

Développement d'applications FLASH® LITE™ 4

Dernière mise à jour le 29/4/2010

© 2010 Adobe Systems Incorporated and its licensors. All rights reserved.

Développement d'applications Adobe® Flash® Lite® 4

This guide is protected under copyright law, furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

This guide is licensed for use under the terms of the Creative Commons Attribution Non-Commercial 3.0 License. This License allows users to copy, distribute, and transmit the user guide for noncommercial purposes only so long as (1) proper attribution to Adobe is given as the owner of the guide; and (2) any reuse or distribution of the user guide contains a notice that use of the guide is governed by these terms. The best way to provide notice is to include the following link. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>

Adobe, the Adobe logo, ActionScript, Flash, and Flash Lite 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. 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.

Updated Information/Additional Third Party Code Information available at <http://www.adobe.com/go/thirdparty>.

Portions include software under the following terms:

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

This software is based in part on the work of the Independent JPEG Group.

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

Video compression and decompression in Flash 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 : Vue d'ensemble de Flash Lite 4

Généralités sur Flash Lite	1
Fonctionnalités de Flash Lite 4	2

Chapitre 2 : Prise en main de Flash Lite

Application Flash Lite Hello World	5
Emulateur Adobe Device Central et base de données de périphériques	6
Flux de travail lors de la création d'applications Flash Lite	6
A propos des types de contenus Flash Lite	7

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

Affichage de l'application terminée	9
Création de l'application	9

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

Interaction de l'utilisateur et touches prises en charge	16
Navigation par défaut	17
Gestion des événements de souris et de touche	23
Prise en charge multipoint	29

Chapitre 5 : Utilisation du texte et des polices

Texte	31
Saisie de texte	32
Moteur de texte Flash	39
Rendu des polices	40
Défilement du texte	43

Chapitre 6 : Utilisation du son

Son de périphérique	45
Son natif	46

Chapitre 7 : Utilisation de la vidéo

Utilisation des fichiers FLV	48
RTMP	50
Améliorations	51

Chapitre 8 : Utilisation des API de périphérique

Classe Geolocation	55
Classe Accelerometer	56

Chapitre 9 : Présentation du code ActionScript pour Flash Lite 4

Éléments de langage et classes ActionScript non pris en charge, et pris en charge intégralement ou partiellement	59
Classes ActionScript partiellement prises en charge : Détails	151
Éléments ActionScript non pris en charge : Détails	165

Chapitre 1 : Vue d'ensemble de Flash Lite 4

Adobe® Flash Lite™ est une version d'Adobe® Flash® Player® conçue pour les périphériques.

Généralités sur Flash Lite

A propos de Flash Lite

Flash Lite 4 s'appuie sur Flash Player 10. Il offre des fonctionnalités visant à améliorer la navigation sur le Web pour les utilisateurs de périphériques mobiles. Nombre de ces fonctionnalités étaient présentes dans Flash® Player 9 mais pas prises en charge par Flash Lite. Elles sont désormais prises en charge par Flash Lite 4. Flash Lite 4 permet désormais de lire le contenu Adobe® ActionScript™ 3.0 qui figure sur les sites Web Flash 10. Il prend également en charge le traitement et la lecture du code ActionScript 3.0, introduit dans Flash Player 9. Pour plus d'informations, voir « [Fonctionnalités de Flash Lite 4](#) » à la page 2.

A propos des composants de Flash Lite

De par leur conception, les composants installés avec Adobe® Flash® Professional CS5 (DataGrid, par exemple) sont destinés aux 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 le contenu destiné aux périphériques mobiles, tenez compte des principes de base du développement de contenu. Les développeurs Flash, par exemple, doivent souvent éviter les graphismes complexes, les interpolations à outrance et une utilisation abusive de l'effet de transparence.

Les développeurs de Flash Lite sont souvent confrontés à des défis découlant des restrictions liées aux périphériques mobiles. Certains périphériques sont plus performants que d'autres, et ce parfois de façon significative. Il est nécessaire de publier le contenu mobile créé sur un grand nombre de périphériques différents. Les développeurs doivent donc parfois se baser sur les périphériques offrant le moins de fonctionnalités.

Pour optimiser le contenu destiné aux périphériques mobiles, il est souvent nécessaire de faire des compromis. Ainsi, telle technique permet d'obtenir un meilleur rendu, mais telle autre offre de meilleures performances. Pour trouver le meilleur compromis à vos yeux, vous serez amené à effectuer une série de tests à répétition dans l'émulateur et sur le périphérique cible.

Adobe Flash Professional CS5 comprend un émulateur Adobe Flash Lite, accessible sur Adobe® Device Central CS5. Cet émulateur permet de tester l'application dans l'outil de création, telle qu'elle se présente et fonctionne sur le périphérique. Lorsque l'application s'exécute dans l'émulateur vous convient, vous pouvez la tester sur un périphérique réel.

Fonctionnalités de Flash Lite 4

Amélioration de l'interactivité et de la navigation

Flash Lite 4 offre diverses fonctions visant à améliorer l'interactivité et la navigation pour les utilisateurs des périphériques mobiles. Certaines d'entre elles étaient prises en charge dans les versions antérieures de Flash Player, mais pas dans Flash Lite. Elles sont maintenant prises en charge dans Flash Lite 4.

Prise en charge multipoint

Flash Lite 4 offre la prise en charge multipoint pour gérer les événements multipoint et les événements de mouvement natifs. Tous les périphériques gérant cette fonction réagissent aux points tactiles multiples à l'écran. En outre, il est possible de traiter une série de données de points tactiles entrantes sous forme de mouvements de plus haut niveau.

Amélioration du texte et des polices

Moteur de texte de Flash Player 10

Flash Lite 4 prend en charge le nouveau moteur de mise en forme de texte de Flash Player 10. Ce moteur prend en charge les langues complexes et bidirectionnelles, les polices TrueType et des fonctions avancées telles que la rotation, le masquage, etc. Le nouveau moteur a été développé en vue d'optimiser la mémoire et les performances. Pour plus d'informations, voir « [Moteur de texte Flash](#) » à la page 39.

Saisie de texte en ligne

Dans Flash Lite 4, les événements d'arrière-plan continuent à s'exécuter lors d'une saisie de texte en ligne, ce qui était impossible dans les versions antérieures. En outre, dans cette version, le mode restreint n'est pas applicable et le code ActionScript est traité au cours de la saisie de texte en ligne. Pour plus d'informations, voir « [Saisie de texte en ligne](#) » à la page 33.

Améliorations vidéo

Canal de données RTMP

ActionScript 3.0 utilise le format AMF 3 pour la sérialisation et la désérialisation des objets ActionScript. Outre les occurrences d'objet, ce format envoie les caractéristiques d'objet et les chaînes. De plus, le format AMF 3 prend en charge certains des types de données introduits dans ActionScript 3.0. Pour plus d'informations, voir « [RTMP](#) » à la page 50.

RTMPE

Flash Lite 4 prend en charge les flux RTMP chiffrés. Cela signifie donc qu'Adobe® Flash® Media Server 3 doit lui envoyer une version chiffrée d'un flux de données multimédias. Flash Lite déchiffre ensuite ce flux sur le périphérique mobile. Pour plus d'informations, voir « [RTMPE](#) » à la page 52.

RTMPT et RTMPTE

Dans Flash Lite 4, RTMPT et RTMPTE sont activés via HTTP. Les transmissions de paquets RTMP sont activées pour les utilisateurs qui essaient de se connecter derrière un pare-feu bloquant toute tentative de transmission. Pour plus d'informations, voir « [RTMPT](#) » à la page 52 et « [RTMPTE](#) » à la page 53.

Diffusion en continu à vitesse de transmission variable

Lors de la diffusion en continu à partir du serveur Flash Media Server, Flash Lite 4 prend en charge le basculement entre différentes vitesses de transmission. L'application peut ainsi s'adapter aux modifications dynamiques des conditions réseau tout en diffusant en continu le contenu enregistré. Pour plus d'informations, voir « [Diffusion en continu à vitesse de transmission variable](#) » à la page 53.

Nouvelles API de périphérique

Flash Lite 4 propose deux nouvelles API de périphérique qui permettent d'obtenir la position géographique et l'accélération directionnelle d'un périphérique mobile.

Geolocation

La classe Geolocation vous permet de vous inscrire pour obtenir les mises à jour émises par le capteur de localisation embarqué d'un périphérique mobile. Le périphérique renvoie sa position géographique sous la forme de coordonnées latitudinales et longitudinales. Pour plus d'informations, voir « [Classe Geolocation](#) » à la page 55.

Accelerometer

Cette fonction permet d'obtenir les données d'accélération d'un périphérique à partir du capteur de l'accéléromètre installé sur celui-ci. Ces données représentent la position ou le mouvement du périphérique le long d'un axe tridimensionnel. Pour plus d'informations, voir « [Classe Accelerometer](#) » à la page 56.

Modifications affectant le comportement des objets partagés

L'implémentation des objets partagés dans Flash Lite 4 est entièrement compatible avec les fichiers SWF de Flash Player 10.

Flash Lite 4 prend en charge toutes les méthodes de la classe SharedObject d'ActionScript 3.0. Un fichier SWF s'exécute de la même façon dans Flash Lite 4 que dans Flash Player10, à la différence près que le périphérique ne dispose pas d'un Gestionnaire des paramètres. La taille de stockage d'un objet partagé doit donc être inférieure ou égale à 100 ko par domaine. Si elle augmente, les transactions associées à l'objet partagé échouent.

L'espace de stockage et la mémoire disponibles sur les périphériques mobiles sont généralement moins importants. En limitant la taille de stockage, vous réduisez l'espace disque et la mémoire supplémentaires nécessaires sur ces périphériques. Pour plus d'informations sur les objets partagés distants, voir www.help.adobe.com.

Remarque : vous devez créer tout nouveau contenu dans ActionScript 3.0 conformément à la documentation Flash Player 10.

Objets partagés distants

Flash Lite 4 permet de créer et d'utiliser des objets partagés distants. Ces objets sont partagés en temps réel par tous les clients connectés à l'application. Toute modification de la propriété d'un objet partagé distant par un client s'applique à tous les autres clients connectés.

Pour plus d'informations sur les objets partagés distants, voir www.help.adobe.com.

Extensions d'ActionScript

La fonction Extensions de Flash Lite 4 permet à des développeurs tiers d'ajouter des fonctionnalités et des API d'ActionScript à Flash Lite à l'aide d'une bibliothèque statique ou de liens dynamiques (DLL). La bibliothèque DLL assure l'implémentation des fonctionnalités et Flash Lite permet son utilisation en appelant des API d'ActionScript. Vous pouvez donc interagir avec les API qui sont exposées par les périphériques hôte. A cet effet, créez une bibliothèque DLL gérant à la fois les fonctionnalités du périphérique et celles de Flash Player. 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 Extensions d'ActionScript n'est disponible pour les développeurs tiers que si elle est activée par le fabricant du périphérique mobile.

Test et débogage

Adobe Device Central CS5 permet de tester et de déboguer des projets Flash Lite sur divers périphériques. Vous pouvez émuler l'affichage, l'utilisation de la mémoire et les performances de périphériques déterminés.

Classes partiellement prises en charge

Vous trouverez une liste détaillée des classes partiellement prises en charge par cette version à la section « [Présentation du code ActionScript pour Flash Lite 4](#) » à la page 59.

Classes non prises en charge

Vous trouverez une liste détaillée des classes non prises en charge par cette version à la section « [Présentation du code ActionScript pour Flash Lite 4](#) » à la page 59.

Chapitre 2 : Prise en main de Flash Lite

Application Flash Lite Hello World

Dans cette section, vous allez créer une application Flash Lite simple pour la tester dans l'émulateur Adobe Device Central CS5. Pour obtenir un exemple d'application plus détaillé, voir « [Didacticiel : Création d'une application Flash Lite](#) » à la page 9.

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 > Flash Lite 4.
- 3 Dans le panneau Outils, sélectionnez l'outil Texte et faites-le glisser sur la scène pour créer un champ de texte.
Tapez **Hello, world !** (ou tout autre texte) dans le champ de texte.
- 4 Dans l'onglet Propriétés, sélectionnez Texte classique dans le menu contextuel Moteur de texte et Texte statique dans le menu contextuel Type de texte.
- 5 Sélectionnez Contrôle > Tester l'animation dans Device Central pour exporter l'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.*

Configuration de paramètres importants pour les applications mobiles

Dans les applications Flash Lite 4 destinées aux périphériques mobiles, certains paramètres exigent une configuration particulière. La plupart des valeurs recommandées sont définies automatiquement lorsque vous choisissez Créer > Flash Lite 4 dans l'écran de bienvenue ou Fichier > Nouveau > Flash Lite 4 dans le menu.

Définition des propriétés du document

- 1 Dans Flash Professional, accédez à la scène. Cliquez sur une partie inoccupée de la scène pour éviter de sélectionner des composants.
- 2 Appuyez sur Ctrl+F3 ou sélectionnez Fenêtre > Propriétés dans le menu pour afficher la fenêtre Propriétés. Si vous avez vraiment désélectionné tous les composants, le panneau supérieur de la fenêtre Propriétés contient le titre « Document » et le nom du projet.
- 3 Dans la fenêtre Propriétés, développez la zone Propriétés.
- 4 Sous I/S, sélectionnez une cadence optimale pour le périphérique cible. La vitesse du processeur, la prise en charge de l'accélération matérielle et la quantité de mémoire peuvent affecter les performances d'exécution de la vidéo. Le fabricant peut fournir les informations nécessaires pour optimiser ce paramètre sur un périphérique déterminé.

- 5 Réglez le paramètre Taille sur les dimensions de l'écran du périphérique par défaut. Cette valeur définit la taille par défaut. Pour les autres périphériques, vous pouvez modifier dynamiquement la taille lors de l'exécution, à l'aide de la classe `flash.display.Stage`.

Définition des paramètres de publication

- 1 Sélectionnez Fichier > Paramètres de publication pour ouvrir la boîte de dialogue correspondante.
- 2 Dans l'onglet Formats, sélectionnez HTML (.html) pour créer un fichier SWF intégré s'exécutant dans un navigateur.
- 3 Dans l'onglet Flash, sélectionnez Flash Lite 4.0 dans la liste déroulante Lecteur, puis ActionScript 3.0 dans la liste déroulante Script.

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.

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 création ou l'implémentation de l'application.

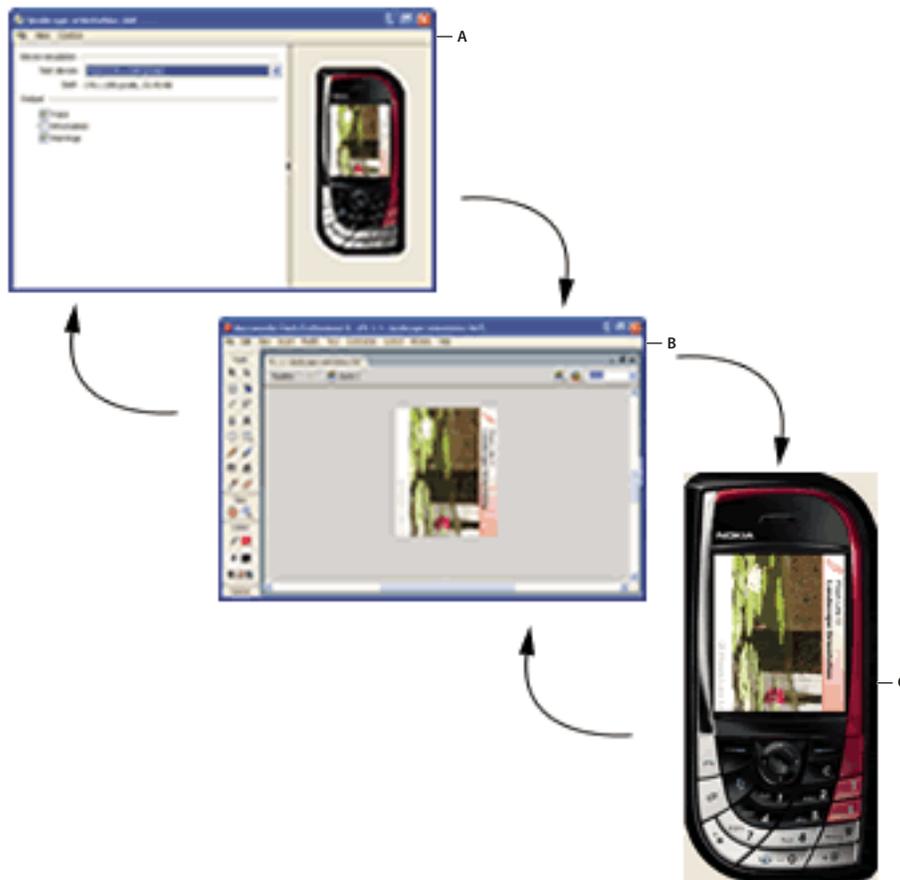
En outre, de nombreux périphériques prennent en charge différents types de contenus Flash Lite, tels que les économiseurs d'écran, les applications autonomes ou les sonneries animées. Le type de contenu pour lequel vous programmez détermine également les fonctionnalités disponibles pour votre application.

Création de l'application dans Flash et test dans Adobe Device Central Adobe Flash Professional CS5 comprend un émulateur dans Adobe Device Central 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 L'émulateur Adobe Device Central ne simule pas tous les aspects du périphérique cible, tels que la vitesse de son processeur, le codage des couleurs ou sa latence réseau. Examinez l'exemple suivant : supposons qu'une animation s'exécute en toute fluidité sur l'émulateur. Il est peu probable qu'elle s'exécute aussi rapidement sur le périphérique du fait de sa vitesse de traitement inférieure. En outre, un dégradé de couleur qui semble continu sur l'émulateur peut contenir des bandes lors de son affichage 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 création 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 programmer une application Flash Lite, tenez compte des considérations suivantes :

- Les périphériques sur lesquels vous souhaitez exécuter le contenu (*périphériques cibles*). Le lecteur Flash Lite est installé sur divers périphériques. Pour obtenir la liste des périphériques sur lesquels Flash Lite est installé, voir la page recensant les périphériques pris en charge sur le site Web d'Adobe à l'adresse www.adobe.com/go/mobile_supported_devices_fr.
- 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 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 de bureau 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 offertes 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.

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

Dans ce didacticiel, vous allez développer une application Adobe Flash Lite, qui récupère des données vidéo en direct sur un site Web en ligne, analyse les données XML et affiche les informations sur l'écran de votre périphérique.

Affichage de l'application terminée

Vous pouvez afficher l'application dans l'émulateur Adobe Device Central. Vous pouvez aussi transférer les applications terminées sur le lecteur Flash Lite autonome installé sur un périphérique mobile.

- 1 Téléchargez puis décompressez les fichiers du tutoriel. Aller à http://www.adobe.com/go/learn_ftl_samples_and_tutorials_fr. Téléchargez le fichier ZIP pour votre version de Flash Lite.
- 2 Dans Flash, choisissez Fichier > Ouvrir et sélectionnez le fichier blogs-rss_complete fla, qui réside dans le dossier Tutorial.
- 3 Choisissez Fichier > Paramètres de publication. Dans l'onglet Flash de la boîte de dialogue Paramètres de publication, assurez-vous que le lecteur correspond à Flash Lite 4.0.
- 4 Sélectionnez Contrôle > Tester l'animation > dans Device Central pour tester l'application dans Device Central. Choisissez le profil de périphérique de test Flash Lite 4.0 32 240x320.

Remarque : l'application simple que vous développez ne prend pas en charge la fonctionnalité multipoint.

- 5 Pour interagir avec l'application, procédez comme suit :
 - Sur l'écran d'accueil, attendez le chargement des titres de blog sur l'écran du périphérique. Cliquez sur la touche Bas pour faire défiler les entrées. Cliquez sur la touche de sélection pour afficher les détails d'une entrée.
 - Cliquez sur la touche Bas pour faire défiler les détails d'une entrée. Appuyez sur la touche 9 du clavier du périphérique pour revenir à la liste de titres. Vous pouvez également appuyer sur 7 pour ouvrir le blog dans un navigateur Web sur le périphérique.

Création de l'application

L'application lit un flux RSS à partir du site de blogs d'Adobe (blogs.adobe.com). Elle répertorie les titres des blogs et permet à l'utilisateur d'afficher les détails d'un blog. Le cas échéant, l'utilisateur peut également lancer le navigateur pour afficher un blog sur le site Web.

Téléchargement du flux RSS

- 1 Dans Flash, sélectionnez Fichier > Ouvrir et sélectionnez le fichier blogs-rss_start fla, qui réside dans le dossier Tutorials.
- 2 Choisissez Fichier > Enregistrer sous et enregistrez le fichier sous le nom blogs-rss fla.

- 3 Dans le scénario (Fenêtre > Scénario), cliquez sur l'image 1 du calque Actions et ouvrez le panneau Actions (Fenêtre > Actions).
- 4 Sous l'instruction import de la ligne 1, ajoutez un composant de préchargement à la scène. Cet objet est un espace réservé graphique qui s'affiche uniquement jusqu'à ce que l'application analyse le contenu XML RSS :

```
var a1:preloader = new preloader();
```

- 5 Définissez les variables globales à utiliser dans l'application :

```
var myNews:Array;
var visi:Number=0;
var selectedNews=0;
var newsshowing:Sprite;
var isSet:Boolean=false;
```

Remarque : les trois étapes suivantes n'exigent aucune intervention de votre part. Elles décrivent simplement les actions exécutées par le code de didacticiel existant.

- 6 Le code existant ci-dessous crée un objet URLLoader et lui ajoute des écouteurs d'événements :

```
var loader:URLLoader = new URLLoader();
loader.addEventListener(Event.COMPLETE, loadxml);
loader.addEventListener(IOErrorEvent.IO_ERROR, ioerror);
loader.addEventListener(SecurityErrorEvent.SECURITY_ERROR, securityerror);
```

- 7 Le code existant ci-dessous appelle la fonction URLLoader.load(), en transmettant l'URL du flux de blogs en direct via un objet URLRequest :

```
loader.load(new URLRequest("http://blogs.adobe.com/index.xml"));
```

- 8 Le code existant ci-dessous définit des fonctions de rappel pour les événements de chargement :

```
function loadxml(e):void{
}
function ioerror(e):void {
    trace(e.type);
}
function securityerror(e):void {
    trace(e.type);
}
```

Analyse du flux RSS téléchargé

Procédez comme suit pour ajouter le code indiqué au corps de la fonction loadxml() :

- 1 Vous êtes maintenant prêt à analyser les données XML et à afficher le flux RSS. Supprimez donc l'objet de préchargement de la scène :

```
stage.removeChild(a1);
```

- 2 Créez un objet XML et affectez-lui le flux RSS :

```
var xml:XML=new XML(loader.data);
```

- 3 Créez un tableau et affectez-le à la variable globale myNews :

```
myNews = new Array();
```

- 4 Analysez les données XML pour afficher le titre, l'URL, la description et la date de publication de l'entrée de blog :

```

for each (var param:XML in xml.channel.item) {
    myNews.push({title:param.title,link:param.link,
                description:param.description,date:param.pubDate} );
}

```

La fonction `loadxml()` se présente maintenant comme suit :

```

function loadxml(e):void {
    stage.removeChild(a1);
    var xml=new XML(loader.data);
    myNews = new Array();
    for each (var param:XML in xml.channel.item) {
        myNews.push({title:param.title,link:param.link,
                    description:param.description,date:param.pubDate});
    }
}

```

Création de l'écran d'accueil

- 1 Dans le scénario, sélectionnez l'image 1 dans le calque d'en-tête. Cliquez sur le panneau Bibliothèque et faites glisser une occurrence de l'objet d'en-tête vers le haut du document sur la scène.
- 2 Sélectionnez l'image 1 dans le calque de la scène. Ajoutez quatre champs de texte dynamiques, alignés verticalement, dans les quatre emplacements sous l'objet d'en-tête. Définissez la largeur de chaque champ de texte sur 230,00 et sa hauteur sur 45,25.
- 3 Dans le panneau Propriétés, attribuez aux occurrences des champs de texte les noms `tf1`, `tf2`, `tf3` et `tf4`, respectivement, de haut en bas.
- 4 Sélectionnez l'image 1 dans le calque Actions et ouvrez le panneau Actions. Créez la fonction `showText()` pour afficher les titres des blogs dans les champs de texte :

```

function showText():void {
    tf1.text=myNews[visi].title;
    tf2.text=myNews[visi+1].title;
    tf3.text=myNews[visi+2].title;
    tf4.text=myNews[visi+3].title;
}

```

Remarque : la variable `visi` stocke la valeur d'index de la première entrée de blog visible.

- 5 Appelez la fonction `showText()` sur la dernière ligne de la fonction `loadxml()` :

```

function loadxml(e):void {
    stage.removeChild(a1);
    var xml=new XML(loader.data);
    myNews = new Array();
    for each (var param:XML in xml.channel.item) {
        myNews.push({title:param.title,link:param.link,
                    description:param.description,date:param.pubDate});
    }
    showText();
}

```

- 6 Depuis le panneau Bibliothèque, faites glisser une occurrence de l'objet de surbrillance vers `tf1`. Dans l'onglet Propriétés, entrez `highlightBar` dans le champ Nom d'occurrence.

Affichage de la date et de l'heure actuelles dans l'en-tête

Le composant d'en-tête contient le titre « Adobe blogs » à gauche et un champ de texte appelé tf à droite. Procédez comme suit pour ajouter un tampon de date/heure à tf.

- 1 Dans le panneau Bibliothèque, double-cliquez sur le composant d'en-tête pour l'ouvrir en vue de le modifier. Le composant est alors le seul objet de la scène.
- 2 Cliquez sur une zone inoccupée de la scène pour désélectionner tous les éléments d'en-tête. Ouvrez le panneau Actions.
- 3 Dans le panneau Actions, ajoutez un écouteur de l'événement ENTER_FRAME. A cet effet, collez le code suivant dans l'image 1 du calque 1 de la sélection active (le composant d'en-tête) :

```
addEventListener(Event.ENTER_FRAME, changeDate);
```

- 4 Sur la ligne suivante, ajoutez le code ci-dessous pour implémenter le gestionnaire d'événement qui affiche la date et l'heure :

```
function changeDate(e):void {  
    var date = new Date();  
    tf.text = date.toString();  
    tf.appendText("\n"+date.toLocaleTimeString());  
}
```

Création d'une liste défilante d'entrées de blog

- 1 Ouvrez la séquence 1 sur la scène.
- 2 Dans le scénario, sélectionnez l'image 1 dans le calque de défilement. Depuis le panneau de la bibliothèque, faites glisser une occurrence de l'objet de défilement vers le bord droit du document de la scène. Positionnez cet objet de telle sorte que sa partie supérieure se trouve sur le bord droit de l'en-tête. Dans l'onglet Propriétés, entrez scrollee dans le champ Nom d'occurrence.
- 3 Sélectionnez l'image 1 dans le calque Actions. Dans le panneau Actions, ajoutez un écouteur d'événement à la scène pour détecter les événements de clavier :

```
stage.addEventListener(KeyboardEvent.KEY_DOWN, onKeyDown);
```

- 4 Ajoutez un gestionnaire de l'événement KEY_DOWN. Par le biais de ce gestionnaire, vous implémentez le défilement des titres de blog. Vous ajoutez également une fonctionnalité permettant de sélectionner un titre de blog et d'afficher les détails correspondants :

```

function onKeyDown(e):void {
    if (e.keyCode==Keyboard.ENTER) {
        trace(myNews[visi+selectedNews].title);
        if(isSet==false){
            showNews();
            isSet=true;
        }
        return;
    }
    if (e.keyCode==57) { // 9 key
        comeBack();
    }
    if (e.keyCode==55) { // 7 key
        goto();
    }
    if (e.keyCode==53) { // 5 key
        if(isSet==false){onrefresh();}
    }
    if (e.keyCode==Keyboard.DOWN) {
        if(isSet==true){return;}
        if (selectedNews<3) {
            highlightBar.y+=62;
            selectedNews+=1;
            scrollee.scroller.y+=(280/myNews.length);
            return;
        }
        if (((visi+4)>=(myNews.length))) {
            return;
        }
        scrollee.scroller.y+=(280/myNews.length);
        visi+=1;
        showText();
    }
    if (e.keyCode==Keyboard.UP) {
        if(isSet==true){return;}
        if (selectedNews>0) {
            highlightBar.y-=62;
            selectedNews-=1;
            scrollee.scroller.y-=(280/myNews.length);
            return;
        }
        if (visi==0) {
            return;
        }
        scrollee.scroller.y-=(280/myNews.length);
        visi-=1;
        showText();
    }
}

```

Ajout de la fonction showNews()

Lorsque vous appuyez sur la touche de sélection pour afficher les détails d'un blog, l'application appelle la classe `readNews`. Cette classe est définie dans le fichier `readNews.as` du dossier `Tutorials`. Elle permet d'afficher le contenu d'un blog. Dans l'image 1 du calque `Actions`, collez le code ci-dessous dans le panneau `Actions` :

```
function showNews():void {
    newsshowing=new readNews(stage,myNews[visi+selectedNews].title,
        myNews[visi+selectedNews].description,myNews[visi+selectedNews].date);
    newsshowing.x=3;
    newsshowing.y=32;
    addChild(newsshowing);
}
```

Ajout de la fonction comeback()

Pour revenir à la liste des entrées de blog, vous appuyez sur la touche 9 du clavier. Dans l'image 1 du calque Actions, insérez le code ci-dessous dans le panneau Actions :

```
function comeBack():void {
    if (isSet==true) {
        removeChild(newsshowing);
        isSet=false;
    }
    return;
}
```

Actualisation de la liste des entrées de blog

Pour actualiser la liste des entrées de blog, vous appuyez sur la touche 5 du clavier. L'application extrait à nouveau le flux RSS et affiche les données XML actives.

- 1 Dans l'image 1 du calque Actions, insérez le code ci-dessous dans le panneau Actions :

```
function onrefresh():void {
    visi=0;
    selectedNews=0;
    scrollee.scroller.y=0;
    highlightBar.y=73.95;
    loader.load(new URLRequest("http://blogs.adobe.com/index.xml"));
    stage.addChild(a1);
}
```

- 2 Près du début du code dans le panneau Actions, appelez la fonction `onrefresh()` immédiatement après l'ajout des écouteurs d'événements de chargement :

```
var loader:URLLoader = new URLLoader();
loader.addEventListener(Event.COMPLETE,loadxml);
loader.addEventListener(IOErrorEvent.IO_ERROR,ioerror);
loader.addEventListener(SecurityErrorEvent.SECURITY_ERROR,securityerror);
onrefresh();
```

Accès à la page Web du blog

Pour afficher un blog dans une fenêtre de navigateur distincte, vous appuyez sur 7 sur le clavier. Dans l'image 1 du calque Actions, insérez le code ci-dessous dans le panneau Actions :

```
function goto():void {
    if (isSet==true) {
        navigateToURL(new URLRequest(myNews[visi+selectedNews].link));
    }
    return;
}
```

Test et publication de l'application

L'application est maintenant terminée. Vous pouvez la tester dans Device Central en choisissant Débogage > Déboguer l'animation > dans Device Central ou Contrôle > Tester l'animation > dans Device Central.

Pour publier l'application, sélectionnez Fichier > Publier. Flash Professional enregistre le fichier SWF intégré sous le nom blogs-rss.html. Pour qu'il soit possible d'afficher le fichier SWF local dans Flash Player, vous devez l'enregistrer en tant que fichier approuvé. Pour plus d'informations à ce sujet, voir www.adobe.com/go/learn_fl_securitypanel_fr.

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

Pour interagir avec votre application Adobe Flash Lite, un utilisateur doit pouvoir effectuer les opérations suivantes :

- Identifier l'objet qui possède le focus à l'écran
- Naviguer au sein des objets
- Initier une action en sélectionnant un objet

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, voir « [A propos des types de contenus Flash Lite](#) » à la page 7.

La méthode la plus simple de mise en place de touches interactives dans une application Flash Lite consiste à appliquer la navigation par défaut, qui exploite le pavé quadridirectionnel du périphérique comme les touches fléchées ou la touche de tabulation et la combinaison Maj+tabulation dans une application de bureau. L'utilisateur place le focus sur l'objet voulu, puis appuie sur la touche de sélection. L'application inclut le code de gestionnaire d'événement pour répondre à ces événements de bouton.

La navigation par défaut de Flash Lite permet d'utiliser des boutons, des champs de texte de saisie et, en option, des clips. Cette fonctionnalité est généralement préférable pour les interactions utilisateur simples, telles que les menus.

Les applications peuvent également répondre aux événements de souris ou de pression de touche que Flash Lite génère lorsqu'un utilisateur effectue un clic de souris ou appuie sur une touche spécifique. 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.

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 de sélectionner des éléments sur l'écran du périphérique.

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. Dans un jeu, par exemple, l'utilisateur déplace un personnage à l'écran à l'aide des touches de navigation. La touche de sélection peut également être affectée à une autre action, comme permettre au personnage de sauter.

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.

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

Description	Touches	Disponibilité
Chiffre, *, #	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 prenant en charge la navigation quadridirectionnelle uniquement (voir « Modes de navigation avec le clavier par défaut » à la page 17)
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

Pour plus de détails sur la classe `System.Capabilities`, voir *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Navigation par défaut

Utilisation de la navigation avec le clavier par défaut dans Flash Lite

Dans les applications de bureau, les touches tabulation et Maj+tabulation permettent aux utilisateurs de déplacer le focus d'un objet à l'autre à l'écran. Les touches fléchées ont la même fonction dans d'autres applications. Dans Flash Lite, les touches de navigation du clavier du périphérique ont la même fonction que la flèche ou les touches de tabulation et Maj+tabulation d'une application de bureau Flash. Lorsque l'objet désiré possède le focus, l'utilisateur peut appuyer sur la touche de sélection pour déclencher une action dans l'application. Vous définissez des gestionnaires d'événement permettant de répondre à la sélection d'un bouton ou d'un clip ; pour plus d'informations, voir « [Gestion des événements de souris et de touche](#) » à la page 23.

La navigation avec le clavier par défaut de Flash Lite utilise des boutons et des champs de texte de saisie. Les clips sont pris en compte s'ils satisfont aux critères suivants :

- Leur propriété `tabEnabled` est définie sur `true`.
- Ils sont associés à des gestionnaires d'événements et leur propriété `tabEnabled` n'est pas définie sur `false`.

Lorsqu'un champ de texte de saisie possède le focus et que l'utilisateur appuie sur la touche de sélection, Flash Lite active le champ pour permettre sa modification en ligne.

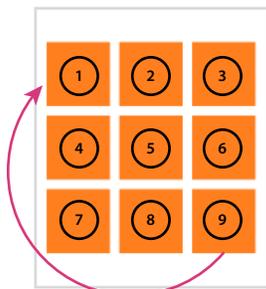
Vous trouverez un exemple de navigation par défaut à la section « [Création d'un menu simple à l'aide de boutons et de la navigation avec la souris](#) » à la page 25.

Modes de navigation avec le clavier par défaut

Flash Lite prend en charge trois modes de navigation avec le clavier 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.

La *navigation bidirectionnelle* dans Flash Lite est similaire à la navigation par tabulation dans les applications de bureau. Dans ces applications, les touches tabulation et Maj+tabulation permettent de naviguer parmi les objets affichés 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 sur 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 du périphérique. Lorsque le bouton situé dans le coin inférieur droit reçoit le focus, celui-ci 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, voir « [Contrôle de l'ordre de tabulation lors de la navigation bidirectionnelle](#) » à la page 22.

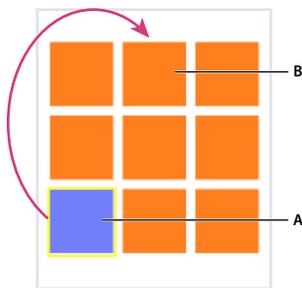
Pour obtenir un exemple de navigation bidirectionnelle, voir la page regroupant des didacticiels et des exemples pour Flash Lite à l'adresse www.adobe.com/go/learn_ftl_samples_and_tutorials_fr. Recherchez, téléchargez puis décompressez le fichier ZIP correspondant à votre version d'ActionScript. Naviguez ensuite jusqu'au dossier Samples pour accéder au fichier d'exemple 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 qui gèrent la navigation quadridirectionnelle, contrairement aux propriétés `tabEnabled` et `tabChildren`. Ce comportement est différent dans les applications de bureau.

Pour obtenir un exemple de navigation quadridirectionnelle, voir la page regroupant des didacticiels et des exemples pour Flash Lite à l'adresse www.adobe.com/go/learn_ftl_samples_and_tutorials_fr. Repérez, téléchargez puis décompressez le fichier ZIP correspondant à votre version d'ActionScript. Affichez ensuite le dossier Samples pour accéder au fichier d'exemple 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 du clavier est situé dans le coin inférieur gauche de l'écran. Si 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 possédant 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. 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 et sélectionnez les exemples qui vous intéressent.

Chaque fichier d'exemple se compose de la même grille à neuf boutons, comme indiqué plus haut, à une différence près. Chaque fichier FLA est configuré de sorte à cibler une combinaison de périphérique et type de contenu Flash Lite prenant en charge le mode de navigation (bidirectionnelle ou quadridirectionnelle).

Pour utiliser un fichier d'exemple, ouvrez-le dans Flash et testez-le dans l'émulateur Adobe Device Central (sélectionnez Contrôle > Tester l'animation > dans Device Central). 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

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`, veillez à offrir un autre indicateur de focus pour les boutons, les champs de texte de saisie et les clips activés par la touche de tabulation. Dans le cas des boutons, il revient au même d'ajouter un élément visuel à l'état « Survol », c'est-à-dire la partie du scénario d'un objet `Button` qui s'affiche lorsque le bouton possède le focus. L'exemple d'application de la section « [Création d'un menu simple à l'aide de boutons et de la navigation avec la souris](#) » à la page 25 illustre cette technique. Dans le cas d'un champ de saisie de texte, vous pouvez utiliser l'objet `Stage` pour déterminer si le champ possède le focus et afficher l'indicateur de focus voulu.
- Il est recommandé d'avoir au moins deux objets (champs de texte de saisie, boutons, clips activés par la touche de tabulation ou toute combinaison de ces objets) à l'écran en même temps. Supposons que l'écran contienne un seul champ de texte de saisie, bouton ou clip activé par tabulation. L'utilisateur ne peut alors pas changer le focus et risque de se sentir enfermé dans l'interface utilisateur.

Si un écran de votre application ne contient qu'un seul bouton d'interaction utilisateur, envisagez de détecter un événement de pression de touche plutôt que d'utiliser des événements de bouton.

- Le cas échéant, pensez à utiliser la propriété `Stage.focus()` pour définir le focus initial sur un objet spécifique à l'écran. Cette propriété permet de guider l'utilisateur au sein de l'interface et de réduire le nombre de touches à utiliser. Par exemple, supposons que l'un des écrans de votre application contienne un champ de saisie de texte. Pour entrer une valeur dans ce champ, l'utilisateur appuie d'abord sur une touche de navigation pour placer le focus sur le champ. Il appuie ensuite sur la touche de sélection pour ouvrir la boîte de dialogue de saisie de texte. Le code ActionScript suivant permet de placer le focus du clavier automatiquement sur le champ de texte de saisie `inputTxt` :

```
stage.focus = inputTxt;
```

Pour plus d'informations, voir l'exemple d'application dans la section « [Contrôle du focus avec le code ActionScript](#) » à la page 21.

- 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. En navigation bidirectionnelle, vous pouvez également contrôler la séquence à l'aide de la propriété `tabIndex`.

A propos du rectangle de focus

Par défaut, Flash Lite dessine un rectangle jaune autour du bouton ou du champ de texte de saisie ayant le focus. Les clips sont également pris en compte s'ils satisfont aux critères suivants :

- Leur propriété `tabEnabled` est définie sur `true`.
- Ils sont associés à des gestionnaires d'événements et leur propriété `tabEnabled` n'est pas définie sur `false`.

Ce rectangle de focus permet à l'utilisateur d'identifier l'objet à 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 actif et rectangle de focus par défaut

Dans le cas des boutons et des clips, le cadre de sélection du rectangle de focus est déterminé par la *zone réactive* de l'objet. Il s'agit de la zone invisible qui (dans les applications de bureau) définit la partie du bouton ou du clip qui répond aux clics de souris. Dans le cas des champs de texte de saisie, ce sont leurs dimensions qui déterminent le cadre de sélection du rectangle de focus.

Vous pouvez personnaliser la couleur du rectangle de focus ou désactiver ce dernier.

Personnalisation du rectangle de focus

Surligné par défaut en jaune, le rectangle de focus désigne le bouton ou le champ de texte de saisie sélectionné. Les clips sont pris en compte si leur propriété `tabEnabled` est définie sur `true`. Ils sont également sélectionnés s'ils sont associés à des gestionnaires d'événement et que leur propriété `tabEnabled` n'est pas définie sur `false`.

Vous pouvez désactiver le comportement par défaut du rectangle de focus en définissant la propriété `Stage.stageFocusRect()` sur `false`. Vous pouvez également désactiver ce comportement pour des objets interactifs individuels, tels que des boutons et des clips, en définissant leur propriété `focusRect`. Pour plus de détails, voir.

Contrôle du focus avec le code ActionScript

L'objet `Stage` permet d'obtenir et de définir le focus du clavier ou d'être averti lorsqu'un objet reçoit ou perd ce focus.

Ceci est utile, par exemple, pour attribuer automatiquement le focus à un bouton particulier lors du premier chargement de l'application. Vous pouvez également faire appel à cet objet pour être averti lorsqu'un objet spécifique à l'écran reçoit ou perd le focus du clavier. Vous pouvez mettre l'écran à jour en fonction de la notification.

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

```
Stage.Focus = login_btn;
```

Les événements `FocusEvent.FOCUS_IN` et `FocusEvent.FOCUS_OUT` permettent de déterminer les changements de focus. Vous pouvez vous en servir pour créer un gestionnaire de focus personnalisé pour tous les objets interactifs, par exemple. 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`), voir 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. Recherchez, téléchargez puis décompressez le fichier ZIP correspondant à votre version d'ActionScript. Affichez ensuite le dossier `Samples` pour accéder à l'exemple.

Création d'un gestionnaire personnalisé de focus sur le champ de 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 de texte sur la scène.
- 3 Sans désélectionner ce champ, dans l'onglet Propriétés, sélectionnez Texte de saisie dans le menu contextuel Type de texte. Entrez `inputTxt_1` dans le champ de texte Nom d'occurrence et sélectionnez l'option Afficher la bordure autour du texte.
- 4 De la même manière, créez un autre champ de texte de saisie sous le premier, attribuez-lui le nom d'occurrence `inputTxt_2`, puis sélectionnez l'option Afficher la bordure autour du 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 :

```
//Set the display to full screen.
stage.displayState = StageDisplayState.FULL_SCREEN
//Add focus in and out event listeners for the current context.
stage.addEventListener(FocusEvent.FOCUS_IN, focushandler);
stage.addEventListener(FocusEvent.FOCUS_OUT, focusouthandler);
function focushandler(e:FocusEvent): void {
    if (e.target is TextField){
        //Set border color of text field with focus to red.
        e.target.borderColor = 0xFF0000;
    }
}
function focusouthandler(e:FocusEvent): void {
    if (e.target is TextField){
        //Set border color of text field without focus to black.
        e.target.borderColor = 0x000000;
    }
}
```

- 7 Enregistrez les modifications et testez l'application dans l'émulateur (Contrôle > Tester l'animation > dans Device Central).
- 8 Appuyez sur les touches Bas et Haut de l'émulateur pour faire basculer le focus d'un champ de texte à l'autre. Le champ de texte possédant le focus a une bordure rouge, tandis que l'autre a alors une bordure noire. Appuyez sur la touche de sélection lorsqu'un champ de 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 `InteractiveObject.tabIndex` permettant de définir avec précision l'ordre des boutons, des clips et des champs de texte de saisie. Les périphériques gérant la navigation quadridirectionnelle ne prennent pas en charge la propriété `tabIndex`. Il est donc impossible de définir l'ordre de tabulation à l'aide de cette propriété pour ce type de navigation.

Pour contrôler l'ordre de tabulation lors de la navigation bidirectionnelle, affectez à la propriété `tabIndex` de chaque objet interactif un numéro spécifiant sa position 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 de 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;
```

Remarque : vous devez définir la propriété `tabEnabled` de la plupart des objets interactifs sur `true`, à quelques exceptions près, dont les boutons et les champs de texte de saisie. Pour plus d'informations, voir la description de la propriété `flash.display.InteractiveObject.tabEnabled()`.

Gestion des événements de souris et de touche

L'architecture d'événements d'ActionScript 3.0 a été complètement réécrite.

Les objets interactifs (tout objet dérivé de `flash.display.InteractiveObject`) permettent d'ajouter de l'interactivité aux applications Flash Lite.

A propos des événements de souris

Les gestionnaires et les écouteurs d'événements déterminent la façon dont l'application réagit aux occurrences générées par l'utilisateur ou le système. Outre la navigation par défaut et les interactions avec les événements connexes, une application Flash Lite permet également d'écouter des événements de souris et d'y répondre.

Gestion des événements de souris

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

Événement de souris	Description
CLICK	Définit la valeur de la propriété de type d'un objet événement <code>click</code> .
DOUBLE_CLICK	Définit la valeur de la propriété de type d'un objet événement <code>doubleClick</code> .
MOUSE_DOWN	Définit la valeur de la propriété de type d'un objet événement <code>mouseDown</code> .
MOUSE_MOVE	Définit la valeur de la propriété de type d'un objet événement <code>mouseMove</code> .
MOUSE_OUT	Définit la valeur de la propriété de type d'un objet événement <code>mouseOut</code> .
MOUSE_OVER	Définit la valeur de la propriété de type d'un objet événement <code>mouseOver</code> .
MOUSE_UP	Définit la valeur de la propriété de type d'un objet événement <code>mouseUp</code> .
MOUSE_WHEEL	Définit la valeur de la propriété de type d'un objet événement <code>mouseWheel</code> .
ROLL_OUT	Définit la valeur de la propriété de type d'un objet événement <code>rollOut</code> .
ROLL_OVER	Définit la valeur de la propriété de type d'un objet événement <code>rollOver</code> .

La procédure ci-après explique comment créer une application simple capable de gérer des événements de souris. Vous trouverez un exemple de création d'un menu simple à l'aide de boutons et de la navigation avec la souris à la section « [Création d'un menu simple à l'aide de boutons et de la navigation avec la souris](#) » à la page 25.

Création d'un gestionnaire d'événement pour des boutons

1 Dans Flash, créez un document mobile et enregistrez-le sous le nom `custom_mouse_handler 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'onglet Propriétés, dans le champ 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'onglet Propriétés, dans le champ 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.
stage.stageFocusRect = false;

//Add event listeners for btn_1.
btn_1.addEventListener(MouseEvent.CLICK, downhandler1);
btn_1.addEventListener(MouseEvent.CLICK, uphandler1);
btn_1.addEventListener(MouseEvent.CLICK, overhandler1);
btn_1.addEventListener(MouseEvent.CLICK, outhandler1);

//Add event handlers for btn_1.
function downhandler1(e:MouseEvent):void {
    trace("you pressed Button 1");
}

function uphandler1(e:MouseEvent):void {
    trace("you released Button 1");
}

function overhandler1(e:MouseEvent):void {
    trace("Button 1 has focus");
}

function outhandler1(e:MouseEvent):void {
    trace("Button 1 lost focus");
}

//Add event listeners for btn_2.
btn_2.addEventListener(MouseEvent.CLICK, downhandler2);
```

```
btn_2.addEventListener(MouseEvent.CLICK, clickHandler2);
btn_2.addEventListener(MouseEvent.MOUSE_UP, uphandler2);
btn_2.addEventListener(MouseEvent.MOUSE_OVER, overhandler2);
btn_2.addEventListener(MouseEvent.MOUSE_OUT, outhandler2);

//Add event handlers for btn_2.
function downhandler2(e:MouseEvent):void {
    trace("you pressed Button 2");
}

function uphandler2(e:MouseEvent):void {
    trace("you released Button 2");
}

function overhandler2(e:MouseEvent):void {
    trace("Button 2 has focus");
}

function outhandler2(e:MouseEvent):void {
    trace("Button 2 lost focus");
}
```

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

Observez les messages qui s'affichent dans le panneau de sortie lorsque vous cliquez sur les boutons sur la scène.

D'autres types d'objets prennent en charge des événements différents. L'objet TextField, par exemple, comporte l'événement `change`, qui s'exécute lors de la modification du contenu d'un champ de texte. Vous pouvez rédiger le code du gestionnaire de ces événements en reprenant le format des gestionnaires d'événement de bouton de cette procédure. Pour plus de détails sur les classes TextField and MovieClip, voir Adobe ActionScript 3.0 Reference for the Adobe Flash Platform.

Création d'un menu simple à l'aide de boutons et de la navigation avec la souris

Pour créer un menu simple à l'aide de boutons et de la navigation avec la souris, mettez le menu au point, puis rédigez le code du gestionnaire des événements de souris. Pour créer le menu, vous utilisez trois symboles de bouton, un pour chaque élément du menu. Vous écrivez ensuite le code du gestionnaire d'événement qui affiche un message lorsque l'utilisateur survole chaque élément de menu ; autrement dit, lorsque l'utilisateur place le focus sur le bouton correspondant et clique sur l'élément de menu donné. Pour plus d'informations sur la gestion des événements de souris dans Flash Lite, voir « [Gestion des événements de souris](#) » à la page 23.

Commencez avec un document Flash partiel. Vous pouvez modifier ces paramètres pour cibler un périphérique et un type de contenu différents (voir *Utilisation d'Adobe Device Central CS5*).

- 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. 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 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'onglet Propriétés, dans le champ 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 Aligned les trois boutons verticalement, comme indiqué dans l'exemple ci-dessous.
- 8 Dans le panneau Outils, sélectionnez l'outil Texte et tracez un champ de texte le long de la partie inférieure de la scène.

Ce champ de texte affiche un court message lorsque l'utilisateur survole les éléments du menu.

- 9 Sans désélectionner le nouveau champ de texte, effectuez les opérations suivantes dans l'onglet Propriétés :
 - a Tapez **txt_status** dans le champ de texte Nom d'occurrence.
 - b Sélectionnez Texte dynamique dans le menu contextuel Type de texte.
 - c Sélectionnez Utiliser les polices de périphérique dans le menu contextuel Méthode de rendu des polices.
- 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.
stage.stageFocusRect = false;

//Add event listeners for each button to listen for roll overs and clicks.
btn_news.addEventListener(MouseEvent.ROLL_OVER, mouserollhandler);
btn_news.addEventListener(MouseEvent.CLICK, mouseclickhandler);

btn_sports.addEventListener(MouseEvent.ROLL_OVER, mouserollhandler);
btn_sports.addEventListener(MouseEvent.CLICK, mouseclickhandler);

btn_weather.addEventListener(MouseEvent.ROLL_OVER, mouserollhandler);
btn_weather.addEventListener(MouseEvent.CLICK, mouseclickhandler);

//Set the function to respond to roll overs.
function mouserollhandler (myEvent:MouseEvent): void {
    //Check if the current button is the btn_news button.
    if (myEvent.target == btn_news){
        //Set the text field text.
        txt_status.text = "Press to select News"
    }
    //Check if the current button is the btn_sports button.
    else if (myEvent.target == btn_sports){
        txt_status.text = "Press to select Sports"
    }
    //Check if the current button is the btn_weather button.
```

```

    else if (myEvent.target == btn_weather){
        txt_status.text = "Press to select Weather"
    }
}

//Set the function to respond to clicks.
function mouseclickhandler (myEvent:MouseEvent): void {
    //Check if the current button is the btn_news button.
    if (myEvent.target == btn_news){
        txt_status.text = "You pressed News"
    }
    //Check if the current button is the btn_sports button.
    else if (myEvent.target == btn_sports){
        txt_status.text = "You pressed Sports"
    }
    //Check if the current button is the btn_weather button.
    else if (myEvent.target == btn_weather){
        txt_status.text = "You pressed Weather"
    }
}
}

```

12 Sélectionnez Contrôle > Tester l'animation > dans Device Central pour afficher un aperçu de l'application dans l'émulateur.

Cliquez sur chaque élément de menu et remarquez le message qui s'affiche.

Gestion des événements de clavier

Flash Lite génère des événements de clavier lorsque l'utilisateur appuie sur les 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, voir « [Touches prises en charge par Flash Lite](#) » à la page 16.

Le tableau suivant recense les événements de clavier d'ActionScript 3.0 pris en charge par Flash Lite :

Événement de clavier	Description
KEY_DOWN	Définit la valeur de la propriété de type d'un objet événement <code>keyDown</code> .
KEY_UP	Définit la valeur de la propriété de type d'un objet événement <code>keyUp</code> .

Création d'un gestionnaire d'événement

Pour gérer les événements de pression de touche, il suffit d'ajouter pour l'événement de clavier `keyDown` un écouteur qui se réfère à une fonction de gestionnaire. Le code suivant associe une fonction de gestionnaire d'événement pour répondre lorsque l'utilisateur appuie sur la touche de navigation vers la droite du périphérique :

```
//Add an event listener to the stage for the keyDown event using the equivalent
//constant value KEY_DOWN from the KeyboardEvent class. This associates the event
//with a "handler" function to respond to the event.
stage.addEventListener(KeyboardEvent.KEY_DOWN, onKeyDownHandler);

//Create a handler function to respond to the event.
function onKeyDownHandler(event:KeyboardEvent):void {
    //Because the right navigation key has the key code 39,
    //check for the keycode value 39, then respond.
    if(event.keyCode == 39){
        trace("You pressed the Right Arrow key");
    }
}
```

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

- 1 Dans Flash, créez un document mobile et enregistrez-le sous keylistener fla.
- 2 Sélectionnez Calque 1 dans le scénario et renommez-le Content.
- 3 Dans le panneau Outils, sélectionnez l'outil Ovale et créez un ovale ou un cercle sur la scène. Convertissez-le ensuite en clip.
- 4 Ce nouveau clip étant sélectionné, dans l'onglet Propriétés, tapez **circle** dans le champ de texte Nom d'occurrence.
- 5 Dans le scénario, insérez un calque et renommez-le ActionScript. Dans ce calque, sélectionnez la première image.
- 6 Ouvrez le panneau Actions (Fenêtre > Actions), puis tapez le code suivant :

```
//Add an event listener to the circle movie clip for the keyDown event using
//the equivalent constant value KEY_DOWN from the KeyboardEvent class.
//This associates the event with a "handler" function to respond to the event.
circle.addEventListener(KeyboardEvent.KEY_DOWN, onKeyDownHandler);
circle.focusRect = false;
stage.focus = circle;

//Create a handler function to respond to the event.
function onKeyDownHandler(event:KeyboardEvent):void {
    //Because the right navigation key has the key code 39,
    //check for the keycode value 39, then respond.
    if(event.keyCode == 39){
        circle.x++;
    }
    //Now respond to the left key.
    else if(event.keyCode == 37){
        circle.x--;
    }
    //Now respond to the up key.
    else if(event.keyCode == 38){
        circle.y--;
    }
    //Now respond to the down key.
    else if(event.keyCode == 40){
        circle.y++;
    }
}
```

- 7 Testez l'application en sélectionnant Contrôle > Tester l'animation > dans Device Central.

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.

Prise en charge multipoint

Flash Lite 4 offre la prise en charge multipoint pour gérer les événements multipoint et les événements de mouvement natifs. Cela signifie que tous les périphériques mobiles pris en charge gèrent les points tactiles multiples et y répondent. En outre, les données de point tactile entrantes sont traitées sous forme de mouvements de plus haut niveau. Pour activer cette fonction, ActionScript 3.0 s'enrichit de nouvelles API d'événements tactiles et de mouvement.

Cette fonction permet d'interagir avec des périphériques en touchant l'écran d'un ou de plusieurs doigts. Le système d'exploitation sous-jacent assure le suivi de tous les points tactiles.

Le contenu Flash Lite 4 a été créé spécifiquement pour les périphériques multipoint. Les API d'événements tactiles permettent d'exposer ces événements au code ActionScript.

La prise en charge multipoint entre dans deux catégories principales. Dans la première catégorie, le contenu Flash Lite répond directement aux événements au fur et à mesure de leur génération par le digitaliseur et les pilotes du périphérique. Dans une application de peinture, par exemple, vous pouvez faire glisser deux doigts sur l'écran pour tracer deux lignes parallèles. Dans la seconde catégorie, le système d'exploitation synthétise un flux de points tactiles entrant sous forme de mouvements de plus haut niveau. Les mouvements du système d'exploitation natifs sont exposés au contenu Flash Lite par le biais des API d'ActionScript 3.0. Ainsi, vous pouvez réduire la taille d'une image en faisant glisser vos doigts sur l'écran dans un mouvement de pince.

Gestion des événements tactiles

Pour répondre aux actions tactiles, les objets ActionScript 3.0 écoutent un certain nombre de types d'événements tactiles. Ces objets reflètent les événements de souris existants et écoutent les événements de survol, de sortie et de défilement, en plus des événements d'état de base, tels que les événements d'appui, de mouvement et de relâchement. Les objets ActionScript 3.0 ne prennent pas en charge les états Survol ni les capteurs de proximité.

Il existe une différence majeure entre un événement tactile et un événement de souris. Chaque événement tactile contient un ID de point tactile unique qui identifie un jeu d'événements connexes. Tous les événements générés par un même doigt sont donc associés à un ID unique, différent de celui des autres doigts. Les événements tactiles d'appui et de relâchement partagent cet ID.

Sur les systèmes uniquement tactiles, le système d'exploitation définit un point tactile primaire. Il mappe également sur ce point un flux parallèle d'événements de souris de niveau système d'exploitation.

Gestion des événements de mouvement natifs

Le digitaliseur génère des mouvements natifs au niveau du système d'exploitation et les transfère au contenu Flash Lite sous la forme d'une nouvelle classe d'événements. Il affecte une phase de mouvement à chaque événement de mouvement. Ces phases sont les suivantes : *begin*, *update* et *end*. Lorsque le système d'exploitation détecte un nouveau mouvement, il le teste par rapport à une pile d'objets à l'emplacement du mouvement. S'il identifie un écouteur, il envoie l'événement de mouvement initial et tous les événements ultérieurs associés au mouvement. Il ne tient pas compte des emplacements des événements ultérieurs même s'ils ne correspondent pas à l'événement initial.

API d'ActionScript 3.0

Pour gérer cette fonction, ActionScript 3.0 propose une nouvelle classe, Multitouch. Cette classe d'aide gère l'état global du système multipoint.

Pour plus d'informations sur la classe Multitouch, ses sous-classes, propriétés et méthodes, voir *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Chapitre 5 : Utilisation du texte et des polices

Vous pouvez insérer des champs de texte statiques et dynamiques ou des champs de texte de saisie dans Adobe Flash Lite 4.

Texte

A propos du texte dans Flash Lite

Flash Lite 4 prend en charge les fonctionnalités de texte suivantes :

- Champs de texte de saisie, statique et dynamique

Lors de l'exécution, le contenu des champs de texte statique ne peut pas changer contrairement à celui des champs de texte dynamique ou de saisie. Les champs de texte de saisie permettent aux utilisateurs d'entrer du texte. Flash Lite 4 prend en charge la saisie de texte en ligne sur la plupart des périphériques. Pour les périphériques qui prennent en charge des langues complexes, les champs de texte de saisie appliquent le mécanisme générique de saisie de texte du périphérique.

- Polices incorporées et polices de périphérique

Flash Lite 4 procède au rendu des champs de texte à l'aide de polices vectorielles incorporées dans le fichier SWF ou des polices disponibles sur le périphérique. Pour plus d'informations, voir « [Méthodes de rendu des polices dans Flash Lite](#) » à la page 40.

- Encodage du texte au format Unicode

Flash Lite 4 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, voir « Création de texte multilingue » dans *Utilisation de Flash*.

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

- Défilement du texte

- Saisie de texte en ligne

La fonctionnalité de saisie de texte en ligne permet à l'utilisateur d'entrer du texte directement dans les champs de texte.

- Prise en charge du moteur de texte de Flash Player 10

Cette version de Flash Lite prend en charge le moteur de texte Flash (flash.text.engine). Ce moteur permet un contrôle avancé de bas niveau des métriques de texte, du formatage et du texte bidirectionnel. Pour plus de détails sur le moteur de texte Flash, voir *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

- Prise en charge de la saisie intuitive

La prise en charge de la saisie intuitive autorise des fonctionnalités telles que la saisie semi-automatique de mots et les listes de choix. Flash Lite 4 prend en charge les principaux moteurs de saisie intuitive (T9, eZiTap/eZiText et iTap, par exemple) sur toute plate-forme. Cette fonction est assurée tant que les implémentations du moteur de texte sont similaires aux API standard fournies par les fournisseurs de moteur de saisie intuitive.

- Prise en charge de XMLSocket

Cette fonction étend la prise en charge du bureau Flash à Flash Lite 4. Elle permet de créer des connexions de données continues à faible latence pour des applications telles que les jeux et les logiciels de dialogue en ligne.

Flash Lite 4 ne prend pas en charge toutes les fonctionnalités de texte de la version de bureau de Flash Player. Flash Lite 4 est soumis aux restrictions suivantes :

- L'anticrênelage personnalisé, une technologie perfectionnée de rendu de police disponible à partir d'Adobe Flash Player 8, 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 de texte de saisie et dynamiques. Le texte est affiché sans les balises HTML. Toutefois, le formatage est uniquement autorisé pour les balises suivantes : `p`, `br`, `sbr`, `font` (avec les attributs `face`, `color` et `size`), `b` et `i`.
- Flash Lite 4 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, voir Adobe ActionScript 3.0 Reference for the Adobe Flash Platform.

Création et formatage de texte

Vous créez et formatez du texte dans Flash Lite 4 de la même façon que dans une application de bureau Flash.

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

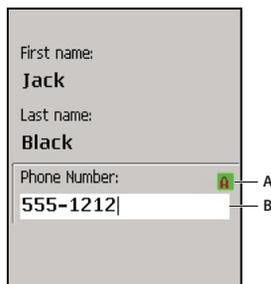
- « Création de texte »
- « Définition des attributs de texte »
- « Manipulation du texte »
- « Contrôle du texte avec ActionScript »

Vous trouverez la liste des fonctions de texte qui ne sont pas prises en charge dans Flash Lite à la section « [A propos du texte dans Flash Lite](#) » à la page 31.

Saisie de texte

Utilisation des champs de texte de saisie

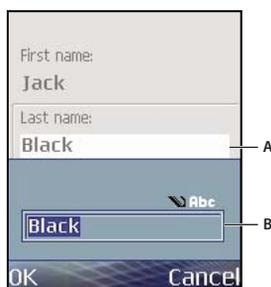
Flash Lite 4 prend en charge la saisie de texte en ligne. Cette fonctionnalité permet aux utilisateurs de modifier directement des champs de texte dans l'application Flash Lite, sans avoir à recourir à des zones de saisie distinctes comme c'était le cas dans les versions précédentes. Par exemple, l'illustration suivante indique comment un champ de texte de saisie en ligne s'affiche sur un périphérique de la gamme Series 60 de Symbian™ exécutant Flash Lite 4 :



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

Flash Lite 4 prend en charge la saisie de texte en ligne des langues latines et asiatiques. Il ne gère cependant pas les langues complexes, notamment les langues s'écrivant de droite à gauche, telles que l'arabe, l'hébreu, l'urdu, le farsi et le yiddish, ni certaines langues asiatiques.

Si un périphérique ne prend pas en charge la saisie de texte en ligne, vous pouvez cependant modifier le contenu des champs de texte de saisie. A cet effet, une boîte de dialogue modale s'affiche par-dessus le contenu Flash Lite. Par exemple, l'illustration suivante donne un exemple de boîte de dialogue de saisie de texte sur un périphérique de la gamme Series 60 de Symbian exécutant Flash Lite 4 :



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

En règle générale, le contenu créé pour les versions antérieures de Flash Lite fonctionne dans Flash Lite 4 sans qu'il soit nécessaire de le modifier. Dans ce cas, la boîte de dialogue modale est remplacée par de simples champs de texte de saisie en ligne. Veillez cependant à recréer le contenu existant de sorte qu'il puisse tirer parti des fonctions propres à Flash Lite 4, telle celle permettant d'exécuter du code ActionScript et des animations en arrière-plan lors de la saisie d'un texte.

Pour plus d'informations sur la boîte de dialogue de saisie de texte, voir « [A propos de la boîte de dialogue de saisie de texte du périphérique \(Flash Lite 4\)](#) » à la page 36.

Saisie de texte en ligne

Flash Lite 4 prend en charge la saisie de texte en ligne, ce qui permet à l'utilisateur d'entrer et de modifier du texte directement dans un champ de 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 saisies de l'utilisateur.

Remarque : dans Adobe Device Central, l'émulateur Flash Lite n'affiche pas le texte en ligne.

Pour que les champs de texte puissent traiter les saisies de l'utilisateur, vous devez les *activer* en leur donnant le focus de sélection. Il suffit ensuite d'appuyer sur la touche de sélection du périphérique ou sur toute touche programmable que l'application définit comme touche d'activation de la saisie de texte. Pour plus d'informations, voir « [Activation des champs de texte de saisie à l'aide d'ActionScript 3.0](#) » à la page 35.

Composants de Flash Lite	Impact
Vidéo	La lecture vidéo se poursuit lorsque le champ de texte est actif. Il est donc possible d'entrer du texte tout en visionnant la vidéo. Dans ce contexte, le terme vidéo désigne les vidéos externes, intégrées et distantes.
Sons	La lecture des sons se poursuit pendant la saisie de texte en ligne.
Réseau	La lecture des flux de données multimédias se poursuit si vous sélectionnez un champ de texte. En outre, il est toujours possible de transférer des données sur le réseau.

Événement `onChange` associé aux champs de texte

Dans les versions antérieures de Flash Lite, l'événement `onChange` est envoyé une seule fois pour la session de saisie de texte, après la désélection du champ de texte. Dans Flash Lite 4, des événements `onChange` sont envoyés pour chaque chaîne de conversion validée et chaque événement de pression de touche. Une chaîne de conversion est validée lorsque l'utilisateur confirme la chaîne. Prenons un exemple.

Supposons que vous saisissez du texte sur un périphérique de la gamme Series 60 de Symbian exécutant Flash Lite 4 en mode d'impression au format texte. Lorsque vous tapez les quelques premières lettres d'une chaîne donnée, la liste des mots complets possibles s'affiche sur le périphérique. En outre, la chaîne est soulignée. Lorsque vous effectuez un choix dans la liste d'options et appuyez sur Entrée, la chaîne est validée. Vous remarquerez que le soulignement disparaît.

Champs de texte de saisie en ligne et navigation

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

Par défaut, Flash Lite dessine un rectangle de focus autour du champ de texte qui a le focus. Ce rectangle peut parfois masquer le pointeur clignotant ou l'indicateur d'insertion du champ de texte activé. Par conséquent, il est recommandé de désactiver le rectangle de focus (en définissant `_focusRect` sur `false`) et d'utiliser un indicateur de focus personnalisé. Pour obtenir un exemple d'application recourant à un indicateur de focus personnalisé, voir 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. Repérez, téléchargez puis décompressez le fichier ZIP correspondant à votre version d'ActionScript. Affichez ensuite le dossier Samples pour accéder à l'exemple.

Activation des champs de texte de saisie à l'aide d'ActionScript 3.0

L'utilisateur peut activer un champ de texte de saisie disposant du focus du clavier en appuyant sur la touche de sélection du périphérique. Une application Flash Lite peut, elle aussi, activer automatiquement un champ de texte de saisie lorsqu'il reçoit le focus, à l'aide de la méthode `setSelection()`. Cette méthode active le champ de texte sélectionné. Si aucun champ n'est sélectionné lorsque la commande s'exécute, aucune action ne se produit.

Pour placer le focus sur un champ de texte et l'activer, vous pouvez utiliser la propriété `focus` de la classe `Stage` et la méthode `setSelection()` de la classe `TextField`. Le fragment de code ci-dessous décrit l'utilisation de ces propriété et méthode.

```
//TextField will have focus after this call
stage.focus = txt;
//TextField will be activated after this call
txt.setSelection(0,0);
```

Pour obtenir un exemple d'application finale utilisant cette technique, voir l'exemple sur la saisie de texte en ligne à l'adresse www.adobe.com/go/learn_ftl_samples_and_tutorials_fr. Repérez, téléchargez puis décompressez le fichier ZIP correspondant à votre version d'ActionScript. Affichez ensuite le dossier Samples pour accéder à l'exemple.

Indicateur du mode de saisie

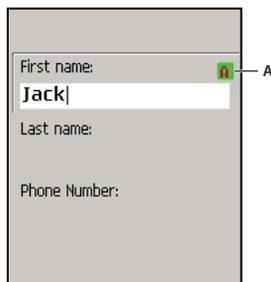
La plupart des périphériques proposent plusieurs modes visant à simplifier les tâches de saisie courantes : saisie de texte intuitive ou manuelle (*triple appui* ou *appuis multiples*) ou saisie numérique uniquement.

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. Dans les autres modes écran, un indicateur de mode de saisie s'affiche sur une barre d'état ou à un autre emplacement sur l'écran. Par exemple, la figure suivante illustre l'indicateur du mode de saisie en mode autre que le mode plein écran avec la version autonome de Flash Lite 4 sur la gamme Series 60 :



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

Lorsque le lecteur s'exécute en mode plein écran, le périphérique affiche 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 4 sur la gamme Series 60 :



A. Indicateur de mode de saisie, mode plein écran

L'indicateur du mode de saisie associé aux applications plein écran illustré plus haut sert uniquement d'exemple pour le lecteur autonome de la gamme Series 60. Le périphérique détermine l'indicateur spécifique, le cas échéant, qui s'affiche.

A propos de la boîte de dialogue de saisie de texte du périphérique (Flash Lite 4)

Pour ouvrir la boîte de dialogue de saisie du périphérique, vous devez d'abord placer le focus sur un champ de texte de saisie, puis appuyer sur la touche de sélection du périphérique.

La boîte de dialogue de saisie de texte est modale, c'est-à-dire que vous ne pouvez pas interagir avec le contenu tant qu'elle possède 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 vous sélectionnez OK (touche programmable Gauche), la boîte de dialogue de saisie se ferme et Flash Lite affecte automatiquement le texte au champ de saisie. Si vous appuyez 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 création 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 de texte de saisie dans une application, voir « [Création d'un exemple d'application de champ de texte](#) » à la page 39.

Spécification des types de champs de texte de saisie

Flash Lite prend en charge les champs de texte de saisie sur une seule ligne, sur plusieurs lignes et de mot de passe. Vous spécifiez le type d'un champ de texte de saisie à l'aide du menu contextuel Type de ligne de l'onglet Propriétés.

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

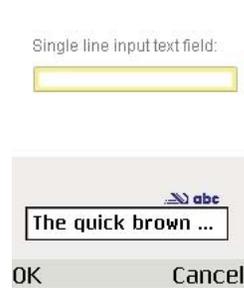
Par exemple, lorsque l'utilisateur modifie un champ de texte de saisie sur une seule ligne, la boîte de dialogue de saisie du périphérique affiche un champ de texte de saisie sur une seule ligne. Le champ de texte de saisie défile horizontalement si l'utilisateur entre plus de caractères qu'il ne peut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

Boîtes de dialogue de saisie du texte (Flash Lite 4)

Dans les versions antérieures de Flash Lite, vous saisissez le texte dans une boîte de dialogue modale distincte plutôt que d'intervenir directement dans la zone de texte à l'écran. L'application hôte contrôle cette boîte de dialogue, pas Flash Lite. Au cours de cette interaction, le lecteur Flash Lite reste en pause jusqu'à ce que l'utilisateur quitte la boîte de dialogue.

Dans Flash Lite 4, l'utilisateur peut entrer du texte directement dans le champ de saisie qui s'affiche à l'écran.

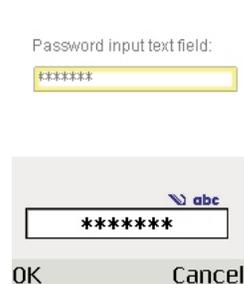
La figure suivante illustre la boîte de dialogue de saisie de texte d'un périphérique pour un champ de texte de saisie sur une seule ligne dans une application Flash Lite 4 :



Lorsque vous modifiez un champ de texte de saisie multiligne, la boîte de dialogue de saisie du périphérique s'agrandit pour afficher la totalité du texte entré, comme illustré ci-dessous :



Lorsque vous modifiez un champ de texte de saisie de mot de passe, la boîte de dialogue de saisie du périphérique affiche chaque caractère entré. Lorsque vous cliquez sur OK, le mot de passe est entièrement remplacé par des astérisques, comme illustré ci-dessous :



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

Cette application simple applique un formatage au texte saisi par l'utilisateur, puis affiche le résultat dans un champ de texte dynamique HTML.

Vous trouverez un exemple d'application finale (textfield_example fla) utilisant cette technique à l'adresse www.adobe.com/go/learn_ftl_samples_and_tutorials_fr. Repérez, téléchargez puis décompressez le fichier ZIP correspondant à votre version d'ActionScript. Affichez ensuite le dossier Samples pour accéder à l'exemple.

- 1 Dans Flash, créez un document mobile et enregistrez-le sous textfield fla.
- 2 Avec l'outil Texte du panneau Outils, créez un champ de texte sur une ligne en haut de la scène.
- 3 Sans désélectionner le champ de texte, dans l'onglet Propriétés, procédez comme suit :
 - a Sélectionnez Texte classique dans le menu contextuel Type de texte.
 - b Sélectionnez Texte de saisie dans le menu contextuel Type de texte.
 - c Sélectionnez Utiliser les polices de périphérique dans le menu contextuel Méthode de rendu des polices.
 - d Entrez **inputTxt** dans le champ de texte Nom d'occurrence.
- 4 Créez un autre champ de texte sous le premier en veillant à ce qu'il soit beaucoup plus long.
- 5 Sans désélectionner le champ de texte, dans l'onglet Propriétés, procédez comme suit :
 - a Sélectionnez Texte classique dans le menu contextuel Type de texte.
 - b Sélectionnez Texte dynamique dans le menu contextuel Type de texte.
 - c Sélectionnez Utiliser les polices de périphérique dans le menu contextuel Méthode de rendu des polices.
 - d Cliquez sur l'icône Rendre le texte au format HTML pour la sélectionner.
 - e Entrez **messageTxt** dans le champ 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 :

```
messageTxt.htmlText = "";
inputTxt.addEventListener(TextEvent.TEXT_INPUT, onTextInput);
function onTextInput(e:TextEvent) {
    messageTxt.htmlText += "You Entered <i>" + e.text + "</i>\n";
}
```

Il est possible d'écouter l'événement `TextInput` du champ de texte de saisie pour toute saisie effectuée dans celui-ci. Dans cet exemple, l'événement `TextInput` permet de recevoir le texte entré et de l'afficher dans le champ de texte dynamique `messageTxt`.

- 8 Enregistrez vos modifications et démarrez l'application (Contrôle > Tester l'animation).
- 9 Pour utiliser l'application, appuyez sur la touche de sélection de l'émulateur pour ouvrir la boîte de dialogue de saisie de texte, puis tapez du 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 de texte `messageTxt` en italique.

Moteur de texte Flash

Le moteur de texte Flash (flash.text.engine) permet de créer et de contrôler des éléments de texte simples, bien qu'il ait été conçu pour créer des composants de gestion de texte. Il se compose de classes servant à créer, formater et contrôler le texte. L'exemple de code ci-dessous illustre l'utilisation du moteur de texte Flash :

```
import flash.text.engine.*;
var str:String="Hello World";
// Fontdescription represents properties necessary to describe a font, for example,
// font name. This example uses the embedded Arial font.
    var fontDescription:FontDescription = new FontDescription();
    fontDescription.fontName = "Arial"
    fontDescription.fontLookup = flash.text.engine.FontLookup.EMBEDDED_CFF;
// The Element format provides formatting information such as color, alpha, size, and
// so on. In this example, the color of text is green and the text size is 18.
    var format:ElementFormat=new ElementFormat(fontDescription);
    format.color = 0x00ff00;
    format.fontSize = 18;
// TextElement represents a formatted string, and is set to the content property
// of textBlock to create a block of text.
    var textElement:TextElement=new TextElement(str,format);
// textBlock represents a paragraph. It is a container for text and graphic elements.
    var textBlock:TextBlock = new TextBlock();
    textBlock.content=textElement;
// Creates a text line with a maximum of 500 characters.
    var textLine:TextLine=textBlock.createTextLine(null,500);
    textLine.y=20;
// Adds the text line to the display object for rendering to make the line display on
// the screen.
    addChild(textLine);
```

Pour plus de détails sur le fonctionnement des classes de la fonction `flash.text.engine`, voir *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Rendu des polices

Méthodes de rendu des polices dans Flash Lite

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

Utilisation des polices de périphérique Vous pouvez appliquer une police à un champ de texte si vous savez qu'elle est présente sur le périphérique. Vous pouvez également spécifier l'une des trois polices de périphérique génériques (*_sans*, *_serif* ou *_typewriter*), accessibles dans le menu contextuel Police. Si vous sélectionnez une police générique, Flash Lite essaie de la faire correspondre à une police présente sur le périphérique lors de l'exécution. La police *_sans*, par exemple, est mappée sur une police *sans serif*, le cas échéant.

Rendu de la police sous forme de bitmap Flash Lite effectue le rendu 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. (Voir « [Incorporation des polices vectorielles dans les fichiers SWF](#) » à la page 42.)

Rendu de la police sous forme de vecteurs anticrênelés Flash Lite rend du texte anticrênelé en utilisant des représentations vectorielles des polices vectorielles incorporées au fichier SWF publié. (Voir « [Incorporation des polices vectorielles dans les fichiers SWF](#) » à la page 42.)

Le menu contextuel Méthode de rendu des polices de l'onglet Propriétés permet de sélectionner une méthode de rendu des polices pour un champ de texte. Ce menu contextuel contient cinq options de rendu ; cependant, trois seulement sont disponibles pour les développeurs Flash Lite. Les deux autres méthodes (Anticrênelage pour la lisibilité et Anticrênelage 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 anticrênelé, bitmap ou de périphérique :

- Si vous utilisez des polices incorporées avec des champs de texte de saisie ou dynamique, incorporez uniquement les vecteurs des caractères à afficher. La réduction du nombre de contours permet d'obtenir un fichier moins volumineux. Par exemple, si vous utilisez un champ de 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 de texte sur le périphérique seront peut-être différents sur l'émulateur.
- Supposons que vous appliquiez un des types de polices de périphérique génériques (*_sans*, *_serif* ou *_typewriter*). Dans ce cas, Flash Lite recherche une police similaire sur le périphérique pour afficher le texte. Le nombre de polices et de styles de police est toutefois généralement moins élevé sur un périphérique mobile que sur un ordinateur de bureau. Il se peut donc qu'une police telle que *_sans* ne soit pas mappée sur une police sans serif. Veillez à tester l'application sur chaque périphérique cible pour déterminer les paramètres de police applicables.

Le texte avec anticrênelage 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 anticrênelés peut entraîner une baisse des performances. Pour optimiser celles-ci, essayez de réduire la qualité de rendu du lecteur Flash Lite pendant l'animation. Lorsqu'elle est terminée, rétablissez une qualité de rendu plus élevée.

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

- 1 Sélectionnez un champ de texte sur la scène.
- 2 Dans l'onglet Propriétés, sélectionnez l'une des options suivantes dans le menu contextuel 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 anticrênelage] 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. (Voir « [Incorporation des polices vectorielles dans les fichiers SWF](#) » à la page 42.)
 - Sélectionnez Anticrênelage pour l'animation afin que Flash Lite procède à l'anticrênelage de la police du champ de texte en fonction du paramètre de qualité de rendu actuel. Cette option nécessite que Flash intègre les polices vectorielles au fichier SWF publié.

Qualité de rendu et texte anticrênelé de Flash Lite

La propriété `quality` de l'objet Stage permet de contrôler la façon dont Flash Lite effectue le rendu des graphiques vectoriels et des bitmaps. Les valeurs possibles sont "LOW", "MEDIUM", "HIGH" et "BEST". La valeur par défaut est "HIGH".

Flash Lite rend le texte anticrênelé 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 HIGH. La figure suivante illustre la qualité de rendu haute (HIGH) pour un champ de texte anticrênelé (Arial, 24 points) :

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

Vous pouvez optimiser les performances d'animation et la cadence. Examinez l'exemple suivant : Pendant une animation ou une séquence d'interpolation intensive, il est possible de réduire temporairement la qualité de rendu. Vous rétablissez ensuite le paramètre précédent, à l'issue de l'animation.

Incorporation des polices vectorielles dans les fichiers SWF

Pour effectuer le rendu d'une police d'un champ de texte, Flash Lite utilise les polices disponibles sur le périphérique ou les polices vectorielles incorporées dans le fichier SWF publié. Lorsque vous incorporez les polices vectorielles dans le fichier SWF, vous avez la garantie que la police du champ de texte se présente de la même façon sur toutes les plateformes cible. En revanche, la taille du fichier augmente de manière significative. Flash Lite oblige les polices vectorielles à rendre soit du texte bitmap (sans anticrênelage) soit du texte anticrênelé.

Pour les champs de texte statique qui utilisent les méthodes de rendu des polices anticrênelées ou bitmap, Flash intègre automatiquement les polices vectorielles requises pour afficher le contenu du champ de texte. Examinez l'exemple suivant : Un champ de texte statique contient le mot *Submit*. Dans ce cas, Flash incorpore automatiquement les polices vectorielles requises pour afficher les six caractères (*S, u, b, m, i, t*). Le contenu d'un champ de texte statique ne pouvant pas être modifié, il est uniquement nécessaire d'incorporer dans le fichier SWF les polices vectorielles de ces caractères.

Vous spécifiez les caractères pour lesquels vous souhaitez incorporer des polices vectorielles dans le fichier SWF publié. Vous devez procéder ainsi pour les champs de texte dynamique et de saisie utilisant les méthodes d'anticrênelage ou de rendu des polices sous forme de bitmaps. Le contenu de ces types de champs de 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 de texte dynamique ou de saisie

- 1 Sélectionnez le champ de texte dynamique ou de saisie sur la scène.
- 2 Dans l'onglet Propriétés, sélectionnez Bitmap (sans anticrênelage) ou Anticrênelage pour l'animation dans le menu contextuel Méthode de rendu des polices.
- 3 Cliquez sur le bouton Incorporer pour ouvrir la boîte de dialogue Incorporation de polices.
- 4 Sélectionnez les caractères que vous souhaitez incorporer dans la liste. Vous pouvez également taper ces caractères dans le champ de texte Inclure également les caractères suivants.
- 5 Cliquez sur OK.

Défilement du texte

Création de texte défilant

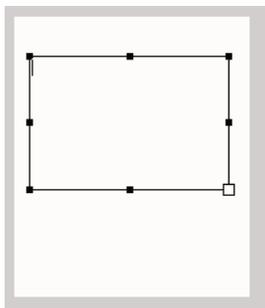
Flash Lite 4 prend en charge les propriétés `TextField.scrollV`, `TextField.scrollH`, `TextField.maxscrollV` et `TextField.maxscrollH`, qui permettent de créer des champs de texte défilants. La propriété `scrollV` désigne la première ligne visible d'un bloc de texte ; vous pouvez lire et définir sa valeur. Par exemple, le code suivant fait défiler de cinq lignes vers le bas le champ de texte dont le nom variable est `story_text` :

```
story_text.scrollV += 5;
```

La propriété `maxscrollV` spécifie la première ligne visible d'un bloc de texte dont la dernière ligne est masquée. Cette propriété est en lecture seule. Vous pouvez comparer la propriété `maxscrollV` d'un champ de texte à sa propriété `scrollV` pour déterminer le niveau de défilement atteint par l'utilisateur dans le champ. Cela peut s'avérer utile si vous souhaitez créer une barre de défilement vertical indiquant la position de défilement actuelle de l'utilisateur par rapport à la position de défilement maximale. Le comportement des propriétés `TextField.scrollH` et `TextField.maxscrollH` est identique.

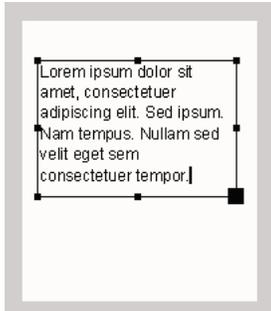
Création d'un champ de texte défilant et contrôle à l'aide du code ActionScript

- 1 Dans Flash, créez un document mobile.
- 2 En utilisant l'outil Texte, faites glisser un champ de texte d'environ la taille indiquée dans l'illustration suivante sur la scène :



- 3 Dans l'onglet Propriétés, sélectionnez Texte classique dans le menu contextuel Moteur de texte.
- 4 Dans l'onglet Propriétés, choisissez Texte dynamique dans le menu contextuel Type de texte.
- 5 Dans l'onglet Propriétés, sélectionnez Utiliser les polices de périphérique dans le menu contextuel Méthode de rendu des polices.
- 6 Dans l'onglet Propriétés, sélectionnez Multiligne dans le menu contextuel Type de ligne.
- 7 Sélectionnez Texte > Défilant pour faire défiler le champ de texte.
- 8 Dans l'onglet Propriétés, tapez **story** dans le champ de texte Nom d'occurrence.

- 9 Double-cliquez dans le champ de texte et tapez un texte suffisamment long pour qu'une ou plusieurs lignes dépassent de sa bordure inférieure.



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

```
stage.focus = story;
story.addEventListener(KeyboardEvent.KEY_DOWN, onKeyEvent);

function onKeyEvent(e:KeyboardEvent): void {
    if (e.keyCode == 38) {
        story.scrollV++;
    }
    else if (e.keyCode == 40) {
        story.scrollV--;
    }
}
```

- 11 Sélectionnez Contrôle > Tester l'animation > dans Device Central pour tester l'application. Déplacez le texte vers le haut ou le bas à l'aide de la flèche vers le bas ou vers le haut.

Chapitre 6 : Utilisation du son

Adobe Flash Lite prend en charge deux types de son : le son de périphérique et le son Flash standard (*natif*). Les sons de périphérique sont joués directement par le périphérique, et non par le lecteur Flash Lite. Les sons natifs, par opposition, sont lus directement par le lecteur Flash Lite, tout comme dans la version de bureau d'Adobe Flash Player.

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, notamment MIDI, MFi ou MP3. Pour incorporer les sons de périphérique au contenu Flash Lite, vous pouvez les inclure dans le fichier SWF publié. Vous pouvez aussi charger des fichiers son externes sur le réseau ou à partir du système de fichiers local du périphérique. Hormis quelques exceptions, vous contrôlez les sons de périphérique à l'aide de la classe `Sound` d'ActionScript, tout comme dans la version de bureau de Flash Player.

Lecture de sons de périphérique externes

Flash Lite permet de charger et de lire des fichiers son externes. Pour lire des sons de périphérique externes, appliquez la méthode `load()` de la classe `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 ce qui se passe dans l'application de bureau, les sons chargés en externe dans le lecteur Flash Lite sont systématiquement traités comme des sons d'événements. Les sons de périphérique ne sont donc pas diffusés en continu (au fur et à mesure de leur téléchargement). Pour que vous puissiez les lire, les données audio doivent être téléchargées dans leur intégralité. Pour les mêmes raisons, vous appelez la méthode `play()` de la classe `Sound` pour lire le son à l'issue de son chargement (voir l'exemple de code ci-après).
- Le lecteur Flash Lite ne lit pas de façon native les fichiers MP3 chargés en externe. Si l'application charge un fichier MP3 externe, Flash Lite transmet les données audio au périphérique, comme s'il s'agissait de tout autre fichier son chargé en externe. C'est le périphérique qui décode et lit le son.

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

```
//The following URLRequest assumes that the mp3 file is in the same directory as
//the SWF file when published. You can use a full URL for a remote file.
var MySound:Sound = new Sound(Areq);
MySound.addEventListener(Event.COMPLETE, onSoundLoaded);
var Areq:URLRequest = new URLRequest("click.mp3");
MySound.load(Areq);
function onSoundLoaded(event:Event):void
{
    var mysound:Sound = event.target as Sound;
    mysound.play();
}
```

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 création de Flash, quel que soit son format. Les sons natifs de Flash Lite peuvent être lus en tant que sons d'événements ou sons synchronisés.

Le flux de travail général d'utilisation des sons natifs dans les applications Flash Lite est identique à celui des applications 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 des fichiers MP3 externes dans l'application.
- Flash Lite ne prend pas en charge l'option de compression audio Discours.

Utilisation de la fonctionnalité d'échantillonnage sur 8 kHz

Par défaut, l'outil de création 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 Symbian prennent uniquement en charge les taux de 8 et 16 kHz. Dans les versions précédentes de Flash Lite, le lecteur ré-échantillonnait les données audio sur 8 kHz lors de l'exécution, avant d'appeler les API de son natif du périphérique pour lire le son. Ce nouvel échantillonnage prenait plus longtemps et consommait davantage de mémoire.

Dans Adobe Flash Professional CS5, vous pouvez échantillonner de nouveau les données audio natives sur 8 kHz à l'aide de l'outil de création pendant le processus de publication du fichier SWF. Vous appliquez ce paramètre à tous les sons d'événements et sons en flux continu de l'application.

La procédure suivante explique comment définir cette option globalement pour tous les sons d'événements et sons en flux continu à l'aide de la boîte de dialogue Paramètres de publication.

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 du taux d'échantillonnage 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 le taux d'échantillonnage 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 le menu contextuel Compression.

le taux d'échantillonnage 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 le taux d'échantillonnage 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 le menu contextuel Compression.

Remarque : le taux d'échantillonnage 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

Flash Lite 4 décode les formats vidéo des codecs *On2* et *Sorenson Spark* et en effectue le rendu. Cette version de Flash Lite prend en charge la diffusion vidéo en continu par le biais des protocoles RTMP, RTMPE, RTMPT et RTMPTE. La fonction de diffusion en continu à vitesse de transmission variable permet d'adapter la vidéo aux variations des conditions réseau sans avoir à mettre la lecture en pause.

Utilisation des fichiers FLV

A propos de la prise en charge de fichiers FLV dans Flash Lite 4

Flash Lite 4 prend en charge les fichiers FLV par le biais des versions des codecs *On2* et *Sorenson* optimisés pour les périphériques portables. Les fichiers FLV sont rendus par Flash Lite 4 et non par le périphérique. Si un périphérique prend en charge Flash Lite 4, 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 Device Profiles dans Adobe Device Central pour vérifier si les fichiers FLV sont pris en charge par un périphérique spécifique.

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

- Intégrer des fichiers FLV dans l'application.
- Lire des fichiers FLV externes à partir d'un disque dur local ou d'un serveur Web via HTTP.
- Lire les fichiers FLV externes en provenance de Flash Media Server sur RTMP, RTMPE, RTMPT ou RTMPTE.

Vous pouvez convertir les fichiers vidéo (QuickTime ou Windows Media, par exemple) en fichiers FLV. Procédez comme suit :

- Utilisez un encodeur (tel qu'Adobe Media Encoder CS5 ou Sorenson™ Squeeze).
- Utilisez le module externe 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 4 :

- La classe Camera ou l'enregistrement de vidéos.
- Les connexions RTMPS et RTMFP.
- La vidéo avec canal alpha.
- Le composant FLVPlayback. Pour plus d'informations, voir « [A propos des composants de Flash Lite](#) » à la page 1.

Intégration de 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 création 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 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.

1 Dans Flash, cliquez sur Flash Lite 4 sous Créer. Un nouveau fichier s'affiche.

2 Sélectionnez Fichier > Importer > Importer de la vidéo.

L'Assistant Importer de la vidéo apparaît.

3 Sur l'écran Sélectionner le fichier vidéo, sélectionnez l'option d'importation d'un fichier sur l'ordinateur, puis cliquez sur Parcourir.

4 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 contextuel Types de fichier (Windows) ou Tous les fichiers à partir du menu contextuel Activer (Macintosh).

5 Cliquez sur Ouvrir. Sélectionnez l'option Incorporer le fichier FLV dans SWF et le diffuser dans le scénario, puis cliquez sur Suivant.

6 Sur l'écran Incorporation, sélectionnez Vidéo intégrée comme type de symbole, puis cliquez sur Suivant.

7 Sur l'écran Terminer l'importation de vidéos, 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'onglet Propriétés.

Vous pouvez utiliser le nom de l'occurrence pour contrôler l'objet vidéo avec les propriétés de la classe Video d'ActionScript.

Lecture de fichiers FLV externes dans une application

Un fichier externe FLV n'est pas intégré dans un fichier SWF. Il 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 4.

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.

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 Flash, cliquez avec le bouton droit de la souris sur la bibliothèque et sélectionnez Nouvelle Vidéo dans les options qui s'affichent pour créer un symbole de vidéo. Assurez-vous que le type de vidéo est « Vidéo (contrôlée par ActionScript) » puis cliquez sur OK.
- 2 Faites glisser le symbole de vidéo sur la scène et entrez un nom d'occurrence dans l'onglet Propriétés. Cet exemple utilise le nom d'occurrence `MyVideo`.
- 3 Dans le scénario, sélectionnez la première image, puis ouvrez le panneau Actions.
- 4 Saisissez le code suivant :

```
var MyNC:NetConnection = new NetConnection();
MyNC.connect(null);

var MyNS:NetStream = new NetStream(MyNC);
MyNS.play("http://www.helpexamples.com/flash/video/clouds.flv");

MyVideo.attachNetStream(MyNS);

//The clouds.flv video has metadata that Adobe is not using, so create
//an error handler to ignore the message generated by the runtime
//about the metadata.
MyNS.addEventListener(AsyncErrorEvent.ASYNC_ERROR, asyncErrorHandler);
function asyncErrorHandler(event:AsyncErrorEvent):void
{
    //Ignore metadata error message.
}
```

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 MIME soit déjà enregistré sur le serveur Web. Référez-vous à la documentation du serveur Web pour de plus amples informations.

Contrôle des 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()`.

Voir *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform* pour plus de détails sur ces classes et pour obtenir des exemples de code illustrant leur utilisation. En outre, Flash Lite 4 prend en charge la totalité des méthodes, propriétés et événements décrits dans *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*. La seule exception est la propriété `MovieClipLoader.checkPolicyFile`.

RTMP

Le canal de données RTMP (Real Time Messaging Protocol) permet de diffuser des données vidéo en continu entre un client (le lecteur Flash Lite) et le serveur (Flash Media Server). Le lecteur Flash Lite et le serveur échangent ces données sous la forme d'objets ActionScript.

Pour sérialiser et désérialiser les objets ActionScript pendant qu'ils transitent par le réseau, vous utilisez le format binaire compact AMF (Action Message Format). La première version de ce format, AMF 0, est utilisée dans ActionScript 2.0. ActionScript 3.0 utilise le format AMF3. Outre les occurrences d'objet, le format AM 3 envoie les caractéristiques d'objet et les chaînes. Il prend également en charge certains des types de données introduits dans ActionScript 3.0.

Pour plus de détails sur le format AMF 3, voir [AMF 3 specification](#). Par défaut, RTMP utilise le port 1935.

L'exemple suivant utilise l'occurrence de `MyVideo` pour charger une vidéo à partir de Flash Media Server via RTMP :

```
var MyVideo:Video = new Video();
addChild(MyVideo);
var nc:NetConnection = new NetConnection();
var ns:NetStream;
// Add event listeners for the connection status and errors.
nc.addEventListener(NetStatusEvent.NET_STATUS, netStatusHandler)
nc.addEventListener(AsyncErrorEvent.ASYNC_ERROR, asyncErrorHandler);
// Provide the video location.
nc.connect("rtmp://fmsexamples.adobe.com/vod");

function netStatusHandler(event:NetStatusEvent):void {
    trace(event.info.code);
    if(event.info.code == "NetConnection.Connect.Success"){
        // Create a NetStream object and pass it the NetConnection object.
        ns = new NetStream(nc);
        // Assign the ns.client property to the object that is handling onMetaData.
        ns.client = this;
        // Attach the NetStream object to the video object on Stage.
        MyVideo.attachNetStream(ns);
        // Play the file "sample.flv". Do not include ".flv" when
        // playing the file on Flash Media Server.
        ns.play("sample");
    }
}

function asyncErrorHandler(event:AsyncErrorEvent):void {
    // ignore error
}

function onMetaData(item:Object):void {
    // See the metadata in the trace output.
    var key:String;
    for (key in item){
        trace(key + ": " + item[key]);
    }
}
```

Améliorations

De nouvelles fonctions liées à la vidéo ont été mises à jour dans cette version de Flash Lite 4.

RTMPE

Pour protéger le transport des flux de données vidéo entre Flash Media Server et le lecteur Flash Lite, vous pouvez activer le chiffrement. Vous pouvez chiffrer les données par le biais de RTMPE.

Flash Media Server chiffre la totalité du contenu à l'exécution ; il est donc inutile de chiffrer le fichier source. RTMPE utilise les primitives cryptographiques standard reposant sur *l'échange de clés Diffie-Hellman* et *HMACSHA256*. Lors du transport des données, RTMPE génère une paire de clés RC4. Une clé chiffre les données envoyées par le serveur, tandis que l'autre chiffre les données qui lui sont envoyées.

RTMPE empêche les applications tierces d'écouter les échanges de données entre le client et le serveur. Pour chiffrer un flux de données, vous faites appel à la syntaxe suivante :

```
nc.connect("rtmpe://localhost/vod/");
```

Si vous utilisez le paramètre source du composant FLVPlayback, le chemin est :

```
rtmpe://localhost/vod/flvFileName
```

Lorsque vous ajoutez la lettre « e », Flash Media Server chiffre le flux de données en temps réel. RTMPE chiffre les données pendant leur transport. Vous n'avez pas besoin d'une clé pour les déchiffrer.

Seul le lecteur Flash Lite peut mettre fin à RTMPE. Si une application utilise RTMPE sans spécifier de port, Flash Player examine les ports dans l'ordre suivant : 1935 (RTMPE), 443 (RTMPE) et 80 (RTMPE). RTMPE est activé par défaut. Pour le désactiver, ouvrez le fichier `fms.ini`, qui réside dans le dossier `C:\Program Files\Adobe\Flash Media Server 3\conf`, et définissez le paramètre `ADAPTOR.RTMPE_ENABLED` sur `off`. Redémarrez ensuite le serveur.

RTMPT

De nombreuses organisations possèdent des pare-feu et des serveurs proxy qui interdisent les connexions socket RTMP directes. Elles étaient donc dans l'impossibilité d'utiliser les applications Flash Media Server.

Pour parer à ce problème, il est possible d'envelopper les données RTMP à envoyer dans des requêtes HTTP. Cette méthode s'appelle le tunnelage. Dans le lecteur Flash Lite, vous pouvez charger le tunnelage explicitement ou automatiquement.

Dans cette version de Flash Lite, `netConnection.connect()` a été étendu pour prendre en charge un nouveau protocole, RTMPT (RTMP Tunneling). RTMPT signifie que RTMP est tunnelé via HTTP ; les données RTMP sont encapsulées sous forme de données HTTP. RTMPT utilise le port 80 par défaut. Pour appeler RTMPT explicitement, utilisez la syntaxe suivante :

```
nc.connect("rtmpt://myserver/myapp"); // defaults to port 80
```

Vous pouvez également spécifier le port :

```
nc.connect("rtmpt://myserver:port/myapp");
```

Nouvelles tentatives automatiques

La fonction de « nouvelles tentatives automatiques » du lecteur Flash Lite a été étendue pour inclure un cas de figure de connexion particulier. Cette fonction est uniquement activée lorsqu'une connexion RTMP associée à un port par défaut est spécifiée. Examinez la syntaxe suivante :

```
nc.connect("rtmpt://myserver/myapp");
```

Cette commande essaie les ports suivants dans l'ordre :

```
nc.connect("rtmp://myserver/myapp"); // Defaults to port 1935.  
nc.connect("rtmp://myserver:443/myapp");  
nc.connect("rtmp://myserver:80/myapp");
```

RTMPTE

Flash Media Server 3 introduit RTMPTE, qui autorise le chiffrement RTMPT. Le port par défaut est 80. RTMPTE sécurise les données même si vous les transférez de votre serveur Flash Media Server vers des serveurs tiers par tunnelage.

Dans cette version de Flash Lite, `netConnection.connect()` a été étendu pour prendre en charge un nouveau protocole RTMPTE (RTMP Tunneling Encrypt). Cela signifie que les données RTMP chiffrées sont encapsulées en tant que données HTTP.

Pour chiffrer un flux de données tunnelées, vous faites appel à la syntaxe suivante :

```
nc.connect("rtmpte://localhost/vod/");
```

Si vous utilisez le paramètre source du composant `FLVPlayback`, le chemin est :

```
rtmpte://localhost/vod/flvFileName
```

Dans le protocole RTMPTE, vous ordonnez à Flash Media Server de chiffrer le flux de données de tunnelage en temps réel. RTMPTE chiffre les données pendant leur transport. Vous n'avez pas besoin d'une clé pour les déchiffrer.

Diffusion en continu à vitesse de transmission variable

Lors de la diffusion en continu à partir de Flash Media Server, Flash Lite 4 prend en charge le basculement dynamique entre différentes vitesses de transmission. L'application peut ainsi s'adapter aux modifications dynamiques des conditions réseau tout en diffusant en continu le contenu enregistré.

Les éditeurs de contenu en ligne font face à d'énormes défis lorsqu'ils essaient de diffuser en continu des vidéos plus longues et de qualité supérieure (notamment la vidéo HD). La fluctuation des connexions réseau des utilisateurs leur pose en effet problème. Un problème qui peut se multiplier lorsque les conditions suivantes s'appliquent à une seule connexion :

- Utilisateurs exécutant plusieurs tâches simultanément
- Fluctuation des réseaux sans fil
- Utilisateurs simultanés

En conséquence, la bande passante disponible n'est jamais constante. Si le réseau ne prend pas en charge la bande passante nécessaire, l'utilisateur finit donc par effectuer une nouvelle mise en mémoire tampon et attendre que la vidéo s'exécute.

La diffusion en continu dynamique permet de résoudre ce problème. Elle détecte les fluctuations de bande passante et bascule d'un flux transmis à une certaine vitesse à un autre, dont la vitesse de transmission est différente, afin de faire coïncider le flux de contenu à la bande passante de l'utilisateur.

Certains utilisateurs, au contraire, lancent le flux avec une faible bande passante et libèrent de la bande passante lorsque la lecture de la vidéo commence. Dans ce cas de figure, la diffusion en continu dynamique permet d'optimiser la qualité de la vidéo, ce qui améliore encore plus l'expérience de l'utilisateur.

Un autre problème susceptible d'entraver la lecture (notamment dans le cas des vidéos HD de grande taille et de la lecture en plein écran) concerne les performances limitées du matériel de l'utilisateur. Si l'unité centrale ne peut pas décoder suffisamment rapidement le flux de données vidéo, des images sont omises. L'affichage de la vidéo est donc moins fluide. Dans ce cas, l'utilisation d'un fichier vidéo de qualité inférieure sollicite moins l'unité centrale, qui peut effectuer un décodage synchronisé et assurer des performances acceptables.

Il était auparavant nécessaire d'exécuter des routines de détection de la bande passante initiales ou fréquentes. Ces tests étaient néanmoins onéreux et n'étaient pas toujours suffisamment précis en raison des fluctuations et des changements affectant normalement la bande passante. Grâce aux fonctionnalités de diffusion en continu dynamique et aux informations de qualité de service (QoS) disponibles, ces tests n'ont maintenant plus grand intérêt.

La diffusion en continu dynamique constitue une solution optimale aux problèmes de fluctuation du réseau et de surcharge de l'unité centrale. En surveillant continuellement les principales métriques QoS sur le client, la diffusion en continu dynamique peut identifier avec précision le moment où il est nécessaire de changer de flux pour passer à une qualité supérieure ou inférieure.

Chapitre 8 : Utilisation des API de périphérique

Vous pouvez obtenir la position géographique et l'accélération directionnelle d'un périphérique à l'aide de Flash Lite 4. C'est dans ce but que Flash Lite 4 prend en charge deux API de périphérique.

Classe Geolocation

La classe Geolocation vous permet de vous inscrire pour obtenir les mises à jour émises par le capteur de localisation embarqué d'un périphérique. Si le périphérique hébergeant Flash Lite prend en charge cette fonction, vous pouvez obtenir sa position géographique actuelle, sous la forme de coordonnées longitudinales et latitudinales.

Vous pouvez recevoir sur votre périphérique les mises à jour correspondant aux changements de position. Cette fonction permet d'obtenir les informations suivantes :

- **Latitude** : indique la latitude de la position du périphérique. Elle est comprise dans la plage suivante :
 $-90^\circ \leq \text{Lat} \leq +90^\circ$
Les valeurs négatives représentent le sud et les valeurs positives le nord.
- **Longitude** : indique la longitude de la position du périphérique. Elle est comprise dans la plage suivante :
 $-180^\circ \leq \text{Long} < +180^\circ$
Les valeurs négatives représentent l'ouest et les valeurs positives l'est.
- **Altitude** : indique l'altitude de la position du périphérique. L'altitude est mesurée en mètres (m).
- **Précision horizontale** : indique l'écart de déviation par rapport à la coordonnée longitudinale exacte et renvoie la valeur en mètres. Exemple : 50 m de 5° O.
- **Précision verticale** : indique l'écart de déviation par rapport à la coordonnée latitudinale exacte et renvoie la valeur en mètres. Exemple : 10 m de 20 N.
- **Direction** : indique la direction du mouvement par rapport au nord géographique et renvoie la valeur en degrés. Exemple : 10° signifie que le périphérique se déplace selon un angle de 10° par rapport au nord géographique.
- **Vitesse** : indique la vitesse de déplacement du périphérique et renvoie la valeur en m/s. Exemple : 0,010 m/s.
- **Horodatage** : indique la durée qui s'est écoulée depuis le démarrage de l'application. Cette valeur est mesurée en millisecondes. Elle est dynamique et varie en fonction des changements de position dans un intervalle de temps spécifié.

Distribution d'événements

La classe Geolocation distribue des événements en fonction de toute activité détectée par le capteur de localisation du périphérique.

Vous pouvez spécifier la fréquence des mises à jour de géolocalisation à l'aide de la méthode `setRequestedUpdateInterval()`. Cette valeur est exprimée en millisecondes.

Remarque : la fréquence de mise à jour est présentée à titre informatif seulement pour économiser la batterie. La fréquence réelle peut être supérieure ou inférieure à cette valeur.

Si vous omettez la méthode `setRequestedUpdateInterval()`, le périphérique hôte applique sa fréquence de mise à jour par défaut.

Utilisation de la géolocalisation

Le code suivant illustre le fonctionnement de la géolocalisation. Il vérifie d'abord que le périphérique prend en charge cette fonction. Si c'est le cas, un écouteur d'événements est enregistré pour écouter les changements de position après le démarrage de l'application. Chaque fois que l'écouteur détecte un changement, la fonction `gpsListener` est appelée et renvoie les nouvelles coordonnées. La fréquence des mises à jour d'accélération est définie sur 5 secondes. Pour plus d'informations, voir « [Classe Geolocation](#) » à la page 55.

```
import flash.sensors.*;
if (Geolocation.isSupported)
{
    var gps:Geolocation = new Geolocation();
    function gpsListener(evt:GeolocationEvent):void
    {
        trace("Latitude: " + evt.latitude.toString());
        trace("Longitude: " + evt.longitude.toString());
        trace("Altitude: " + evt.altitude.toString());
        trace("Horizontal Accuracy: " + evt.horizontalAccuracy.toString());
        trace("Vertical Accuracy: " + evt.verticalAccuracy.toString());
        trace("Speed: " + evt.speed.toString());
        trace("Heading: " + evt.heading.toString());
        trace("Timestamp: " + evt.timestamp.toString());
    }
    gps.addEventListener(GeolocationEvent.UPDATE, gpsListener);
    gps.setRequestedUpdateInterval(5000);
    gps.removeEventListener(GeolocationEvent.UPDATE, gpsListener);
}
else
{
    trace("GPS not supported.");
}
```

Pour plus de détails sur les classes `Geolocation` et `GeolocationEvent`, voir *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Classe Accelerometer

La classe `Accelerometer` permet d'obtenir les données d'accélération d'un périphérique à partir du capteur de l'accéléromètre installé sur celui-ci. Ces données peuvent représenter :

- la position du périphérique dans l'espace le long d'un axe tridimensionnel ;
- le mouvement du périphérique le long d'un axe tridimensionnel.

L'axe tridimensionnel se compose des axes X, Y et Z. Lors du déplacement du périphérique, le capteur détecte le mouvement et renvoie les coordonnées d'accélération correspondantes. Pour plus d'informations sur l'orientation des axes tridimensionnels, voir « [Orientation des axes](#) » à la page 57.

Vous pouvez aussi obtenir la valeur d'horodatage des changements d'accélération. Elle indique la durée qui s'est écoulée depuis le démarrage de l'application. Cette valeur est mesurée en millisecondes. Elle est dynamique et varie en fonction des changements d'accélération dans un intervalle de temps spécifié.

Distribution d'événements

La classe `Accelerometer` distribue des événements en fonction de toute activité détectée par le capteur de mouvement du périphérique.

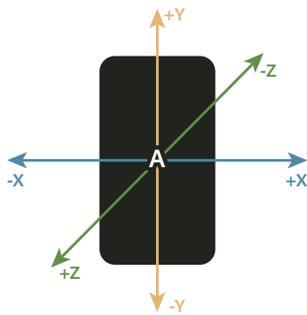
Vous pouvez spécifier la fréquence des mises à jour d'accélération à l'aide de la méthode `setRequestedUpdateInterval()`. Cette valeur est exprimée en millisecondes.

Remarque : la fréquence de mise à jour est présentée à titre informatif seulement pour économiser la batterie. La fréquence réelle peut être supérieure ou inférieure à cette valeur.

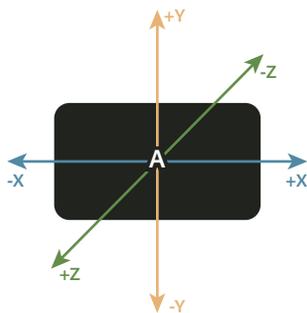
Si vous omettez la méthode `setRequestedUpdateInterval()`, le périphérique hôte applique sa fréquence de mise à jour par défaut.

Orientation des axes

Les axes tridimensionnels sont fondés sur l'orientation du contenu affiché sur le périphérique, comme illustré ci-après.



A. Axe orienté en mode portrait



A. Axe orienté en mode paysage

L'accélération le long des axes est définie comme suit :

- Accélération le long de l'axe X : la valeur est basée sur la valeur de G (9,81 m/s²). L'axe X est orienté de gauche à droite sur le périphérique, celui-ci étant parallèle au sol. L'accélération est positive si le périphérique se déplace vers la droite.

- Accélération le long de l'axe Y : la valeur est basée sur la valeur de G (9,81 m/s²). L'axe Y est orienté de bas en haut sur le périphérique, celui-ci étant parallèle au sol. L'accélération est positive si le périphérique se déplace vers l'avant.
- Accélération le long de l'axe Z : la valeur est basée sur la valeur de G (9,81 m/s²). L'axe Z est perpendiculaire à la face du périphérique, celui-ci étant parallèle au sol. L'accélération est positive si vous éloignez le périphérique du sol.

Utilisation de la classe Accelerometer

Le code suivant illustre le fonctionnement de l'accéléromètre. Il vérifie d'abord que le périphérique prend en charge un accéléromètre. Si c'est le cas, un écouteur d'événements est enregistré pour écouter les changements d'accélération après le démarrage de l'application. Chaque fois que l'écouteur détecte un changement, la fonction `accListener` est appelée et renvoie les nouvelles données d'accélération. La fréquence des mises à jour d'accélération est définie sur 5 secondes. Pour plus d'informations, voir « [Classe Accelerometer](#) » à la page 56.

```
import flash.sensors.*;
if ((Accelerometer.isSupported)
{
    var acc:Accelerometer = new Accelerometer();
    function accListener(evt:AccelerometerEvent)
    {
        trace("AccelerationX: " + evt.accelerationX.toString());
        trace("AccelerationY: " + evt.accelerationY.toString());
        trace("AccelerationZ: " + evt.accelerationZ.toString());
        trace("Timestamp: "+evt.timestamp.toString());
    }
    acc.addEventListener(AccelerometerEvent.UPDATE, accListener);
    acc.setRequestedUpdateInterval(5000);
    acc.removeEventListener(AccelerometerEvent.UPDATE, accListener);
}
else
{
    trace("Accelerometer not supported.");
}
```

Pour plus de détails sur les classes `Accelerometer` et `AccelerometerEvent`, voir [Adobe ActionScript 3.0 Reference for the Adobe Flash Platform](#).

Chapitre 9 : Présentation du code ActionScript pour Flash Lite 4

Le logiciel Adobe Flash Lite 4 est fondé sur Flash Player 10, mais présente les différences suivantes :

- Flash Lite ne prend en charge certaines fonctionnalités que de façon partielle.
- Flash Lite dispose de fonctionnalités propres aux périphériques mobiles.

Éléments de langage et classes ActionScript non pris en charge, et pris en charge intégralement ou partiellement

Accelerometer

Le tableau ci-dessous recense les membres de la classe Accelerometer pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
Accelerometer()	Oui	Oui
isSupported : Boolean	Oui	Oui
muted : Boolean	Oui	Non
setRequestedUpdateInterval(interval:Number):void	Oui	Oui
update	Oui	Oui

AccelerometerEvent

Le tableau ci-dessous recense les membres de la classe AccelerometerEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
accelerationX : Number	Oui	Oui
accelerationY : Number	Oui	Oui
accelerationZ : Number	Oui	Oui
AccelerometerEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, timestamp:Number = 0, accelerationX:Number = 0, accelerationY:Number = 0, accelerationZ:Number = 0)	Oui	Oui
timestamp : Number	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Non
toString():String	Oui	Non
UPDATE : String = "accelerometerUpdate"	Oui	Oui

ActionGenerator

Le tableau ci-dessous recense les membres de la classe ActionGenerator pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ActionGenerator()	Oui	Oui
generateAction(a:AutomationAction):void	Oui	Oui
generateActions(actionList:Array):void	Oui	Oui

ActionScriptVersion

Le tableau ci-dessous recense les membres de la classe ActionScriptVersion pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ACTIONSCRIPT3 : uint = 3	Oui	Oui

ApplicationDomain

Le tableau ci-dessous recense les membres de la classe ApplicationDomain pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
currentDomain : ApplicationDomain	Oui	Oui
domainMemory : ByteArray	Oui	Oui
MIN_DOMAIN_MEMORY_LENGTH : uint	Oui	Oui
parentDomain : ApplicationDomain	Oui	Oui
ApplicationDomain(parentDomain:ApplicationDomain = null)	Oui	Oui
constructApplicationDomainObject(parentDomain:ApplicationDomain = null):*	Oui	Oui
getDefinition(name:String):Object	Oui	Oui
hasDefinition(name:String):Boolean	Oui	Oui

ArgumentError

Le tableau ci-dessous recense les membres de la classe ArgumentError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ArgumentError(message:String = "")	Oui	Oui

arguments

Le tableau ci-dessous recense les membres de la classe arguments pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
callee : Function	Oui	Oui
length : Number	Oui	Oui

Array

Le tableau ci-dessous recense les membres de la classe Array pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
length : uint	Oui	Oui
Array(... values)	Oui	Oui
Array(numElements:int = 0)	Oui	Oui
concat(... args):Array	Oui	Oui
every(callback:Function, thisObject:* = null):Boolean	Oui	Oui
filter(callback:Function, thisObject:* = null):Array	Oui	Oui
forEach(callback:Function, thisObject:* = null):void	Oui	Oui
indexOf(searchElement:*, fromIndex:int = 0):int	Oui	Oui
join(sep:String):String	Oui	Oui
lastIndexOf(searchElement:*, fromIndex:int = 0x7fffffff):int	Oui	Oui
map(callback:Function, thisObject:* = null):Array	Oui	Oui
pop():*	Oui	Oui
push(... args):uint	Oui	Oui
reverse():Array	Oui	Oui
shift():*	Oui	Oui
slice(startIndex:int = 0, endIndex:int = 16777215):Array	Oui	Oui
some(callback:Function, thisObject:* = null):Boolean	Oui	Oui
sort(... args):Array	Oui	Oui
sortOn(fieldName:Object, options:Object = null):Array	Oui	Oui
splice(startIndex:int, deleteCount:uint, ... values):Array	Oui	Oui
toLocaleString():String	Oui	Oui
toString():String	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
unshift(... args):uint	Oui	Oui
CASEINSENSITIVE : uint = 1	Oui	Oui
DESCENDING : uint = 2	Oui	Oui
NUMERIC : uint = 16	Oui	Oui
RETURNINDEXEDARRAY : uint = 8	Oui	Oui
UNIQUESORT : uint = 4	Oui	Oui

AsyncErrorEvent

Le tableau ci-dessous recense les membres de la classe AsyncErrorEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
AsyncErrorEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, text:String = "", error:Error = null)	Oui	Oui
clone():Event	Oui	Oui
toString():String	Oui	Oui
ASYNC_ERROR : String = "asynError"	Oui	Oui

AutomationAction

Le tableau ci-dessous recense les membres de la classe AutomationAction pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
type : String	Oui	Oui
AutomationAction(type:String)	Oui	Oui

AVM1Movie

Pour plus d'informations sur les membres de la classe AVM1Movie que prennent en charge ActionScript 3.0 et Flash Lite 4, voir *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Bitmap

Le tableau ci-dessous recense les membres de la classe Bitmap pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
bitmapData : BitmapData	Oui	Oui
pixelSnapping : String	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
smoothing : Boolean	Oui	Oui
Bitmap(bitmapData:BitmapData = null, pixelSnapping:String = "auto", smoothing:Boolean = false)	Oui	Oui
constructBitmapObject(bitmapData:BitmapData = null, pixelSnapping:String = "auto", smoothing:Boolean = false):void	Oui	Oui

BitmapData

Le tableau ci-dessous recense les membres de la classe BitmapData pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
height : int	Oui	Oui
rect : Rectangle	Oui	Oui
transparent : Boolean	Oui	Oui
width : int	Oui	Oui
BitmapData(width:int, height:int, transparent:Boolean = true, fillColor:uint = 0xFFFFFFFF)	Oui	Oui
applyFilter(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, filter:BitmapFilter):void	Oui	Non
clone():BitmapData	Oui	Oui
colorTransform(rect:Rectangle, colorTransform:flash.geom.ColorTransform):void	Oui	Oui
compare(otherBitmapData:BitmapData):Object	Oui	Oui
constructBitmapDataObject(width:int, height:int, transparent:Boolean = true, fillColor:uint = 0xFFFFFFFF):void	Oui	Oui
copyChannel(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, sourceChannel:uint, destChannel:uint):void	Oui	Oui
copyPixels(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, alphaBitmapData:BitmapData = null, alphaPoint:Point = null, mergeAlpha:Boolean = false):void	Oui	Oui
dispose():void	Oui	Oui
draw(source:IBitmapDrawable, matrix:Matrix = null, colorTransform:flash.geom.ColorTransform = null, blendMode:String = null, clipRect:Rectangle = null, smoothing:Boolean = false):void	Oui	Oui
fillRect(rect:Rectangle, color:uint):void	Oui	Oui
floodFill(x:int, y:int, color:uint):void	Oui	Oui
generateFilterRect(sourceRect:Rectangle, filter:BitmapFilter):Rectangle	Oui	Non

Nom du membre	ActionScript 3.0	Flash Lite 4
getColorBoundsRect(mask:uint, color:uint, findColor:Boolean = true):Rectangle	Oui	Oui
getPixel(x:int, y:int):uint	Oui	Oui
getPixel32(x:int, y:int):uint	Oui	Oui
getPixels(rect:Rectangle):ByteArray	Oui	Oui
getVector(rect:Rectangle):Vector.<uint>	Oui	Oui
histogram(hRect:Rectangle = null):Vector.<Vector.<Number>>	Oui	Oui
hitTest(firstPoint:Point, firstAlphaThreshold:uint, secondObject:Object, secondBitmapDataPoint:Point = null, secondAlphaThreshold:uint = 1):Boolean	Oui	Oui
lock():void	Oui	Non
merge(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, redMultiplier:uint, greenMultiplier:uint, blueMultiplier:uint, alphaMultiplier:uint):void	Oui	Oui
noise(randomSeed:int, low:uint = 0, high:uint = 255, channelOptions:uint = 7, grayScale:Boolean = false):void	Oui	Non
paletteMap(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, redArray:Array = null, greenArray:Array = null, blueArray:Array = null, alphaArray:Array = null):void	Oui	Oui
perlinNoise(baseX:Number, baseY:Number, numOctaves:uint, randomSeed:int, stitch:Boolean, fractalNoise:Boolean, channelOptions:uint = 7, grayScale:Boolean = false, offsets:Array = null):void	Oui	Non
pixelDissolve(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, randomSeed:int = 0, numPixels:int = 0, fillColor:uint = 0):int	Oui	Non
scroll(x:int, y:int):void	Oui	Non
setPixel(x:int, y:int, color:uint):void	Oui	Oui
setPixel32(x:int, y:int, color:uint):void	Oui	Oui
setPixels(rect:Rectangle, inputByteArray:ByteArray):void	Oui	Oui
setVector(rect:Rectangle, inputVector:Vector.<uint>):void	Oui	Oui
threshold(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, operation:String, threshold:uint, color:uint = 0, mask:uint = 0xFFFFFFFF, copySource:Boolean = false):uint	Oui	Non
unlock(changeRect:Rectangle = null):void	Oui	Non

BitmapDataChannel

Le tableau ci-dessous recense les membres de la classe BitmapDataChannel pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ALPHA : uint = 8	Oui	Oui
BLUE : uint = 4	Oui	Oui
GREEN : uint = 2	Oui	Oui
RED : uint = 1	Oui	Oui

Boolean

Le tableau ci-dessous recense les membres de la classe Boolean pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
Boolean(expression:Object = false)	Oui	Oui
toString():String	Oui	Oui
valueOf():Boolean	Oui	Oui

BreakOpportunity

Le tableau ci-dessous recense les membres de la classe BreakOpportunity pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ALL : String = "all"	Oui	Oui
ANY : String = "any"	Oui	Oui
AUTO : String = "auto"	Oui	Oui
NONE : String = "none"	Oui	Oui

ByteArray

Le tableau ci-dessous recense les membres de la classe ByteArray pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ByteArray()	Oui	Oui
bytesAvailable : uint	Oui	Oui
clear():void	Oui	Oui
compress():void	Oui	Non
defaultObjectEncoding : uint	Oui	Oui
deflate():void	Oui	Oui
endian : String	Oui	Oui
inflate():void	Oui	Non
length : uint	Oui	Oui
objectEncoding : uint	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
position : uint	Oui	Oui
readBoolean():Boolean	Oui	Oui
readByte():int	Oui	Oui
readBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void	Oui	Oui
readDouble():Number	Oui	Oui
readFloat():Number	Oui	Oui
readInt():int	Oui	Oui
readMultiByte(length:uint, charSet:String):String	Oui	Oui
readObject():*	Oui	Oui
readShort():int	Oui	Oui
readUnsignedByte():uint	Oui	Oui
readUnsignedInt():uint	Oui	Oui
readUnsignedShort():uint	Oui	Oui
readUTF():String	Oui	Oui
readUTFBytes(length:uint):String	Oui	Oui
toString():String	Oui	Oui
uncompress():void	Oui	Oui
writeBoolean(value:Boolean):void	Oui	Oui
writeByte(value:int):void	Oui	Oui
writeBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void	Oui	Oui
writeDouble(value:Number):void	Oui	Oui
writeFloat(value:Number):void	Oui	Oui
writeInt(value:int):void	Oui	Oui
writeMultiByte(value:String, charSet:String):void	Oui	Oui
writeObject(object:*) :void	Oui	Oui
writeShort(value:int):void	Oui	Oui
writeUnsignedInt(value:uint):void	Oui	Oui
writeUTF(value:String):void	Oui	Oui
writeUTFBytes(value:String):void	Oui	Oui

Capabilities

Le tableau ci-dessous recense les membres de la classe Capabilities pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
avHardwareDisable : Boolean	Oui	Oui
cpuArchitecture	Oui	Non
hasAccessibility : Boolean	Oui	Oui
hasAudio : Boolean	Oui	Oui
hasAudioEncoder : Boolean	Oui	Oui
hasEmbeddedVideo : Boolean	Oui	Oui
hasIME : Boolean	Oui	Oui
hasMP3 : Boolean	Oui	Oui
hasPrinting : Boolean	Oui	Oui
hasScreenBroadcast : Boolean	Oui	Oui
hasScreenPlayback : Boolean	Oui	Oui
hasStreamingAudio : Boolean	Oui	Oui
hasStreamingVideo : Boolean	Oui	Oui
hasTLS : Boolean	Oui	Oui
hasVideoEncoder : Boolean	Oui	Oui
isDebugger : Boolean	Oui	Oui
isEmbeddedInAcrobat : Boolean	Oui	Oui
language : String	Oui	Oui
localFileReadDisable : Boolean	Oui	Oui
manufacturer : String	Oui	Oui
maxLevelIDC : String	Oui	Oui
os : String	Oui	Oui
pixelAspectRatio : Number	Oui	Oui
playerType : String	Oui	Oui
screenColor : String	Oui	Oui
screenDPI : Number	Oui	Oui
screenResolutionX : Number	Oui	Oui
screenResolutionY : Number	Oui	Oui
serverString : String	Oui	Oui
supports32BitProcesses : Boolean	Oui	Non
supports64BitProcesses : Boolean	Oui	Non
touchscreenType : String	Oui	Non
version : String	Oui	Oui

CFFHinting

Le tableau ci-dessous recense les membres de la classe CFFHinting pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
HORIZONTAL_STEM : String = "horizontalStem"	Oui	Oui
NONE : String = "none"	Oui	Oui

Class

Pour plus d'informations sur les membres de la classe Class que prennent en charge ActionScript 3.0 et Flash Lite 4, voir *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

ColorTransform

Le tableau ci-dessous recense les membres de la classe ColorTransform pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
alphaMultiplieur : Number	Oui	Oui
alphaOffset : Number	Oui	Oui
blueMultiplieur : Number	Oui	Oui
blueOffset : Number	Oui	Oui
color : uint	Oui	Oui
ColorTransform(redMultiplieur:Number = 1.0, greenMultiplieur:Number = 1.0, blueMultiplieur:Number = 1.0, alphaMultiplieur:Number = 1.0, redOffset:Number = 0, greenOffset:Number = 0, blueOffset:Number = 0, alphaOffset:Number = 0)	Oui	Oui
concat(second:flash.geom:ColorTransform):void	Oui	Oui
greenMultiplieur : Number	Oui	Oui
greenOffset : Number	Oui	Oui
redMultiplieur : Number	Oui	Oui
redOffset : Number	Oui	Oui
toString():String	Oui	Oui

ContentElement

Le tableau ci-dessous recense les membres de la classe ContentElement pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ContentElement(elementFormat:ElementFormat = null, eventMirror:EventDispatcher = null, textRotation:String = "rotate0")	Oui	Oui
elementFormat : ElementFormat	Oui	Oui
eventMirror : EventDispatcher	Oui	Oui
GRAPHIC_ELEMENT : uint = 0xFDEF	Oui	Oui
groupElement : GroupElement	Oui	Oui
rawText : String	Oui	Oui
text : String	Oui	Oui
textBlock : flash.text.engine:TextBlock	Oui	Oui
textBlockBeginIndex : int	Oui	Oui
textRotation : String	Oui	Oui
userData : *	Oui	Oui

DataEvent

Le tableau ci-dessous recense les membres de la classe DataEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Oui
data : String	Oui	Oui
DATA : String = "data"	Oui	Oui
DataEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, data:String = "")	Oui	Oui
toString():String	Oui	Oui
UPLOAD_COMPLETE_DATA : String = "uploadCompleteData"	Oui	Oui

Date

Le tableau ci-dessous recense les membres de la classe Date pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
date : Number	Oui	Oui
Date(yearOrTimevalue:Object, month:Number, date:Number = 1, hour:Number = 0, minute:Number = 0, second:Number = 0, millisecond:Number = 0)	Oui	Oui
dateUTC : Number	Oui	Oui
day : Number	Oui	Oui
dayUTC : Number	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
fullYear : Number	Oui	Oui
fullYearUTC : Number	Oui	Oui
getDate():Number	Oui	Oui
getDay():Number	Oui	Oui
getFullYear():Number	Oui	Oui
getHours():Number	Oui	Oui
getMilliseconds():Number	Oui	Oui
getMinutes():Number	Oui	Oui
getMonth():Number	Oui	Oui
getSeconds():Number	Oui	Oui
getTime():Number	Oui	Oui
getTimezoneOffset():Number	Oui	Oui
getUTCDate():Number	Oui	Oui
getUTCDay():Number	Oui	Oui
getUTCFullYear():Number	Oui	Oui
getUTCHours():Number	Oui	Oui
getUTCMilliseconds():Number	Oui	Oui
getUTCMinutes():Number	Oui	Oui
getUTCMonth():Number	Oui	Oui
getUTCSeconds():Number	Oui	Oui
hours : Number	Oui	Oui
hoursUTC : Number	Oui	Oui
milliseconds : Number	Oui	Oui
millisecondsUTC : Number	Oui	Oui
minutes : Number	Oui	Oui
minutesUTC : Number	Oui	Oui
month : Number	Oui	Oui
monthUTC : Number	Oui	Oui
parse(date:String):Number	Oui	Oui
seconds : Number	Oui	Oui
secondsUTC : Number	Oui	Oui
setDate(day:Number):Number	Oui	Oui
setFullYear(year:Number, month:Number, day:Number):Number	Oui	Oui
setHours(hour:Number, minute:Number, second:Number, millisecond:Number):Number	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
setMilliseconds(milliseconds:Number):Number	Oui	Oui
setMinutes(minute:Number, second:Number, milliseconds:Number):Number	Oui	Oui
setMonth(month:Number, day:Number):Number	Oui	Oui
setSeconds(second:Number, milliseconds:Number):Number	Oui	Oui
setTime(milliseconds:Number):Number	Oui	Oui
setUTCDate(day:Number):Number	Oui	Oui
setUTCFullYear(year:Number, month:Number, day:Number):Number	Oui	Oui
setUTCHours(hour:Number, minute:Number, second:Number, milliseconds:Number):Number	Oui	Oui
setUTCMilliseconds(milliseconds:Number):Number	Oui	Oui
setUTCMinutes(minute:Number, [second:Number], [milliseconds:Number]) : Number	Oui	Oui
setUTCMinutes(minute:Number, second:Number, milliseconds:Number):Number	Oui	Oui
setUTCMonth(month:Number, day:Number):Number	Oui	Oui
setUTCSeconds(second:Number, milliseconds:Number):Number	Oui	Oui
setYear(year:Number) : Number	Oui	Oui
time : Number	Oui	Oui
timezoneOffset : Number	Oui	Oui
toDateString():String	Oui	Oui
toLocaleDateString():String	Oui	Oui
toLocaleString():String	Oui	Oui
toLocaleTimeString():String	Oui	Oui
toString():String	Oui	Oui
toTimeString():String	Oui	Oui
toUTCString():String	Oui	Oui
UTC(year:Number, month:Number, date:Number = 1, hour:Number = 0, minute:Number = 0, second:Number = 0, milliseconds:Number = 0):Number	Oui	Oui
valueOf():Number	Oui	Oui

DefinitionError

Le tableau ci-dessous recense les membres de la classe DefinitionError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
DefinitionError(message:String = "")	Oui	Oui

Dictionary

Le tableau ci-dessous recense les membres de la classe Dictionary pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
Dictionary(weakKeys:Boolean = false)	Oui	Oui

DigitCase

Le tableau ci-dessous recense les membres de la classe DigitCase pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
DEFAULT : String = "default"	Oui	Oui
LINING : String = "lining"	Oui	Oui
OLD_STYLE : String = "oldStyle"	Oui	Oui

DigitWidth

Le tableau ci-dessous recense les membres de la classe DigitWidth pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
DEFAULT : String = "default"	Oui	Oui
PROPORTIONAL : String = "proportional"	Oui	Oui
TABULAR : String = "tabular"	Oui	Oui

DisplayObject

Le tableau ci-dessous recense les membres de la classe DisplayObject pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
accessibilityProperties : AccessibilityProperties	Oui	Non
added	Oui	Oui
addedToStage	Oui	Oui
alpha : Number	Oui	Oui
blendMode : String	Oui	Non
blendShader : Shader	Oui	Non
cacheAsBitmap : Boolean	Oui	Oui
enterFrame	Oui	Oui
exitFrame	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
filters : Array	Oui	Non
frameConstructed	Oui	Oui
getBounds(targetCoordinateSpace:DisplayObject):Rectangle	Oui	Oui
getRect(targetCoordinateSpace:DisplayObject):Rectangle	Oui	Oui
globalToLocal(point:Point):Point	Oui	Oui
globalToLocal3D(point:Point):Vector3D	Oui	Oui
height : Number	Oui	Oui
hitTestObject(obj:DisplayObject):Boolean	Oui	Oui
hitTestPoint(x:Number, y:Number, shapeFlag:Boolean = false):Boolean	Oui	Oui
loaderInfo : LoaderInfo	Oui	Oui
local3DToGlobal(point3d:Vector3D):Point	Oui	Oui
localToGlobal(point:Point):Point	Oui	Oui
mask : DisplayObject	Oui	Oui
modified : Boolean	Oui	Oui
mouseX : Number	Oui	Oui
mouseY : Number	Oui	Oui
name : String	Oui	Oui
opaqueBackground : Object	Oui	Oui
parent : DisplayObjectContainer	Oui	Oui
removed	Oui	Oui
removedFromStage	Oui	Oui
render	Oui	Oui
root : DisplayObject	Oui	Oui
rotation : Number	Oui	Oui
rotationX : Number	Oui	Oui
rotationY : Number	Oui	Oui
rotationZ : Number	Oui	Oui
scale9Grid : Rectangle	Oui	Oui
scaleX : Number	Oui	Oui
scaleY : Number	Oui	Oui
scaleZ : Number	Oui	Non
scrollRect : Rectangle	Oui	Non
stage : Stage	Oui	Oui
transform : flash.geom:Transform	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
visible : Boolean	Oui	Oui
width : Number	Oui	Oui
x : Number	Oui	Oui
y : Number	Oui	Oui
z : Number	Oui	Non

DisplayObjectContainer

Le tableau ci-dessous recense les membres de la classe DisplayObjectContainer pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
addChild(child:DisplayObject):DisplayObject	Oui	Oui
addChildAt(child:DisplayObject, index:int):DisplayObject	Oui	Oui
areInaccessibleObjectsUnderPoint(point:Point):Boolean	Oui	Oui
contains(child:DisplayObject):Boolean	Oui	Oui
getChildAt(index:int):DisplayObject	Oui	Oui
getChildByName(name:String):DisplayObject	Oui	Oui
getChildIndex(child:DisplayObject):int	Oui	Oui
getObjectsUnderPoint(point:Point):Array	Oui	Oui
mouseChildren : Boolean	Oui	Oui
numChildren : int	Oui	Oui
removeChild(child:DisplayObject):DisplayObject	Oui	Oui
removeChildAt(index:int):DisplayObject	Oui	Oui
setChildIndex(child:DisplayObject, index:int):void	Oui	Oui
swapChildren(child1:DisplayObject, child2:DisplayObject):void	Oui	Oui
swapChildrenAt(index1:int, index2:int):void	Oui	Oui
tabChildren : Boolean	Oui	Oui
textSnapshot : flash.text:TextSnapshot	Oui	Non

EastAsianJustifier

Le tableau ci-dessous recense les membres de la classe EastAsianJustifier pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():flash.text.engine.TextJustifier	Oui	Oui
EastAsianJustifier(locale:String = "ja", lineJustification:String = "allButLast", justificationStyle:String = "pushInKinsoku")	Oui	Oui
justificationStyle : String	Oui	Oui

ElementFormat

Le tableau ci-dessous recense les membres de la classe ElementFormat pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
alignmentBaseline : String	Oui	Oui
alpha : Number	Oui	Oui
baselineShift : Number	Oui	Oui
breakOpportunity : String	Oui	Oui
clone():ElementFormat	Oui	Oui
color : uint	Oui	Oui
digitCase : String	Oui	Oui
digitWidth : String	Oui	Oui
dominantBaseline : String	Oui	Oui
ElementFormat(fontDescription:FontDescription = null, fontSize:Number = 12.0, color:uint = 0x000000, alpha:Number = 1.0, textRotation:String = "auto", dominantBaseline:String = "roman", alignmentBaseline:String = "useDominantBaseline", baselineShift:Number = 0.0, kerning:String = "on", trackingRight:Number = 0.0, trackingLeft:Number = 0.0, locale:String = "en", breakOpportunity:String = "auto", digitCase:String = "default", digitWidth:String = "default", ligatureLevel:String = "common", typographicCase:String = "default")	Oui	Oui
fontDescription : FontDescription	Oui	Oui
fontSize : Number	Oui	Oui
getFontMetrics():FontMetrics	Oui	Oui
kerning : String	Oui	Oui
ligatureLevel : String	Oui	Oui
locale : String	Oui	Oui
locked : Boolean	Oui	Oui
textRotation : String	Oui	Oui
trackingLeft : Number	Oui	Oui
trackingRight : Number	Oui	Oui
typographicCase : String	Oui	Oui

Endian

Le tableau ci-dessous recense les membres de la classe Endian pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
BIG_ENDIAN : String = "bigEndian"	Oui	Oui
LITTLE_ENDIAN : String = "littleEndian"	Oui	Oui

EOFError

Le tableau ci-dessous recense les membres de la classe EOFError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
EOFError(message:String = "")	Oui	Oui

Error

Le tableau ci-dessous recense les membres de la classe Error pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
Error(message:String = "", id:int = 0)	Oui	Oui
errorID : int	Oui	Oui
getStackTrace():String	Oui	Oui
message : String	Oui	Oui
name : String	Oui	Oui
toString():String	Oui	Oui

ErrorEvent

Le tableau ci-dessous recense les membres de la classe ErrorEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Oui
ERROR : String = "error"	Oui	Oui
ErrorEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, text:String = "")	Oui	Oui
errorID : int	Oui	Non
toString():String	Oui	Oui

EvalError

Le tableau ci-dessous recense les membres de la classe EvalError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
EvalError(message:String = "")	Oui	Oui

Event

Le tableau ci-dessous recense les membres de la classe Event pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ACTIVATE : String = "activate"	Oui	Oui
ADDED : String = "added"	Oui	Oui
ADDED_TO_STAGE : String = "addedToStage"	Oui	Oui
bubbles : Boolean	Oui	Oui
CANCEL : String = "cancel"	Oui	Oui
cancelable : Boolean	Oui	Oui
CHANGE : String = "change"	Oui	Oui
CLEAR : String = "clear"	Oui	Oui
clone():Event	Oui	Oui
CLOSE : String = "close"	Oui	Oui
COMPLETE : String = "complete"	Oui	Oui
CONNECT : String = "connect"	Oui	Oui
COPY : String = "copy"	Oui	Oui
currentTarget : Object	Oui	Oui
CUT : String = "cut"	Oui	Oui
DEACTIVATE : String = "deactivate"	Oui	Oui
ENTER_FRAME : String = "enterFrame"	Oui	Oui
Event(type:String, bubbles:Boolean = false, cancelable:Boolean = false)	Oui	Oui
eventPhase : uint	Oui	Oui
EXIT_FRAME : String = "exitFrame"	Oui	Oui
formatToString(className:String, ... arguments):String	Oui	Oui
FRAME_CONSTRUCTED : String = "frameConstructed"	Oui	Oui
FULLSCREEN : String = "fullScreen"	Oui	Oui
ID3 : String = "id3"	Oui	Oui
INIT : String = "init"	Oui	Oui
isDefaultPrevented():Boolean	Oui	Oui
MOUSE_LEAVE : String = "mouseLeave"	Oui	Oui
OPEN : String = "open"	Oui	Oui
PASTE : String = "paste"	Oui	Oui
preventDefault():void	Oui	Oui
REMOVED : String = "removed"	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
REMOVED_FROM_STAGE : String = "removedFromStage"	Oui	Oui
RENDER : String = "render"	Oui	Oui
RESIZE : String = "resize"	Oui	Oui
SCROLL : String = "scroll"	Oui	Oui
SELECT : String = "select"	Oui	Oui
SELECT_ALL : String = "selectAll"	Oui	Oui
SOUND_COMPLETE : String = "soundComplete"	Oui	Oui
stopImmediatePropagation():void	Oui	Oui
stopPropagation():void	Oui	Oui
TAB_CHILDREN_CHANGE : String = "tabChildrenChange"	Oui	Oui
TAB_ENABLED_CHANGE : String = "tabEnabledChange"	Oui	Oui
TAB_INDEX_CHANGE : String = "tabIndexChange"	Oui	Oui
target : Object	Oui	Oui
toString():String	Oui	Oui
type : String	Oui	Oui
UNLOAD : String = "unload"	Oui	Oui

EventDispatcher

Le tableau ci-dessous recense les membres de la classe EventDispatcher pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
activate	Oui	Oui
addEventListener(type:String, listener:Function, useCapture:Boolean = false, priority:int = 0, useWeakReference:Boolean = false):void	Oui	Oui
deactivate	Oui	Oui
dispatchEvent(event:Event):Boolean	Oui	Oui
EventDispatcher(target:IEventDispatcher = null)	Oui	Oui
hasEventListener(type:String):Boolean	Oui	Oui
removeEventListener(type:String, listener:Function, useCapture:Boolean = false):void	Oui	Oui
willTrigger(type:String):Boolean	Oui	Oui

EventPhase

Le tableau ci-dessous recense les membres de la classe EventPhase pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
AT_TARGET : uint = 2	Oui	Oui
BUBBLING_PHASE : uint = 3	Oui	Oui
CAPTURING_PHASE : uint = 1	Oui	Oui

Extension

Le tableau ci-dessous recense les membres de la classe Extension pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
callMethod(methodName:String, ... rest):*	Oui	Oui
Extension(extensionName:String, ... rest)	Oui	Oui

ExternalInterface

Le tableau ci-dessous recense les membres de la classe ExternalInterface pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
addCallback(functionName:String, closure:Function):void	Oui	Oui
available : Boolean	Oui	Oui
call(functionName:String, ... arguments):*	Oui	Oui
marshallExceptions : Boolean = false	Oui	Oui
objectID : String	Oui	Oui

FocusDirection

Le tableau ci-dessous recense les membres de la classe FocusDirection pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
BOTTOM : String = "bottom"	Oui	Non
NONE : String = "none"	Oui	Non
TOP : String = "top"	Oui	Non

FocusEvent

Le tableau ci-dessous recense les membres de la classe FocusEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Oui
FocusEvent(type:String, bubbles:Boolean = true, cancelable:Boolean = false, relatedObject:InteractiveObject = null, shiftKey:Boolean = false, keyCode:uint = 0)	Oui	Oui
FOCUS_IN : String = "focusIn"	Oui	Oui
FOCUS_OUT : String = "focusOut"	Oui	Oui
isRelatedObjectInaccessible : Boolean	Oui	Oui
keyCode : uint	Oui	Oui
KEY_FOCUS_CHANGE : String = "keyFocusChange"	Oui	Oui
MOUSE_FOCUS_CHANGE : String = "mouseFocusChange"	Oui	Oui
relatedObject : InteractiveObject	Oui	Oui
shiftKey : Boolean	Oui	Oui
toString():String	Oui	Oui

Font

Le tableau ci-dessous recense les membres de la classe Font pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
enumerateFonts(enumerateDeviceFonts:Boolean = false):Array	Oui	Partielle
fontName : String	Oui	Oui
fontStyle : String	Oui	Oui
fontType : String	Oui	Oui
hasGlyphs(str:String):Boolean	Oui	Oui
registerFont(font:Class):void	Oui	Oui

FontDescription

Le tableau ci-dessous recense les membres de la classe FontDescription pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
cffHinting : String	Oui	Oui
clone():FontDescription	Oui	Oui
FontDescription(fontName:String = "_serif", fontWeight:String = "normal", fontPosture:String = "normal", fontLookup:String = "device", renderingMode:String = "cff", cffHinting:String = "horizontalStem")	Oui	Oui
fontLookup : String	Oui	Oui
fontName : String	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
fontPosture : String	Oui	Oui
fontWeight : String	Oui	Oui
isFontCompatible(fontName:String, fontWeight:String, fontPosture:String):Boolean	Oui	Oui
relatedObject : InteractiveObject	Oui	Oui
locked : Boolean	Oui	Oui
renderingMode : String	Oui	Oui

FontLookup

Le tableau ci-dessous recense les membres de la classe FontLookup pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
DEVICE : String = "device"	Oui	Oui
EMBEDDED_CFF : String = "embeddedCFF"	Oui	Oui

FontMetrics

Le tableau ci-dessous recense les membres de la classe FontMetrics pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
emBox : Rectangle	Oui	Oui
FontMetrics(emBox:Rectangle, strikethroughOffset:Number, strikethroughThickness:Number, underlineOffset:Number, underlineThickness:Number, subscriptOffset:Number, subscriptScale:Number, superscriptOffset:Number, superscriptScale:Number)	Oui	Oui
strikethroughOffset : Number	Oui	Oui
strikethroughThickness : Number	Oui	Oui
subscriptOffset : Number	Oui	Oui
subscriptScale : Number	Oui	Oui
superscriptOffset : Number	Oui	Oui
superscriptScale : Number	Oui	Oui
underlineOffset : Number	Oui	Oui
underlineThickness : Number	Oui	Oui

FontPosture

Le tableau ci-dessous recense les membres de la classe FontPosture pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ITALIC : String = "italic"	Oui	Oui
NORMAL : String = "normal"	Oui	Oui

FontType

Le tableau ci-dessous recense les membres de la classe FontType pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
DEVICE : String = "device"	Oui	Oui
EMBEDDED : String = "embedded"	Oui	Oui
EMBEDDED_CFF : String = "embeddedCFF"	Oui	Oui

FontWeight

Le tableau ci-dessous recense les membres de la classe FontWeight pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
BOLD : String = "bold"	Oui	Oui
NORMAL : String = "normal"	Oui	Oui

FrameLabel

Le tableau ci-dessous recense les membres de la classe FrameLabel pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
frame : int	Oui	Oui
name : String	Oui	Oui

FullScreenEvent

Le tableau ci-dessous recense les membres de la classe FullScreenEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Oui
fullScreen : Boolean	Oui	Oui
FullScreenEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, fullScreen:Boolean = false)	Oui	Oui
FULL_SCREEN : String = "fullScreen"	Oui	Oui
toString():String	Oui	Oui

Function

Le tableau ci-dessous recense les membres de la classe Function pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
apply(thisArg:*, argArray:*)*	Oui	Oui
call(thisArg:*, ... args):*	Oui	Oui

Geolocation

Le tableau ci-dessous recense les membres de la classe Geolocation pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
Geolocation()	Oui	Oui
muted : Boolean	Oui	Non
update	Oui	Oui
isSupported : Boolean	Oui	Oui
setRequestedUpdateInterval(time:Number):*	Oui	Oui
status	Oui	Oui

GeolocationEvent

Le tableau ci-dessous recense les membres de la classe GeolocationEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
altitude : Number	Oui	Oui
GeolocationEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, latitude:Number = 0, longitude:Number = 0, altitude:Number = 0, hAccuracy:Number = 0, vAccuracy:Number = 0, speed:Number = 0, heading:Number = 0, timestamp:int = 0)	Oui	Oui
heading : Number	Oui	Oui
horizontalAccuracy : Number	Oui	Oui
latitude : Number	Oui	Oui
longitude : Number	Oui	Oui
speed : Number	Oui	Oui
timestamp : Number	Oui	Oui
verticalAccuracy : Number	Oui	Oui
clone() : Event	Oui	Non
toString() : String	Oui	Non
UPDATE : String = "geolocationUpdate"	Oui	Oui

GestureEvent

Le tableau ci-dessous recense les membres de la classe GestureEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
altKey : Boolean	Oui	Non
clone():Event	Oui	Oui
commandKey : Boolean	Oui	Non
controlKey : Boolean	Oui	Non
ctrlKey : Boolean	Oui	Non
GestureEvent(type:String, bubbles:Boolean = true, cancelable:Boolean = false, phase:String = null, localX:Number = 0, localY:Number = 0, ctrlKey:Boolean = false, altKey:Boolean = false, shiftKey:Boolean = false)	Oui	Oui
GESTURE_TWO_FINGER_TAP : String = "gestureTwoFingerTap"	Oui	Oui
localX : Number	Oui	Oui
localY : Number	Oui	Oui
phase : String	Oui	Oui
shiftKey : Boolean	Oui	Non
stageX : Number	Oui	Oui
stageY : Number	Oui	Oui
toString():String	Oui	Oui
updateAfterEvent():void	Oui	Oui

GesturePhase

Le tableau ci-dessous recense les membres de la classe GesturePhase pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ALL : String = "all"	Oui	Non
BEGIN : String = "begin"	Oui	Oui
END : String = "end"	Oui	Oui
UPDATE : String = "update"	Oui	Oui

GradientType

Le tableau ci-dessous recense les membres de la classe GradientType pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
LINEAR : String = "linear"	Oui	Oui
RADIAL : String = "radial"	Oui	Oui

GraphicElement

Le tableau ci-dessous recense les membres de la classe GraphicElement pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
elementHeight : Number	Oui	Oui
elementWidth : Number	Oui	Oui
graphic : DisplayObject	Oui	Oui
GraphicElement(graphic:DisplayObject = null, elementWidth:Number = 15.0, elementHeight:Number = 15.0, elementFormat:ElementFormat = null, eventMirror:EventDispatcher = null, textRotation:String = "rotate0")	Oui	Oui

Graphics

Le tableau ci-dessous recense les membres de la classe Graphics pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
beginBitmapFill(bitmap:BitmapData, matrix:Matrix = null, repeat:Boolean = true, smooth:Boolean = false):void	Oui	Oui
beginFill(color:uint, alpha:Number = 1.0):void	Oui	Oui
beginGradientFill(type:String, colors:Array, alphas:Array, ratios:Array, matrix:Matrix = null, spreadMethod:String = "pad", interpolationMethod:String = "rgb", focalPointRatio:Number = 0):void	Oui	Oui
beginShaderFill(shader:Shader, matrix:Matrix = null):void	Oui	Non
clear():void	Oui	Oui
copyFrom(sourceGraphics:Graphics):void	Oui	Oui
curveTo(controlX:Number, controlY:Number, anchorX:Number, anchorY:Number):void	Oui	Oui
drawCircle(x:Number, y:Number, radius:Number):void	Oui	Oui
drawEllipse(x:Number, y:Number, width:Number, height:Number):void	Oui	Oui
drawGraphicsData(graphicsData:Vector.<IGraphicsData>):void	Oui	Non
drawPath(commands:Vector.<int>, data:Vector.<Number>, winding:String = "evenOdd"):void	Oui	Non
drawRect(x:Number, y:Number, width:Number, height:Number):void	Oui	Oui
drawRoundRect(x:Number, y:Number, width:Number, height:Number, ellipseWidth:Number, ellipseHeight:Number = NaN):void	Oui	Oui
drawTriangles(vertices:Vector.<Number>, indices:Vector.<int> = null, uvData:Vector.<Number> = null, culling:String = "none"):void	Oui	Non
endFill():void	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
lineBitmapStyle(bitmap:BitmapData, matrix:Matrix = null, repeat:Boolean = true, smooth:Boolean = false):void	Oui	Non
lineGradientStyle(type:String, colors:Array, alphas:Array, ratios:Array, matrix:Matrix = null, spreadMethod:String = "pad", interpolationMethod:String = "rgb", focalPointRatio:Number = 0):void	Oui	Non
lineShaderStyle(shader:Shader, matrix:Matrix = null):void	Oui	Non
lineStyle(thickness:Number = NaN, color:uint = 0, alpha:Number = 1.0, pixelHinting:Boolean = false, scaleMode:String = "normal", caps:String = null, joints:String = null, miterLimit:Number = 3):void	Oui	Partielle
lineTo(x:Number, y:Number):void	Oui	Oui
moveTo(x:Number, y:Number):void	Oui	Oui

GroupElement

Le tableau ci-dessous recense les membres de la classe GroupElement pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
elementCount : int	Oui	Oui
getElementAt(index:int):ContentElement	Oui	Oui
getElementAtCharIndex(charIndex:int):ContentElement	Oui	Oui
getElementIndex(element:ContentElement):int	Oui	Oui
GroupElement(elements:Vector.<ContentElement> = null, elementFormat:ElementFormat = null, eventMirror:EventDispatcher = null, textRotation:String = "rotate0")	Oui	Oui
groupElements(beginIndex:int, endIndex:int):GroupElement	Oui	Oui
mergeTextElements(beginIndex:int, endIndex:int):flash.text.engine:TextElement	Oui	Oui
replaceElements(beginIndex:int, endIndex:int, newElements:Vector.<ContentElement>):Vector.<ContentElement>	Oui	Oui
setElements(value:Vector.<ContentElement>):void	Oui	Oui
splitTextElement(elementIndex:int, splitIndex:int):flash.text.engine:TextElement	Oui	Oui
ungroupElements(groupIndex:int):void	Oui	Oui

HTTPStatusEvent

Le tableau ci-dessous recense les membres de la classe HTTPStatusEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Oui
HTTPStatusEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, status:int = 0)	Oui	Oui
HTTP_STATUS : String = "httpStatus"	Oui	Oui
status : int	Oui	Oui
toString():String	Oui	Oui

IBitmapDrawable

Pour plus d'informations sur l'interface IBitmapDrawable que prennent en charge ActionScript 3.0 et Flash Lite 4, voir *Adobe ActionScript 0 Reference for the Adobe Flash Platform*.

ID3Info

Le tableau ci-dessous recense les membres de la classe ID3Info pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
album : String	Oui	Oui
artist : String	Oui	Oui
comment : String	Oui	Oui
genre : String	Oui	Oui
songName : String	Oui	Oui
track : String	Oui	Oui
year : String	Oui	Oui

IDataInput

Le tableau ci-dessous recense les membres de l'interface IDataInput pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
bytesAvailable : uint	Oui	Oui
endian : String	Oui	Oui
objectEncoding : uint	Oui	Oui
readBoolean():Boolean	Oui	Oui
readByte():int	Oui	Oui
readBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void	Oui	Oui
readDouble():Number	Oui	Oui
readFloat():Number	Oui	Oui
readInt():int	Oui	Oui
readMultiByte(length:uint, charSet:String):String	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
readObject():*	Oui	Oui
readShort():int	Oui	Oui
readUnsignedByte():uint	Oui	Oui
readUnsignedInt():uint	Oui	Oui
readUnsignedShort():uint	Oui	Oui
readUTF():String	Oui	Ye
readUTFBytes(length:uint):String	Oui	Oui

IDataOutput

Le tableau ci-dessous recense les membres de l'interface IDataOutput pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
endian : String	Oui	Oui
objectEncoding : uint	Oui	Oui
writeBoolean(value:Boolean):void	Oui	Oui
writeByte(value:int):void	Oui	Oui
writeBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void	Oui	Oui
writeDouble(value:Number):void	Oui	Oui
writeFloat(value:Number):void	Oui	Oui
writeInt(value:int):void	Oui	Oui
writeMultiByte(value:String, charSet:String):void	Oui	Oui
writeObject(object:*) :void	Oui	Oui
writeShort(value:int):void	Oui	Oui
writeUnsignedInt(value:uint):void	Oui	Oui
writeUTF(value:String):void	Oui	Oui
writeUTFBytes(value:String):void	Oui	Oui

IDynamicPropertyOutput

Le tableau ci-dessous recense les membres de l'interface IDynamicPropertyOutput pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
writeDynamicProperty(name:String, value:*) :void	Oui	Oui

IDynamicPropertyWriter

Le tableau ci-dessous recense les membres de l'interface IDynamicPropertyWriter pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
writeDynamicProperties(obj:Object, output:IDynamicPropertyOutput):void	Oui	Oui

IEventDispatcher

Le tableau ci-dessous recense les membres de l'interface IEventDispatcher pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
addEventListener(type:String, listener:Function, useCapture:Boolean = false, priority:int = 0, useWeakReference:Boolean = false):void	Oui	Oui
dispatchEvent(event:Event):Boolean	Oui	Oui
hasEventListener(type:String):Boolean	Oui	Oui
removeEventListener(type:String, listener:Function, useCapture:Boolean = false):void	Oui	Oui
willTrigger(type:String):Boolean	Oui	Oui

IExternalizable

Le tableau ci-dessous recense les membres de l'interface IExternalizable pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
readExternal(input:IDataInput):void	Oui	Oui
writeExternal(output:IDataOutput):void	Oui	Oui

IllegalOperationError

Le tableau ci-dessous recense les membres de la classe IllegalOperationError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
IllegalOperationError(message:String = "")	Oui	Oui

IMEEvent

Le tableau ci-dessous recense les membres de la classe IMEEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Oui
imeClient : IIMEClient	Oui	Non
IMEEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, text:String = "")	Oui	Oui
IME_COMPOSITION : String = "imeComposition"	Oui	Oui
IME_START_COMPOSITION : String = "imeStartComposition"	Oui	Non
toString():String	Oui	Oui

int

Le tableau ci-dessous recense les membres de la classe int pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
int(num:Object)	Oui	Oui
MAX_VALUE : int = 2147483648	Oui	Oui
MIN_VALUE : int = -2147483648	Oui	Oui
toExponential(fractionDigits:uint):String	Oui	Oui
toFixed(fractionDigits:uint):String	Oui	Oui
toPrecision(precision:uint):String	Oui	Oui
toString(radix:uint):String	Oui	Oui
valueOf():int	Oui	Oui

InteractiveObject

Le tableau ci-dessous recense les membres de la classe InteractiveObject pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clear	Oui	Oui
click	Oui	Oui
contextMenu : ContextMenu	Oui	Non
copy	Oui	Non
cut	Oui	Non
doubleClick	Oui	Non
doubleClickEnabled : Boolean	Oui	Non
focusIn	Oui	Oui
focusOut	Oui	Oui
focusRect : Object	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
InteractiveObject()	Oui	Oui
keyDown	Oui	Oui
keyFocusChange	Oui	Oui
keyUp	Oui	Oui
mouseDown	Oui	Oui
mouseEnabled : Boolean	Oui	Oui
mouseFocusChange	Oui	Oui
mouseMove	Oui	Oui
mouseOut	Oui	Oui
mouseOver	Oui	Oui
mouseUp	Oui	Oui
mouseWheel	Oui	Non
paste	Oui	Non
rollOut	Oui	Oui
rollOver	Oui	Oui
selectAll	Oui	Non
tabChildrenChange	Oui	Non
tabEnabled : Boolean	Oui	Non
tabEnabledChange	Oui	Non
tabIndex : int	Oui	Non
tabIndexChange	Oui	Non

InvalidSWFError

Le tableau ci-dessous recense les membres de la classe InvalidSWFError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
InvalidSWFError(message:String = "", id:int = 0)	Oui	Oui

IOError

Le tableau ci-dessous recense les membres de la classe IOError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
IOError(message:String = "")	Oui	Oui

IOErrorEvent

Le tableau ci-dessous recense les membres de la classe IOErrorEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Oui
IOErrorEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, text:String = "")	Oui	Oui
IO_ERROR : String = "ioError"	Oui	Oui
toString():String	Oui	Oui

JPEGLoaderContext

Le tableau ci-dessous recense les membres de la classe JPEGLoaderContext pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
deblockingFilter : Number = 0.0	Oui	Oui
JPEGLoaderContext(deblockingFilter:Number = 0.0, checkPolicyFile:Boolean = false, applicationDomain:ApplicationDomain = null, securityDomain:SecurityDomain = null)	Oui	Oui

JustificationStyle

Le tableau ci-dessous recense les membres de la classe JustificationStyle pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
PRIORITIZE_LEAST_ADJUSTMENT : String = "prioritizeLeastAdjustment"	Oui	Oui
PUSH_IN_KINSOKU : String = "pushInKinsoku"	Oui	Oui
PUSH_OUT_ONLY : String = "pushOutOnly"	Oui	Oui

Kerning

Le tableau ci-dessous recense les membres de la classe Kerning pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
AUTO : String = "auto"	Oui	Oui
OFF : String = "off"	Oui	Oui
ON : String = "on"	Oui	Oui

Keyboard

Le tableau ci-dessous recense les membres de la classe Keyboard pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
BACKSPACE : uint = 8	Oui	Oui
capsLock : Boolean	Oui	Oui
CAPS_LOCK : uint = 20	Oui	Oui
CONTROL : uint = 17	Oui	Oui
DELETE : uint = 46	Oui	Oui
DOWN : uint = 40	Oui	Oui
END : uint = 35	Oui	Oui
ENTER : uint = 13	Oui	Oui
ESCAPE : uint = 27	Oui	Oui
F1 : uint = 112	Oui	Oui
F10 : uint = 121	Oui	Oui
F11 : uint = 122	Oui	Oui
F12 : uint = 123	Oui	Oui
F13 : uint = 124	Oui	Oui
F14 : uint = 125	Oui	Oui
F15 : uint = 126	Oui	Oui
F2 : uint = 113	Oui	Oui
F3 : uint = 114	Oui	Oui
F4 : uint = 115	Oui	Oui
F5 : uint = 116	Oui	Oui
F6 : uint = 117	Oui	Oui
F7 : uint = 118	Oui	Oui
F8 : uint = 119	Oui	Oui
F9 : uint = 120	Oui	Oui
hasVirtualKeyboard : Boolean	Oui	Non
HOME : uint = 36	Oui	Oui
INSERT : uint = 45	Oui	Oui
isAccessible():Boolean	Oui	Non
LEFT : uint = 37	Oui	Oui
numLock : Boolean	Oui	Oui
NUMPAD_0 : uint = 96	Oui	Oui
NUMPAD_1 : uint = 97	Oui	Oui
NUMPAD_2 : uint = 98	Oui	Oui
NUMPAD_3 : uint = 99	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
NUMPAD_4 : uint = 100	Oui	Oui
NUMPAD_5 : uint = 101	Oui	Oui
NUMPAD_6 : uint = 102	Oui	Oui
NUMPAD_7 : uint = 103	Oui	Oui
NUMPAD_8 : uint = 104	Oui	Oui
NUMPAD_9 : uint = 105	Oui	Oui
NUMPAD_ADD : uint = 107	Oui	Oui
NUMPAD_DECIMAL : uint = 110	Oui	Oui
NUMPAD_DIVIDE : uint = 111	Oui	Oui
NUMPAD_ENTER : uint = 108	Oui	Oui
NUMPAD_MULTIPLY : uint = 106	Oui	Oui
NUMPAD_SUBTRACT : uint = 109	Oui	Oui
PAGE_DOWN : uint = 34	Oui	Oui
PAGE_UP : uint = 33	Oui	Oui
physicalKeyboardType : String	Oui	Non
RIGHT : uint = 39	Oui	Oui
SHIFT : uint = 16	Oui	Oui
SPACE : uint = 32	Oui	Oui
TAB : uint = 9	Oui	Oui
UP : uint = 38	Oui	Oui

KeyboardAutomationAction

Le tableau ci-dessous recense les membres de la classe KeyboardAutomationAction pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
KeyboardAutomationAction(type:String, keyCode:uint = 0)	Oui	Oui
IOErrorEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, text:String = "")	Oui	Oui
IO_ERROR : String = "ioError"	Oui	Oui
toString():String	Oui	Oui

KeyboardEvent

Le tableau ci-dessous recense les membres de la classe KeyboardEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
altKey : Boolean	Oui	Non
charCode : uint	Oui	Oui
clone():Event	Oui	Oui
ctrlKey : Boolean	Oui	Non
KeyboardEvent(type:String, bubbles:Boolean = true, cancelable:Boolean = false, charCode:uint = 0, keyCode:uint = 0, keyLocation:uint = 0, ctrlKey:Boolean = false, altKey:Boolean = false, shiftKey:Boolean = false)	Oui	Oui
keyCode : uint	Oui	Oui
KEY_DOWN : String = "keyDown"	Oui	Oui
keyLocation : uint	Oui	Oui
KEY_UP : String = "keyUp"	Oui	Oui
shiftKey : Boolean	Oui	Non
toString():String	Oui	Oui
updateAfterEvent():void	Oui	Non

LigatureLevel

Le tableau ci-dessous recense les membres de la classe LigatureLevel pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
COMMON : String = "common"	Oui	Oui
EXOTIC : String = "exotic"	Oui	Oui
MINIMUM : String = "minimum"	Oui	Oui
NONE : String = "none"	Oui	Oui
UNCOMMON : String = "uncommon"	Oui	Oui

LineJustification

Le tableau ci-dessous recense les membres de la classe LineJustification pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ALL_BUT_LAST : String = "allButLast"	Oui	Oui
ALL_INCLUDING_LAST : String = "allIncludingLast"	Oui	Oui
UNJUSTIFIED : String = "unjustified"	Oui	Oui

Loader

Le tableau ci-dessous recense les membres de la classe Loader pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
close():void	Oui	Oui
content : DisplayObject	Oui	Oui
contentLoaderInfo : LoaderInfo	Oui	Oui
load(request:URLRequest, context:LoaderContext = null):void	Oui	Oui
loadBytes(bytes:ByteArray, context:LoaderContext = null):void	Oui	Oui
Loader()	Oui	Oui
uncaughtErrorEvents : UncaughtErrorEvents	Oui	Non
unload():void	Oui	Oui
unloadAndStop(gc:Boolean = true):void	Oui	Oui

LoaderContext

Le tableau ci-dessous recense les membres de la classe LoaderContext pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
applicationDomain : ApplicationDomain = null	Oui	Oui
checkPolicyFile : Boolean = false	Oui	Oui
LoaderContext(checkPolicyFile:Boolean = false, applicationDomain:ApplicationDomain = null, securityDomain:SecurityDomain = null)	Oui	Oui
securityDomain : SecurityDomain = null	Oui	Oui

LoaderInfo

Le tableau ci-dessous recense les membres de la classe LoaderInfo pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
actionScriptVersion : uint	Oui	Oui
applicationDomain : ApplicationDomain	Oui	Oui
bytes : ByteArray	Oui	Oui
bytesLoaded : uint	Oui	Oui
bytesTotal : uint	Oui	Oui
childAllowsParent : Boolean	Oui	Oui
complete	Oui	Oui
content : DisplayObject	Oui	Oui
contentType : String	Oui	Oui
frameRate : Number	Oui	Oui
getLoaderInfoByDefinition(object:Object):LoaderInfo	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
height : int	Oui	Oui
HttpStatus	Oui	Oui
init	Oui	Oui
ioError	Oui	Oui
loader : Loader	Oui	Oui
loaderURL : String	Oui	Oui
open	Oui	Oui
parameters : Object	Oui	Oui
parentAllowsChild : Boolean	Oui	Oui
progress	Oui	Oui
sameDomain : Boolean	Oui	Oui
sharedEvents : EventDispatcher	Oui	Oui
swfVersion : uint	Oui	Oui
uncaughtErrorEvents : UncaughtErrorEvents	Oui	Non
unload	Oui	Oui
url : String	Oui	Oui
width : int	Oui	Oui

LocalConnection

Le tableau ci-dessous recense les membres de la classe LocalConnection pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
allowDomain(... domains):void	Oui	Oui
allowInsecureDomain(... domains):void	Oui	Oui
asyncError	Oui	Oui
client : Object	Oui	Oui
close():void	Oui	Oui
connect(connectionName:String):void	Oui	Oui
domain : String	Oui	Oui
isPerUser : Boolean	Oui	Non
isSupported : Boolean	Oui	Non
LocalConnection()	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
securityError	Oui	Oui
send(connectionName:String, methodName:String, ... arguments):void	Oui	Oui
status	Oui	Oui

Math

Le tableau ci-dessous recense les membres de la classe Math pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
abs(val:Number):Number	Oui	Oui
acos(val:Number):Number	Oui	Oui
asin(val:Number):Number	Oui	Oui
atan(val:Number):Number	Oui	Oui
atan2(y:Number, x:Number):Number	Oui	Oui
ceil(val:Number):Number	Oui	Oui
cos(angleRadians:Number):Number	Oui	Oui
E : Number = 2,71828182845905	Oui	Oui
exp(val:Number):Number	Oui	Oui
floor(val:Number):Number	Oui	Oui
LN10 : Number = 2,302585092994046	Oui	Oui
LN2 : Number = 0,6931471805599453	Oui	Oui
log(val:Number):Number	Oui	Oui
LOG10E : Number = 0,4342944819032518	Oui	Oui
LOG2E : Number = 1,442695040888963387	Oui	Oui
max(val1:Number, val2:Number, ... rest):Number	Oui	Oui
min(val1:Number, val2:Number, ... rest):Number	Oui	Oui
PI : Number = 3,141592653589793	Oui	Oui
pow(base:Number, pow:Number):Number	Oui	Oui
random():Number	Oui	Oui
round(val:Number):Number	Oui	Oui
sin(angleRadians:Number):Number	Oui	Oui
sqrt(val:Number):Number	Oui	Oui
SQRT1_2 : Number = 0,7071067811865476	Oui	Oui
SQRT2 : Number = 1,4142135623730951	Oui	Oui
tan(angleRadians:Number):Number	Oui	Oui

Matrix

Le tableau ci-dessous recense les membres de la classe Matrix pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
a : Number	Oui	Oui
b : Number	Oui	Oui
c : Number	Oui	Oui
clone():Matrix	Oui	Oui
concat(m:Matrix):void	Oui	Oui
createBox(scaleX:Number, scaleY:Number, rotation:Number = 0, tx:Number = 0, ty:Number = 0):void	Oui	Oui
createGradientBox(width:Number, height:Number, [rotation:Number], [tx:Number], [ty:Number]) : Void	Oui	Oui
d : Number	Oui	Oui
deltaTransformPoint(point:Point):Point	Oui	Oui
identity():void	Oui	Oui
invert():void	Oui	Oui
Matrix(a:Number = 1, b:Number = 0, c:Number = 0, d:Number = 1, tx:Number = 0, ty:Number = 0)	Oui	Oui
rotate(angle:Number):void	Oui	Oui
scale(sx:Number, sy:Number):void	Oui	Oui
toString():String	Oui	Oui
transformPoint(point:Point):Point	Oui	Oui
translate(dx:Number, dy:Number):void	Oui	Oui
tx : Number	Oui	Oui
ty : Number	Oui	Oui

MemoryError

Le tableau ci-dessous recense les membres de la classe MemoryError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
MemoryError(message:String = "")	Oui	Oui

MorphShape

Pour plus d'informations sur la classe MorphShape que prennent en charge ActionScript 3.0 et Flash Lite 4, voir *Adobe ActionScript 3 Reference for the Adobe Flash Platform*.

MouseAutomationAction

Le tableau ci-dessous recense les membres de la classe MouseAutomationAction pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
delta : int	Oui	Oui
MIDDLE_MOUSE_DOWN : String = "middleMouseDown"	Oui	Oui
MIDDLE_MOUSE_UP : String = "middleMouseUp"	Oui	Oui
MouseAutomationAction(type:String, stageX:Number = 0, stageY:Number = 0, delta:int = 0)	Oui	Oui
MOUSE_DOWN : String = "mouseDown"	Oui	Oui
MOUSE_MOVE : String = "mouseMove"	Oui	Oui
MOUSE_UP : String = "mouseUp"	Oui	Oui
MOUSE_WHEEL : String = "mouseWheel"	Oui	Oui
RIGHT_MOUSE_DOWN : String = "rightMouseDown"	Oui	Oui
RIGHT_MOUSE_UP : String = "rightMouseUp"	Oui	Oui
stageX : Number	Oui	Oui
stageY : Number	Oui	Oui

MouseEvent

Le tableau ci-dessous recense les membres de la classe MouseEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
altKey : Boolean	Oui	Oui
buttonDown : Boolean	Oui	Oui
CLICK : String = "click"	Oui	Oui
clone():Event	Oui	Oui
ctrlKey : Boolean	Oui	Oui
delta : int	Oui	Oui
DOUBLE_CLICK : String = "doubleClick"	Oui	Oui
isRelatedObjectInaccessible : Boolean	Oui	Oui
localX : Number	Oui	Oui
localY : Number	Oui	Oui
MouseEvent(type:String, bubbles:Boolean = true, cancelable:Boolean = false, localX:Number = NaN, localY:Number = NaN, relatedObject:InteractiveObject = null, ctrlKey:Boolean = false, altKey:Boolean = false, shiftKey:Boolean = false, buttonDown:Boolean = false, delta:int = 0)	Oui	Oui
MOUSE_DOWN : String = "mouseDown"	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
MOUSE_MOVE : String = "mouseMove"	Oui	Oui
MOUSE_OUT : String = "mouseOut"	Oui	Oui
MOUSE_OVER : String = "mouseOver"	Oui	Oui
MOUSE_UP : String = "mouseUp"	Oui	Oui
MOUSE_WHEEL : String = "mouseWheel"	Oui	Non
relatedObject : InteractiveObject	Oui	Oui
ROLL_OUT : String = "rollOut"	Oui	Oui
ROLL_OVER : String = "rollOver"	Oui	Oui
shiftKey : Boolean	Oui	Oui
stageX : Number	Oui	Oui
stageY : Number	Oui	Oui
toString():String	Oui	Oui
updateAfterEvent():void	Oui	Non

MovieClip

Le tableau ci-dessous recense les membres de la classe MovieClip pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
currentFrame : int	Oui	Oui
currentFrameLabel : String	Oui	Oui
currentLabel : String	Oui	Oui
currentLabels : Array	Oui	Oui
currentScene : Scene	Oui	Oui
enabled : Boolean	Oui	Oui
framesLoaded : int	Oui	Oui
gotoAndPlay(frame:Object, scene:String = null):void	Oui	Oui
gotoAndStop(frame:Object, scene:String = null):void	Oui	Oui
MovieClip()	Oui	Oui
nextFrame():void	Oui	Oui
nextScene():void	Oui	Oui
play():void	Oui	Oui
prevFrame():void	Oui	Oui
prevScene():void	Oui	Oui
scenes : Array	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
stop():void	Oui	Oui
totalFrames : int	Oui	Oui
trackAsMenu : Boolean	Oui	Non

Multitouch

Le tableau ci-dessous recense les membres de la classe Multitouch pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
inputMode : String	Oui	Oui
maxTouchPoints : int	Oui	Oui
supportedGestures : Vector.<String>	Oui	Oui
supportsGestureEvents : Boolean	Oui	Oui
supportsTouchEvent : Boolean	Oui	Oui

MultitouchInputMode

Le tableau ci-dessous recense les membres de la classe MultitouchInputMode pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
GESTURE : String = "gesture"	Oui	Oui
NONE : String = "none"	Oui	Oui
TOUCH_POINT : String = "touchPoint"	Oui	Oui

Namespace

Le tableau ci-dessous recense les membres de la classe Namespace pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
Namespace(prefixValue:*, uriValue:*)	Oui	Oui
Namespace(uriValue:*)	Oui	Oui
prefix : String	Oui	Oui
toString():String	Oui	Oui
uri : String	Oui	Oui
valueOf():String	Oui	Oui

NetConnection

Le tableau ci-dessous recense les membres de la classe NetConnection pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
asyncError	Oui	Oui
addHeader(operation:String, mustUnderstand:Boolean = false, param:Object = null):void	Oui	Oui
call(command:String, responder:Responder, ... arguments):void	Oui	Oui
client : Object	Oui	Oui
close():void	Oui	Oui
connect(command:String, ... arguments):void	Oui	Oui
connected : Boolean	Oui	Oui
connectedProxyType : String	Oui	Oui
defaultObjectEncoding : uint	Oui	Oui
farID : String	Oui	Non
farNonce : String	Oui	Non
ioError	Oui	Oui
maxPeerConnections : uint	Oui	Non
nearID : String	Oui	Non
nearNonce : String	Oui	Non
NetConnection()	Oui	Oui
netStatus	Oui	Oui
objectEncoding : uint	Oui	Oui
protocol : String	Oui	Non
proxyType : String	Oui	Oui
securityError	Oui	Oui
unconnectedPeerStreams : Array	Oui	Non
uri : String	Oui	Oui
usingTLS : Boolean	Oui	Oui

NetGroup

Le tableau ci-dessous recense les membres de la classe NetGroup pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
estimatedMemberCount : Number	Oui	Non
info : NetGroupInfo	Oui	Non
localCoverageFrom : String	Oui	Non
localCoverageTo : String	Oui	Non
neighborCount : Number	Oui	Non

Nom du membre	ActionScript 3.0	Flash Lite 4
receiveMode : String	Oui	Non
replicationStrategy : String	Oui	Non
NetGroup(connection:NetConnection, groupspec:String)	Oui	Non
addHaveObjects(startIndex:Number, endIndex:Number):void	Oui	Non
addMemberHint(peerID:String):Boolean	Oui	Non
addNeighbor(peerID:String):Boolean	Oui	Non
addWantObjects(startIndex:Number, endIndex:Number):void	Oui	Non
close():void	Oui	Non
convertPeerIDToGroupAddress(peerID:String):String	Oui	Non
denyRequestedObject(requestID:int):void	Oui	Non
post(message:Object):String	Oui	Non
removeHaveObjects(startIndex:Number, endIndex:Number):void	Oui	Non
removeWantObjects(startIndex:Number, endIndex:Number):void	Oui	Non
sendToAllNeighbors(message:Object):String	Oui	Non
sendToNearest(message:Object, groupAddress:String):String	Oui	Non
sendToNeighbor(message:Object, sendMode:String):String	Oui	Non
writeRequestedObject(requestID:int, object:Object):void	Oui	Non

NetGroupInfo

Le tableau ci-dessous recense les membres de la classe NetGroupInfo pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
objectReplicationReceiveBytesPerSecond : Number	Oui	Non
objectReplicationSendBytesPerSecond : Number	Oui	Non
postingReceiveControlBytesPerSecond : Number	Oui	Non
postingReceiveDataBytesPerSecond : Number	Oui	Non
postingSendControlBytesPerSecond : Number	Oui	Non
postingSendDataBytesPerSecond : Number	Oui	Non
routingReceiveBytesPerSecond : Number	Oui	Non
routingSendBytesPerSecond : Number	Oui	Non
toString():String	Oui	Non

NetGroupReceiveMode

Le tableau ci-dessous recense les membres de la classe NetGroupReceiveMode pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
EXACT : String = "exact"	Oui	Non
NEAREST : String = "nearest"	Oui	Non

NetGroupReplicationStrategy

Le tableau ci-dessous recense les membres de la classe NetGroupReplicationStrategy pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
LOWEST_FIRST : String = "lowestFirst"	Oui	Non
RAREST_FIRST : String = "rarestFirst"	Oui	Non

NetGroupSendMode

Le tableau ci-dessous recense les membres de la classe NetGroupSendMode pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
NEXT_DECREASING : String = "nextDecreasing"	Oui	Non
NEXT_INCREASING : String = "nextIncreasing"	Oui	Non

NetGroupSendResult

Le tableau ci-dessous recense les membres de la classe NetGroupSendResult pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ERROR : String = "error"	Oui	Non
NO_ROUTE : String = "no route"	Oui	Non
SENT : String = "sent"	Oui	Non

NetStatusEvent

Le tableau ci-dessous recense les membres de la classe NetStatusEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Oui
info : Object	Oui	Oui
NetStatusEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, info:Object = null)	Oui	Oui
NET_STATUS : String = "netStatus"	Oui	Oui
toString():String	Oui	Oui

NetStream

Le tableau ci-dessous recense les membres de la classe NetStream pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
asyncError	Oui	Oui
attach(connection:NetConnection) : void	Oui	Non
attachAudio	Oui	Non
attachCamera	Oui	Non
audioReliable : Boolean	Oui	Non
audioSampleAccess : Boolean	Oui	Non
backBufferLength : Number	Oui	Non
backBufferTime : Number	Oui	Non
bufferLength : Number	Oui	Oui
bufferTime : Number	Oui	Oui
bufferTimeMax : Number	Oui	Non
bytesLoaded : uint	Oui	Oui
bytesTotal : uint	Oui	Oui
checkPolicyFile : Boolean	Oui	Non
client : Object	Oui	Oui
close() : Void	Oui	Oui
CONNECT_TO_FMS : String = "connectToFMS"	Oui	Oui
currentFps : Number	Oui	Oui
dataReliable : Boolean	Oui	Non
DIRECT_CONNECTIONS : String = "directConnections"	Oui	Oui
farID : String	Oui	Non
farNonce : String	Oui	Non
inBufferSeek : Boolean	Oui	Non
info : NetStreamInfo	Oui	Oui
ioError	Oui	Oui
liveDelay : Number	Oui	Oui
maxPauseBufferTime : Number	Oui	Oui
multicastAvailabilitySendToAll : Boolean	Oui	Non
multicastAvailabilityUpdatePeriod : Number	Oui	Non
multicastFetchPeriod : Number	Oui	Non
multicastInfo : NetStreamMulticastInfo	Oui	Non
multicastPushNeighborLimit : Number	Oui	Non

Nom du membre	ActionScript 3.0	Flash Lite 4
multicastRelayMarginDuration : Number	Oui	Non
multicastWindowDuration : Number	Oui	Non
nearNonce : String	Oui	Oui
netStatus	Oui	Oui
objectEncoding : uint	Oui	Oui
onPeerConnect(subscriber:NetStream):Boolean	Oui	Oui
onCuePoint	Oui	Oui
onImageData	Oui	Oui
onMetaData	Oui	Oui
onPlayStatus	Oui	Oui
onPlayStatus	Oui	Oui
onXMPData	Oui	Oui
pause():void	Oui	Oui
play(... arguments):void	Oui	Oui
peerStreams : Array	Oui	Non
play2(param:NetStreamPlayOptions):void	Oui	Oui
publish(name:String = null, type:String = null):void	Oui	Non
receiveAudio(flag:Boolean):void	Oui	Oui
receiveVideo(flag:Boolean):void	Oui	Oui
receiveVideoFPS(FPS:Number):void	Oui	Oui
resume():void	Oui	Oui
seek(offset:Number):void	Oui	Oui
send(handlerName:String, ... arguments):void	Oui	Non
soundTransform : flash.media:SoundTransform	Oui	Non
step(frames:int):void	Oui	Non
time : Number	Oui	Oui
togglePause():void	Oui	Oui
videoReliable : Boolean	Oui	Non
videoSampleAccess : Boolean	Oui	Non

NetStreamInfo

Le tableau ci-dessous recense les membres de la classe NetStreamInfo pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
audioBufferByteLength : Number	Oui	Oui
audioBufferLength : Number	Oui	Oui
audioByteCount : Number	Oui	Oui
audioBytesPerSecond : Number	Oui	Oui
audioLossRate : Number	Oui	Oui
byteCount : Number	Oui	Oui
currentBytesPerSecond : Number	Oui	Oui
dataBufferByteLength : Number	Oui	Oui
dataBufferLength : Number	Oui	Oui
dataByteCount : Number	Oui	Oui
dataBytesPerSecond : Number"	Oui	Oui
droppedFrames : Number	Oui	Oui
NetStreamInfo(curBPS:Number, byteCount:Number, maxBPS:Number, audioBPS:Number, audioByteCount:Number, videoBPS:Number, videoByteCount:Number, dataBPS:Number, dataByteCount:Number, playbackBPS:Number, droppedFrames:Number, audioBufferByteLength:Number, videoBufferByteLength:Number, dataBufferByteLength:Number, audioBufferLength:Number, videoBufferLength:Number, dataBufferLength:Number, srtt:Number, audioLossRate:Number)	Oui	Oui
maxBytesPerSecond : Number	Oui	Oui
playbackBytesPerSecond : Number	Oui	Oui
toString():String	Oui	Oui
SRTT : Number	Oui	Oui
videoBufferByteLength : Number	Oui	Oui
videoBufferLength : Number	Oui	Oui
videoByteCount : Number	Oui	Oui
videoBytesPerSecond : Number	Oui	Oui
videoLossRate : Number	Oui	Non

NetStreamPlayOptions

Le tableau ci-dessous recense les membres de la classe NetStreamPlayOptions pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
len : Number	Oui	Oui
NetStreamPlayOptions()	Oui	Oui
oldStreamName : String	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
start : Number	Oui	Oui
streamName : String	Oui	Oui
transition : String	Oui	Oui

NetStreamPlayTransitions

Le tableau ci-dessous recense les membres de la classe NetStreamPlayTransitions pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
APPEND : * = append	Oui	Oui
RESET : * = reset	Oui	Oui
STOP : * = stop	Oui	Oui
SWAP : * = swap	Oui	Oui
SWITCH : * = switch	Oui	Oui

Number

Le tableau ci-dessous recense les membres de la classe Number pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
MAX_VALUE : Number	Oui	Oui
MIN_VALUE : Number	Oui	Oui
NaN : Number	Oui	Oui
NEGATIVE_INFINITY : Number	Oui	Oui
Number(num:Object)	Oui	Oui
POSITIVE_INFINITY : Number	Oui	Oui
toExponential(fractionDigits:uint):String	Oui	Oui
toFixed(fractionDigits:uint):String	Oui	Oui
toPrecision(precision:uint):String	Oui	Oui
toString(radix:Number) : String	Oui	Oui
valueOf() : Number	Oui	Oui

Object

Le tableau ci-dessous recense les membres de la classe Object pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
constructor : Object	Oui	Oui
hasOwnProperty(name:String):Boolean	Oui	Oui
isPrototypeOf(theClass:Object):Boolean	Oui	Oui
Object()	Oui	Oui
propertyIsEnumerable(name:String):Boolean	Oui	Oui
prototype : Object	Oui	Oui
prototype:Object	Oui	Oui
setPropertyIsEnumerable(name:String, isEnum:Boolean = true):void	Oui	Oui
toLocaleString():String	Oui	Oui
toString():String	Oui	Oui
valueOf():Object	Oui	Oui

ObjectEncoding

Le tableau ci-dessous recense les membres de la classe ObjectEncoding pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
AMF0 : uint = 0	Oui	Oui
AMF3 : uint = 3	Oui	Oui
DEFAULT : uint = 3	Oui	Oui
dynamicPropertyWriter : IDynamicPropertyWriter	Oui	Oui

PixelSnapping

Le tableau ci-dessous recense les membres de la classe PixelSnapping pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ALWAYS : String = "always"	Oui	Oui
AUTO : String = "auto"	Oui	Oui
NEVER : String = "never"	Oui	Oui

Point

Le tableau ci-dessous recense les membres de la classe Point pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
add(v:Point) : Point	Oui	Oui
clone() : Point	Oui	Oui
distance(pt1:Point, pt2:Point):Number	Oui	Oui
equals(toCompare:Point):Boolean	Oui	Oui
interpolate(pt1:Point, pt2:Point, f:Number):Point	Oui	Oui
length : Number	Oui	Oui
normalize(thickness:Number):void	Oui	Oui
offset(dx:Number, dy:Number):void	Oui	Oui
Point(x:Number = 0, y:Number = 0)	Oui	Oui
polar(len:Number, angle:Number):Point	Oui	Oui
subtract(v:Point):Point	Oui	Oui
toString():String	Oui	Oui
x : Number	Oui	Oui
y : Number	Oui	Oui

PressAndTapGestureEvent

Le tableau ci-dessous recense les membres de la classe PressAndTapGestureEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Oui
GESTURE_PRESS_AND_TAP : String = "gesturePressAndTap"	Oui	Oui
PressAndTapGestureEvent(type:String, bubbles:Boolean = true, cancelable:Boolean = false, phase:String = null, localX:Number = 0, localY:Number = 0, tapLocalX:Number = 0, tapLocalY:Number = 0, ctrlKey:Boolean = false, altKey:Boolean = false, shiftKey:Boolean = false)	Oui	Oui
tapLocalX : Number	Oui	Oui
tapLocalY : Number	Oui	Oui
tapStageX : Number	Oui	Oui
tapStageY : Number	Oui	Oui
toString():String	Oui	Oui

ProgressEvent

Le tableau ci-dessous recense les membres de la classe ProgressEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
bytesLoaded : uint	Oui	Oui
bytesTotal : uint	Oui	Oui
clone():Event	Oui	Oui
PROGRESS : String = "progress"	Oui	Oui
ProgressEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, bytesLoaded:uint = 0, bytesTotal:uint = 0)	Oui	Oui
SOCKET_DATA : String = "socketData"	Oui	Oui
toString():String	Oui	Oui

Proxy

Le tableau ci-dessous recense les membres de la classe Proxy pris en charge par ActionScript 3.0 et Flash Lite 4.

Remarque : les membres répertoriés dans le tableau suivant sont tous des méthodes protégées.

Nom du membre	ActionScript 3.0	Flash Lite 4
callProperty(name:*, ... rest):*	Oui	Oui
deleteProperty(name:*)Boolean	Oui	Oui
getDescendants(name:*):*	Oui	Oui
getProperty(name:*):*	Oui	Oui
hasProperty(name:*)Boolean	Oui	Oui
isAttribute(name:*)Boolean	Oui	Oui
nextName(index:int):String	Oui	Oui
nextNameIndex(index:int):int	Oui	Oui
nextValue(index:int):*	Oui	Oui
setProperty(name:*, value:*)void	Oui	Oui
valueOf():Object	Oui	Oui

QName

Le tableau ci-dessous recense les membres de la classe QName pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
localName : String	Oui	Oui
QName(qname:QName)	Oui	Oui
QName(uri:Namespace, localName:QName)	Oui	Oui
toString():String	Oui	Oui
uri : String	Oui	Oui
valueOf():QName	Oui	Oui

RangeError

Le tableau ci-dessous recense les membres de la classe RangeError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
RangeError(message:String = "")	Oui	Oui

Rectangle

Le tableau ci-dessous recense les membres de la classe Rectangle pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
bottom : Number	Oui	Oui
bottomRight : Point	Oui	Oui
clone():Rectangle	Oui	Oui
contains(x:Number, y:Number):Boolean	Oui	Oui
containsPoint(point:Point):Boolean	Oui	Oui
containsRect(rect:Rectangle):Boolean	Oui	Oui
equals(toCompare:Rectangle):Boolean	Oui	Oui
height : Number	Oui	Oui
inflate(dx:Number, dy:Number):void	Oui	Oui
inflatePoint(point:Point):void	Oui	Oui
intersection(toIntersect:Rectangle):Rectangle	Oui	Oui
intersects(toIntersect:Rectangle):Boolean	Oui	Oui
isEmpty():Boolean	Oui	Oui
left : Number	Oui	Oui
offset(dx:Number, dy:Number):void	Oui	Oui
offsetPoint(point:Point):void	Oui	Oui
Rectangle(x:Number = 0, y:Number = 0, width:Number = 0, height:Number = 0)	Oui	Oui
right : Number	Oui	Oui
setEmpty():void	Oui	Oui
size : Point	Oui	Oui
top : Number	Oui	Oui
topLeft : Point	Oui	Oui
toString():String	Oui	Oui
union(toUnion:Rectangle):Rectangle	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
width : Number	Oui	Oui
x : Number	Oui	Oui
y : Number	Oui	Oui

ReferenceError

Le tableau ci-dessous recense les membres de la classe ReferenceError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ReferenceError(message:String = "")	Oui	Oui

RegExp

Le tableau ci-dessous recense les membres de la classe RegExp pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
dotall : Boolean	Oui	Oui
exec(str:String):Object	Oui	Oui
extended : Boolean	Oui	Oui
global : Boolean	Oui	Oui
ignoreCase : Boolean	Oui	Oui
lastIndex : Number	Oui	Oui
multiline : Boolean	Oui	Oui
RegExp(re:String, flags:String)	Oui	Oui
source : String	Oui	Oui
test(str:String):Boolean	Oui	Oui

RenderingMode

Le tableau ci-dessous recense les membres de la classe RenderingMode pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
CFF : String = "cff"	Oui	Oui
NORMAL : String = "normal"	Oui	Oui

Responder

Le tableau ci-dessous recense les membres de la classe Responder pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
constructResponderObject(result:Function, status:Function = null):void	Oui	Oui
Responder(result:Function, status:Function = null)	Oui	Oui

Scene

Le tableau ci-dessous recense les membres de la classe Scene pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
labels : Array	Oui	Oui
name : String	Oui	Oui
numFrames : int	Oui	Oui

ScriptTimeoutError

Le tableau ci-dessous recense les membres de la classe ScriptTimeoutError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ScriptTimeoutError(message:String = "")	Oui	Oui

Security

Le tableau ci-dessous recense les membres de la classe Security pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
allowDomain(... domains):void	Oui	Oui
allowInsecureDomain(... domains):void	Oui	Oui
exactSettings : Boolean	Oui	Oui
loadPolicyFile(url:String):void	Oui	Oui
LOCAL_TRUSTED : String = "localTrusted"	Oui	Oui
LOCAL_WITH_FILE : String = "localWithFile"	Oui	Oui
LOCAL_WITH_NETWORK : String = "localWithNetwork"	Oui	Oui
REMOTE : String = "remote"	Oui	Oui
sandboxType : String	Oui	Oui
showSettings(panel:String = "default"):void	Oui	Oui

SecurityDomain

Le tableau ci-dessous recense les membres de la classe SecurityDomain pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
currentDomain : SecurityDomain	Oui	Oui

SecurityError

Le tableau ci-dessous recense les membres de la classe SecurityError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
SecurityError(message:String = "")	Oui	Oui

SecurityErrorEvent

Le tableau ci-dessous recense les membres de la classe SecurityErrorEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Oui
SecurityErrorEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, text:String = "")	Oui	Oui
SECURITY_ERROR : String = "securityError"	Oui	Oui
toString():String	Oui	Oui

SecurityPanel

Le tableau ci-dessous recense les membres de la classe SecurityPanel pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
CAMERA : String = "camera"	Oui	Oui
DEFAULT : String = "default"	Oui	Oui
DISPLAY : String = "display"	Oui	Oui
LOCAL_STORAGE : String = "localStorage"	Oui	Oui
MICROPHONE : String = "microphone"	Oui	Oui
PRIVACY : String = "privacy"	Oui	Oui
SETTINGS_MANAGER : String = "settingsManager"	Oui	Oui

Shape

Le tableau ci-dessous recense les membres de la classe Shape pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
graphics : Graphics	Oui	Oui
Shape()	Oui	Oui

SharedObject

Le tableau ci-dessous recense les membres de la classe SharedObject pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
asyncError	Oui	Oui
client : Object	Oui	Oui
clear():void	Oui	Oui
close():void	Oui	Oui
connect(myConnection:NetConnection, params:String = null):void	Oui	Oui
data : Object	Oui	Oui
defaultObjectEncoding : uint	Oui	Oui
flush(minDiskSpace:int = 0):String	Oui	Oui
fps : Number	Oui	Oui
getLocal(name:String, localPath:String = null, secure:Boolean = false):SharedObject	Oui	Oui
getRemote(name:String, remotePath:String = null, persistence:Object = false, secure:Boolean = false):SharedObject	Non	Oui
netStatus	Oui	Oui
objectEncoding : uint	Oui	Oui
send(... arguments):void	Oui	Oui
setDirty(propertyName:String):void	Oui	Oui
setProperty(propertyName:String, value:Object = null):void	Oui	Oui
size : uint	Oui	Oui
sync	Oui	Oui

SharedObjectFlushStatus

Le tableau ci-dessous recense les membres de la classe SharedObjectFlushStatus pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
FLUSHED : String = "flushed"	Oui	Oui
PENDING : String = "pending"	Oui	Oui

SimpleButton

Le tableau ci-dessous recense les membres de la classe SimpleButton pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
downState : DisplayObject	Oui	Oui
enabled : Boolean	Oui	Oui
hitTestState : DisplayObject	Oui	Oui
overState : DisplayObject	Oui	Oui
SimpleButton(upState:DisplayObject = null, overState:DisplayObject = null, downState:DisplayObject = null, hitTestState:DisplayObject = null)	Oui	Oui
soundTransform : flash.media:SoundTransform	Oui	Non
trackAsMenu : Boolean	Oui	Non
upState : DisplayObject	Oui	Oui
useHandCursor : Boolean	Oui	Non

Socket

Le tableau ci-dessous recense les membres de la classe Socket pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
bytesAvailable : uint	Oui	Oui
close	Oui	Oui
close():void	Oui	Oui
connect	Oui	Oui
connect(host:String, port:int):void	Oui	Oui
connected : Boolean	Oui	Oui
endian : String	Oui	Oui
flush():void	Oui	Oui
ioError	Oui	Oui
objectEncoding : uint	Oui	Oui
readBoolean():Boolean	Oui	Oui
readByte():int	Oui	Oui
readBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void	Oui	Oui
readDouble():Number	Oui	Oui
readFloat():Number	Oui	Oui
readInt():int	Oui	Oui
readMultiByte(length:uint, charSet:String):String	Oui	Oui
readObject():*	Oui	Oui
readShort():int	Oui	Oui
readUnsignedByte():uint	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
readUnsignedInt():uint	Oui	Oui
readUnsignedShort():uint	Oui	Oui
readUTF():String	Oui	Oui
readUTFBytes(length:uint):String	Oui	Oui
securityError	Oui	Oui
Socket(host:String = null, port:int = 0)	Oui	Oui
socketData	Oui	Oui
timeout : uint	Oui	Oui
writeBoolean(value:Boolean):void	Oui	Oui
writeByte(value:int):void	Oui	Oui
writeBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void	Oui	Oui
writeDouble(value:Number):void	Oui	Oui
writeFloat(value:Number):void	Oui	Oui
writeInt(value:int):void	Oui	Oui
writeMultiByte(value:String, charSet:String):void	Oui	Oui
writeObject(object:*) :void	Oui	Oui
writeShort(value:int):void	Oui	Oui
writeUnsignedInt(value:uint):void	Oui	Oui
writeUTF(value:String):void	Oui	Oui
writeUTFBytes(value:String):void	Oui	Oui

Sound

Le tableau ci-dessous recense les membres de la classe Sound pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
bytesLoaded : uint	Oui	Oui
bytesTotal : int	Oui	Oui
close():void	Oui	Oui
complete	Oui	Oui
extract(target:ByteArray, length:Number, startPosition:Number = -1):Number	Oui	Non
id3 : Event	Oui	Oui
id3 : ID3Info	Partielle	Partielle
isBuffering : Boolean	Oui	Oui
ioError	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
length : Number	Oui	Oui
load(stream:URLRequest, context:SoundLoaderContext = null):void	Oui	Oui
open	Oui	Oui
play(startTime:Number = 0, loops:int = 0, sndTransform:flash.media.SoundTransform = null):SoundChannel	Oui	Oui
progress	Oui	Oui
sampleData	Oui	Non
Sound(stream:URLRequest = null, context:SoundLoaderContext = null)	Oui	Oui
url : String	Oui	Oui

SoundChannel

Le tableau ci-dessous recense les membres de la classe SoundChannel pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
leftPeak : Number	Oui	Oui
position : Number	Oui	Oui
rightPeak : Number	Oui	Oui
soundComplete	Oui	Oui
soundTransform : flash.media.SoundTransform	Oui	Oui
stop():void	Oui	Oui

SoundLoaderContext

Le tableau ci-dessous recense les membres de la classe SoundLoaderContext pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
bufferTime : Number = 1000	Oui	Oui
checkPolicyFile : Boolean = false	Oui	Oui
SoundLoaderContext(bufferTime:Number = 1000, checkPolicyFile:Boolean = false)	Oui	Oui

SoundMixer

Le tableau ci-dessous recense les membres de la classe SoundMixer pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
areSoundsInaccessible():Boolean	Oui	Non
bufferTime : int	Oui	Oui
computeSpectrum(outputArray:ByteArray, FFTMode:Boolean = false, stretchFactor:int = 0):void	Oui	Non
stopAll():void	Oui	Oui

SoundTransform

Le tableau ci-dessous recense les membres de la classe SoundTransform pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
leftToLeft : Number	Oui	Oui
leftToRight : Number	Oui	Oui
pan : Number	Oui	Oui
rightToLeft : Number	Oui	Oui
rightToRight : Number	Oui	Oui
SoundTransform(vol:Number = 1, panning:Number = 0)	Oui	Oui
volume : Number	Oui	Oui

SpaceJustifier

Le tableau ci-dessous recense les membres de la classe SpaceJustifier pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():flash.text.engine.TextJustifier	Oui	Oui
letterSpacing : Boolean	Oui	Oui
maximumSpacing : Number	Oui	Non
minimumSpacing : Number	Oui	Non
optimumSpacing : Number	Oui	Non
SpaceJustifier(locale:String = "en", lineJustification:String = "unjustified", letterSpacing:Boolean = false)	Oui	Oui
volume : Number	Oui	Oui

Sprite

Le tableau ci-dessous recense les membres de la classe Sprite pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
buttonMode : Boolean	Oui	Oui
dropTarget : DisplayObject	Oui	Oui
graphics : Graphics	Oui	Oui
hitArea : Sprite	Oui	Oui
soundTransform : flash.media:SoundTransform	Oui	Oui
Sprite()	Oui	Oui
startDrag(lockCenter:Boolean = false, bounds:Rectangle = null):void	Oui	Oui
startTouchDrag(touchPointID:int, lockCenter:Boolean = false, bounds:Rectangle = null):void	Oui	Oui
stopDrag():void	Oui	Oui
stopTouchDrag(touchPointID:int):void	Oui	Oui
useHandCursor : Boolean	Oui	Non

StackOverflowError

Le tableau ci-dessous recense les membres de la classe StackOverflowError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
StackOverflowError(message:String = "")	Oui	Oui

Stage

Le tableau ci-dessous recense les membres de la classe Stage pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
addChild(child:DisplayObject):DisplayObject	Oui	Oui
addChildAt(child:DisplayObject, index:int):DisplayObject	Oui	Oui
addEventListener(type:String, listener:Function, useCapture:Boolean = false, priority:int = 0, useWeakReference:Boolean = false):void	Oui	Oui
align : String	Oui	Oui
assignFocus(objectToFocus:InteractiveObject, direction:String):void	Oui	Non
colorCorrection : String		Non
colorCorrectionSupport : String		Non
dispatchEvent(event:Event):Boolean	Oui	Oui
displayState : String	Oui	Oui
focus : InteractiveObject	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
frameRate : Number	Oui	Oui
fullScreen	Oui	Oui
fullScreenHeight : uint	Oui	Oui
fullScreenSourceRect : Rectangle	Oui	Oui
fullScreenWidth : uint	Oui	Oui
hasEventListener(type:String):Boolean	Oui	Oui
height : Number	Oui	Oui
isFocusInaccessible():Boolean	Oui	Oui
mouseChildren : Boolean	Oui	Oui
mouseLeave	Oui	Oui
numChildren : int	Oui	Oui
quality : String	Oui	Oui
removeChildAt(index:int):DisplayObject	Oui	Oui
resize	Oui	Oui
scaleMode : String	Oui	Oui
setChildIndex(child:DisplayObject, index:int):void	Oui	Oui
showDefaultContextMenu : Boolean	Oui	Non
stageFocusRect : Boolean	Oui	Oui
stageHeight : int	Oui	Oui
stageWidth : int	Oui	Oui
swapChildrenAt(index1:int, index2:int):void	Oui	Oui
tabChildren : Boolean	Oui	Non
textSnapshot : flash.text:TextSnapshot	Oui	Non
width : Number	Oui	Oui
willTrigger(type:String):Boolean	Oui	Non

StageAlign

Le tableau ci-dessous recense les membres de la classe StageAlign pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
BOTTOM : String = "B"	Oui	Oui
BOTTOM_LEFT : String = "BL"	Oui	Oui
BOTTOM_RIGHT : String = "BR"	Oui	Oui
LEFT : String = "L"	Oui	Oui
RIGHT : String = "R"	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
TOP : String = "T"	Oui	Oui
TOP_LEFT : String = "TL"	Oui	Oui
TOP_RIGHT : String = "TR"	Oui	Oui

StageCapture

Le tableau ci-dessous recense les membres de la classe StageCapture pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
addEventListener(type:String, listener:Function, useCapture:Boolean = false, priority:int = 0, useWeakReference:Boolean = false):void	Oui	Oui
cancel():void	Oui	Oui
capture	Oui	Oui
capture(type:String):void	Oui	Oui
clipRect : Rectangle	Oui	Oui
CURRENT : String = "current"	Oui	Oui
fileNameBase : String	Oui	Oui
MULTIPLE : String = "multiple"	Oui	Oui
NEXT : String = "next"	Oui	Oui
removeEventListener(type:String, listener:Function, useCapture:Boolean = false):void	Oui	Oui

StageCaptureEvent

Le tableau ci-dessous recense les membres de la classe StageCaptureEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
CAPTURE : String = "capture"	Oui	Oui
checksum : uint	Oui	Oui
StageCaptureEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, url:String = null, checksum:uint = 0)	Oui	Oui
url : String	Oui	Oui

StageDisplayState

Le tableau ci-dessous recense les membres de la classe StageDisplayState pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
FULL_SCREEN : String = "fullScreen"	Oui	Oui
NORMAL : String = "normal"	Oui	Oui

StageQuality

Le tableau ci-dessous recense les membres de la classe StageQuality pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
BEST : String = "best"	Oui	Oui
HIGH : String = "high"	Oui	Oui
LOW : String = "low"	Oui	Oui
MEDIUM : String = "medium"	Oui	Oui

StageScaleMode

Le tableau ci-dessous recense les membres de la classe StageScaleMode pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
EXACT_FIT : String = "exactFit"	Oui	Oui
NO_BORDER : String = "noBorder"	Oui	Oui
NO_SCALE : String = "noScale"	Oui	Oui
SHOW_ALL : String = "showAll"	Oui	Oui

StaticText

Le tableau ci-dessous recense les membres de la classe StaticText pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
text : String	Oui	Oui

StatusEvent

Le tableau ci-dessous recense les membres de la classe StatusEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Oui
code : String	Oui	Oui
level : String	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
STATUS : String = "status"	Oui	Oui
StatusEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, code:String = "", level:String = "")	Oui	Oui
toString():String	Oui	Oui

String

Le tableau ci-dessous recense les membres de la classe String pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
charAt(index:Number = 0):String	Oui	Oui
charCodeAt(index:Number = 0):Number	Oui	Oui
concat(... args):String	Oui	Oui
fromCharCode(... charCodes):String	Oui	Oui
indexOf(val:String, startIndex:Number = 0):int	Oui	Oui
lastIndexOf(val:String, startIndex:Number = 0x7FFFFFFF):int	Oui	Oui
length : int	Oui	Oui
localeCompare(other:String, ... values):int	Oui	Oui
match(pattern:*):Array	Oui	Oui
replace(pattern:*, repl:Object):String	Oui	Oui
search(pattern:*):int	Oui	Oui
slice(startIndex:Number = 0, endIndex:Number = 0x7FFFFFFF):String	Oui	Oui
split(delimiter:*, limit:Number = 0x7FFFFFFF):Array	Oui	Oui
String(val:String)	Oui	Oui
substr(startIndex:Number = 0, len:Number = 0x7FFFFFFF):String	Oui	Oui
substring(startIndex:Number = 0, endIndex:Number = 0x7FFFFFFF):String	Oui	Oui
toLocaleLowerCase():String	Oui	Oui
toLocaleUpperCase():String	Oui	Oui
toLowerCase():String	Oui	Oui
toUpperCase():String	Oui	Oui
valueOf():String	Oui	Oui

StyleSheet

Le tableau ci-dessous recense les membres de la classe StyleSheet pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clear():void	Oui	Oui
getStyle(styleName:String):Object	Oui	Oui
parseCSS(CSSText:String):void	Oui	Oui
setStyle(styleName:String, styleObject:Object):void	Oui	Oui
styleNames : Array	Oui	Oui
transform(formatObject:Object):flash.text:TextFormat	Oui	Oui

SWFVersion

Le tableau ci-dessous recense les membres de la classe SWFVersion pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
FLASH1 : uint = 1	Oui	Oui
FLASH10 : uint = 10	Oui	Oui
FLASH2 : uint = 2	Oui	Oui
FLASH3 : uint = 3	Oui	Oui
FLASH4 : uint = 4	Oui	Oui
FLASH5 : uint = 5	Oui	Oui
FLASH6 : uint = 6	Oui	Oui
FLASH7 : uint = 7	Oui	Oui
FLASH8 : uint = 8	Oui	Oui
FLASH9 : uint = 9	Oui	Oui

SyncEvent

Le tableau ci-dessous recense les membres de la classe SyncEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
changeList : Array	Oui	Oui
clone():Event	Oui	Oui
SYNC : String = "sync"	Oui	Oui
SyncEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, changeList:Array = null)	Oui	Oui
toString():String	Oui	Oui

SyntaxError

Le tableau ci-dessous recense les membres de la classe SyntaxError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
SyntaxError(message:String = "")	Oui	Oui

System

Le tableau ci-dessous recense les membres de la classe System pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
disposeXML	Oui	Non
exit(code:uint):void	Oui	Non
freeMemory : Number	Oui	Non
gc():void	Oui	Oui
ime : IME	Oui	Oui
pause():void	Oui	Oui
privateMemory : Number	Oui	Non
resume():void	Oui	Oui
setClipboard(string:String):void	Oui	Non
totalMemory : uint	Oui	Oui
totalMemoryNumber : Number	Oui	Non
useCodePage : Boolean	Oui	Oui

TabAlignment

Le tableau ci-dessous recense les membres de la classe TabAlignment pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
CENTER : String = "center"	Oui	Oui
DECIMAL : String = "decimal"	Oui	Oui
END : String = "end"	Oui	Oui
START : String = "start"	Oui	Oui

TabStop

Le tableau ci-dessous recense les membres de la classe TabStop pris en charge par ActionScript 3.0 et Flash Lite 4.

Member Name	ActionScript 3.0	Flash Lite 4
alignment : String	Oui	Oui
decimalAlignmentToken : String	Oui	Oui
position : Number	Oui	Oui
TabStop(alignment:String = "start", position:Number = 0.0, decimalAlignmentToken:String = "")	Oui	Oui

TextBaseline

Le tableau ci-dessous recense les membres de la classe TextBaseline pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ASCENT : String = "ascent"	Oui	Oui
DESCENT : String = "descent"	Oui	Oui
IDEOGRAPHIC_BOTTOM : String = "ideographicBottom"	Oui	Oui
IDEOGRAPHIC_CENTER : String = "ideographicCenter"	Oui	Oui
IDEOGRAPHIC_TOP : String = "ideographicTop"	Oui	Oui
ROMAN : String = "roman"	Oui	Oui
USE_DOMINANT_BASELINE : String = "useDominantBaseline"	Oui	Oui

TextBlock

Le tableau ci-dessous recense les membres de la classe TextBlock pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
applyNonLinearFontScaling : Boolean	Oui	Oui
baselineFontDescription : FontDescription	Oui	Oui
baselineFontSize : Number	Oui	Oui
baselineZero : String	Oui	Oui
bidilLevel : int	Oui	Oui
content : ContentElement	Oui	Oui
createTextLine(previousLine:flash.text.engine:TextLine = null, width:Number = 1000000, lineOffset:Number = 0.0, fitSomething:Boolean = false):flash.text.engine:TextLine	Oui	Oui
dump():String	Oui	Non
findNextAtomBoundary(afterCharIndex:int):int	Oui	Oui
findNextWordBoundary(afterCharIndex:int):int	Oui	Oui
findPreviousAtomBoundary(beforeCharIndex:int):int	Oui	Oui
findPreviousWordBoundary(beforeCharIndex:int):int	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
firstInvalidLine : flash.text.engine:TextLine	Oui	Oui
firstLine : flash.text.engine:TextLine	Oui	Oui
getTextLineAtCharIndex(charIndex:int):flash.text.engine:TextLine	Oui	Oui
lastLine : flash.text.engine:TextLine	Oui	Oui
lineRotation : String	Oui	Oui
recreateTextLine(textLine:flash.text.engine:TextLine, previousLine:flash.text.engine:TextLine = null, width:Number = 1000000, lineOffset:Number = 0.0, fitSomething:Boolean = false):flash.text.engine:TextLine	Oui	Oui
releaseLineCreationData():void	Oui	Oui
releaseLines(firstLine:flash.text.engine:TextLine, lastLine:flash.text.engine:TextLine):void	Oui	Oui
tabStops : Vector.<flash.text.engine:TabStop>	Oui	Oui
TextBlock(content:ContentElement = null, tabStops:Vector.<flash.text.engine:TabStop> = null, textJustifier:flash.text.engine:TextJustifier = null, lineRotation:String = "rotate0", baselineZero:String = "roman", bidiLevel:int = 0, applyNonLinearFontScaling:Boolean = true, baselineFontDescription:FontDescription = null, baselineFontSize:Number = 12.0)	Oui	Oui
textJustifier : flash.text.engine:TextJustifier	Oui	Oui
textLineCreationResult : String	Oui	Oui
userData : *	Oui	Oui

TextElement

Le tableau ci-dessous recense les membres de la classe TextElement pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
replaceText(beginIndex:int, endIndex:int, newText:String):void	Oui	Oui
text : String	Oui	Oui
TextElement(text:String = null, elementFormat:ElementFormat = null, eventMirror:EventDispatcher = null, textRotation:String = "rotate0")	Oui	Oui

TextEvent

Le tableau ci-dessous recense les membres de la classe TextEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Oui
LINK : String = "link"	Oui	Oui
text : String	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
TextEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, text:String = "")	Oui	Oui
TEXT_INPUT : String = "textInput"	Oui	Oui
toString():String	Oui	Oui

TextField

Le tableau ci-dessous recense les membres de la classe TextField pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
alwaysShowSelection : Boolean	Oui	Oui
antiAliasType : String	Oui	Non
appendText(newText:String):void	Oui	Oui
autoSize : String	Oui	Oui
background : Boolean	Oui	Oui
backgroundColor : uint	Oui	Oui
border : Boolean	Oui	Oui
borderColor : uint	Oui	Oui
bottomScrollV : int	Oui	Oui
caretIndex : int	Oui	Oui
change	Oui	Oui
condenseWhite : Boolean	Oui	Oui
defaultTextFormat : flash.text:TextFormat	Oui	Oui
displayAsPassword : Boolean	Oui	Oui
embedFonts : Boolean	Oui	Oui
getCharBoundaries(charIndex:int):Rectangle	Oui	Oui
getCharIndexAtPoint(x:Number, y:Number):int	Oui	Oui
getFirstCharInParagraph(charIndex:int):int	Oui	Oui
getImageReference(id:String):DisplayObject	Oui	Oui
getLineIndexAtPoint(x:Number, y:Number):int	Oui	Oui
getLineIndexOfChar(charIndex:int):int	Oui	Oui
getLineLength(lineIndex:int):int	Oui	Oui
getLineMetrics(lineIndex:int):flash.text:TextLineMetrics	Oui	Oui
getLineOffset(lineIndex:int):int	Oui	Oui
getLineText(lineIndex:int):String	Oui	Oui
getParagraphLength(charIndex:int):int	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
getTextFormat(beginIndex:int = -1, endIndex:int = -1):flash.text:TextFormat	Oui	Oui
gridFitType : String	Oui	Non
htmlText : String	Oui	Oui
isFontCompatible(fontName:String, fontStyle:String):Boolean	Oui	Oui
length : int	Oui	Oui
link	Oui	Oui
maxChars : int	Oui	Oui
maxScrollH : int	Oui	Oui
maxScrollV : int	Oui	Oui
mouseWheelEnabled : Boolean	Oui	Non
multiline : Boolean	Oui	Oui
numLines : int	Oui	Oui
replaceSelectedText(value:String):void	Oui	Oui
replaceText(beginIndex:int, endIndex:int, newText:String):void	Oui	Oui
restrict : String	Oui	Oui
scroll	Oui	Oui
scrollH : int	Oui	Oui
scrollV : int	Oui	Oui
selectable : Boolean	Oui	Oui
selectionBeginIndex : int	Oui	Oui
selectionEndIndex : int	Oui	Oui
setSelection(beginIndex:int, endIndex:int):void	Oui	Oui
setTextFormat(format:flash.text:TextFormat, beginIndex:int = -1, endIndex:int = -1):void	Oui	Oui
sharpness : Number	Oui	Non
styleSheet : StyleSheet	Oui	Oui
text : String	Oui	Oui
textColor : uint	Oui	Oui
TextField()	Oui	Oui
textHeight : Number	Oui	Oui
textInput	Oui	Oui
textWidth : Number	Oui	Oui
thickness : Number	Oui	Non

Nom du membre	ActionScript 3.0	Flash Lite 4
type : String	Oui	Oui
useRichTextClipboard : Boolean	Oui	Non
wordWrap : Boolean	Oui	Oui

TextFieldAutoSize

Le tableau ci-dessous recense les membres de la classe TextFieldAutoSize pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
CENTER : String = "center"	Oui	Oui
LEFT : String = "left"	Oui	Oui
NONE : String = "none"	Oui	Oui
RIGHT : String = "right"	Oui	Oui

TextFieldType

Le tableau ci-dessous recense les membres de la classe TextFieldType pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
DYNAMIC : String = "dynamic"	Oui	Oui
INPUT : String = "input"	Oui	Oui

TextFormat

Le tableau ci-dessous recense les membres de la classe TextFormat pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
align : String	Oui	Oui
blockIndent : Object	Oui	Oui
bold : Object	Oui	Oui
bullet : Object	Oui	Oui
color : Object	Oui	Oui
font : String	Oui	Oui
indent : Object	Oui	Oui
italic : Object	Oui	Oui
kerning : Object	Oui	Non
leading : Object	Oui	Oui
leftMargin : Object	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
letterSpacing : Object	Oui	Non
rightMargin : Object	Oui	Oui
size : Object	Oui	Oui
tabStops : Array	Oui	Oui
target : String	Oui	Oui
TextFormat(font:String = null, size:Object = null, color:Object = null, bold:Object = null, italic:Object = null, underline:Object = null, url:String = null, target:String = null, align:String = null, leftMargin:Object = null, rightMargin:Object = null, indent:Object = null, leading:Object = null)	Oui	Oui
underline : Object	Oui	Oui
url : String	Oui	Oui

TextFormatAlign

Le tableau ci-dessous recense les membres de la classe TextFormatAlign pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
CENTER : String = "center"	Oui	Oui
JUSTIFY : String = "justify"	Oui	Oui
LEFT : String = "left" NONE : String = "none"	Oui	Oui
RIGHT : String = "right"	Oui	Oui

TextJustifier

Le tableau ci-dessous recense les membres de la classe TextJustifier pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():flash.text.engine:TextJustifier	Oui	Oui
getJustifierForLocale(locale:String):flash.text.engine:TextJustifier	Oui	Oui
lineJustification : String	Oui	Oui
locale : String	Oui	Oui
TextJustifier(locale:String, lineJustification:String)	Oui	Oui

TextLine

Le tableau ci-dessous recense les membres de la classe TextLine pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ascent : Number	Oui	Oui
atomCount : int	Oui	Oui
descent : Number	Oui	Oui
dump():String	Oui	Non
flushAtomData():void	Oui	Oui
getAtomBidiLevel(atomIndex:int):int	Oui	Oui
getAtomBounds(atomIndex:int):Rectangle	Oui	Oui
getAtomCenter(atomIndex:int):Number	Oui	Oui
getAtomGraphic(atomIndex:int):DisplayObject	Oui	Oui
getAtomIndexAtCharIndex(charIndex:int):int	Oui	Oui
getAtomIndexAtPoint(stageX:Number, stageY:Number):int	Oui	Oui
getAtomTextBlockBeginIndex(atomIndex:int):int	Oui	Oui
getAtomTextBlockEndIndex(atomIndex:int):int	Oui	Oui
getAtomTextRotation(atomIndex:int):String	Oui	Oui
getAtomWordBoundaryOnLeft(atomIndex:int):Boolean	Oui	Oui
getBaselinePosition(baseline:String):Number	Oui	Oui
getMirrorRegion(mirror:EventDispatcher):flash.text.engine:TextLineMirrorRegion	Oui	Oui
hasGraphicElement : Boolean	Oui	Oui
hasTabs : Boolean	Oui	Non
MAX_LINE_WIDTH : int = 1000000	Oui	Oui
mirrorRegions : Vector.<flash.text.engine:TextLineMirrorRegion>	Oui	Oui
nextLine : flash.text.engine:TextLine	Oui	Oui
previousLine : flash.text.engine:TextLine	Oui	Oui
rawTextLength : int	Oui	Oui
specifiedWidth : Number	Oui	Oui
textBlock : flash.text.engine:TextBlock	Oui	Oui
textBlockBeginIndex : int	Oui	Oui
textHeight : Number	Oui	Oui
textWidth : Number	Oui	Oui
unjustifiedTextWidth : Number	Oui	Oui
userData : *	Oui	Oui
validity : String	Oui	Oui

TextLineCreationResult

Le tableau ci-dessous recense les membres de la classe TextLineCreationResult pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
COMPLETE : String = "complete"	Oui	Oui
EMERGENCY : String = "emergency"	Oui	Oui
INSUFFICIENT_WIDTH : String = "insufficientWidth"	Oui	Oui
SUCCESS : String = "success"	Oui	Oui

TextLineMetrics

Le tableau ci-dessous recense les membres de la classe TextLineMetrics pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ascent : Number	Oui	Oui
descent : Number	Oui	Oui
height : Number	Oui	Oui
leading : Number	Oui	Oui
TextLineMetrics(x:Number, width:Number, height:Number, ascent:Number, descent:Number, leading:Number)	Oui	Oui
width : Number	Oui	Oui
x : Number	Oui	Oui

TextLineMirrorRegion

Le tableau ci-dessous recense les membres de la classe TextLineMirrorRegion pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
bounds : Rectangle	Oui	Oui
element : ContentElement	Oui	Oui
mirror : EventDispatcher	Oui	Oui
nextRegion : flash.text.engine:TextLineMirrorRegion	Oui	Oui
previousRegion : flash.text.engine:TextLineMirrorRegion	Oui	Oui
textLine : flash.text.engine:TextLine	Oui	Oui

TextLineValidity

Le tableau ci-dessous recense les membres de la classe TextLineValidity pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
INVALID : String = "invalid"	Oui	Oui
POSSIBLY_INVALID : String = "possiblyInvalid"	Oui	Oui
STATIC : String = "static"	Oui	Oui
VALID : String = "valid"	Oui	Oui

TextRotation

Le tableau ci-dessous recense les membres de la classe TextRotation pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
AUTO : String = "auto"	Oui	Oui
ROTATE_0 : String = "rotate0"	Oui	Oui
ROTATE_180 : String = "rotate180"	Oui	Oui
ROTATE_270 : String = "rotate270"	Oui	Oui
ROTATE_90 : String = "rotate90"	Oui	Oui

TextSnapshot

Le tableau ci-dessous recense les membres de la classe TextSnapshot pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
charCount : int	Oui	Oui
findText(beginIndex:int, textToFind:String, caseSensitive:Boolean):int	Oui	Oui
getSelected(beginIndex:int, endIndex:int):Boolean	Oui	Oui
getSelectedText(includeLineEndings:Boolean = false):String	Oui	Oui
getText(beginIndex:int, endIndex:int, includeLineEndings:Boolean = false):String	Oui	Oui
getTextRunInfo(beginIndex:int, endIndex:int):Array	Oui	Oui
hitTestTextNearPos(x:Number, y:Number, maxDistance:Number = 0):Number	Oui	Oui
setSelectColor(hexColor:uint = 0xFFFF00):void	Oui	Oui
setSelected(beginIndex:int, endIndex:int, select:Boolean):void	Oui	Oui

Timer

Le tableau ci-dessous recense les membres de la classe Timer pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
currentCount : int	Oui	Oui
delay : Number	Oui	Oui
repeatCount : int	Oui	Oui
reset():void	Oui	Oui
running : Boolean	Oui	Oui
start():void	Oui	Oui
stop():void	Oui	Oui
timer	Oui	Oui
Timer(delay:Number, repeatCount:int = 0)	Oui	Oui
timerComplete	Oui	Oui

TimerEvent

Le tableau ci-dessous recense les membres de la classe TimerEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Oui
TIMER : String = "timer"	Oui	Oui
TimerEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false)	Oui	Oui
TIMER_COMPLETE : String = "timerComplete"	Oui	Oui
toString():String	Oui	Oui
updateAfterEvent():void	Oui	Oui

TouchEvent

Le tableau ci-dessous recense les membres de la classe TouchEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
altKey : Boolean	Oui	Non
clone():Event	Oui	Oui
ctrlKey : Boolean	Oui	Non
isPrimaryTouchPoint : Boolean	Oui	Oui
isRelatedObjectInaccessible : Boolean	Oui	Oui
localX : Number	Oui	Oui
localY : Number	Oui	Oui
pressure : Number	Oui	Oui
relatedObject : InteractiveObject	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
shiftKey : Boolean	Oui	Non
sizeX : Number	Oui	Oui
sizeY : Number	Oui	Oui
stageX : Number	Oui	Oui
stageY : Number	Oui	Oui
toString():String	Oui	Oui
TouchEvent(type:String, bubbles:Boolean = true, cancelable:Boolean = false, touchPointID:int = 0, isPrimaryTouchPoint:Boolean = false, localX:Number = NaN, localY:Number = NaN, sizeX:Number = NaN, sizeY:Number = NaN, pressure:Number = NaN, relatedObject:InteractiveObject = null, ctrlKey:Boolean = false, altKey:Boolean = false, shiftKey:Boolean = false)	Oui	Oui
touchPointID : int	Oui	Oui
TOUCH_BEGIN : String = "touchBegin"	Oui	Oui
TOUCH_END : String = "touchEnd"	Oui	Oui
TOUCH_MOVE : String = "touchMove"	Oui	Oui
TOUCH_OUT : String = "touchOut"	Oui	Oui
TOUCH_OVER : String = "touchOver"	Oui	Oui
TOUCH_ROLL_OUT : String = "touchRollOut"	Oui	Oui
TOUCH_ROLL_OVER : String = "touchRollOver"	Oui	Oui
TOUCH_TAP : String = "touchTap"	Oui	Oui
updateAfterEvent():void	Oui	Oui

Transform

Le tableau ci-dessous recense les membres de la classe Transform pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
colorTransform : flash.geom:ColorTransform	Oui	Oui
concatenatedColorTransform : flash.geom:ColorTransform	Oui	Oui
concatenatedMatrix : Matrix	Oui	Oui
getRelativeMatrix3D(relativeTo:DisplayObject):Matrix3D	Oui	Non
matrix : Matrix	Oui	Oui
matrix3D : Matrix3D	Oui	Non
perspectiveProjection : PerspectiveProjection	Oui	Non
pixelBounds : Rectangle	Oui	Oui

TransformGestureEvent

Le tableau ci-dessous recense les membres de la classe TransformGestureEvent pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clone():Event	Oui	Oui
GESTURE_PAN : String = "gesturePan"	Oui	Oui
GESTURE_ROTATE : String = "gestureRotate"	Oui	Oui
GESTURE_SWIPE : String = "gestureSwipe"	Oui	Oui
GESTURE_ZOOM : String = "gestureZoom"	Oui	Oui
offsetX : Number	Oui	Oui
offsetY : Number	Oui	Oui
rotation : Number	Oui	Oui
scaleX : Number	Oui	Oui
scaleY : Number	Oui	Oui
TransformGestureEvent(type:String, bubbles:Boolean = true, cancelable:Boolean = false, phase:String = null, localX:Number = 0, localY:Number = 0, scaleX:Number = 1.0, scaleY:Number = 1.0, rotation:Number = 0, offsetX:Number = 0, offsetY:Number = 0, ctrlKey:Boolean = false, altKey:Boolean = false, shiftKey:Boolean = false)	Oui	Oui
toString():String	Oui	Oui

TypeError

Le tableau ci-dessous recense les membres de la classe TypeError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
TypeError(message:String = "")	Oui	Oui

TypographicCase

Le tableau ci-dessous recense les membres de la classe TypographicCase pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
CAPS : String = "caps"	Oui	Oui
CAPS_AND_SMALL_CAPS : String = "capsAndSmallCaps"	Oui	Oui
DEFAULT : String = "default"	Oui	Oui
LOWERCASE : String = "lowercase"	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
SMALL_CAPS : String = "smallCaps"	Oui	Oui
TITLE : String = "title"	Oui	Oui
UPPERCASE : String = "uppercase"	Oui	Oui

uint

Le tableau ci-dessous recense les membres de la classe uint pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
MAX_VALUE : uint = 4294967295	Oui	Oui
MIN_VALUE : uint = 0	Oui	Oui
uint(num:Object)	Oui	Oui
toExponential(fractionDigits:uint):String	Oui	Oui
toFixed(fractionDigits:uint):String	Oui	Oui
toPrecision(precision:uint):String	Oui	Oui
toString(radix:uint):String	Oui	Oui
valueOf():uint	Oui	Oui

URIError

Le tableau ci-dessous recense les membres de la classe URIError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
URIError(message:String = "")	Oui	Oui

URLLoader

Le tableau ci-dessous recense les membres de la classe URLLoader pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
bytesLoaded : uint = 0	Oui	Oui
bytesTotal : uint = 0	Oui	Oui
close():void	Oui	Oui
complete	Oui	Oui
data : *	Oui	Oui
dataFormat : String = "text"	Oui	Oui
httpStatus	Oui	Oui
ioError	Oui	Oui
load(request:URLRequest):void	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
open	Oui	Oui
progress	Oui	Oui
securityError	Oui	Oui

URLLoaderDataFormat

Le tableau ci-dessous recense les membres de la classe URLLoaderDataFormat pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
BINARY : String = "binary"	Oui	Oui
TEXT : String = "text"	Oui	Oui
VARIABLES : String = "variables"	Oui	Oui

URLRequest

Le tableau ci-dessous recense les membres de la classe URLRequest pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
authenticate : Boolean	Oui	Non
cacheResponse : Boolean	Oui	Non
contentType : String	Oui	Oui
data : Object	Oui	Oui
digest : String	Oui	Oui
manageCookies : Boolean	Oui	Non
method : String	Oui	Oui
requestHeaders : Array	Oui	Oui
url : String	Oui	Oui
URLRequest(url:String = null)	Oui	Oui
useCache : Boolean	Oui	Non

URLRequestHeader

Le tableau ci-dessous recense les membres de la classe URLRequestHeader pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
name : String	Oui	Oui
URLRequestHeader(name:String = "", value:String = "")	Oui	Oui
value : String	Oui	Oui

URLRequestMethod

Le tableau ci-dessous recense les membres de la classe URLRequestMethod pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
DELETE : String = "DELETE"	Oui	Non
GET : String = "GET"	Oui	Oui
HEAD : String = "HEAD"	Oui	Non
OPTIONS : String = "OPTIONS"	Oui	Non
POST : String = "POST"	Oui	Oui
PUT : String = "PUT"	Oui	Non

URLStream

Le tableau ci-dessous recense les membres de la classe URLStream pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
bytesAvailable : uint	Oui	Oui
close():void	Oui	Oui
complete	Oui	Oui
connected : Boolean	Oui	Oui
endian : String	Oui	Oui
httpStatus	Oui	Oui
ioError	Oui	Oui
load(request:URLRequest):void	Oui	Oui
objectEncoding : uint	Oui	Oui
open	Oui	Oui
progress	Oui	Oui
readBoolean():Boolean	Oui	Oui
readByte():int	Oui	Oui
readBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void	Oui	Oui
readDouble():Number	Oui	Oui
readFloat():Number	Oui	Oui
readInt():int	Oui	Oui
readMultiByte(length:uint, charSet:String):String	Oui	Oui
readObject():*	Oui	Oui
readShort():int	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
readUnsignedByte():uint	Oui	Oui
readUnsignedInt():uint	Oui	Oui
readUnsignedShort():uint	Oui	Oui
readUTF():String	Oui	Oui
readUTFBytes(length:uint):String	Oui	Oui
securityError	Oui	Oui

URLVariables

Le tableau ci-dessous recense les membres de la classe URLVariables pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
decode(source:String):void	Oui	Oui
toString():String	Oui	Oui
URLVariables(source:String = null)	Oui	Oui

Vector

Le tableau ci-dessous recense les membres de la classe Vector pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
concat(... args):Vector.<T>	Oui	Oui
every(callback:Function, thisObject:Object = null):Boolean	Oui	Oui
filter(callback:Function, thisObject:Object = null):Vector.<T>	Oui	Oui
fixed : Boolean	Oui	Oui
forEach(callback:Function, thisObject:Object = null):void	Oui	Oui
indexOf(searchElement:T, fromIndex:int = 0):int	Oui	Oui
join(sep:String = ","):String	Oui	Oui
lastIndexOf(searchElement:T, fromIndex:int = 0x7fffffff):int	Oui	Oui
length : uint	Oui	Oui
map(callback:Function, thisObject:Object = null):Vector.<T>	Oui	Oui
pop():T	Oui	Oui
push(... args):uint	Oui	Oui
reverse():Vector.<T>	Oui	Oui
shift():T	Oui	Oui
slice(startIndex:int = 0, endIndex:int = 16777215):Vector.<T>	Oui	Oui
some(callback:Function, thisObject:Object = null):Boolean	Non	Oui
sort(compareFunction:Function):Vector.<T>	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
splice(startIndex:int, deleteCount:uint = 4294967295, ... items):Vector.<T>	Oui	Oui
toLocaleString():String	Non	Oui
toString():String	Non	Oui
unshift(... args):uint	Non	Oui

VerifyError

Le tableau ci-dessous recense les membres de la classe VerifyError pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
VerifyError(message:String = "")	Oui	Oui

Video

Le tableau ci-dessous recense les membres de la classe Video pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
attachCamera(camera:Camera):void	Oui	Non
attachNetStream(netStream:NetStream):void	Oui	Oui
clear():void	Oui	Oui
constructVideoObject(width:int = 320, height:int = 240):void	Oui	Oui
deblocking : int	Oui	Oui
smoothing : Boolean	Oui	Oui
Video(width:int = 320, height:int = 240)	Oui	Oui
videoHeight : int	Oui	Oui
videoWidth : int	Oui	Oui

XML

Le tableau ci-dessous recense les membres de la classe XML pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
addNamespace(ns:Object):XML	Oui	Oui
appendChild(child:Object):XML	Oui	Oui
attribute(attributeName:*):XMLList	Oui	Oui
attributes():XMLList	Oui	Oui
child(propertyName:Object):XMLList	Oui	Oui
childIndex():int	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
children():XMLList	Oui	Oui
comments():XMLList	Oui	Oui
contains(value:XML):Boolean	Oui	Oui
copy():XML	Oui	Oui
defaultSettings():Object	Oui	
descendants(name:Object = *):XMLList	Oui	Oui
elements(name:Object = *):XMLList	Oui	Oui
hasComplexContent():Boolean	Oui	Oui
hasOwnProperty(p:String):Boolean	Oui	Oui
hasSimpleContent():Boolean	Oui	Oui
ignoreComments : Boolean	Oui	Oui
ignoreProcessingInstructions : Boolean	Oui	Oui
ignoreWhitespace : Boolean	Oui	Oui
inScopeNamespaces():Array	Oui	Oui
insertChildAfter(child1:Object, child2:Object):*	Oui	Oui
insertChildBefore(child1:Object, child2:Object):*	Oui	Oui
length():int	Oui	Oui
localName():Object	Oui	Oui
name():Object	Oui	Oui
namespace(prefix:String = null):*	Oui	Oui
namespaceDeclarations():Array	Oui	Oui
nodeKind():String	Oui	Oui
normalize():XML	Oui	Oui
parent():*	Oui	Oui
prependChild(value:Object):XML	Oui	Oui
prettyIndent : int	Oui	Oui
prettyPrinting : Boolean	Oui	Oui
processingInstructions(name:String = "*"):XMLList	Oui	Oui
propertyIsEnumerable(p:String):Boolean	Oui	Oui
removeNamespace(ns:Namespace):XML	Oui	Oui
replace(propertyName:Object, value:XML):XML	Oui	Oui
setChildren(value:Object):XML	Oui	Oui
setLocalName(name:String):void	Oui	Oui
setName(name:String):void	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
setNamespace(ns:Namespace):void	Oui	Oui
setSettings(... rest):void	Oui	Oui
settings():Object	Oui	Oui
text():XMLList	Oui	Oui
toString():String	Oui	Oui
toXMLString():String	Oui	Oui
valueOf():XML	Oui	Oui

XMLDocument

Le tableau ci-dessous recense les membres de la classe XMLDocument pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
createElement(name:String):XMLNode	Oui	Oui
createTextNode(text:String):XMLNode	Oui	Oui
docTypeDecl : Object = null	Oui	Oui
idMap : Object	Oui	Oui
ignoreWhite : Boolean = false	Oui	Oui
parseXML(source:String):void	Oui	Oui
toString():String	Oui	Oui
xmlDecl : Object = null	Oui	Oui
XMLDocument(source:String = null)	Oui	Oui

XMLList

Le tableau ci-dessous recense les membres de la classe XMLList pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
attribute(attributeName*):XMLList	Oui	Oui
attributes():XMLList	Oui	Oui
child(propertyName:Object):XMLList	Oui	Oui
children():XMLList	Oui	Oui
comments():XMLList	Oui	Oui
comments():XMLList	Oui	Oui
contains(value:XML):Boolean	Oui	Oui
copy():XML	Oui	Oui
descendants(name:Object = *):XMLList	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
elements(name:Object = *):XMLList	Oui	Oui
hasComplexContent():Boolean	Oui	Oui
hasOwnProperty(p:String):Boolean	Oui	Oui
hasSimpleContent():Boolean	Oui	Oui
length():int	Oui	Oui
normalize():XMLList	Oui	Oui
parent():Object	Oui	Oui
processingInstructions(name:String = "*"):XMLList	Oui	Oui
propertyIsEnumerable(p:String):Boolean	Oui	Oui
text():XMLList	Oui	Oui
toString():String	Oui	Oui
toXMLString():String	Oui	Oui
valueOf():XMLList	Oui	Oui

XMLNode

Le tableau ci-dessous recense les membres de la classe XMLNode pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
appendChild(node:XMLNode):void	Oui	Oui
attributes : Object	Oui	Oui
childNodes : Array	Oui	Oui
cloneNode(deep:Boolean):XMLNode	Oui	Oui
firstChild : XMLNode	Oui	Oui
getNamespaceForPrefix(prefix:String):String	Oui	Oui
getPrefixForNamespace(ns:String):String	Oui	Oui
hasChildNodes():Boolean	Oui	Oui
insertBefore(node:XMLNode, before:XMLNode):void	Oui	Oui
lastChild : XMLNode	Oui	Oui
localName : String	Oui	Oui
namespaceURI : String	Oui	Oui
nextSibling : XMLNode	Oui	Oui
nodeName : String	Oui	Oui
nodeType : uint	Oui	Oui
nodeValue : String	Oui	Oui
parentNode : XMLNode	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
prefix : String	Oui	Oui
previousSibling : XMLNode	Oui	Oui
removeNode():void	Oui	Oui
toString():String	Oui	Oui
XMLNode(type:uint, value:String)	Oui	Oui

XMLNodeType

Le tableau ci-dessous recense les membres de la classe XMLNodeType pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
ELEMENT_NODE : uint = 1	Oui	Oui
TEXT_NODE : uint = 3	Oui	Oui

XMLSocket

Le tableau ci-dessous recense les membres de la classe XMLSocket pris en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
close	Oui	Oui
close():void	Oui	Oui
connect	Oui	Oui
connect(host:String, port:int):void	Oui	Oui
connected : Boolean	Oui	Oui
data	Oui	Oui
ioError	Oui	Oui
securityError	Oui	Oui
send(object:*):void	Oui	Oui
timeout : int	Oui	Oui
XMLSocket(host:String = null, port:int = 0)	Oui	Oui

Fonctions globales

flash.net

Le tableau suivant recense les fonctions globales du package flash.net prises en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
getClassByAlias(aliasName:String):Class	Oui	Oui
navigateToURL(request:URLRequest, window:String = null):void	Oui	Oui
registerClassAlias(aliasName:String, classObject:Class):void	Oui	Oui
sendToURL(request:URLRequest):void	Oui	Oui

flash.utils

Le tableau suivant recense les fonctions globales du package flash.utils prises en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
clearInterval(id:uint):void	Oui	Oui
clearTimeout(id:uint):void	Oui	Oui
describeType(value:*)::XML	Oui	Oui
escapeMultiByte(value:String):String	Oui	Oui
getAliasName(value:*)::String	Oui	Oui
getDefinitionByName(name:String):Object	Oui	Oui
getQualifiedClassName(value:*)::String	Oui	Oui
getQualifiedSuperclassName(value:*)::String	Oui	Oui
getTimer():int	Oui	Oui
setInterval(closure:Function, delay:Number, ... arguments):uint	Oui	Oui
setTimeout(closure:Function, delay:Number, ... arguments):uint	Oui	Oui
unescapeMultiByte(value:String):String	Oui	Oui

Premier niveau

Le tableau ci-dessous recense les fonctions globales de premier niveau prises en charge par ActionScript 3.0 et Flash Lite 4.

Nom du membre	ActionScript 3.0	Flash Lite 4
Array(... args):Array	Oui	Oui
Boolean(expression:Object):Boolean	Oui	Oui
decodeURI(uri:String):String	Oui	Oui
decodeURIComponent(uri:String):String	Oui	Oui
encodeURI(uri:String):String	Oui	Oui
encodeURIComponent(uri:String):String	Oui	Oui
escape(str:String):String	Oui	Oui

Nom du membre	ActionScript 3.0	Flash Lite 4
int(value:Number):int	Oui	Oui
isFinite(num:Number):Boolean	Oui	Oui
isNaN(num:Number):Boolean	Oui	Oui
isXMLName(str:String):Boolean	Oui	Oui
Number(expression:Object):Number	Oui	Oui
Object(value:Object):Object	Oui	Oui
parseFloat(str:String):Number	Oui	Oui
parseInt(str:String, radix:uint = 0):Number	Oui	Oui
String(expression:Object):String	Oui	Oui
trace(... arguments):void	Oui	Oui
uint(value:Number):uint	Oui	Oui
unescape(str:String):String	Oui	Oui
Vector(sourceArray:Object):Vector.<T>	Oui	Oui
XML(expression:Object):XML	Oui	Oui
XMLList(expression:Object):XMLList	Oui	Oui

Classes ActionScript partiellement prises en charge : Détails

Quelques classes ActionScript 3.0 ne sont que partiellement ou pas du tout prises en charge par Flash Lite 4. Pour plus de détails sur ces classes, voir *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Classe BitmapData

La classe BitmapData permet d'intervenir sur les données (pixels) d'un objet Bitmap. Vous pouvez utiliser ses méthodes pour créer des images bitmap transparentes ou opaques de taille arbitraire, ou encore pour manipuler ces images de diverses façons lors de l'exécution. En outre, vous pouvez accéder à l'objet BitmapData d'une image bitmap que vous chargez à l'aide de la classe flash.display.Loader.

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe BitmapData qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
applyFilter()	Génère une image filtrée à partir d'une image source et d'un objet filtre.	Pas de prise en charge
generateFilterRect()	Détermine le rectangle de destination affecté par l'appel de la méthode applyFilter(), en fonction d'un objet BitmapData, d'un rectangle source et d'un objet filtre spécifiés.	Pas de prise en charge
lock()	Verrouille une image de sorte que tout objet référençant l'objet BitmapData, tels que des objets Bitmap, ne soit pas mis à jour lorsque l'objet BitmapData change.	Pas de prise en charge
noise()	Remplit une image de pixels représentant un bruit aléatoire.	Pas de prise en charge
perlinNoise()	Génère une image de bruit Perlin.	Pas de prise en charge
pixelDissolve()	Procède à la dissolution de pixels, soit d'une image source vers une image de destination, soit en utilisant la même image.	Pas de prise en charge
scroll()	Fait défiler une image d'un certain nombre de pixels (x, y).	Pas de prise en charge
threshold()	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.	Pas de prise en charge
unlock()	Déverrouille une image de sorte que tout objet référençant l'objet BitmapData, tels que des objets Bitmap, soit mis à jour lorsque l'objet BitmapData change.	Pas de prise en charge

Classe ByteArray

La classe ByteArray offre des méthodes et des propriétés visant à optimiser la lecture, l'écriture et l'utilisation des données binaires.

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe ByteArray qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
compress	Comprime le tableau d'octets à l'aide du format de données compressées zlib.	Pas de prise en charge
inflate	Décompresse le tableau d'octets.	Pas de prise en charge

Classe Capabilities

La classe Capabilities offre des méthodes et des propriétés visant à optimiser la lecture, l'écriture et l'utilisation des données binaires.

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe Capabilities qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
supports32BitProcesses	Spécifie si le système prend en charge l'exécution des processus 32 bits.	Pas de prise en charge
supports64BitProcesses	Spécifie si le système prend en charge l'exécution des processus 64 bits.	Pas de prise en charge

Classe DisplayObject

La classe `DisplayObject` est la classe de base de tous les objets qu'il est possible d'insérer dans la liste d'affichage. Celle-ci gère tous les objets affichés dans Flash Player. La classe `DisplayObjectContainer` permet d'agencer les objets dans la liste d'affichage.

Les objets `DisplayObjectContainer` peuvent posséder des objets d'affichage enfants, alors que d'autres objets d'affichage, tels que `Shape` et `TextField`, constituent des nœuds terminaux, dotés de parents et de frères, mais pas d'enfants.

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe `DisplayObject` qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
accessibilityProperties	Options d'accessibilité en cours de cet objet d'affichage.	Pas de prise en charge
blendMode	Valeur de la classe <code>blendMode</code> qui spécifie le mode de fusion à utiliser.	Pas de prise en charge
blendShader	Définit un shader utilisé pour fusionner le premier plan et l'arrière-plan.	Pas de prise en charge
filters	Tableau indexé contenant chaque objet de filtre actuellement associé à l'objet d'affichage.	Pas de prise en charge
scaleZ	Indique l'échelle de profondeur (pourcentage) d'un objet telle qu'elle est appliquée à partir du point d'alignement de celui-ci.	Pas de prise en charge
scrollRect	Limites du rectangle de défilement de l'objet d'affichage	Pas de prise en charge
z	Indique la position de la coordonnée z sur l'axe z de l'occurrence de <code>DisplayObject</code> relativement au conteneur parent 3D.	Pas de prise en charge

Classe DisplayObjectContainer

La classe `DisplayObjectContainer` est la classe de base de tous les objets qu'il est possible d'utiliser comme conteneurs d'objets d'affichage sur la liste d'affichage. Celle-ci gère tous les objets affichés dans Flash Player.

La classe `DisplayObjectContainer` permet d'agencer les objets dans la liste d'affichage. Chaque objet `DisplayObjectContainer` possède sa propre liste d'enfants qui permet de définir l'ordre z des objets. L'ordre z correspond à l'ordre d'empilement du premier plan vers l'arrière-plan qui détermine l'objet dessiné au premier plan, l'objet dessiné derrière lui, etc.

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe `DisplayObjectContainer` qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
<code>textSnapshot</code>	Renvoie un objet <code>TextSnapshot</code> pour cette occurrence de <code>DisplayObjectContainer</code> .	Pas de prise en charge

Classe Font

La classe `Font` permet de gérer les polices incorporées aux fichiers SWF. Les polices incorporées sont représentées par une sous-classe de la classe `Font`. La classe `Font` a actuellement pour unique rôle de trouver des informations sur les polices incorporées. Elle ne permet pas de modifier une police.

Elle ne permet pas non plus de charger des polices externes ni de créer une occurrence d'un objet `Font`. Utilisez-la en tant que classe de base abstraite.

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe `Font` qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
<code>enumerateFonts()</code>	Spécifie s'il faut fournir la liste des polices incorporées actuellement disponibles.	Prise en charge partielle

Classe Graphics

La classe `Graphics` contient un ensemble de méthodes permettant de créer une forme vectorielle. Parmi les objets d'affichage dont le traçage est pris en charge figurent les objets `Sprite` et `Shape`. Chacune de ces classes comprend une propriété graphique qui correspond à un objet `Graphics`. Pour vous faciliter la tâche, vous disposez des fonctions d'aide suivantes : `drawRect()`, `drawRoundRect()`, `drawCircle()` et `drawEllipse()`.

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe `Graphics` qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
<code>beginShaderFill()</code>	Spécifie un remplissage de shader que Flash Player utilise pour les appels suivants des autres méthodes <code>Graphics</code> (telles que <code>lineTo()</code> ou <code>drawCircle()</code>) de l'objet.	Pas de prise en charge
<code>drawGraphicsData()</code>	Envoie une série d'occurrences d' <code>IGraphicsData</code> à des fins de dessin.	Pas de prise en charge
<code>drawPath()</code>	Envoie une série de commandes à des fins de dessin.	Pas de prise en charge
<code>drawTriangles()</code>	Effectue le rendu d'un ensemble de triangles, généralement pour déformer des bitmaps et leur donner un aspect tridimensionnel.	Pas de prise en charge
<code>lineBitmapStyle()</code>	Spécifie un bitmap à utiliser pour le trait lors du traçage de lignes.	Pas de prise en charge

Méthode	Description	Prise en charge
lineGradientStyle()	Spécifie un dégradé à utiliser pour le trait lors du traçage de lignes.	Pas de prise en charge
lineShaderStyle()	Spécifie un shader à utiliser pour le trait lors du traçage de lignes.	Pas de prise en charge
lineStyle()	Spécifie un style de ligne utilisé pour les appels suivants de méthodes Graphics telles que lineTo() ou drawCircle()	Prise en charge partielle

Classe InteractiveObject

La classe InteractiveObject est la classe de base abstraite de tous les objets d'affichage avec lesquels l'utilisateur peut interagir à l'aide de la souris et du clavier.

Il est impossible d'instancier directement la classe InteractiveObject. Un appel du nouveau constructeur InteractiveObject() renvoie une exception ArgumentError.

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe InteractiveObject qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
contextMenu	Spécifie le menu contextuel associé à l'objet.	Pas de prise en charge
doubleClickEnabled	Spécifie si l'objet reçoit les événements doubleClick.	Pas de prise en charge
tabEnabled	Indique si l'objet est spécifié dans l'ordre de tabulation.	Pas de prise en charge
tabIndex	Spécifie l'ordre de tabulation des objets dans un fichier SWF.	Pas de prise en charge

Résumé des événements

Le tableau suivant recense les événements de la classe InteractiveObject qui ne sont pas pris en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Événement	Description	Prise en charge
copy	Distribué lorsque l'utilisateur active la combinaison de touches de raccourci propre à la plate-forme représentant une opération de copie ou sélectionne « Copier » dans le menu contextuel du texte.	Pas de prise en charge
cut	Distribué lorsque l'utilisateur active la combinaison de touches de raccourci propre à la plate-forme représentant une opération de coupage ou sélectionne « Couper » dans le menu contextuel du texte.	Pas de prise en charge
doubleClick	Distribué lorsque l'utilisateur appuie sur le bouton principal d'un périphérique de pointage et le relâche deux fois de suite sur un même objet InteractiveObject, l'indicateur doubleClickEnabled de cet objet étant défini sur true.	Pas de prise en charge

Événement	Description	Prise en charge
mouseWheel	Distribué lorsque l'utilisateur actionne la molette d'une souris sur une occurrence d'InteractiveObject dans la fenêtre Flash Player.	Pas de prise en charge
paste	Distribué lorsque l'utilisateur active la combinaison de touches de raccourci propre à la plate-forme représentant une opération de collage ou sélectionne « Coller » dans le menu contextuel du texte.	Pas de prise en charge
selectAll	Distribué lorsque l'utilisateur active la combinaison de touches de raccourci propre à la plate-forme pour tout sélectionner ou choisit « Sélectionner tout » dans le menu contextuel du texte.	Pas de prise en charge
tabChildrenChange	Distribué lorsque la valeur de l'indicateur tabChildren de l'objet change.	Pas de prise en charge
tabEnabledChange	Distribué lorsque l'indicateur tabEnabled de l'objet change.	Pas de prise en charge
tabIndexChange	Distribué lorsque la valeur de la propriété tabIndex de l'objet change.	Pas de prise en charge

Classe Keyboard

La classe Keyboard permet de créer une interface que l'utilisateur peut gérer à l'aide d'un clavier. Vous pouvez utiliser les méthodes et propriétés de la classe Keyboard sans recourir à un constructeur. Les propriétés de la classe Keyboard sont des constantes représentant les touches les plus couramment utilisées dans les jeux.

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe Keyboard qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
isAccessible()	Spécifie si d'autres fichiers SWF ont accès à la dernière touche actionnée.	Pas de prise en charge

Classe KeyboardEvent

Flash Player distribue des objets KeyboardEvent en réponse aux entrées de l'utilisateur effectuées par le biais d'un clavier. Il existe deux types d'événements de clavier : KeyboardEvent.KEY_DOWN et KeyboardEvent.KEY_UP.

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe KeyboardEvent qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
updateAfterEvent()	Impose à Flash Player d'effectuer un rendu au terme du traitement de cet événement si la liste d'affichage a été modifiée.	Pas de prise en charge

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe KeyboardEvent qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
altKey	Indique si la touche Alt est active (true) ou inactive (false).	Pas de prise en charge
ctrlKey	Indique si la touche Ctrl est active (true) ou inactive (false).	Pas de prise en charge
shiftKey	Indique si la touche Maj est active (true) ou inactive (false).	Pas de prise en charge

Classe LocalConnection

La classe LocalConnection permet de créer un objet LocalConnection qui peut appeler une méthode dans un autre objet du même type.

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe LocalConnection qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
isPerUser	Indique si l'objet LocalConnection est limité à l'utilisateur en cours (true) ou est accessible globalement à tous les utilisateurs sur l'ordinateur (false).	Pas de prise en charge

Classe MouseEvent

Flash Player distribue des objets MouseEvent dans le flux d'événements chaque fois qu'un événement de souris se produit. En règle générale, un événement de souris est généré par un périphérique d'entrée utilisateur, tel qu'une souris ou un trackball, utilisant un pointeur.

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe MouseEvent qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
updateAfterEvent()	Impose à Flash Player d'effectuer un rendu au terme du traitement de cet événement si la liste d'affichage a été modifiée.	Pas de prise en charge

Résumé des événements

Le tableau suivant indique l'événement de la classe MouseEvent qui n'est pas pris en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Événement	Description	Prise en charge
MOUSE_WHEEL()	Définit la valeur de la propriété de type d'un objet événement mouseWheel.	Pas de prise en charge

Classe MovieClip

La classe MovieClip hérite des classes suivantes : Sprite, DisplayObjectContainer, InteractiveObject, DisplayObject et EventDispatcher.

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe MovieClip qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
trackAsMenu	Indique si d'autres objets d'affichage de type SimpleButton ou MovieClip peuvent recevoir des événements de relâchement de la souris.	Pas de prise en charge

Classe NetConnection

La classe NetConnection crée une connexion bidirectionnelle entre Flash Player et une application Flash Media Server ou entre Flash Player et un serveur d'applications exécutant Flash Remoting.

Un objet NetConnection est assimilable à un conduit entre le client et le serveur. Les objets NetStream permettent d'envoyer des flux via ce conduit.

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe NetConnection qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
farID	Identifiant de l'occurrence de Flash Media Server à laquelle cette occurrence de Flash Player ou d'Adobe AIR est connectée.	Pas de prise en charge
farNonce	Valeur choisie principalement par Flash Media Server et réservée exclusivement à cette connexion.	Pas de prise en charge
maxPeerConnections	Nombre total de connexions d'homologue entrantes et sortantes que cette occurrence de Flash Player ou d'Adobe AIR autorise.	Pas de prise en charge
nearID	Identifiant de cette occurrence de Flash Player ou d'Adobe AIR pour cette occurrence de NetConnection.	Pas de prise en charge
nearNonce	Valeur choisie principalement par cette occurrence de Flash Player ou d'Adobe AIR et réservée exclusivement à cette connexion.	Pas de prise en charge
protocol	Protocole utilisé pour établir la connexion.	Pas de prise en charge
unconnectedPeerStreams	Objet contenant tous les objets NetStream d'homologues abonnés qui ne sont pas associés aux objets NetStream de publication.	Pas de prise en charge

Classe NetStream

La classe NetStream ouvre une connexion en flux en continu monodirectionnelle entre Flash Player et Flash Media Server ou entre Flash Player et le système de fichiers local. Un objet NetStream est un canal au sein d'un objet NetConnection. Ce canal peut publier un flux en diffusion continue, à l'aide de NetStream.publish(), ou s'abonner à un flux publié et recevoir des données, à l'aide de NetStream.play().

Vous pouvez publier ou lire des données en direct (en temps réel) et des données préalablement enregistrées. Les objets NetStream permettent également d'envoyer des messages texte à tous les clients abonnés (voir méthode NetStream.send()).

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe NetStream qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
checkPolicyFile	Spécifie si Flash Player doit tenter de télécharger un fichier de régulation d'URL à partir du serveur du fichier vidéo chargé avant de commencer à charger ce dernier.	Pas de prise en charge
publish()	Envoie des données audio, vidéo et des messages texte en diffusion continue à partir d'un client et à destination de Flash Media Server, tout en laissant la possibilité d'enregistrer ce flux pendant la transmission.	Pas de prise en charge
send()	Envoie un message sur un flux continu publié à tous les clients abonnés.	Pas de prise en charge

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe NetStream qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
checkPolicyFile	Spécifie si Flash Player doit tenter de télécharger un fichier de régulation d'URL à partir du serveur du fichier vidéo chargé avant de commencer à charger ce dernier.	Pas de prise en charge
farID	Identifiant de l'extrémité connectée à cette occurrence de NetStream.	Pas de prise en charge
farNonce	Valeur choisie principalement par l'autre extrémité de ce flux et réservée exclusivement à cette connexion.	Pas de prise en charge
peerStreams	Objet qui détient toutes les occurrences de NetStream d'abonnement qui écoutent cette occurrence de NetStream de publication.	Pas de prise en charge
soundTransform	Contrôle le son dans cet objet NetStream.	Pas de prise en charge

Classe SimpleButton

La classe SimpleButton permet de contrôler toutes les occurrences de symboles de bouton dans un fichier SWF. Après avoir créé une occurrence de bouton dans l'outil de création, vous pouvez utiliser les méthodes et propriétés de la classe SimpleButton pour manipuler les boutons à l'aide d'ActionScript

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe NetStream qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
soundTransform	Objet soundTransform affecté à ce bouton.	Pas de prise en charge
trackAsMenu	Indique si d'autres objets d'affichage de type SimpleButton ou MovieClip peuvent recevoir des événements de relâchement de la souris.	Pas de prise en charge
useHandCursor	Valeur booléenne qui, lorsqu'elle est réglée sur true, indique si Flash Player affiche un curseur en forme de main lorsque la souris survole un bouton.	Pas de prise en charge

Classe Sound

La classe Sound permet de contrôler le son dans une application. Cette classe permet de créer un objet Sound, de charger et de lire un fichier MP3 externe dans cet objet, de fermer le flux audio et d'accéder aux données relatives au son (nombre d'octets que contient le flux, informations sur les métadonnées ID3, etc.).

Un contrôle plus précis du son est possible via sa source (objet SoundChannel ou Microphone correspondant) et des propriétés de la classe SoundTransform qui contrôlent l'émission du son par les haut-parleurs de l'ordinateur.

Résumé des événements

Le tableau suivant recense les événements de la classe Sound qui ne sont pas pris en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Événement	Description	Prise en charge
sampleData	Distribué lorsque le lecteur demande de nouvelles données audio.	Pas de prise en charge

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe Sound qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
extract()	Extrait les données audio brutes d'un objet Sound.	Pas de prise en charge

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe Sound qui sont partiellement prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
id3	Donne accès aux métadonnées faisant partie d'un fichier MP3.	Prise en charge partielle

Classe SoundMixer

La classe SoundMixer contient des propriétés et des méthodes statiques permettant de contrôler globalement le son dans l'application. Elle contrôle les sons en flux continu et intégrés de l'application. Elle ne contrôle pas les sons créés dynamiquement (autrement dit, les sons générés en réponse à la distribution d'un événement sampleData par un objet Sound).

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe SoundMixer qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
areSoundsInaccessible()	Détermine si des sons sont inaccessibles en raison de restrictions sécuritaires.	Pas de prise en charge
computeSpectrum()	Génère un instantané de l'onde sonore active et le place dans l'objet ByteArray spécifié.	Pas de prise en charge

Classe Sprite

La classe Sprite est un bloc constitutif de base de la liste d'affichage : un nœud de liste d'affichage qui permet d'afficher des graphiques et peut également contenir des enfants.

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe Sprite qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
useHandCursor	Valeur booléenne indiquant si le curseur en forme de main apparaît lorsque la souris survole un sprite dont la propriété buttonMode est réglée sur true.	Pas de prise en charge

Classe Stage

La classe Stage représente la zone de dessin principale. La scène correspond à l'intégralité de la zone d'affichage du contenu Flash. L'objet Stage n'est pas accessible globalement. Vous devez y accéder par le biais de la propriété stage d'une occurrence de DisplayObject.

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe Stage qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
assignFocus()	Définit le focus du clavier sur l'objet interactif spécifié par objectToFocus, la direction du focus étant spécifiée par le paramètre direction.	Pas de prise en charge
willTrigger()	Vérifie si un écouteur d'événements est enregistré auprès de cet objet EventDispatcher ou d'un de ces ancêtres pour le type d'événement spécifié.	Pas de prise en charge

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe Stage qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
colorCorrection	Contrôle la correction des couleurs de Flash Player pour les affichages.	Pas de prise en charge
colorCorrectionSupport	Indique si Flash Player s'exécute sur un système d'exploitation qui prend en charge la correction des couleurs et s'il peut lire et comprendre le profil de couleurs de l'écran principal	Pas de prise en charge
showDefaultContextMenu	Spécifie l'affichage ou le masquage des éléments par défaut dans le menu contextuel de Flash Player.	Pas de prise en charge
tabChildren	Détermine si les enfants de l'objet prennent en charge la tabulation.	Pas de prise en charge
textSnapshot	Renvoie un objet TextSnapshot pour cette occurrence de DisplayObjectContainer.	Pas de prise en charge

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, ainsi que de l'utilisation du Presse-papiers.

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe System qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
exit()	Ferme Flash Player.	Pas de prise en charge
setClipboard()	Remplace le contenu du Presse-papiers par une chaîne de texte spécifiée.	Pas de prise en charge

Classe TextBlock

La classe TextBlock est une usine de création d'objets TextLine, dont vous pouvez effectuer le rendu en les plaçant dans la liste d'affichage.

Elle est conçue pour contenir un paragraphe unique car c'est le maximum que peuvent traiter les algorithmes bidirectionnels Unicode et de retour à la ligne. Si une application se compose de plusieurs paragraphes de texte, utilisez un langage de marquage ou analysez le texte pour le diviser en paragraphes et créer un objet TextBlock par paragraphe.

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe TextBlock qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
dump()	Libère le contenu sous-jacent de l'objet TextBlock sous forme de chaîne XML.	Pas de prise en charge

Classe TextField

La classe TextField permet de créer des objets d'affichage destinés à l'affichage ou l'entrée de texte. Tous les champs de 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 de 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.

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe TextField qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
antiAliasType	Type d'antirénelage appliqué à ce champ de texte.	Pas de prise en charge
gridFitType	Type d'adaptation à la grille appliqué à ce champ de texte.	Pas de prise en charge
mouseWheelEnabled	Valeur booléenne qui indique si Flash Player doit automatiquement faire défiler des champs de texte multilignes lorsque l'utilisateur clique sur un champ de texte et actionne la molette de la souris.	Pas de prise en charge
sharpness	Netteté des bords du glyphe dans ce champ de texte.	Pas de prise en charge
thickness	Épaisseur des bords du glyphe dans ce champ de texte.	Pas de prise en charge
useRichTextClipboard	Spécifie si le formatage doit être copié ou collé en même temps que le texte.	Pas de prise en charge

Classe TextFormat

La classe TextFormat regroupe les informations de mise en forme de caractères. La classe TextFormat permet de personnaliser la mise en forme des champs de texte. Vous pouvez formater le texte des champs statiques et dynamiques. Les propriétés de la classe TextFormat s'appliquent aux polices incorporées et aux polices de périphérique.

Toutefois, dans le cas des polices incorporées, le texte en gras et en italique nécessite des polices spécifiques. Pour afficher du texte en gras ou en italique avec une police incorporée, vous devez incorporer les options gras et italique correspondantes.

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe TextFormat qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
kerning	Valeur booléenne indiquant si le crénage est activé (true) ou désactivé (false).	Pas de prise en charge
letterSpacing	Nombre représentant la quantité d'espace répartie uniformément entre tous les caractères.	Pas de prise en charge

Classe TextLine

La classe TextLine permet d'afficher du texte dans la liste d'affichage.

Il est impossible de créer un objet TextLine directement à partir de code ActionScript. Si vous appelez un nouvel objet `TextLine()`, une exception est renvoyée. Pour créer un objet TextLine, appelez la méthode `createTextLine()` d'un objet TextBlock.

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe TextLine qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
dump()	Libère le contenu sous-jacent de l'objet TextLine sous forme de chaîne XML.	Pas de prise en charge

Classe Transform

La classe Transform permet d'accéder aux propriétés de réglage des couleurs et aux objets de transformation bi ou tridimensionnels pouvant s'appliquer à un objet d'affichage. Au cours de la transformation, la couleur ou l'orientation et la position d'un objet d'affichage sont ajustées (décalées) par rapport aux valeurs ou coordonnées actives.

La classe Transform collecte également des données sur les transformations de couleur et de matrice bidimensionnelle qui sont appliquées à un objet d'affichage et à tous ses objets parents. Vous pouvez accéder à ces transformations conjuguées par le biais des propriétés `concatenatedColorTransform` et `concatenatedMatrix`.

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe Transform qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
getRelativeMatrix3D()	Renvoie un objet Matrix3D, capable de transformer l'espace de l'objet d'affichage spécifié par rapport à l'espace de l'objet d'affichage actif.	Pas de prise en charge

Résumé des propriétés

Le tableau suivant recense les propriétés de la classe Transform qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Propriété	Description	Prise en charge
matrix3D	Permet d'accéder à l'objet Matrix3D d'un objet d'affichage tridimensionnel.	Pas de prise en charge
perspectiveProjection	Permet d'accéder à l'objet PerspectiveProjection d'un objet d'affichage tridimensionnel.	Pas de prise en charge

Classe Video

La classe Video affiche une vidéo en direct ou enregistrée dans une application sans l'intégrer au fichier SWF. Cette classe crée un objet Video dans une interface Adobe Flash ou Adobe Flex qui lit l'un ou l'autre des deux types de vidéo suivants :

- fichiers FLV enregistrés stockés localement ou sur un serveur ;
- vidéo en direct capturée sur l'ordinateur de l'utilisateur.

Un objet Video est un objet d'affichage qui figure sur la liste d'affichage de l'application et représente l'espace visuel dans laquelle la vidéo s'exécute dans une interface utilisateur.

Résumé des méthodes

Le tableau suivant recense les méthodes de la classe Video qui ne sont pas prises en charge par Flash Lite 4 lors de l'utilisation d'ActionScript 3.0.

Méthode	Description	Prise en charge
attachCamera()	Spécifie un flux vidéo issu d'une caméra à afficher dans les limites de l'objet Video dans l'application.	Pas de prise en charge

Éléments ActionScript non pris en charge : Détails

Quelques classes ActionScript 3.0 ne sont pas prises en charge par ActionScript 3 pour Flash Lite 4. Elles sont isolées en fonction des packages auxquels elles appartiennent.

Pour plus de détails sur ces classes, voir *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

adobe.utils

Le package adobe.utils contient des fonctions et des classes dont se servent les développeurs utilisant l'outil de création Flash.

Les classes suivantes ne sont pas prises en charge par ActionScript 3.0 pour Flash Lite 4.

- CustomActions
- XMLUI

flash.accessibility

Le package flash.accessibility contient des classes de prise en charge de l'accessibilité dans le contenu et les applications Flash.

Les classes suivantes ne sont pas prises en charge par ActionScript 3.0 pour Flash Lite 4.

- Accessibility
- AccessibilityProperties

flash.desktop

Le package flash.desktop contient des classes utilisées pour les opérations de copier-coller et glisser-déplacer, ainsi que la classe Icon, qui permet de définir les icônes système utilisées par un fichier.

Les classes suivantes ne sont pas prises en charge par ActionScript 3.0 pour Flash Lite 4.

- Clipboard
- ClipboardFormats
- ClipboardTransferMode

flash.display

Le package flash.display contient des classes utilisées pour les opérations de copier-coller et glisser-déplacer, ainsi que la classe Icon, qui permet de définir les icônes système utilisées par un fichier.

Les classes suivantes ne sont pas prises en charge par ActionScript 3.0 pour Flash Lite 4.

- BlendMode
- CapsStyle
- ColorCorrection
- ColorCorrectionSupport
- FocusDirection
- GraphicsBitmapFill
- GraphicsEndFill
- GraphicsGradientFill
- GraphicsPath
- GraphicsPathCommand
- GraphicsPathWinding
- GraphicsShaderFill
- GraphicsSolidFill
- GraphicsStroke
- GraphicsTrianglePath
- IGraphicsData
- IGraphicsFill
- IGraphicsPath
- IGraphicsStroke
- InterpolationMethod
- JointStyle

- LineScaleMode
- NativeMenu
- NativeMenuItem
- NativeWindow
- NativeWindowDisplayState
- NativeWindowInitOptions
- NativeWindowResize
- NativeWindowSystemChrome
- NativeWindowType
- Screen
- Shader
- ShaderData
- ShaderInput
- ShaderJob
- ShaderParameter
- ShaderParameterType
- ShaderPrecision
- SpreadMethod
- TriangleCulling

flash.errors

Le package flash.errors contient des classes d'erreur qui font partie de l'interface de programmation d'application (API) de Flash Player, plutôt que du langage ActionScript de base.

Les classes suivantes ne sont pas prises en charge par ActionScript 3.0 pour Flash Lite 4.

- DRMMManagerError
- SQLError
- SQLErrorOperation

flash.events

Le package flash.events prend en charge le nouveau modèle d'événement DOM et comprend la classe de base EventDispatcher.

Les classes suivantes ne sont pas prises en charge par ActionScript 3.0 pour Flash Lite 4.

- ActivityEvent
- BrowserInvokeEvent
- DRMAAuthenticateEvent
- DRMAAuthenticationCompleteEvent
- DRMAAuthenticationErrorEvent

- DRMErrorEvent
- DRMStatusEvent
- FileListEvent
- HTMLUncaughtScriptExceptionEvent
- InvokeEvent
- NativeDragEvent
- NativeWindowBoundsEvent
- NativeWindowDisplayStateEvent
- OutputProgressEvent
- SampleDataEvent
- ScreenMouseEvent
- ShaderEvent
- SQLErrorEvent
- SQLEvent
- SQLUpdateEvent

flash.filters

Le package flash.filters contient des classes destinées aux effets de filtrage de bitmaps. Les filtres permettent d'appliquer des effets visuels riches, tels que les effets de flou, biseau, rayonnement et ombres portées, pour afficher des objets.

Les classes suivantes ne sont pas prises en charge par ActionScript 3.0 pour Flash Lite 4.

- BevelFilter
- BitmapFilter
- BitmapFilterQuality
- BitmapFilterType
- BlurFilter
- ColorMatrixFilter
- ConvolutionFilter
- DisplacementMapFilter
- DisplacementMapFilterMode
- DropShadowFilter
- GlowFilter
- GradientBevelFilter
- GradientGlowFilter
- ShaderFilter

flash.geom

Le package flash.geom contient des classes géométriques (points, rectangles et matrices de transformation, par exemple) pour prendre en charge la classe BitmapData et la fonctionnalité de mise en cache de bitmaps.

Les classes suivantes ne sont pas prises en charge par ActionScript 3.0 pour Flash Lite 4.

- Matrix3D
- Orientation3D
- PerspectiveProjection
- Utils3D
- Vector3D

flash.media

Le package flash.media contient des classes permettant de manipuler des actifs multimédias, tels que des sons et des vidéos. Il comporte également les classes vidéo et audio disponibles dans Flash Media Server.

Les classes suivantes ne sont pas prises en charge par ActionScript 3.0 pour Flash Lite 4.

- Camera
- Microphone
- scanHardware
- SoundCodec

flash.net

Le package flash.net contient des classes permettant d'envoyer et de recevoir des données sur le réseau (téléchargement d'URL et Flash Remoting, par exemple).

Les classes suivantes ne sont pas prises en charge par ActionScript 3.0 pour Flash Lite 4.

- FileFilter
- FileReference
- FileReferenceList

flash.printing

Le package flash.printing contient des classes permettant d'imprimer le contenu Flash.

Les classes suivantes ne sont pas prises en charge par ActionScript 3.0 pour Flash Lite 4.

- PrintJob
- PrintJobOptions
- PrintJobOrientation

flash.profiler

Le package flash.profiler contient des fonctions qui permettent de déboguer et tester le code ActionScript.

La classe ShowRedrawRegions n'est pas prise en charge par ActionScript 3.0 pour Flash Lite 4.

flash.sampler

Le package flash.sampler regroupe les méthodes et les classes qui permettent d'assurer le suivi des appels de procédure, afin que vous puissiez profiler l'utilisation de la mémoire et optimiser les applications. Ce package est utilisé par l'agent de profilage distribué avec Adobe Flex Builder ; il a pour but de vous permettre de personnaliser cet agent ou de créer vos propres tests de mémoire. Lorsque vous identifiez les parties d'une application qui consomment le plus de mémoire, vous pouvez cibler l'optimisation et accélérer les performances de l'application.

La classe StackFrame n'est pas prise en charge par ActionScript 3.0 pour Flash Lite 4.

flash.system

Le package flash.system contient des classes permettant d'accéder aux fonctionnalités de niveau système, telles que la sécurité, le nettoyage de la mémoire, etc.

Les classes suivantes ne sont pas prises en charge par ActionScript 3.0 pour Flash Lite 4.

- IME
- IMEConversionMode

flash.text

Le package flash.text contient des classes permettant de manipuler des champs de texte, de formater du texte, de modifier les dimensions du texte, d'appliquer des feuilles de style et des mises en page. Des fonctions d'anticrênelage avancées sont disponibles à partir de Flash Player 8 via les classes flash.text.TextFormat et flash.text.TextRenderer.

Les classes suivantes ne sont pas prises en charge par ActionScript 3.0 pour Flash Lite 4.

- AntiAliasType
- CSMSettings
- FontStyle
- GridFitType
- TextColorType
- TextDisplayMode
- TextExtent
- TextRenderer

flash.ui

Le package flash.ui contient des classes d'interface utilisateur, telles que les classes permettant d'interagir avec la souris et le clavier.

Les classes suivantes ne sont pas prises en charge par ActionScript 3.0 pour Flash Lite 4.

- ContextMenu
- ContextMenuBuiltInItems

- ContextMenuClipboardItems
- ContextMenuItem
- KeyLocation
- Mouse
- MouseCursor