niveau de Flash Player en tant que bitmap en fonction des limites spécifiées dans le paramètre ( <code>bmovie</code> , <code>bmax</code> ou <code>bframe</code> ).	Pas de prise en charge

Commande, fonction ou mot-clé	Description	Prise en charge
<code>printNum()</code>	Fonction qui imprime le niveau de Flash Player en fonction des limites spécifiées dans le paramètre <code>boundingBox</code> ( <code>bmovie</code> , <code>bmax</code> ou <code>bframe</code> ).	Pas de prise en charge
<code>startDrag()</code>	Fonction qui rend le clip cible déplaçable pendant la lecture du fichier SWF. Vous ne pouvez déplacer qu'un seul clip à la fois. <b>Limitation</b> : Prise en charge si l'interface de la souris ou du stylet est prise en charge.	Prise en charge partielle
<code>stopDrag()</code>	Fonction qui arrête l'opération de déplacement en cours. <b>Limitation</b> : Prise en charge si l'interface de la souris ou du stylet est prise en charge.	Prise en charge partielle
<code>updateAfterEvent()</code>	Fonction qui met à jour l'affichage (indépendamment du nombre d'images par seconde défini pour le fichier SWF) lorsque vous l'appellez dans un gestionnaire <code>onClipEvent()</code> ou dans le cadre d'une fonction ou d'une méthode que vous transmettez à <code>setInterval()</code> .	Pas de prise en charge

## Propriétés globales

Le tableau ci-dessous répertorie les propriétés globales d'ActionScript qui sont prises en charge de façon partielle ou ne sont pas prises en charge par Flash Lite 2.x et versions ultérieures.

Propriétés	Description	Prise en charge
<code>_droptarget</code>	Propriété en lecture seule qui renvoie le chemin absolu en appliquant la syntaxe à barre oblique (/) de l'occurrence de clip sur laquelle <code>draggableInstanceName</code> (le nom d'une occurrence de clip qui était la cible de la fonction <code>startDrag()</code> ) a été déposé. Cette propriété renvoie toujours un chemin qui commence par une barre oblique (/).	Pas de prise en charge
<code>_highquality</code>	Propriété globale qui spécifie le niveau d'anti-aliasing appliqué au fichier SWF actuel. Cette propriété permet également de contrôler le lissage de bitmaps. <b>Limitation</b> : Flash Lite ne prend pas en charge le lissage de bitmaps.	Prise en charge partielle
<code>_url</code>	Propriété en lecture seule qui extrait l'URL du fichier SWF à partir duquel le clip a été téléchargé.	Pas de prise en charge

# Chapitre 11 : Messages d'erreur et d'avertissement

Cette annexe présente les messages d'avertissement ou d'informations que le périphérique et l'émulateur Adobe Device Central peuvent produire lors du test d'une application Flash Lite.

## Codes d'erreur de périphérique

Pendant le test du contenu, l'émulateur d'Adobe Device Central peut générer des alertes. Le premier type d'alerte s'affiche uniquement dans l'émulateur et fournit des informations sur les erreurs réelles ou potentielles ; le deuxième s'affiche à la fois dans l'émulateur et sur le périphérique réel.

Le premier type de message d'alerte fournit des informations de débogage concernant votre fichier SWF. Par exemple, si le fichier SWF contient du code Adobe ActionScript non pris en charge par Flash Lite (ou par la version de Flash Lite disponible sur le périphérique de test sélectionné), l'émulateur déclenche une alerte.

L'autre type de message peut survenir aussi bien dans l'émulateur que sur le périphérique réel. Ces types d'erreurs s'affichent dans une boîte de dialogue d'erreur que l'utilisateur doit fermer pour permettre à l'application de poursuivre. L'illustration suivante présente un exemple de boîte de dialogue d'erreur, telle qu'elle est affichée dans l'émulateur :



Sur le périphérique, la boîte de dialogue d'erreur qui s'affiche contient la chaîne « Problème de contenu » suivie d'un numéro d'erreur. Dans l'émulateur, la boîte de dialogue d'erreur contient également une courte chaîne d'erreur. Enfin, l'émulateur affiche une description plus détaillée de l'erreur dans le panneau Sortie.

Le tableau suivant répertorie toutes les erreurs survenant dans le lecteur Flash Lite, y compris les numéros d'erreur, les descriptions brèves qui s'affichent dans la boîte de dialogue d'erreur et les descriptions détaillées qui s'affichent dans le panneau Sortie.

Numéro de l'erreur	Chaîne d'erreurs	Description et causes possibles
1	Mémoire insuffisante	L'émulateur ne dispose pas de suffisamment de mémoire heap. Sauf mention contraire, l'émulateur alloue au fichier SWF 1 Mo de mémoire à utiliser.
2	Limite d'empilement atteinte	L'émulateur a détecté que sa limite d'empilement a été atteinte ou dépassée. Les causes peuvent être diverses : plusieurs niveaux de clips imbriqués ou dessins vectoriels complexes.
3	Fichier SWF corrompu	L'émulateur a détecté que les données du fichier SWF sont corrompues.
4	Blocage du code ActionScript	L'émulateur a détecté que l'exécution d'une portion de code ActionScript figurant dans le fichier SWF prenait trop de temps. Par conséquent, l'émulateur a arrêté l'exécution de ce code ActionScript.
5	S/O	S/O
6	Données d'image incorrectes	Le fichier SWF contient une image que Flash Lite ou le décodeur d'image natif de la plate-forme n'a pas pu décoder.
7	Données audio incorrectes	Le fichier SWF a tenté de charger un son à un format non pris en charge ou les données audio sont corrompues.
8	Clip racine purgé	Cette erreur se produit lorsque le fichier SWF racine (niveau 0) est remplacé par un autre fichier SWF.
9	S/O	S/O
10	Chaîne getURL trop longue	La chaîne URL de l'appel getURL est trop longue.
11	Mémoire insuffisante pour décompresser l'image.	Flash Lite ne dispose pas de suffisamment de mémoire pour décoder une image dans le fichier SWF.
12	Mauvaises données SVG	Flash Lite tente de charger les données SVG qui sont corrompues.
13	Mémoire insuffisante pour le chargement du flux de diffusion	Flash Lite ne dispose pas de suffisamment de mémoire pour traiter les données chargées en flux continu à partir d'une URL donnée. Par exemple, cette erreur peut se produire si vous tentez de charger sur le réseau un fichier XML qui est trop volumineux pour être traité par Flash Lite en une seule opération. Si possible, tentez de fractionner le fichier de données en plusieurs fichiers, puis chargez ces fichiers de façon séquentielle.

## Messages d'informations et d'erreur de l'émulateur Adobe Device Central

Ces messages s'affichent dans une fenêtre contextuelle Sortie dans l'émulateur tandis que Adobe Flash les signale dans son panneau Sortie. Le tableau suivant répertorie tous les messages à contenu informatif de l'émulateur Adobe Device Central :



Code d'erreur	Message	Description
FTPA002	FSCOMMAND est ignoré.	L'émulateur a détecté un appel de la fonction <code>fscommand()</code> , qui n'est pas prise en charge par le périphérique de test sélectionné. Aucune modification n'est apportée au fichier SWF spécifique du périphérique ; il ne s'agit que d'un avertissement.
FTPA003	loadVariables est ignoré.	L'émulateur a détecté un appel de la fonction <code>loadVariables()</code> , qui n'est pas prise en charge par le périphérique de test et le type de contenu sélectionnés. Aucune modification n'est apportée au fichier SWF spécifique du périphérique ; il ne s'agit que d'un avertissement.
FTPA004	loadMovie est ignoré.	L'émulateur a détecté un appel de la fonction <code>loadMovie()</code> , qui n'est pas prise en charge par le périphérique de test et le type de contenu sélectionnés. Aucune modification n'est apportée au fichier SWF spécifique du périphérique ; il ne s'agit que d'un avertissement.
FTPA005	L'appel à GetURL pour URL a été ignoré parce qu'il y a eu plusieurs requêtes par pression de touche.	Flash Lite n'autorise qu'un seul appel de la fonction <code>getURL()</code> par pression de touche ; l'émulateur a détecté qu'il y a eu plusieurs appels <code>getURL()</code> , par conséquent seule la première commande est traitée et les autres sont ignorées.
FTPA006	L'appel à GetURL pour URL a été ignoré dans la mesure où il n'a pas été associé à une pression de touche.	Le périphérique de test et le type de contenu actuellement sélectionnés ne traitent que les appels de la fonction <code>getURL()</code> résultant de la pression d'une touche de périphérique par l'utilisateur. L'émulateur a détecté que votre application a effectué un appel à la fonction <code>getURL()</code> qui n'était pas associé à une pression de touche.
FTPA007	getProperty ou setProperty non pris en charge pour : <i>propriété MovieClip</i> .	Flash Lite ne prend pas en charge la propriété de clip spécifiée.
FTPA008	getProperty ou setProperty pris partiellement en charge pour : <i>propriété MovieClip</i> .	Flash Lite ne prend pas totalement en charge la propriété de clip spécifiée. Pour plus d'informations, consultez l'entrée correspondant à la propriété spécifiée dans le <i>Guide de référence du langage ActionScript Flash Lite 2.x et 3.0</i> .
FTPA009	startDrag et stopDrag ne sont pas pris en charge.	L'émulateur a détecté un appel de la fonction <code>startDrag()</code> ou <code>stopDrag()</code> , qui n'est pas prise en charge par Flash Lite.
FTPA014	getURL est ignoré.	L'émulateur a détecté un appel de la fonction <code>getURL()</code> , qui n'est pas prise en charge par le périphérique de test et le type de contenu sélectionnés. Aucune modification n'est apportée au fichier SWF spécifique du périphérique ; il ne s'agit que d'un avertissement.

Code d'erreur	Message	Description
FTPA015	L'appel à <code>loadMovie</code> pour <i>URL</i> a été ignoré parce qu'il y a eu plusieurs requêtes par pression de touche.	Flash Lite n'autorise qu'un seul appel de la fonction <code>loadMovie()</code> par pression de touche ; l'émulateur a détecté qu'il y a eu plusieurs appels <code>loadMovie()</code> , par conséquent seule la première commande est traitée et les autres sont ignorées.
FTPA016	L'appel à <code>loadMovie</code> pour <i>URL</i> a été ignoré dans la mesure où il n'a pas été associé à une pression de touche.	Le périphérique de test et le type de contenu actuellement sélectionnés ne traitent que les appels de la fonction <code>loadMovie()</code> résultant de la pression d'une touche sur le périphérique par l'utilisateur. L'émulateur a détecté que votre application a effectué un appel de la fonction <code>loadMovie()</code> qui n'était pas associé à une pression de touche.
FTPA017	L'appel à <code>loadVariables</code> pour <i>URL</i> a été ignoré parce qu'il y a eu plusieurs requêtes par pression de touche.	Votre application a effectué plusieurs appels de la fonction <code>loadVariables()</code> au cours d'un seul événement de pression de touche. Flash Lite n'autorise qu'une seule commande <code>loadVariables()</code> par pression de touche ; par conséquent, seule la première commande est traitée et les autres sont ignorées.
FTPA018	L'appel à <code>loadVariables</code> pour <i>URL</i> a été ignoré dans la mesure où il n'a pas été associé à une pression de touche.	Le périphérique de test et le type de contenu actuellement sélectionnés ne traitent que les appels de la fonction <code>loadVariables()</code> résultant de la pression d'une touche sur le périphérique par l'utilisateur. L'émulateur a détecté que votre application a effectué un appel de la fonction <code>loadVariables()</code> qui n'était pas associé à une pression de touche.
FTPA019	L'appel à <code>FSCommand</code> avec les arguments <i>arguments-commande</i> a été ignoré parce qu'il y a eu plusieurs requêtes par pression de touche.	Flash Lite n'autorise qu'un seul appel de la fonction <code>fscommand()</code> par pression de touche ; l'émulateur a détecté qu'il y a eu plusieurs appels <code>fscommand()</code> , par conséquent seule la première commande est traitée et les autres sont ignorées.
FTPA020	L'appel à <code>FSCommand</code> avec les arguments <i>arguments-commande</i> a été ignoré parce qu'il n'était pas associé à une pression de touche.	Le périphérique de test et le type de contenu actuellement sélectionnés ne traitent que les appels de la fonction <code>fscommand()</code> résultant de la pression d'une touche sur le périphérique par l'utilisateur. L'émulateur a détecté que votre application a effectué un appel de la fonction <code>fscommand()</code> qui n'était pas associé à une pression de touche.
FTPE001	La touche ne sera pas traitée : <i>nom_touche</i> Valeur ASCII : <i>valeur</i> .	L'émulateur a détecté que l'utilisateur a appuyé sur une touche du périphérique non prise en charge par Flash Lite, cette action est ignorée.
FTPE013	Les champs texte de saisie ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	Le périphérique de test et le type de contenu courants ne prennent pas en charge les champs texte de saisie.
FTPS010	Le son en diffusion continue n'est pas pris en charge.	Le périphérique de test et le type de contenu sélectionnés ne prennent pas en charge le son en diffusion continue.

Code d'erreur	Message	Description
FTPS011	Vous ne pouvez jouer qu'un seul son à la fois (pas de mixage).	L'émulateur a détecté la lecture simultanée de plusieurs sons dans le fichier SWF ; cette fonctionnalité n'est pas prise en charge dans Flash Lite.
FTPS012	L'événement audio a été ignoré parce qu'il n'est pas associé à une touche.	Dans Flash Lite 1.0, un son ne peut être lu qu'en réponse à la pression d'une touche par l'utilisateur sur son périphérique. (Cette restriction ne s'applique pas à Flash Lite version 2.0 et ultérieure.)
FTPS021	Son non pris en charge pour le type de contenu sélectionné sur ce périphérique.	Le périphérique de test et le type de contenu sélectionnés ne prennent pas en charge le son.
FTPS022	Les sons ADPCM ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence dans le fichier SWF d'un son natif (par opposition au son de périphérique) compressé au format ADPCM, qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique. Aucune modification n'est apportée au fichier SWF spécifique du périphérique ; il ne s'agit que d'un avertissement.
FTPS023	Les sons MP3 ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence dans le fichier SWF d'un son natif (par opposition au son de périphérique) compressé au format MP3, qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique. Aucune modification n'est apportée au fichier SWF spécifique du périphérique ; il ne s'agit que d'un avertissement.
FTPS024	Les sons MIDI ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence d'un son de périphérique MIDI, qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique.
FTPS025	Les sons PCM ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence d'un son Flash natif compressé au format PCM, qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique.
FTPS026	La fonction de débogage d'animation n'est pas prise en charge par le lecteur d'animation de test spécifié.	L'émulateur Adobe Device Central ne prend pas en charge la commande de menu Contrôle > Déboguer l'animation.
FTPS027	Package audio détecté.	L'émulateur a détecté la présence dans le fichier SWF d'un fichier de package audio.
FTPS028	Commande FSCommand2 <i>nom-commande</i> incorrecte détectée.	La commande <code>fscommand2()</code> spécifiée n'est pas une chaîne de commande valide. Pour obtenir la liste des commandes <code>fscommand2()</code> valides, consultez la section Fonction <code>fscommand2</code> dans le <i>Guide de référence du langage ActionScript Flash Lite 2.x et 3.0</i> .
FTPS029	Commande FSCommand2 <i>nom-commande</i> détectée.	L'émulateur a détecté la commande <code>fscommand2()</code> spécifiée.

Code d'erreur	Message	Description
FTPS030	Commande <code>FSCommand2</code> <i>nom-commande</i> non prise en charge dans l' ; veuillez la tester sur le périphérique.	L'émulateur ne prend pas en charge la commande <code>fscommand2()</code> spécifiée. Vous devez tester ce fichier SWF sur un périphérique où Flash Lite est installé pour vérifier si la commande spécifiée fonctionne comme vous le souhaitez.
FTPS031	Plusieurs instances d'appels de requête URL détectés, un seul est autorisé par pression de touche/image.	Flash Lite n'autorise qu'un seul appel de la fonction <code>getURL()</code> par pression de touche ou image ; l'émulateur a détecté plusieurs appels <code>getURL()</code> , par conséquent seule la première commande est traitée et les autres sont ignorées.
FTPS032	Un appel à <code>GetURL(URL)</code> a été détecté ; ce dernier peut être soumis à des restrictions.	L'émulateur a détecté un appel de la fonction <code>getURL()</code> susceptible de présenter certaines restrictions d'exécution lors de sa lecture sur le périphérique sélectionné. Testez votre fichier SWF sur un périphérique réel pour vérifier si la commande fonctionne comme vous le souhaitez.
FTPS033	Un appel à <code>loadVariables(URL)</code> a été détecté ; ce dernier peut être soumis à des limitations.	L'émulateur a détecté un appel de la fonction <code>loadVariables()</code> susceptible de présenter certaines restrictions d'exécution lors de sa lecture sur le périphérique sélectionné. Testez votre fichier SWF sur un périphérique pour vérifier si la commande fonctionne comme vous le souhaitez.
FTPS034	Un appel à <code>FSCommand(nom-commande)</code> a été détecté ; ce dernier peut être soumis à des restrictions.	Il s'agit juste d'un avertissement indiquant que certains périphériques et types de contenus Flash Lite peuvent ne pas prendre en charge la commande <code>fscommand()</code> dans l'application. Testez votre fichier SWF sur un périphérique pour vérifier si la commande fonctionne comme vous le souhaitez.
FTPS035	Un appel à <code>loadMovie(URL)</code> a été détecté ; ce dernier peut être soumis à des restrictions.	L'émulateur a détecté un appel de la fonction <code>loadMovie()</code> susceptible de présenter certaines restrictions d'exécution lors de sa lecture sur le périphérique sélectionné. Testez votre fichier SWF sur un périphérique pour vérifier si la commande fonctionne comme vous le souhaitez.
FTPS036	<i>N</i> kilo-octets de <i>son-périphérique</i> détecté dans le regroupement audio.	Pour chaque son contenu dans un regroupement audio, l'émulateur en signale le type (par exemple, MIDI ou SMAF) et la taille.
FTPS037	Les sons SMAF ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence d'un son de périphérique SMAF, qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique.
FTPS038	L'appel à <code>StartVibrate</code> a été ignoré parce qu'il y a eu plusieurs requêtes par image ou événement.	Flash Lite n'autorise qu'un seul appel de la fonction <code>fscommand2("StartVibrate")</code> par pression de touche ou image ; l'émulateur a détecté qu'il y a eu plusieurs appels, par conséquent seule la première commande est traitée et les autres sont ignorées.

Code d'erreur	Message	Description
FTPS039	FSCommand2 SetInputTextType ( <i>arguments-commande</i> ) détectée, non prise en charge dans l'émulateur ; veuillez la tester sur le périphérique.	La commande <code>SetInputTextType</code> n'est pas prise en charge dans l'émulateur. Vous devez la tester sur un périphérique réel.
FTPS048	Navigation quadridirectionnelle non prise en charge pour ce périphérique.	Le périphérique de test et le type de contenu actuellement sélectionnés prennent en charge la navigation bidirectionnelle. Vous avez appuyé sur les flèches de direction Gauche ou Droite du clavier à cinq directions de l'émulateur, qui ne sont pas prises en charge dans la navigation bidirectionnelle. Pour plus d'informations, consultez la section « <a href="#">Modes de navigation par défaut</a> » à la page 37.
FTPS049	Navigation quadridirectionnelle en boucle non prise en charge pour ce périphérique.	Le périphérique de test et le type de contenu actuellement sélectionnés prennent en charge la navigation quadridirectionnelle. Vous avez appuyé sur l'une des flèches de direction du périphérique alors qu'il n'y avait aucun objet sur la scène pour recevoir le focus, en fonction du sens de la flèche sur laquelle vous avez appuyé. Pour plus d'informations, consultez la section « <a href="#">Modes de navigation par défaut</a> » à la page 37.
FTPS050	Les sons MFI génériques ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence d'un son de périphérique MFI générique, qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique.
FTPS051	Événement de souris ( <i>nom-événement</i> ) non pris en charge détecté.	L'événement de souris spécifié n'est pas pris en charge par le périphérique de test et le type de contenu sélectionnés.
FTPS067	Les sons SMAF (MA-2) ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence d'un son de périphérique SMAF (MA-2) qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique.
FTPS068	Les sons SMAF (MA-3) ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence d'un son de périphérique SMAF (MA-3) qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique.
FTPS069	Les sons SMAF (MA-5) ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence d'un son de périphérique SMAF (MA-5) qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique.
FTPS070	Les sons MFI avec extension Fujitsu ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence d'un son de périphérique MFI doté d'une extension Fujitsu, qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique.
FTPS071	Les sons MFI avec extension Mitsubishi ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence d'un son de périphérique MFI doté d'une extension Mitsubishi, qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique.

Code d'erreur	Message	Description
FTPS072	Les sons MFI avec extension NEC ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence d'un son de périphérique MFI doté d'une extension NEC, qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique.
FTPS073	Les sons MFI avec extension Panasonic ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence d'un son de périphérique MFI doté d'une extension Panasonic, qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique.
FTPS074	Les sons MFI avec extension Sharp ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence d'un son de périphérique MFI doté d'une extension Sharp, qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique.
FTPS075	Les sons MFI avec extension Sony ne sont pas pris en charge pour le type de contenu sélectionné sur ce périphérique.	L'émulateur a détecté la présence d'un son de périphérique MFI doté d'une extension Sony, qui n'est pas pris en charge par le type de contenu sélectionné sur ce périphérique.
FTPS099	Les commandes d'impression ne sont pas prises en charge.	Votre application contient l'une des commandes d'impression ActionScript (par exemple, <code>print()</code> ou <code>printAsBitmap()</code> ), qui ne sont pas prises en charge par Flash Lite.
FTPS100	Le paramètre <i>Son-périphérique</i> est sélectionné dans le regroupement audio.	Indique le nom du son de périphérique que l'émulateur a choisi de lire dans un regroupement audio.
FTPS101	Aucun des formats du package audio n'est pris en charge sur ce périphérique.	Indique qu'aucun des sons de périphérique du regroupement audio n'est pris en charge par le type de contenu sélectionné sur ce périphérique.
FTPS102	Echec de la lecture du son SMAF.	L'émulateur n'a pas pu lire le son SMAF.
FTPS105	Ce fichier SWF n'est pas au format Flash Lite.	Votre application a tenté de charger un fichier SWF dont la version n'était pas au format Flash Lite ; Flash Lite peut charger uniquement des fichiers SWF Flash Lite ou des fichiers au format Flash 4.
FTPS106	L'événement de souris ( <i>nom-événement</i> ) a été ignoré dans la mesure où il n'a pas été déclenché par une pression de touche.	L'émulateur a détecté un événement de souris sur un bouton dans votre application Flash Lite. Le périphérique de test actuel ne prend pas en charge les interfaces de type stylet ou tactile, par conséquent, vous ne pouvez interagir avec les boutons de l'écran qu'en utilisant le clavier de l'émulateur ou les raccourcis clavier correspondants.
FTPS107	La touche ne sera pas traitée : <i>touche-périphérique</i> . Utilisez la commande <code>FSCommand2 SetSoftKeys</code> pour activer cette touche.	Vous avez appuyé sur l'une des touches programmables du clavier de l'émulateur sans avoir appelé au préalable la commande <code>SetSoftKeys</code> . Pour plus d'informations, consultez la section « <a href="#">Utilisation des touches programmables</a> » à la page 50.

Code d'erreur	Message	Description
FTPS108	Commande FSCommand ( <i>nom-commande</i> ) incorrecte détectée.	La commande <code>fscommand()</code> spécifiée n'est pas une chaîne de commande valide.
FTPS109	Commande FSCommand ( <i>nom-commande</i> ) non prise en charge dans l'émulateur ; veuillez la tester sur le périphérique.	L'émulateur ne prend pas en charge la commande <code>fscommand()</code> spécifiée. Testez ce fichier SWF sur un périphérique où Flash Lite est installé pour vérifier si la commande spécifiée fonctionne comme vous le souhaitez.
FTPS110	Le lecteur Flash Lite 1.0 ne prend pas en charge les touches programmables.	L'émulateur a détecté que vous avez appuyé sur l'une des touches programmables alors que le paramètre de publication de la version du fichier SWF de votre document est défini sur Flash Lite 1.0. Flash Lite 1.0 ne prend pas en charge les touches programmables.

# Index

## Symboles

%%% 104

## A

ActionScript

contrôle de vidéo de périphérique avec 93

contrôle de vidéo Flash avec 80

ADS, définition 95

animation interpolée 23

Application Download Server (*voir* ADS)

application Hello World 15

applications (BREW)

définition en tant que type de contenu 109

objectif du développement 101

assistant (BREW)

fenêtre Applet Icon 109

fenêtre Define Output Settings 109

fenêtre Identify Applet 108

fenêtre Include Applet Information 108

fenêtre Summary of Your Suggestions 110

## B

BDS, définition 95

Binary Runtime Environment for Wireless  
(*voir* BREW)

BREW

téléchargement des fichiers sur les  
périphériques 112

types de fichiers 103

BREW 2.x

chargement des fichiers sur les  
périphériques 114

structure des fichiers de périphérique 106

BREW 3.x

chargement des fichiers sur les  
périphériques 115

structure des fichiers de périphérique 106

BREW Delivery System (*voir* BDS)

BREW SDK

versions prises en charge 98

Button, classe

gestionnaires d'événements non pris en  
charge ou partiellement pris en  
charge 175

propriétés non prises en charge ou  
partiellement prises en charge 174

## C

Café Townsend, application

création d'éléments de navigation 25

création de l'animation Specials 23

calques de masque 23

champs texte

champs texte de saisie, utilisation 55

création de texte défilant 66

dynamique 25

limitation des caractères dans les champs  
texte de saisie 60

propriétés des paramètres 25

qualité de rendu 64

champs texte de saisie

exemple d'application 61

limitation des caractères dans 60

chargement des fichiers sur les  
périphériques BREW

BREW 2.x 114

BREW 3.x 115

Classe Selection 180

Commandes FS ActionScript (BREW) 97

configuration logicielle requise (BREW) 96

configuration matérielle requise (BREW) 96

configuration requise (BREW)

logicielle 96

matérielle 96

conseils d'utilisation de l'outil  
AppLoader 113

contenu Web, navigation 8

création d'éléments de navigation

création d'un bouton de détecteur de  
touches 25

utilisation des touches programmables 25

## D

Date, classe

méthodes non prises en charge ou  
partiellement prises en charge 175

diffusion en continu (BREW)

son 97

vidéo 97

données permanentes (BREW) 97

données permanentes (non BREW) 14

données, permanentes (BREW) 97

## E

écrans de veille (BREW)

définition en tant que type de contenu 109

objectif du développement 101

émulateur Adobe Device Central

définition 1, 117

options de débogage 118

versus simulateur BREW 101

émulateur Flash Lite

fonctionnalités non prises en charge 117

interaction 119

messages d'avertissement et d'erreur 194

émulateur versus simulateur (BREW) 101

événements de bouton 43

événements de pression de touche

codes de touche ActionScript 36

création d'un écouteur de touche 49

rédaction d'un script de gestionnaire de  
touches 36

touches prises en charge 35

traitement avec ActionScript 43

Extension, installation sur les périphériques  
(BREW) 113

## F

fenêtre Identify Applet (assistant) 108

Fenêtre Include Applet Information  
(assistant) 108

fenêtre Summary of Your Suggestions  
(assistant) 110

fichiers BAR, définis 106

fichiers DLL

conventions d'appellation 104

définition 103

fichiers MIF

conventions d'appellation 104

définition 103

fichiers MOD

conventions d'appellation 104

définition 103

fichiers partagés (BREW) 107

fichiers SIG

conventions d'appellation 104

définition 103

obtention 106

fichiers, partagés (BREW) 107



- Flash Lite
  - émulateur Adobe Device Central 15
  - présentation de la programmation 16
  - types de contenus 17
- Flash Lite 2.1 pour BREW
  - définition 95
- Flash Lite Publisher pour BREW
  - définition 95
- Flash Media Server 12
- Flash Player 4 et Flash Player 7
  - différences entre 125
- fsCommand() et fsCommand2() 191
- fscommand() et fscommand2()
  - commandes non prises en charge 191
  - commandes partiellement prises en charge 192
- G**
  - gestion du contenu 10
  - getURL() (BREW)
    - limites 101
    - présentation 98
- I**
  - identification des formats de fichier audio
    - pris en charge 75
  - inscription en tant que développeur BREW 99
  - intégration de polices vectorielles, à propos 65
  - interactivité
    - création à l'aide de boutons 43
    - création d'un menu à l'aide de boutons 46
    - détection des pressions de touche 49
    - gestion des événements de pression de touche 43
    - navigation par tabulation 37
- K**
  - Key, classe
    - nouvelles méthodes 176
    - nouvelles propriétés 176
  - kit de développement BREW
    - installation 99
- L**
  - limitations
    - vidéo de périphérique 82
- limites (BREW)
  - getURL() 101
  - loadVars() 101
  - XMLSend() 101
- M**
  - menus, création avec des boutons 46
  - MMS 98
  - modes d'application, dans Flash Lite 119
  - Mouse, classe
    - événements non pris en charge ou partiellement pris en charge 177
    - méthodes non prises en charge ou partiellement prises en charge 176
  - MovieClip, classe
    - gestionnaires d'événements pris en charge partiellement 179
    - méthodes non prises en charge ou partiellement prises en charge 178
- N**
  - National Software Testing Laboratories (*voir* NSTL)
  - navigation par tabulation
    - exemple d'application utilisant 46
    - modes de 37
    - présentation 37
    - quadririrectionnelle en boucle 38
    - recommandations pour 40
    - rectangle de focus 39
  - navigation, contenu Web 8
  - navigation. *Voir* navigation par tabulation
  - NetConnection
    - utilisation de vidéo Flash 79
  - NetStream
    - utilisation de vidéo Flash 79
  - NSTL, définition 95
- O**
  - objectif, périphériques visés par le développement (BREW) 101
  - outil AppLoader, conseils 113
- P**
  - pack de périphérique (BREW)
    - générique versus spécifique 102
    - générique vs. spécifique 111
    - téléchargement 102, 111
  - papier peint (BREW) 98
  - paramètres de publication (BREW)
    - liste des post-processeurs 108
  - périphériques (BREW)
    - objectif du développement 101
    - pris en charge 98
    - structure des fichiers pour BREW 2.x 106
    - structure des fichiers pour BREW 3.x 106
  - périphériques BREW, test des applications sur 116
  - périphériques cibles 17
  - périphériques de test 17
  - pilotes (USB), installation 100
  - pilotes USB, installation (BREW) 100
  - polices vectorielles, intégration dans les fichiers SWF 65
  - polices, méthodes de rendu appliquées aux champs texte 63
  - prise en charge (BREW)
    - périphériques 98
    - versions du SDK 98
  - processus de création de contenu 16
  - propriétés globales
    - partiellement prises en charge 193
- Q**
  - qualité du rendu, qualité par défaut 64
- S**
  - scène, taille à l'écran et disponibilité 122
  - SDK (BREW)
    - installation 99
    - versions prises en charge 98
  - sécurité
    - nouveau modèle de sécurité de Flash 3.0 9
    - sandboxes 9
  - Selection, classe
    - méthodes non prises en charge 180
  - SharedObject, classe
    - méthodes partiellement prises en charge 181
    - nouvelles méthodes 181
  - simulateur BREW
    - versus émulateur Adobe Device Central 101
  - simulateur versus émulateur (BREW) 101
  - SMS 98
  - son
    - externes 74
    - natifs 75
    - périphérique 68
    - regroupement 68

- son (BREW)
  - décodage 97
  - diffusion en continu (BREW) 97
- son de périphérique 75
  - \_forceframerate, propriété 74
  - synchronisation avec animation 74
- son natif
  - nouvel échantillonnage sur 8 kHz 76
  - présentation 75
- sonneries animées (BREW) 98
- Sound, classe
  - méthodes partiellement prises en charge 185
  - nouvelles méthodes 186
  - propriétés partiellement prises en charge 186
- Stage, classe
  - nouvelles propriétés 187
- structure des fichiers de périphérique
  - BREW 2.x 106
  - BREW 3.x 106
- Suite d'outils (BREW), installation 99
- Suite d'outils BREW, installation 99
- System, classe
  - événements partiellement pris en charge 182
  - méthodes non prises en charge 182
  - propriétés non prises en charge 182
- System.capabilities, classe
  - nouvelles propriétés 183
  - propriétés non prises en charge 182

**T**

- taille de scène disponible 15
- tailles, icônes 109
- TBT, définition 95
- téléchargement des fichiers sur les périphériques BREW
  - flux de travail 112
- test (BREW)
  - simulateur versus émulateur 101
  - sur les périphériques BREW 116
- texte défilant 66
- texte dynamique, champ 25
- texte en ligne (BREW) 97
- texte, en ligne (BREW) 97
- TextField, classe
  - méthodes non prises en charge 188
  - propriétés non prises en charge 188

- TextFormat, classe
  - propriétés partiellement prises en charge 189
- touches programmables 25
- True BREW Testing (*voir* TBT)
- types de contenus 17
- types de contenus (BREW)
  - applications 101
  - écrans de veille 101
- types de contenus dans Flash Lite, description 119
- types de fichiers
  - BREW 103
  - conventions d'appellation 104
  - icônes 109

## V

- vidéo
  - diffusion continue (BREW) 97
  - Flash (FLV) 79
  - périphérique 82
- vidéo de périphérique
  - contrôle avec ActionScript 93
  - émulateur 124
  - groupé 83
  - importation 83
  - présentation 82
- vidéo de périphérique, restrictions 82
- vidéo Flash
  - à propos de 79
  - contrôle avec ActionScript 80
  - incorporée 80
- vidéo Flash incorporée 80
- Video, classe
  - méthodes non prises en charge 190
  - nouvelles méthodes 190
  - propriétés non prises en charge 190

## X

- XML (BREW)
  - gestion des données 97
  - sockets 97
- XML.Send(), limites (BREW) 101
- XMLSocket 12