

Développement d'applications FLASH® LITE™ 1.x

© 2008 Adobe Systems Incorporated. Tous droits réservés.

Développement d'applications Flash® Lite™ 1.x d'Adobe®

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

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

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

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

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

Windows is either a registered trademark or a trademark 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. All other trademarks are the property of their respective owners.

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

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

Portions licensed from Nellymoser, Inc. (www.nellymoser.com).

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

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

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

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

Sommaire

Chapitre 1 : Présentation

Nouveautés de la programmation Flash Lite	1
Guide des supports de formation	1
Ressources supplémentaires	1
Conventions typographiques	2

Chapitre 2 : Présentation de Flash Lite

A propos de la technologie Flash Lite	3
Disponibilité de Flash Lite 1.x	3
A propos des types de contenus Flash Lite	4
Flux de travail lors de la création d'applications Flash Lite	5
Fonctionnalités de création Flash Lite	6
Application Flash Lite Hello World	7
Présentation du lecteur Flash Lite autonome	8

Chapitre 3 : Création d'une application Flash Lite

Présentation de l'application Café	10
Affichage de l'application terminée	11
Création de l'application	11

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

Touches prises en charge par Flash Lite	20
Utilisation de la navigation par tabulation dans Flash Lite	21
Gestion des événements de touche	24
Gestion des événements de bouton	30
Utilisation des touches programmables	35

Chapitre 5 : Utilisation du texte et des polices

A propos du texte dans Flash Lite	40
Utilisation des champs texte de saisie	41
Méthodes de rendu des polices dans Flash Lite	45
Qualité de rendu et texte anti-aliasé de Flash Lite	47
Intégration des polices vectorielles dans les fichiers SWF	47
Exemple d'application de champ texte	48
Création de texte défilant	51

Chapitre 6 : Utilisation du son

A propos du son dans Flash Lite	53
Utilisation des sons de périphérique	53
Utilisation de sons Flash natifs	61

Chapitre 7 : Optimisation du contenu pour de meilleures performances et une taille de fichier réduite

Optimisation des performances	62
-------------------------------------	----

Chapitre 8 : Test du contenu Flash Lite

Présentation des fonctionnalités de test Flash Lite 63
Fonctionnalités de test non prises en charge par l'émulateur 63
Utilisation de l'émulateur 64
Types de contenus Flash Lite 65
Erreurs 68
Détermination des fonctionnalités de la plate-forme 71

Chapitre 9 : Formation à ActionScript Flash Lite 1.x

A propos d'ActionScript Flash Lite 1.x 74
Guide d'introduction au code ActionScript Flash 4 76
Tâches de script courantes 83

Index 88

Chapitre 1 : Présentation

Ce manuel décrit le développement d'applications pour périphériques portables à l'aide de Macromedia® Flash® Lite™ 1.0 et Macromedia® Flash® Lite™ 1.1 d'Adobe, dont l'ensemble porte le nom de Flash Lite 1.x. Le manuel explique comment utiliser divers modes de navigation pour les différents périphériques et comment vous servir du texte et des polices. Ce manuel explique également comment gérer la mémoire d'exécution disponible dans les applications Flash Lite fonctionnant sur différents modèles de téléphones portables. Grâce à Adobe® Device Central CS4, l'émulateur mobile intégré à Adobe® Flash® CS4 Professional, vous pouvez tester et déboguer votre application dans l'outil de programmation avant de la tester sur un périphérique réel.

Nouveautés de la programmation Flash Lite

Flash inclut les nouvelles fonctionnalités suivantes qui permettent aux développeurs de créer des applications Flash Lite :

Adobe Device Central Adobe Device Central comporte un émulateur qui permet de prévisualiser la manière dont le contenu sera exécuté sur un périphérique. L'émulateur peut être configuré de manière à reproduire les fonctionnalités disponibles sur tout périphérique pris en charge. Il propose également des informations de débogage qui vous renseignent sur d'éventuels problèmes ou incompatibilités sur le périphérique cible.

Modèles de document de périphérique Flash comprend des modèles de document permettant de vous lancer rapidement dans la création de contenu pour des périphériques et des types de contenus spécifiques.

Guide des supports de formation

La documentation de Flash Lite inclut les supports suivants qui vous permettront de vous familiariser avec la création d'applications Flash Lite :

- *Développement d'applications Flash Lite 1.x* est un guide complet sur la création de contenu Flash Lite 1.x. Il contient un didacticiel détaillé de création d'une application Flash Lite 1.x et des instructions pour tester vos applications dans l'émulateur Adobe Device Central.
- *Guide de référence du langage ActionScript Flash Lite 1.x* décrit toutes les fonctionnalités du langage ActionScript disponibles pour les développeurs de Flash Lite et fournit des exemples de code.
- Les exemples d'applications Flash Lite, disponibles à l'adresse www.adobe.com/go/learn_ftl_samples_and_tutorials, illustrent les meilleures pratiques et les concepts indispensables qui sont traités ou évoqués dans la présente documentation.

Ressources supplémentaires

Pour obtenir les toutes dernières informations sur le développement d'applications Flash Lite, avec des avis d'utilisateurs experts, des rubriques avancées, des exemples, des conseils et d'autres mises à jour, consultez la page du pôle de développement des périphériques et terminaux mobiles à l'adresse www.adobe.com/go/developer_flashlite_fr.

Pour bénéficier des TechNotes, des mises à jour de la documentation et des liens utiles vers des ressources supplémentaires dans la communauté des développeurs de Flash Lite, visitez le Support Flash Lite à l'adresse www.adobe.com/go/support_flashlite_fr.

Conventions typographiques

Ce manuel utilise les conventions typographiques suivantes :

- La *police en italique* indique une valeur à remplacer (par exemple, dans le chemin d'un dossier).
- La `police de code` indique le code ActionScript.
- La **police en gras** désigne une entrée littérale.
- Les guillemets droits (" "), dans les exemples de code, séparent les chaînes de caractères. Cependant, les programmeurs peuvent également employer des guillemets simples.

Chapitre 2 : Présentation de Flash Lite

Macromedia Flash Lite d'Adobe est une version d'Adobe Flash Player conçue pour les périphériques portables.

A propos de la technologie Flash Lite

Macromedia Flash Lite d'Adobe est une version d'Adobe Flash Player conçue pour les périphériques portables. Elle associe les fonctionnalités Flash à la puissance de traitement et aux configurations des périphériques portables les plus vendus actuellement sur le marché. Cette documentation s'applique aux deux versions de Flash Lite 1 : Flash Lite 1.0 et Flash Lite 1.1, dénommées collectivement Flash Lite 1.x. Flash Lite 1.x est composé des éléments suivants :

Moteur de rendu principal Le moteur de rendu gère l'ensemble du rendu vectoriel et bitmap.

Interprète d'ActionScript Flash Lite prend en charge la version du langage ActionScript utilisé dans Macromedia® Flash® Player 4 d'Adobe, y compris de nombreuses commandes spécifiques des téléphones portables, comme la récupération des informations relatives à l'heure et à la date à partir du périphérique. Ce code ActionScript hybride de Flash Player 4, ainsi que les commandes et propriétés spécifiques de Flash Lite, forment ce que l'on appelle le code ActionScript Flash Lite 1.x.

Pour de plus amples informations sur ActionScript Flash Lite 1.x, reportez-vous au *Guide de référence du langage ActionScript Flash Lite 1.x*.

Texte et polices Flash Lite prend en charge les champs texte statique, dynamique et de saisie. Vous pouvez utiliser les polices disponibles sur le périphérique ou intégrer des données de police dans votre fichier SWF publié. Pour plus d'informations sur l'utilisation du texte et des polices dans Flash Lite, consultez la section « [Utilisation du texte et des polices](#) » à la page 40.

Son Flash Lite 1.0 et Flash Lite 1.1 prennent tous les deux en charge les formats audio de périphérique (par exemple, MIDI ou MFi). Flash Lite 1.1 prend également en charge les paramètres audio Flash standard. Pour plus d'informations sur l'utilisation du son dans Flash Lite, consultez la section « [Utilisation du son](#) » à la page 53.

Connectivité réseau Flash Lite 1.1 prend en charge le chargement de données externes et de fichiers SWF, ainsi que les commandes et propriétés permettant d'obtenir des informations relatives à l'état des requêtes HTTP et de la connectivité.

Intégration de plates-formes et de périphériques Flash Lite permet d'accéder à différentes commandes et fonctionnalités système, comme le lancement d'appels téléphoniques et l'envoi de SMS, ainsi que l'obtention d'informations sur les capacités de la plate-forme ou sur les actions de l'utilisateur grâce aux boîtes de dialogue de saisie standard du périphérique.

Disponibilité de Flash Lite 1.x

Flash Lite 1.0 et Flash Lite 1.1 sont pris en charge sur de nombreux périphériques portables vendus dans diverses régions et marchés dans le monde. Certains de ces périphériques sont disponibles dans le monde entier. D'autres ne sont vendus que dans certaines zones géographiques ou par des opérateurs de télécommunications mobiles spécifiques. Sur certains périphériques, Flash Lite est pré-installé par le fabricant ou l'opérateur de télécommunications mobiles ; sur les autres en revanche, il faut l'installer après avoir acheté le périphérique.

Pour obtenir la liste la plus récente des périphériques prenant en charge Flash Lite, consultez la page www.adobe.com/go/mobile_supported_devices_fr.

Les périphériques disponibles dans le monde entier prenant en charge Flash Lite sont les suivants : les modèles Symbian Series 60 des marques Nokia, Sendo et Siemens, ainsi que les modèles Symbian UIQ de Sony-Ericsson. A l'heure actuelle, la plupart des périphériques disponibles dans le monde entier ne prennent en charge que le lecteur Flash Lite autonome. Le lecteur autonome est installé comme une application de « niveau supérieur » que l'utilisateur peut lancer depuis le menu d'applications de son périphérique (de la même manière qu'une application de messagerie texte ou un navigateur Web mobile, par exemple). Pour plus d'informations sur le lecteur Flash Lite autonome, consultez la section « [Présentation du lecteur Flash Lite autonome](#) » à la page 8.

A l'heure actuelle, le lecteur Flash Lite autonome n'est pré-installé sur aucun périphérique disponible dans le monde entier. Vous pouvez acheter ce lecteur autonome à des fins de développement dans le magasin en ligne d'Adobe à l'adresse www.adobe.com/go/store_fr.

Les périphériques disponibles au niveau régional prenant en charge Flash Lite sont plus nombreux que ceux disponibles dans le monde entier. A l'heure actuelle, ces périphériques sont vendus principalement au Japon et fournis avec Flash Lite pré-installé. Sur ces périphériques, Flash Lite permet d'afficher plusieurs types de contenus, tels que des économiseurs d'écran Flash et des sonneries animées. Pour plus d'informations sur les types de contenus Flash Lite, consultez la section « [A propos des types de contenus Flash Lite](#) » à la page 4.

A propos des types de contenus Flash Lite

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

- Le ou les périphériques sur lesquels le contenu sera exécuté, ou *périphériques cibles*. Le lecteur Flash Lite est installé sur de nombreux périphériques. Pour obtenir une liste complète des périphériques équipés de Flash Lite, consultez la page regroupant 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 ou *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 utilisent Flash Lite pour afficher du contenu Flash intégré dans des pages Web mobiles. Certains types de contenus ne prennent pas en charge toutes les fonctionnalités Flash Lite.

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

Les fonctionnalités de test de Flash Lite sous Flash permettent de tester différents périphériques et types de contenus Flash Lite. Vous pouvez ainsi déterminer si votre application fait appel à des fonctionnalités non disponibles pour le type de contenu que vous développez. Pour plus d'informations sur le choix des périphériques cibles et des types de contenus, consultez la section « [Test du contenu Flash Lite](#) » à la page 63.

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

La procédure de création de contenu Flash Lite est une méthode itérative qui implique les étapes suivantes :

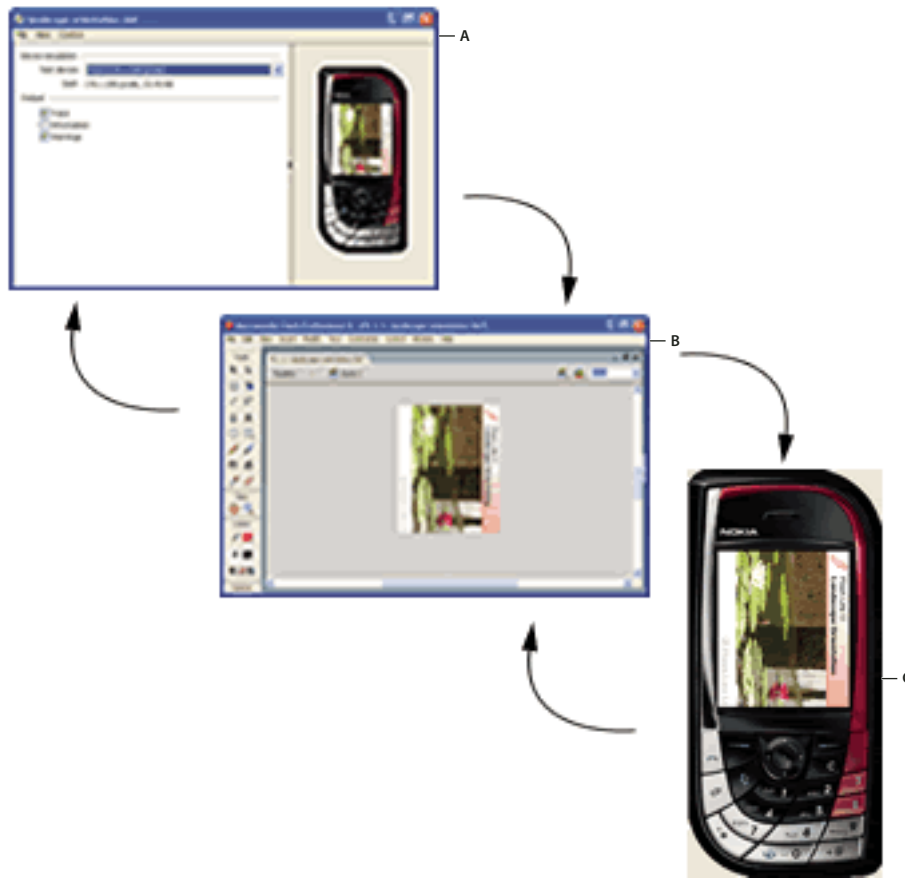
Identification des périphériques cibles et du type de contenu Flash Lite De nombreux périphériques présentent des caractéristiques différentes, par exemple en termes de taille d'écran, de prise en charge des formats audio et de profondeur de couleur d'écran. Ces facteurs peuvent influencer sur la conception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

En outre, de nombreux périphériques prennent en charge différents types de contenus Flash Lite, tels que les économiseurs d'écran, les applications autonomes ou les sonneries animées. Le type de contenu pour lequel vous programmez détermine également les fonctionnalités disponibles pour votre application. Pour plus d'informations sur les types de contenus Flash Lite, consultez la section « [A propos des types de contenus Flash Lite](#) » à la page 4.

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

Test de l'application sur les périphériques cibles Cette opération est importante car l'émulateur ne simule pas tous les aspects du périphérique cible. Par exemple, un dégradé de couleur qui semble continu sur l'émulateur peut apparaître avec 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 programmation Flash.

La figure suivante 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

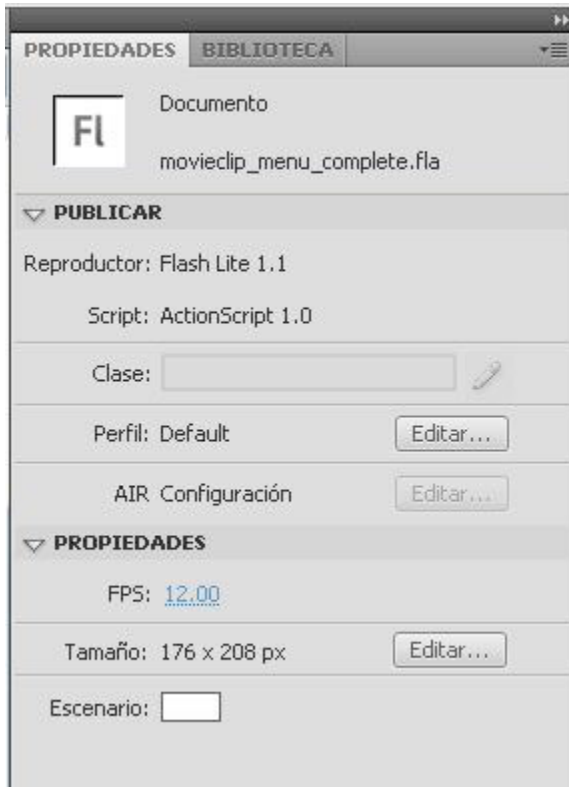
Fonctionnalités de création Flash Lite

Cette section présente les fonctionnalités de Flash spécifiquement conçues pour les développeurs Flash Lite. A l'exception de la fonctionnalité de modèles de périphériques, les fonctionnalités suivantes ne sont disponibles que si le paramètre Version de votre document est défini sur Flash Lite 1.0 ou Flash Lite 1.1 dans l'onglet Flash de la boîte de dialogue Paramètres de publication. Pour plus d'informations sur la spécification de la version du fichier SWF de vos documents, consultez la section « Définition d'options de publication pour le format de fichier Flash SWF » dans l'aide de Flash.

Adobe Device Central permet de tester la manière dont le contenu sera exécuté et affiché sur un périphérique. Il permet aussi de sélectionner un autre périphérique de test, d'afficher des informations sur votre application et de définir les options de débogage de l'émulateur. Certains périphériques prennent en charge des types de média spécifiques (par exemple, différents types de formats audio de périphérique), ainsi que divers types de contenus Flash Lite, tels qu'un lecteur autonome, un économiseur d'écran ou un navigateur. Lorsque vous prévisualisez votre application, l'émulateur simule comme type de contenu spécifié les fonctionnalités disponibles pour le périphérique de test sélectionné.

Pour plus d'informations sur les types de contenus Flash Lite, consultez la section « [A propos des types de contenus Flash Lite](#) » à la page 4.

L'inspecteur **Propriétés** contient une section fournissant des informations sur les paramètres de votre périphérique actuel, ainsi qu'un bouton qui permet d'ouvrir la boîte de dialogue Paramètres du périphérique. Ce bouton n'est actif que si le paramètre Version de votre document est défini sur Flash Lite 1.0 ou Flash Lite 1.1 dans l'onglet Flash de la boîte de dialogue Paramètres de publication.



A. Bouton Paramètres du périphérique B. Type de contenu et périphérique de test actifs

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. L'objectif de ce didacticiel est de vous familiariser avec les fonctionnalités de test et de programmation pour périphériques portables d'Adobe Flash CS4 Professional, ainsi qu'avec le processus de création général du contenu Flash Lite. Pour obtenir un exemple d'application plus détaillé, consultez la section « [Création d'une application Flash Lite](#) » à la page 10.

Vous devez d'abord choisir les périphériques et le type de contenu Flash Lite que vous ciblez. Dans le cadre de ce didacticiel, nous allons supposer que vous développez du contenu pour le lecteur autonome Flash Lite 1.1 sur les périphériques de la gamme Series 60 de Nokia. Tous les périphériques de la gamme Series 60 de Nokia pris en charge ont la même taille de scène disponible (176 x 208 pixels). Par conséquent, la même application (fichier SWF) doit en principe s'exécuter sur tous ces périphériques.

Pour commencer le développement, vous devez d'abord configurer les paramètres de publication de votre document Flash, les paramètres du document, ainsi que les paramètres du périphérique en fonction du périphérique et du type de contenu ciblés. Vous pouvez effectuer ce paramétrage manuellement à l'aide d'un nouveau document vierge ou vous pouvez utiliser un modèle Flash Lite pour créer un nouveau document préconfiguré dont les paramètres sont adaptés au périphérique et au type de contenu ciblés. La procédure suivante explique comment créer une application simple de type « Hello World ».

Pour configurer et créer une application Flash Lite simple :

- 1 Démarrez Flash.
- 2 Dans l'écran Flash principal, sélectionnez Créer > Document mobile Flash. Flash ouvre Adobe Device Central et affiche l'onglet Nouveau document.
- 3 Dans Device Central, repérez la zone Version du lecteur et sélectionnez-y Flash Lite 1.1; faites de même pour la zone Version d'ActionScript avec ActionScript 2.0 et la zone Type de contenu avec Lecteur autonome.
- 4 Cliquez sur Taille personnalisée pour tous les périphériques sélectionnés, en bas de l'écran. Ceci vous permet de créer du contenu pour le lecteur Flash Lite autonome.
- 5 Cliquez sur Créer. Vous revenez à Flash qui lance la création d'un document en utilisant les paramètres de publication prédéfinis et (lorsque vous spécifiez un périphérique) la taille adéquate pour le périphérique sélectionné.
- 6 Dans le panneau Outils, sélectionnez l'outil Texte et faites-le glisser sur la scène pour créer une zone de texte. Tapez **Hello, world !** (ou tout autre texte) dans la zone de texte.
- 7 Sélectionnez Contrôle > Tester l'animation pour exporter votre application vers Adobe Device Central et l'afficher dans son émulateur.
Remarque : Au cours de la phase de test effectué dans Device Central, il est possible de modifier le périphérique et le type de contenu pour voir l'application sur une autre plate-forme. Pour cela, double-cliquez sur un périphérique dans le panneau de bibliothèque et sélectionnez un nouveau type de contenu dans la zone Type de contenu. Lorsque vous revenez à Flash, les paramètres que vous avez utilisés la dernière fois dans l'émulateur sont conservés.
- 8 Pour revenir à Flash, sélectionnez Fichier > Retourner à Flash.
Sélectionnez Contrôle > Tester l'animation pour afficher l'application dans l'émulateur Adobe Device Central.

Présentation du lecteur Flash Lite autonome

Le lecteur Flash Lite 1.1 autonome est une application qui permet d'ouvrir et d'afficher les fichiers SWF se trouvant sur la carte mémoire de votre périphérique, accessibles depuis son navigateur Web mobile ou arrivés dans sa boîte de réception grâce à la technologie sans fil Bluetooth® ou à une connexion infrarouge.

A l'heure actuelle, le lecteur autonome est disponible dans le monde entier pour les plates-formes et périphériques suivants :

Plate-forme Series 60 :

- Nokia 3600, 3620, 3650, 3660, 6260, 6600, 6620, 6630, 6670, 6680, 6681, 7610, N-Gage, N-Gage QD
- Sendo X
- Siemens SX1

Plate-forme UIQ :

- Sony Ericsson P900, P910

En tant que développeur, vous pouvez acheter le lecteur Flash Lite 1.1 autonome pour les périphériques pris en charge dans le magasin en ligne d'Adobe à l'adresse www.adobe.com/go/store_fr. Pour obtenir une liste des questions les plus fréquemment posées concernant l'achat du lecteur autonome, consultez la section FAQ de Flash Lite 1.1 à l'adresse www.adobe.com/go/flt_faqs_fr. Pour obtenir de l'aide sur l'installation du lecteur, consultez la TechNote 4632f5aa à l'adresse www.adobe.com/go/4632f5aa_fr.

Chapitre 3 : Création d'une application Flash Lite

La présente section porte sur le développement d'une application Adobe Flash Lite de promotion pour un restaurant fictif, le Café Townsend. Les utilisateurs peuvent afficher une liste des spécialités servies et appeler le restaurant pour réserver une table.

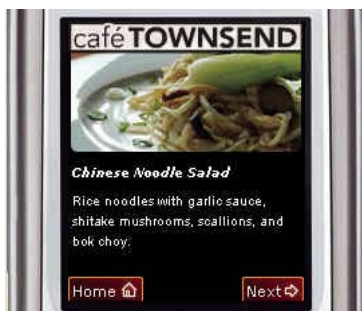
Présentation de l'application Café

L'écran initial de l'application Café contient un texte d'introduction sur le restaurant et un menu à deux éléments : Specials et Reservations. Pour sélectionner une option de menu, l'utilisateur doit appuyer sur les flèches Haut ou Bas de son périphérique pour définir le focus, puis appuyer sur la touche de sélection pour confirmer son choix.



Ecran principal de l'application Café

Si l'utilisateur sélectionne l'option de menu Specials, un écran qui permet de naviguer dans la liste des plats du jour apparaît. Pour parcourir les images et descriptions des plats, l'utilisateur appuie sur la touche programmable Droite du périphérique (intitulée Next). Pour revenir à l'application principale, il appuie sur la touche programmable Gauche (intitulée Home).



Ecran Specials de l'application Café

Si l'utilisateur sélectionne l'option Reservations dans l'écran principal, l'application lance directement un appel téléphonique vers le restaurant. Avant de composer le numéro, Flash Lite demande toujours à l'utilisateur de confirmer qu'il souhaite lancer l'appel.

Affichage de l'application terminée

Une version complète de l'application Café est installée avec Flash. Vous pouvez afficher l'application finalisée dans Device Central, ou si une version autonome de Flash Lite 1.1 est installée sur votre périphérique portable, vous pouvez transférer le fichier SWF sur ce périphérique pour l'y afficher.

Pour afficher l'application finalisée dans Adobe Device Central :

- 1 Dans Flash, ouvrez le fichier `cafe_tutorial_complete fla` qui se trouve à l'adresse www.adobe.com/go/learn_ftl_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 Tutorial Assets qu'il contient et sélectionnez le fichier qui vous intéresse.
- 2 Sélectionnez Contrôle > Tester l'animation pour lancer l'application dans l'émulateur.
- 3 Pour interagir avec l'application, procédez comme suit :
 - Dans l'écran principal, cliquez sur la flèche Bas du clavier de l'émulateur pour sélectionner l'élément de menu Specials. Cliquez ensuite sur la touche de sélection de l'émulateur pour afficher l'écran.
 - Dans l'écran Specials, cliquez sur la touche programmable Droite (intitulée Next) de l'émulateur pour afficher l'image et la description de chaque plat. Cliquez sur la touche programmable Gauche (intitulée Home) pour retourner dans l'écran principal.
 - De retour dans l'écran principal, sélectionnez l'élément de menu Reservations pour passer un appel téléphonique au restaurant.

Création de l'application

Cette section contient des procédures détaillées qui permettent de recréer l'application Café. Le didacticiel est organisé en trois parties :

- Sélection des périphériques de test et des types de contenus. Dans cette section, vous allez configurer les paramètres de publication, de document et de périphérique de votre document Flash.
- Création du menu pour l'écran principal de l'application. Dans cet écran, l'utilisateur peut afficher les images et les descriptions des plats du jour ou appeler le restaurant pour réserver une table à partir d'un même menu.
- Création de l'écran Specials. Dans cet écran, l'utilisateur peut appuyer sur la touche programmable Gauche du périphérique pour parcourir les images et les descriptions des plats servis chaque jour au déjeuner dans le café, ou bien il peut appuyer sur la touche programmable Droite pour retourner à l'écran principal.

Sélection des périphériques de test et des types de contenus

Vous utilisez Device Central pour sélectionner les périphériques et le type de contenu que vous ciblez. Lorsque vous testez votre application dans l'émulateur Adobe Device Central, ce dernier est configuré pour adapter la configuration du lecteur au périphérique et au type de contenu cibles.

Vous indiquez ces paramètres dès le départ lorsque vous créez votre document mobile Flash. Pour plus d'informations sur la création d'un document du début à la fin, consultez la section « [Application Flash Lite Hello World](#) » à la page 7.

Création du menu pour l'écran principal

Dans cette section, vous allez créer le menu de l'écran principal de l'application. Ce menu se compose de deux options : Specials et Reservations. L'option Specials dirige l'utilisateur vers un écran affichant les images et descriptions des plats du jour du café. L'option Reservations lance un appel téléphonique vers le café pour permettre à l'utilisateur de faire une réservation.

Le menu comporte deux boutons Flash standard qui définissent les états Haut, Dessus et Abaissé. L'utilisateur donne le focus à l'un de ces boutons en appuyant sur la flèche Haut ou Bas du périphérique. Lorsqu'un bouton a le focus, il s'affiche à l'état Dessus. Il génère un événement de bouton `press` lorsque l'utilisateur appuie sur la touche de sélection du périphérique. Cette navigation par tabulation par défaut permet de créer facilement une interface utilisateur simple pour une application Flash Lite. Pour plus d'informations sur l'utilisation de la navigation par tabulation, consultez la section « [Utilisation de la navigation par tabulation dans Flash Lite](#) » à la page 21.

Pour créer le menu de l'écran principal :

- 1 Dans Flash, ouvrez le fichier que vous avez enregistré dans la section précédente (consultez la section « [Sélection des périphériques de test et des types de contenus](#) » à la page 11).
- 2 Dans la fenêtre Scénario (Fenêtre > Scénario), sélectionnez l'image 1 dans le calque du menu.
- 3 Pour créer le menu, ouvrez le panneau Bibliothèque (Fenêtre > Bibliothèque), puis faites glisser une occurrence du symbole de bouton Specials sur la scène.

Positionnez le bouton sous le champ texte (déjà en place) présentant le restaurant.

- 4 Faites glisser l'occurrence du symbole de bouton Reservations sur la scène et placez-la sous le bouton Specials, comme indiqué dans l'illustration suivante :



- 5 Sélectionnez le bouton Specials et ouvrez le panneau Actions (Fenêtre > Actions).
- 6 Ajoutez le code suivant dans le panneau Actions :

```
on (press) {  
    gotoAndStop ("specials");  
}
```

Ce code de gestionnaire d'événements place la tête de lecture sur l'image `Specials` lorsque l'utilisateur sélectionne ce bouton. Vous allez créer le contenu de cette image dans la section suivante (consultez la section « [Création de l'écran Specials](#) » à la page 13).

- 7 Dans la scène, sélectionnez le bouton Reservations et ouvrez de nouveau le panneau Actions.
- 8 Dans le panneau Actions, tapez le code suivant :


```
on(press) {  
    getURL("tel:1-415-555-1212");  
}
```

Lorsque l'utilisateur sélectionne l'élément de menu Reservations, Flash Lite lance directement un appel téléphonique vers le numéro spécifié. Flash Lite demande toujours à l'utilisateur d'autoriser ou de refuser la composition d'un numéro à partir d'un fichier SWF. Pour plus d'informations, consultez la section « [Lancement d'un appel téléphonique](#) » à la page 84.

9 Dans le scénario, sélectionnez l'image 1 dans le calque Actions.

10 Ouvrez le panneau Actions et entrez le code suivant :

```
stop();  
_focusRect = false;  
fscommand2("resetsoftkeys");  
fscommand2("setquality", "high");  
fscommand2("fullscreen", "true");
```

Ce code produit les résultats suivants :

- Arrête la tête de lecture sur cette image.
- Désactive l'affichage du rectangle de focus jaune dessiné par défaut par Flash Lite autour du bouton ou du champ texte de saisie ayant le focus (consultez la section « [A propos du rectangle de focus](#) » à la page 22).
- Rétablit l'état par défaut des touches programmables. (Dans la suite de ce didacticiel, vous allez ajouter du code permettant d'enregistrer les touches programmables utilisées par votre application.)
- Définit la qualité du rendu du lecteur sur « high » (élevée). Par défaut, Flash Lite affiche une qualité moyenne pour le contenu graphique.
- Oblige le lecteur à afficher l'application en plein écran.

11 Pour tester votre travail, sélectionnez Contrôle > Tester l'animation.

12 Dans l'émulateur, cliquez sur la flèche Haut ou Bas du pavé numérique (ou appuyez sur la flèche Haut ou Bas sur le clavier de l'ordinateur) pour donner le focus au bouton Specials.

Lorsque le bouton Specials a le focus, il s'affiche à l'état Dessus.

13 Cliquez sur la touche de sélection du pavé numérique de l'émulateur (ou appuyez sur la touche Entrée de votre clavier) pour sélectionner l'élément de menu.

A ce stade, l'écran Specials ne contient aucune fonctionnalité. Dans la section suivante, vous allez ajouter des éléments d'interactivité et de l'animation pour créer l'écran Specials (consultez la section « [Création de l'écran Specials](#) » à la page 13).

Création de l'écran Specials

Dans cette section, vous allez créer les éléments de l'interface utilisateur permettant de parcourir les images et les descriptions de chaque plat. L'écran Specials comprend les éléments suivants :

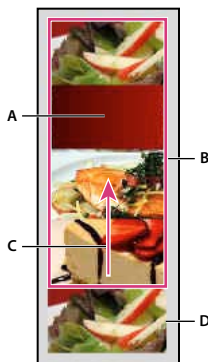
- Une animation qui assure la transition entre les images de chaque plat.
- Des champs texte dynamique qui affichent le nom et la description de chaque plat.
- Les éléments de l'interface utilisateur permettant de naviguer entre les différents plats et de retourner à l'écran principal de l'application.

Cette section du didacticiel est divisée en deux parties. Dans la première partie, vous allez créer l'animation qui assure la transition entre les images de chaque plat. Dans la seconde partie, vous allez ajouter des éléments d'interface utilisateur et du code ActionScript pour permettre à l'utilisateur de naviguer entre les images et d'afficher le nom et la description de chaque plat.

Création de l'animation d'images

Dans cette section, vous allez créer l'animation interpolée qui assure la transition entre les images de chaque plat. Lorsque vous aurez terminé, l'animation sera lue sans discontinuer. Dans la suite de ce didacticiel, vous allez ajouter des éléments de navigation et du code ActionScript pour permettre à l'utilisateur de contrôler l'animation à l'aide de la touche programmable Droite du périphérique.

Pour créer l'animation, vous allez utiliser un clip prédéfini contenant les images de tous les plats, regroupées en une colonne verticale. Vous allez utiliser un calque de masque pour n'afficher qu'une seule des images. Vous allez ensuite créer une série d'interpolations permettant de déplacer le clip vers le haut et d'afficher une image différente. La dernière image du clip est une duplication de la première, ce qui permet à la séquence d'animation de reprendre son état initial une fois que l'utilisateur a consulté la dernière image. L'illustration suivante illustre ces concepts :

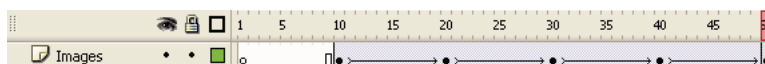


A. Objet calque de masque B. Clip d'images masqué C. Sens de l'interpolation D. Image dupliquée

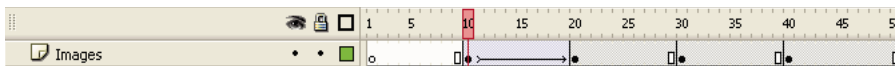
Dans la dernière section de ce didacticiel, vous allez ajouter du code ActionScript et des éléments d'interface utilisateur pour faciliter le contrôle de la séquence d'animation.

Pour créer l'animation d'images :

- 1 Ouvrez le fichier que vous avez enregistré dans la section précédente (consultez la section « [Création du menu pour l'écran principal](#) » à la page 12).
- 2 Dans le scénario, sélectionnez l'image-clé de l'image 10 dans le calque Images.
- 3 Ouvrez le panneau Bibliothèque et faites glisser le symbole Specials Images movie clip sur la scène.
La suite de ce didacticiel fait référence à ce clip simplement sous le nom *clip d'images*.
- 4 Une fois la nouvelle occurrence de clip sélectionnée, définissez les coordonnées *x* et *y* du clip sur **0** dans l'inspecteur Propriétés.
Cette action aligne le coin supérieur gauche du clip d'images sur le coin supérieur gauche de la scène.
- 5 Dans le calque Images, insérez des images-clés dans les images 20, 30, 40 et 50, comme indiqué dans l'illustration ci-dessous :



- 6 Dans le scénario, sélectionnez l'image-clé sur l'image 20.
- 7 Dans la scène, sélectionnez le clip d'images et définissez sa coordonnée *y* sur **-100** dans l'inspecteur Propriétés.
Cette action déplace le clip de 100 pixels vers le haut sur la scène.
- 8 Sélectionnez l'image-clé de l'image 30 dans le scénario, puis le clip d'images, et définissez sa coordonnée *y* sur **-200** dans l'inspecteur Propriétés.
- 9 Sélectionnez l'image-clé de l'image 40, puis le clip d'images, et définissez sa coordonnée *y* sur **-300** dans l'inspecteur Propriétés.
- 10 Sélectionnez l'image-clé de l'image 50, puis le clip d'images, et définissez sa coordonnée *y* sur **-400** dans l'inspecteur Propriétés.
- 11 Sélectionnez l'image-clé de l'image 10, puis l'option Mouvement dans le menu Interpolation de l'inspecteur Propriétés.
Cette action interpole la position du clip d'images entre les images-clés des images 10 et 20.



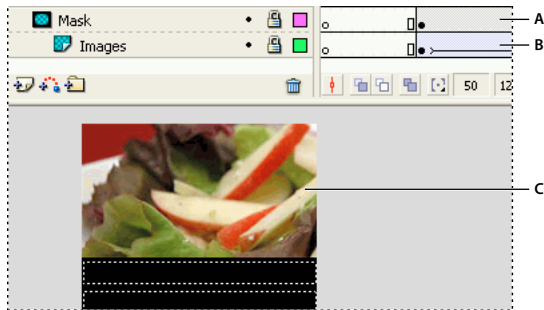
- 12 Pour créer des transitions entre les autres images, répétez l'étape 11 pour les images-clés des images 20, 30 et 40.
- 13 Pour créer le calque de masque, sélectionnez le calque Images dans le scénario, puis l'option Insérer > Scénario > Nouveau calque (ou cliquez sur le bouton Insérer un calque dans le scénario).
- 14 Insérez une image-clé dans l'image 10 du nouveau calque de masque.
- 15 Avec l'outil Rectangle du panneau Outils, créez un rectangle au-dessus de la première image (la plus au-dessus) dans le clip d'images.

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



Rectangle de masquage

- 16 Pour vous assurer que le rectangle recouvre toute la zone de l'image, double-cliquez dessus pour le sélectionner, puis utilisez l'inspecteur des propriétés pour définir ses coordonnées *x* et *y* sur **0**, sa largeur sur **176** et sa hauteur sur **100**.
- 17 Cliquez avec le bouton droit de la souris (Windows®) ou en appuyant sur la touche Contrôle (Macintosh®) sur le calque Specials dans le scénario, puis sélectionnez Masque dans le menu contextuel.
Le calque est converti en calque de masque, signalé par l'icône correspondante. Le calque situé immédiatement au-dessous est lié au calque de masque et son contenu affiché à travers la zone remplie sur le masque. Pour plus d'informations sur l'utilisation des calques de masque dans Flash, consultez la section « Utilisation des calques de masque » du guide *d'utilisation de Flash*.



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

18 Enregistrez vos modifications (Fichier > Enregistrer).

A ce stade, le test de l'application dans l'émulateur lit l'animation jusqu'au bout et s'arrête. Dans la section suivante (consultez la section « [Ajout d'éléments de navigation et de texte dans l'écran Specials](#) » à la page 16), vous allez ajouter du code ActionScript permettant d'arrêter l'animation à chaque image-clé, ainsi que des éléments d'interface utilisateur permettant de naviguer entre les images.

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

Dans cette section, vous allez ajouter des éléments d'interactivité à l'écran Specials pour permettre à l'utilisateur de contrôler la transition entre chaque animation. Vous allez également ajouter des champs texte dynamique affichant le nom et la description de chaque image.

Pour ajouter du texte devant afficher les noms et les descriptions des plats :

- 1 Dans Flash, ouvrez le fichier que vous avez enregistré dans la section précédente (consultez la section « [Création du menu pour l'écran principal](#) » à la page 12).
- 2 Dans le scénario, sélectionnez l'image 10 dans le calque Text (Texte).
- 3 Dans le panneau Outils, sélectionnez l'outil Texte et tracez un champ texte sous la première image de plat masquée.

Ce champ texte affiche le nom du plat dont l'image est affichée.



Champ texte devant afficher le nom du plat

- 4 En maintenant le champ texte sélectionné sur la scène, effectuez les modifications suivantes dans l'inspecteur Propriétés :
 - Sélectionnez Texte dynamique dans le menu déroulant Type de texte.
 - Sélectionnez Verdana dans le menu contextuel Police.
 - Sélectionnez l'option de style de texte Italique.
 - Définissez la taille de la police sur 10.
 - Sélectionnez Bitmap (sans anti-alias) dans le menu contextuel Méthode de rendu des polices.
 - Tapez **title** dans la zone de texte Var. Il s'agit du nom de variable attribué au champ texte dynamique.
- 5 Créez un autre champ texte sous le premier pour afficher une brève description du plat consulté par l'utilisateur.
- 6 A l'aide de l'outil Sélection, redimensionnez le champ texte afin que sa taille soit environ trois fois plus grande que celle de l'autre champ texte.



Champ texte devant afficher la description du plat

- 7 En maintenant le champ texte sélectionné sur la scène, effectuez les modifications suivantes dans l'inspecteur Propriétés :
 - Sélectionnez Texte dynamique dans le menu déroulant Type de texte.
 - Sélectionnez Multiligne dans le menu contextuel Type de ligne.
 - Sélectionnez Verdana dans le menu contextuel Police.
 - Définissez la taille de la police sur 10.
 - Sélectionnez Bitmap (sans anti-alias) dans le menu contextuel Méthode de rendu des polices.
 - Tapez **description** dans la zone de texte Var.
- 8 Dans le scénario, sélectionnez l'image-clé de l'image 10 dans le calque Actions.
- 9 Ouvrez le panneau Actions et ajoutez le code suivant :

```
title = "Summer salad";  
description = "Butter lettuce with apples, blood orange segments, gorgonzola, and raspberry  
vinaigrette.";  
fscommand2("SetSoftKeys", "Home", "Next");  
stop();
```

Ce code affiche le nom et la description du plat consulté par l'utilisateur et arrête la tête de lecture. La commande `SetSoftKeys` enregistre les touches programmables du périphérique qui permettront à l'utilisateur de revenir à l'écran d'accueil ou de naviguer entre les plats.

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

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

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

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

12 Dans le calque Actions, sélectionnez l'image-clé de l'image 40 et saisissez le code suivant dans le panneau Actions :

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

13 Dans le calque Actions, sélectionnez l'image-clé de l'image 50 et saisissez le code suivant dans le panneau Actions :

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

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

14 Enregistrez vos modifications.

Vous allez ensuite ajouter des éléments de navigation à l'écran **Specials** pour permettre à l'utilisateur de parcourir les images et les descriptions de chaque plat.

Pour ajouter des éléments de navigation dans l'écran **Specials :**

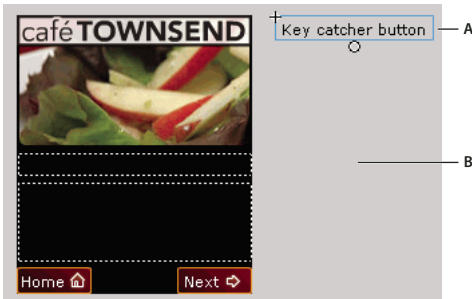
- 1** Ouvrez le fichier que vous avez enregistré dans la section précédente.
- 2** Dans le panneau Bibliothèque (Fenêtre > Bibliothèque), recherchez le symbole **Home** et faites-le glisser vers le coin inférieur gauche de la scène.
- 3** Dans l'inspecteur Propriétés, définissez la coordonnée *x* du graphique **Home** sur **0** et sa coordonnée *y* sur **188**.
- 4** Faites glisser le symbole **Next** de la Bibliothèque jusqu'au coin inférieur droit de la scène.
- 5** Dans l'inspecteur Propriétés, définissez la coordonnée *x* du graphique sur **120** et sa coordonnée *y* sur **188**.

La scène de votre application doit ressembler à l'illustration ci-dessous :



- 6 Dans le scénario, sélectionnez l'image-clé de l'image 10 dans le calque Key Catcher (Détecteur de touches).
- 7 Depuis la Bibliothèque, faites glisser le symbole du bouton de détecteur de touches et placez-le dans la zone de travail en dehors de la scène.

Remarque : Pour afficher l'espace de travail, sélectionnez *Affichage > Zone de travail* dans Flash.



A. Bouton de détecteur de touches B. Zone de travail

Ce bouton a pour but de « détecter » les événements de pression de touche ActionScript initiés par l'utilisateur, puis de répondre par l'action appropriée. Pour plus d'informations sur l'utilisation des boutons de détecteur de touches, consultez la section « [Création d'un bouton de détecteur de touches](#) » à la page 25.

- 8 Sélectionnez le bouton de détecteur de touches puis, dans le panneau Actions, entrez le code suivant :

```
// Handle right soft key event ("Next" button):
on(keyPress "<PageDown>") {
    play();
}
// Handle left soft key event ("Home" button):
on(keyPress "<PageUp>") {
    gotoAndStop("main");
}
```

Le premier gestionnaire `on(keyPress)` avance l'animation d'images jusqu'à l'image suivante dans la séquence ; le deuxième gestionnaire renvoie la tête de lecture sur l'écran principal de l'application.

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

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

Macromedia Flash Lite 1.0 et Flash Lite 1.1 d'Adobe prennent en charge l'interaction de l'utilisateur par le biais du clavier du périphérique, ou bien d'une interface de type stilet ou tactile sur les périphériques qui en sont dotés.

Deux méthodes permettent d'ajouter de l'interactivité basée sur des touches à une application Flash Lite. Vous pouvez utiliser la navigation par tabulation Flash Lite par défaut ou créer un système de navigation personnalisé basé sur des touches.

Le fonctionnement de la navigation par tabulation est le même dans Flash Lite que dans les applications Flash pour ordinateurs de bureau, où les touches Tab et Maj+Tab permettent à l'utilisateur de passer d'un objet à l'autre à l'écran. Dans Flash Lite, les quatre touches de direction du périphérique ont la même fonction que les touches Tab et Maj+Tab. La navigation par tabulation dans Flash Lite utilise uniquement des boutons et des champs texte de saisie ; elle convient généralement aux interactions utilisateur simples, telles que les menus. Pour plus d'informations, consultez la section « [Utilisation de la navigation par tabulation dans Flash Lite](#) » à la page 21.

Plutôt que d'utiliser la navigation par tabulation, vous pouvez utiliser une navigation par touche personnalisée. Dans ce cas, votre application gère les événements de touche générés par Flash Lite en réponse à la pression d'une touche sur le périphérique de l'utilisateur, puis répond par l'action appropriée. Vous pouvez utiliser ce type de navigation par exemple pour créer un jeu Flash Lite ou toute autre application dont le modèle d'interaction est plus complexe que celui d'un simple menu.

Touches prises en charge par Flash Lite

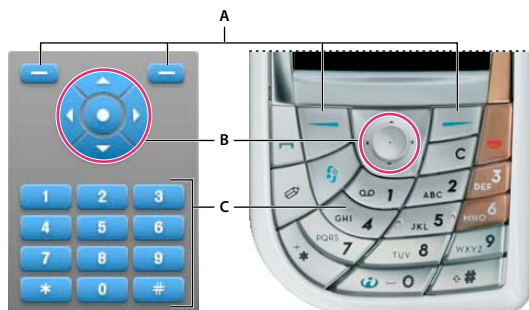
Outre les touches alphanumériques disponibles sur les téléphones standard, la plupart des périphériques mobiles possèdent un clavier à cinq directions qui permet à l'utilisateur de parcourir et de sélectionner des éléments sur l'écran du périphérique, ainsi qu'au moins deux touches programmables Gauche et Droite. Les *touches programmables* d'un périphérique sont des touches multifonctionnelles qui utilisent l'écran pour identifier leur cible à tout moment.

Le clavier à cinq directions se compose de quatre flèches (Haut, Bas, Gauche et Droite) et d'une touche de sélection, généralement située au centre du clavier. Ces touches peuvent être exploitées différemment selon l'application active. Par exemple, dans un jeu Flash Lite, l'utilisateur peut utiliser les flèches pour déplacer un personnage sur l'écran, puis utiliser la touche de sélection pour exécuter une autre action, telle que faire sauter le personnage.

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

- Touches du clavier à cinq directions (Haut, Bas, Gauche, Droite et Sélection)
- Touches programmables Gauche et Droite
- Touches 0 à 9, * et #

Les illustrations suivantes montrent la disposition de ces touches sur un clavier générique et sur un véritable téléphone :



A. Touches programmables B. Clavier à cinq directions C. Touches numériques, * et #

Certains périphériques et types de contenus Flash Lite ne prennent pas en charge toutes ces touches. Par exemple, sur certains périphériques, une application Flash Lite peut utiliser les flèches Haut et Bas, mais pas les flèches Gauche et Droite. De même, certains périphériques et types de contenus ne permettent pas aux applications Flash d'utiliser les touches programmables Gauche et Droite. Lorsque vous testez votre application dans Adobe Device Central, ce dernier génère des messages d'avertissement si vous appuyez sur des touches non disponibles pour le périphérique et le type de contenu cibles.

Utilisation de la navigation par tabulation dans Flash Lite

Dans les applications de bureau Flash, les touches Tab et Maj+Tab permettent aux utilisateurs de sélectionner les objets affichés à l'écran. L'objet qui a le focus répond aux pressions de touche suivantes. Dans Flash Lite, les touches de navigation du pavé à cinq directions du périphérique ont la même fonction que les touches Tab et Maj+Tab.

Flash Lite prend en charge trois modes de navigation par tabulation différents : navigation bidirectionnelle, navigation quadridirectionnelle et navigation quadridirectionnelle en boucle. Différents périphériques et types de contenus Flash Lite prennent en charge différents modes de navigation. Pour plus d'informations, consultez la section « [Modes de navigation par tabulation](#) » à la page 21.

La navigation par tabulation de Flash Lite utilise des boutons et des champs texte. Lorsqu'un champ texte de saisie a le focus et que l'utilisateur appuie sur la touche de sélection, Flash Lite ouvre la boîte de dialogue de saisie de texte du périphérique dans laquelle l'utilisateur peut saisir du texte. Pour obtenir un exemple d'utilisation des champs texte de saisie devant recevoir les entrées de l'utilisateur, consultez la section « [Exemple d'application de champ texte](#) » à la page 48.

Modes de navigation par tabulation

Flash Lite prend en charge trois modes de navigation par tabulation : navigation bidirectionnelle, navigation quadridirectionnelle et navigation quadridirectionnelle en boucle. Différents périphériques et types de contenus Flash Lite prennent en charge différents modes de navigation. Pour plus d'informations sur le choix du mode de navigation pour un type de périphérique ou de contenu spécifique, consultez la section « [Détermination des fonctionnalités de la plate-forme](#) » à la page 71.

Chaque mode de navigation présenté dans les sections suivantes fait référence à un exemple de fichier que vous pouvez visualiser au moyen d'Adobe Device Central. Chaque fichier d'exemple se compose de la même grille à neuf boutons, comme indiqué ci-après. La seule différence entre les fichiers d'exemple est que chacun est configuré pour cibler à la fois un périphérique et un type de contenu Flash Lite prenant en charge le mode de navigation concerné.



Tableau de boutons d'un fichier d'exemple

Pour utiliser un fichier d'exemple, ouvrez-le dans Adobe Flash CS4 Professionnel et testez-le dans Adobe Device Central (sélectionnez Contrôle > Tester l'animation). Les touches de direction du clavier du périphérique émulé (ou bien celles de votre clavier) permettent de voir dans quelle mesure chaque mode de navigation affecte la navigation de l'utilisateur.

Navigation bidirectionnelle En mode de navigation bidirectionnelle, les flèches Haut et Bas du périphérique déplacent le focus d'un objet à un autre (bouton ou champ texte de saisie) ; les flèches Gauche et Droite n'ont aucun effet. La touche Bas déplace le focus vers l'objet suivant situé à droite. Si aucun objet ne se trouve à droite de l'objet ayant le focus courant, ce dernier se déplace vers l'objet le plus à gauche situé sous l'objet sélectionné. Si aucun objet ne se trouve sous l'objet le plus à droite ayant le focus, ce dernier se déplace jusqu'à l'objet le plus à gauche situé au-dessus. La touche Haut déplace le focus vers l'objet suivant situé à gauche. Si aucun objet ne se trouve à gauche de l'objet ayant le focus courant, ce dernier se déplace vers l'objet le plus à droite situé au-dessus de l'objet sélectionné.

Pour obtenir un exemple de navigation bidirectionnelle, consultez le fichier d'exemple 2-way.fla qui se trouve à l'adresse www.adobe.com/go/learn_ftl_samples_and_tutorials_fr. Cette page Web regroupe des didacticiels et des exemples : repérez, téléchargez et décompressez le fichier .zip correspondant à votre version de Flash Lite, puis affichez le dossier Samples qu'il contient et sélectionnez l'exemple qui vous intéresse.

Navigation quadridirectionnelle En mode de navigation quadridirectionnelle, l'utilisateur peut déplacer le focus d'un objet à un autre à l'aide des touches Gauche, Droite, Haut et Bas. Appuyer sur la touche Gauche déplace le focus de l'objet actuellement sélectionné vers l'objet situé à gauche du bouton ayant le focus. La touche Droite déplace le focus vers le bouton suivant situé à droite du bouton ayant le focus. De la même manière, les touches Haut et Bas déplacent le focus vers le bouton situé au-dessus et en-dessous du bouton ayant le focus.

Pour obtenir un exemple de navigation quadridirectionnelle, consultez le fichier d'exemple 4-way.fla qui se trouve à l'adresse www.adobe.com/go/learn_ftl_samples_and_tutorials_fr. Cette page Web regroupe des didacticiels et des exemples : repérez, téléchargez et décompressez le fichier .zip correspondant à votre version de Flash Lite, puis affichez le dossier Samples qu'il contient et sélectionnez l'exemple qui vous intéresse.

Navigation quadridirectionnelle en boucle Ce mode est identique au mode de navigation quadridirectionnelle, mis à part qu'en l'absence de bouton sous le bouton le plus à droite ayant le focus, ce dernier se déplace vers le bouton le plus à gauche situé au-dessus.

A propos du rectangle de focus

Par défaut, Flash Lite dessine un rectangle jaune autour du bouton ou du champ texte de saisie ayant le focus. Ce rectangle de focus permet à l'utilisateur de connaître 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 courant et rectangle de focus par défaut

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

Vous pouvez désactiver le comportement par défaut du rectangle de focus en définissant la propriété `_focusRect` sur `false`. Si vous utilisez des boutons qui définissent les *étatsDessus*, Flash Lite les affiche lorsque le bouton reçoit le focus. Pour cette raison, le rectangle de focus s'avère souvent inutile lorsque vous utilisez des boutons. Par exemple, l'illustration suivante montre la même application que celle de l'illustration précédente, excepté que le rectangle de focus est désactivé ; le bouton ayant le focus indique son état « Dessus » :



Bouton ayant le focus courant (rectangle de focus désactivé)

Si votre application contient des champs texte de saisie, Adobe recommande de *ne pas* désactiver le rectangle de focus qui est le seul élément à pouvoir indiquer visuellement si un champ texte a le focus ou non. Par exemple, l'illustration suivante montre un champ texte de saisie ayant le focus courant :



Champ texte de saisie ayant le focus courant

Si votre application contient des boutons (dont les états « Dessus » sont définis) et des champs texte de saisie dans le même écran, vous pouvez définir pour chaque bouton la propriété `_focusRect` sur `false` dans le gestionnaire d'événements `on(rollOver)` et sur `true` dans le gestionnaire d'événements `on(rollOut)`, comme indiqué dans l'exemple de code suivant. Cette action oblige le rectangle de focus à apparaître lorsqu'un champ texte de saisie a le focus, mais pas quand il s'agit d'un bouton.

```
// Attach this code to each button on the Stage.  
on(rollOver) {  
    _focusRect = false;  
}  
on(rollOut) {  
    _focusRect = true;  
}
```

Pour plus d'informations sur l'utilisation des champs texte de saisie, consultez la section « [Utilisation des champs texte de saisie](#) » à la page 41.

Conseils pour l'utilisation de la navigation par tabulation

Lorsque vous utilisez la navigation par tabulation pour créer de l'interactivité, vous devez positionner au moins deux objets (champs texte de saisie, boutons ou une combinaison des deux) simultanément dans l'écran. Si l'écran ne contient qu'un seul bouton ou champ texte de saisie, l'utilisateur ne peut pas modifier le focus ; il risque alors de se sentir « coincé » dans l'interface utilisateur.

Si un écran de votre application ne contient qu'un seul bouton d'interaction utilisateur, préférez la détection d'un événement de pression de touche à l'utilisation d'événements de bouton. Pour plus d'informations, consultez la section « [Gestion des événements de touche](#) » à la page 24.

Gestion des événements de touche

Outre l'utilisation de la navigation par tabulation entre les boutons et les champs texte de saisie, une application Flash Lite peut également réagir à des événements de pression de touche arbitraires.

Certains périphériques et types de contenus ne prennent pas en charge toutes les touches de périphérique. Par exemple, sur un périphérique prenant en charge la navigation bidirectionnelle (voir « [Modes de navigation par tabulation](#) » à la page 21) Flash Lite ne génère pas d'événements de pression de touche pour les flèches Gauche et Droite.

Sur tous les périphériques, Flash Lite prend en charge les touches suivantes :

- Touches 0 à 9, * et #
- Touche de sélection

Sur les périphériques prenant en charge la navigation bidirectionnelle, Flash Lite prend également en charge les flèches Haut et Bas du clavier à cinq directions. Sur les périphériques prenant en charge la navigation quadridirectionnelle, Flash Lite prend en charge les flèches Haut, Bas, Gauche et Droite.

Sur les périphériques prenant en charge la commande `SetSoftKeys`, Flash Lite prend également en charge les touches programmables Gauche et Droite.

Rédaction d'un script de gestionnaire de touches

Pour gérer un événement de pression de touche, associez un gestionnaire `on(keyPress "key")` à une occurrence de bouton, dans lequel `key` correspond à l'un des noms d'événements de touche pris en charge. Par exemple, le code suivant, associé à une occurrence de bouton sur la scène, s'exécute lorsque l'utilisateur appuie sur la flèche Droite de son périphérique :

```
on(keyPress "<Right>") {
    trace("You pressed the right arrow key");
}
```

Le tableau suivant répertorie les événements de pression de touche ActionScript générés par Flash Lite en réponse à la pression d'une touche sur le périphérique de l'utilisateur :

Touche du périphérique	Événement de touche ActionScript	Disponibilité
0-9, *, #	0, 1, 2, 3, 4, 5, 6, 7, 8, -9, *, #	Tous les périphériques
Touche de sélection	<Entrée>	Tous les périphériques
Flèche gauche	<Gauche>	Périphériques prenant en charge la navigation quadridirectionnelle uniquement.
Flèche droite	<Aligner les bords droits>	Périphériques prenant en charge la navigation quadridirectionnelle uniquement.
Flèche haut	<Haut>	Périphériques prenant en charge la navigation quadridirectionnelle uniquement.
Flèche bas	<Enfoncé>	Périphériques prenant en charge la navigation quadridirectionnelle uniquement.
Touche programmable Gauche	<Page précédente>	Périphériques prenant en charge la commande <code>SetSoftKeys</code> uniquement.
Touche programmable Droite	<Page suivante>	Périphériques prenant en charge la commande <code>SetSoftKeys</code> uniquement.

Création d'un bouton de détecteur de touches

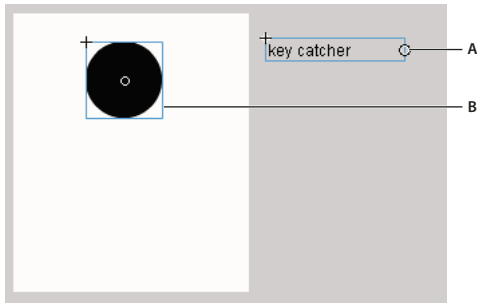
Si votre application doit gérer plusieurs événements de pression de touche différents, vous pouvez soit créer un bouton pour chaque événement de pression de touche, soit utiliser un bouton unique pour gérer tous ces événements. Ce type de bouton est généralement appelé bouton de *détecteur de touches* (ou écouteur de touches). L'utilisateur n'étant pas supposé voir le bouton de détecteur de touches, ce dernier est généralement placé hors de la scène (dans la zone de travail).

Les procédures suivantes décrivent comment utiliser un bouton de détecteur de touches pour gérer des événements de pression de touche dans une application simple. L'application permet à l'utilisateur de déplacer un cercle autour de la scène en appuyant sur les quatre touches de direction du périphérique.

Pour créer et utiliser un bouton de détecteur de touches :

- 1 Dans Flash, créez un document mobile et enregistrez-le sous `keycatcher fla`.
- 2 Sélectionnez le calque intitulé Contenu dans le scénario.
- 3 A l'aide de l'outil Ovale, créez un ovale ou un cercle sur la scène puis convertissez-le en clip.
- 4 Ce nouveau clip étant sélectionné, dans l'inspecteur des propriétés, tapez **circle** dans la zone de texte Nom de l'occurrence.

- 5 A l'aide de l'outil Texte, créez un champ texte contenant le texte **détecteur de touches**, puis convertissez-le en symbole de bouton.
- 6 Positionnez ce nouveau symbole de bouton dans la zone de travail entourant la scène. Pour afficher l'espace de travail entourant la scène, sélectionnez Affichage > Zone de travail.



A. Bouton de détecteur de touches B. Cercle de l'occurrence de clip

- 7 Sélectionnez l'occurrence du bouton de détection de touches et ouvrez le panneau Actions (Fenêtre > Actions).
- 8 Entrez le code suivant dans le panneau Actions :

```
on(keyPress "<Left>") {  
    circle._x -= 10;  
}  
on(keyPress "<Right>") {  
    circle._x += 10;  
}  
on(keyPress "<Up>") {  
    circle._y -= 10;  
}  
on(keyPress "<Down>") {  
    circle._y += 10;  
}
```

- 9 Testez l'application en sélectionnant Contrôle > Tester l'animation.

Appuyez sur les quatre flèches du clavier de l'émulateur Adobe Device Central pour déplacer le cercle autour de la scène.

Pour obtenir un autre exemple d'utilisation du bouton de détecteur de touches, consultez la section « [Création d'un menu simple à l'aide de clips](#) » à la page 26.

Création d'un menu simple à l'aide de clips

Dans cette section, vous découvrirez comment créer un menu simple en utilisant des clips. Plutôt que de recourir à la navigation par tabulation par défaut entre les boutons, qui implique l'association de code à chaque bouton, cette méthode utilise un bouton de détecteur de touches pour écouter les événements de pression de touche et mettre à jour l'interface utilisateur le cas échéant. Cette technique nécessite certes davantage de développement que la méthode de menu impliquant les boutons (voir « [Gestion des événements de touche](#) » à la page 24), mais elle offre certains avantages :

- Contrôler l'ordre de tabulation. Plutôt que d'avoir un ordre de tabulation déterminé par le lecteur Flash Lite (comme pour les boutons), le développeur peut choisir l'objet qui a le focus et sa manière de réagir aux événements de pression de touche.

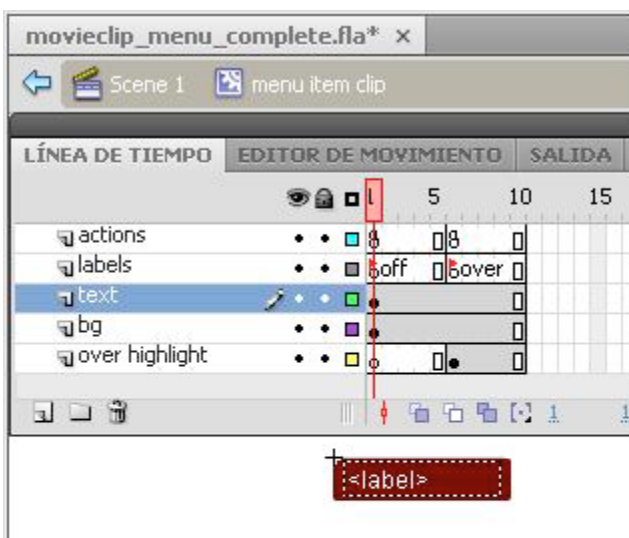
- Gérer la sélection du menu entre les états de l'application. Par exemple, vous souhaitez que votre application se « rappelle » du dernier élément de menu sélectionné par l'utilisateur pour vous permettre de renvoyer le focus sur le même élément ultérieurement. Ceci n'est pas possible si vous utilisez un menu impliquant les boutons car vous ne pouvez pas affecter de focus à un bouton à l'aide de code ActionScript.

Dans la procédure suivante, vous démarrez avec un document Flash partiellement terminé. Pour obtenir un exemple d'application finalisée (movieclip_menu_complete fla), consultez la page qui regroupe des didacticiels et des exemples pour Flash Lite à l'adresse www.adobe.com/go/learn_ftl_samples_and_tutorials_fr. Repérez et téléchargez le fichier .zip correspondant à votre version d'ActionScript, puis décompressez-le. Recherchez ensuite le dossier Samples et ouvrez-le pour afficher l'exemple qui vous intéresse.

Pour créer un menu simple à l'aide de clips :

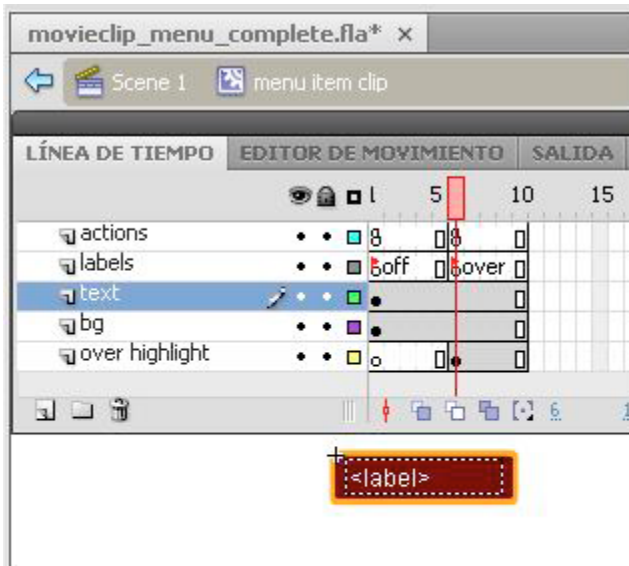
- 1 Téléchargez et ouvrez le fichier movieclip_menu_start fla qui se trouve à l'adresse www.adobe.com/go/learn_ftl_samples_and_tutorials_fr. Cette page Web regroupe des didacticiels et des exemples : repérez, téléchargez et décompressez le fichier .zip correspondant à votre version de Flash Lite, puis affichez le dossier Samples qu'il contient et sélectionnez l'exemple qui vous intéresse.
- 2 Dans le scénario, sélectionnez le calque intitulé Menu Items.
- 3 Ouvrez le panneau Bibliothèque (Fenêtre > Bibliothèque) et faites glisser une occurrence du symbole du clip intitulé Élément de menu depuis la bibliothèque vers la scène.

Ce clip contient deux images-clés, également appelées états visuels : une pour l'état initial non sélectionné de l'élément de menu, et l'autre pour son état sélectionné qui apparaît lorsque l'élément de menu reçoit le focus. L'illustration suivante présente la première image-clé du scénario du clip. Elle contient un champ texte dynamique devant afficher l'étiquette de l'élément de menu ainsi qu'un graphique d'arrière-plan rouge. Le champ texte et le graphique d'arrière-plan s'étendent sur toutes les images dans le scénario du clip.

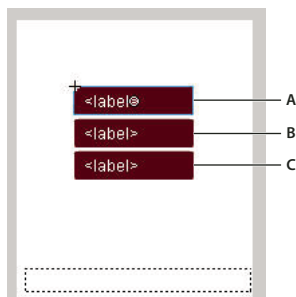


A. Etiquettes d'images. B. Champ texte dynamique ayant pour nom de variable label

L'illustration suivante présente l'image 5 du scénario du clip. La seule différence visuelle entre cette image et la première est la bordure en relief jaune qui entoure l'arrière-plan rouge de l'élément de menu.



- 4 Faites glisser deux autres occurrences du clip de l'élément de menu sur la scène et alignez-les verticalement dans une colonne.
- 5 Sélectionnez le clip supérieur et, dans l'inspecteur des propriétés, tapez **menu_1** dans la zone de texte Nom de l'occurrence.
- 6 Affectez respectivement les noms d'occurrences **menu_2** et **menu_3** aux clips intermédiaire et inférieur.



A. menu_1 B. menu_2 C. menu_3

Le suffixe numérique ajouté à chaque occurrence de nom vous permet de faire référence de manière dynamique à chaque clip dans le code que vous allez bientôt ajouter.

- 7 En utilisant l'outil Texte, créez un champ texte le long de la bordure inférieure de la scène.
- 8 Dans l'inspecteur des propriétés, choisissez Dynamique dans le menu déroulant Type de texte, puis tapez **status** dans la zone de texte Var.

Comme dans l'exemple du menu simple, ce champ texte affiche un message de statut sur l'élément de menu actuellement sélectionné.

9 Dans le scénario, sélectionnez la première image dans le calque intitulé Actions.

10 Ouvrez le panneau Actions (Fenêtre > Actions), puis tapez le code suivant :

```
// Initialize menu item labels:
menu_1.label = "News";
menu_2.label = "Sports";
menu_3.label = "Weather";

// Initialize variable that specifies number of menu items
numItems = 3;

// Initialize selectedItem variable, which contains
// the index of the current menu selection
selectedItem = 1;

// Initialize status text field
currentLabel = eval("menu_" + selectedItem + ".label");
status = "Press to select " + currentLabel;

// Send the first menu item to its "over" state
tellTarget("menu_1") {
    gotoAndStop("over");
}
```

11 Dans le scénario, sélectionnez le calque intitulé Key Catcher.

12 Ouvrez le panneau Bibliothèque et faites glisser une occurrence du bouton nommé Détecteur de touches sur la scène.

Vous allez ensuite associer un code de gestionnaire d'événements à ce bouton qui gère les événements de pression de touche de l'utilisateur, puis mettre à jour l'interface utilisateur.

13 Avec le bouton sélectionné sur la scène, ouvrez le panneau Actions.

14 Tapez (ou copiez et collez) le code suivant dans le panneau Actions :

```
on(keyPress "<Down>") {
    if(selectedItem < numItems) {
        // Turn off highlight on previously selected menu item:
        tellTarget("menu_" + selectedItem) {
            gotoAndStop("off");
        }
        // Increment selectedItem variable
        // and turn on highlight for new selection
        selectedItem++;
        tellTarget("menu_" + selectedItem) {
            gotoAndStop("over");
        }
        // Update status text field with label of selected item:
        currentLabel = eval("menu_" + selectedItem + ".label");
        status = "Press to select " + currentLabel;
    }
}

on(keyPress "<Up>") {
    if(selectedItem > 1) {
```

```

// Turn off highlight on previously selected item:
tellTarget("menu_" add selectedItem) {
    gotoAndStop("off");
}
// Increment selectedItem and turn on highlight for new selection
selectedItem++;
tellTarget("menu_" add selectedItem) {
    gotoAndStop("over");
}
// Update status text field with label of selected item:
currentLabel = eval("menu_" add selectedItem add ":label");
status = "Press to select " add currentLabel;
}
}

on(keyPress "<Enter>") {
    // Update status field with selected item
    status = "You selected " add currentLabel;
}

```

15 Sélectionnez Contrôle > Tester l'animation pour vérifier l'application dans l'émulateur Adobe Device Central.

Pour interagir avec le menu, cliquez sur les flèches Haut et Bas de l'émulateur ou appuyez sur les flèches correspondantes de votre clavier.

Gestion des événements de bouton

Flash Lite prend en charge les événements ActionScript de bouton suivants : `press`, `release`, `rollOver` et `rollOut`. Pour gérer ces événements, associez un gestionnaire `on` (*event*) à une occurrence de bouton, dans lequel *event* correspond à l'un des événements de bouton pris en charge et répertoriés dans le tableau suivant :

Événement de bouton	Moment où l'événement est généré
<code>press</code>	L'utilisateur appuie sur la touche de sélection du périphérique lorsque le bouton a le focus.
<code>release</code>	L'utilisateur relâche la touche de sélection du périphérique lorsque le bouton a le focus.
<code>rollOver</code>	Le bouton reçoit le focus.
<code>rollOut</code>	Le bouton perd le focus.

La procédure ci-après explique comment créer une application simple capable de gérer des événements de bouton. Pour visualiser un exemple d'utilisation des boutons en vue de créer un menu, consultez la section « [Création d'un menu simple à l'aide des boutons et de la navigation par tabulation](#) » à la page 32.

Pour créer un script de gestionnaire d'événements de bouton :

- 1 Dans Flash, créez un document mobile et enregistrez-le sous `button_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 Boutons du cercle pour l'ouvrir.

- 4 Faites glisser une occurrence du symbole de bouton Menu sur la scène.
- 5 Sélectionnez le bouton et ouvrez le panneau Actions (Fenêtre > Actions).
- 6 Tapez le code suivant dans le panneau Actions :

```
on(press) {  
    trace("You pressed Button 1");  
}  
on(release) {  
    trace("You released Button 1");  
}  
on(rollOver) {  
    trace("Button 1 has focus");  
}  
on(rollOut) {  
    trace("Button 1 lost focus");  
}
```

- 7 Faites glisser une autre occurrence du même bouton sur la scène et placez-la directement en dessous du premier bouton.
- 8 Avec le second bouton sélectionné sur la scène, ouvrez le panneau Actions et saisissez le code suivant :

```
on(press) {  
    trace("You pressed Button 2");  
}  
on(release) {  
    trace("You released Button 2");  
}  
on(rollOver) {  
    trace("Button 2 has focus");  
}  
on(rollOut) {  
    trace("Button 2 lost focus");  
}
```

- 9 Dans le scénario, sélectionnez l'image 1 dans le calque intitulé ActionScript.
- 10 Tapez le code suivant dans le panneau Actions :

```
_focusRect = false;
```

Cette action désactive le rectangle de focus jaune dessiné par Flash Lite autour du bouton ayant le focus. Dans ce cas, le rectangle de focus par défaut n'est pas nécessaire car le bouton contient un état « Dessus » qui s'affiche lorsqu'il a le focus.

- 11 Testez l'application dans l'émulateur Adobe Device Central (Contrôle > Tester l'animation).

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



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

Cette section explique comment créer un menu simple à l'aide de boutons et de la navigation par tabulation. Pour créer le menu, vous allez utiliser trois symboles de bouton, un pour chaque option du menu. Vous allez associer un code de gestion d'événements à chaque occurrence de bouton afin d'afficher un message lorsque l'utilisateur survole un élément de menu (autrement dit, lorsque l'utilisateur donne le focus au bouton correspondant) et qu'il sélectionne l'élément de menu en appuyant sur la touche de sélection de son périphérique. Pour plus d'informations sur la gestion des événements de bouton dans Flash Lite, consultez la section « [Gestion des événements de bouton](#) » à la page 30.

Vous allez commencer avec un document Flash partiellement terminé et pré-configuré pour cibler le téléphone Nokia 7610 et le type de contenu du lecteur autonome. Vous pouvez modifier ces paramètres pour cibler un périphérique ou un type de contenu différent (voir l'aide d'Adobe Device Central pour plus d'informations).

Pour créer un menu simple à l'aide de boutons :

- 1 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 qu'il contient et sélectionnez l'exemple qui vous intéresse.
- 2 Ouvrez le panneau Bibliothèque (Fenêtre > Bibliothèque).
La Bibliothèque contient trois symboles de bouton nommés News Button, Weather Button et Sports Button.
- 3 Dans le scénario (Fenêtre > Scénario), sélectionnez le calque intitulé Menu Buttons.
- 4 Faites glisser une occurrence du symbole News Button (Bouton Informations) du panneau Bibliothèque sur la scène.
- 5 Répétez l'étape 4 pour les boutons Sports et Weather (Météo).
- 6 Aligned les trois boutons verticalement, comme indiqué dans l'illustration suivante :



- 7 Dans le panneau Outils, sélectionnez l'outil Texte et tracez un champ texte le long de la partie inférieure de la scène.
Ce champ texte affiche un court message lorsque l'utilisateur survole les éléments du menu.
- 8 Avec le nouveau champ texte toujours sélectionné, effectuez les opérations suivantes dans l'inspecteur des propriétés :
 - a Sélectionnez Texte dynamique dans le menu déroulant Type de texte.
 - b Tapez **status** dans la zone de texte Var.
- 9 Sur la scène, sélectionnez le bouton News (Informations) et ouvrez le panneau Actions (Fenêtre > Actions).
- 10 Dans le panneau Actions, tapez le code suivant :

```

on(rollOver) {
    status = "Press to select News"
}
on(press) {
    status = "You selected news"
}

```

Ce code affecte un texte au champ texte dynamique si l'utilisateur survole le bouton de menu News.

11 Sélectionnez le bouton Sports puis, dans le panneau Actions, entrez le code suivant :

```

on(rollOver) {
    status = "Press to select Sports";
}
on(press) {
    status = "You selected Sports";
}

```

12 Sélectionnez le bouton Weather (Météo) puis, dans le panneau Actions, entrez le code suivant :

```

on(rollOver) {
    status = "Press to select Weather";
}
on(press) {
    status = "You selected Weather";
}

```

13 Dans le scénario, sélectionnez l'image 1 du calque intitulé Actions.

14 Dans le panneau Actions, tapez le code suivant :

```
_focusRect = false;
```

Il désactive l'affichage du rectangle de focus jaune que Flash Lite dessine par défaut autour des boutons et des champs texte de saisie ayant le focus (consultez la section « [A propos du rectangle de focus](#) » à la page 22).

A ce stade, la scène doit ressembler à l'illustration ci-dessous :



15 Sélectionnez Contrôle > Tester l'animation pour prévisualiser l'application dans l'émulateur Adobe Device Central.

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



Utilisation des touches programmables

Les touches programmables d'un périphérique sont des touches multifonctionnelles qui utilisent l'écran pour identifier leur cible à tout moment. Par exemple, dans l'application suivante, les étiquettes situées au-dessus des touches programmables indiquent que l'utilisateur peut appuyer sur la touche programmable Droite pour afficher le prochain plat, ou appuyer sur la touche Gauche pour retourner à l'écran d'accueil de l'application :



A. Etiquettes de touches programmables

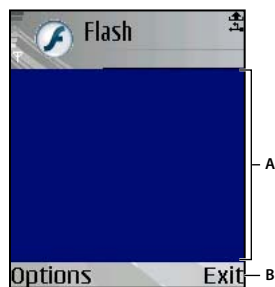
Pour utiliser les touches programmables Gauche et Droite, vous devez d'abord appeler la commande `SetSoftKeys` (voir `SetSoftKeys` dans le *Guide de référence du langage ActionScript Flash Lite 1.x*). Flash Lite génère alors un événement `PageDown` lorsque l'utilisateur appuie sur la touche programmable Droite et un événement `PageUp` lorsqu'il appuie sur la touche programmable Gauche. Vous pouvez rédiger un code de gestionnaire d'événements ActionScript qui réagit à ces événements et répond par l'action désirée.

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

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

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

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



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

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

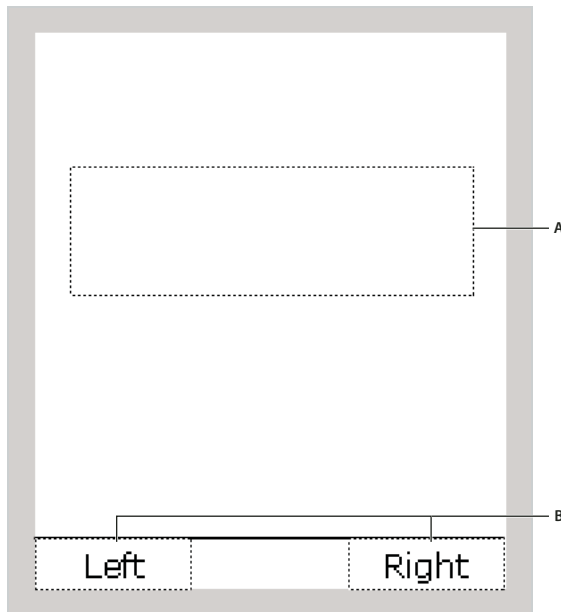


Étiquettes de touches programmables personnalisées

Pour utiliser les touches programmables dans une application :

- 1 Dans Flash, créez un document mobile.
- 2 Sous l'onglet Nouveau document de Device Central, sélectionnez le type de contenu du lecteur autonome. Sous l'onglet Profils de périphérique, ajoutez un des périphériques du dossier Nokia à votre liste de périphériques de test.
- 3 Dans le scénario, sélectionnez le calque intitulé Content.
- 4 A l'aide de l'outil Texte, créez un champ texte statique nommé Gauche (ou du nom de votre choix) et placez-le dans le coin inférieur gauche de la scène, au-dessus de la touche programmable Gauche du périphérique.
- 5 Créez un autre champ texte statique nommé Droite et placez-le dans le coin inférieur droit de la scène, au-dessus de la touche programmable Droite du périphérique.
- 6 En utilisant l'outil Texte, créez un autre champ texte dynamique et placez-le au milieu de la scène.

Ce champ texte affiche un message lorsque vous exécutez l'application et appuyez sur les touches programmables Gauche et Droite. La scène de votre document doit avoir l'aspect suivant :



A. Champ texte du message de statut B. Etiquettes de touches programmables

Remarque : Dans une véritable application, il pourrait être souhaitable d'utiliser d'autres éléments que de simples champs texte pour les étiquettes des touches programmables, comme par exemple des symboles de graphiques ou de clips.

- 7 Le champ texte d'état étant toujours sélectionné, dans l'inspecteur des propriétés tapez **status** dans la zone de texte Var.
- 8 Créez un bouton de détecteur de touches (consultez la section « [Création d'un bouton de détecteur de touches](#) » à la page 25). Dans le panneau Actions, associez le code suivant au bouton :

```
// Handle Left soft keypress event
on(keyPress "<PageUp>") {
    status = "You pressed the Left soft key.";
}
// Handle Right soft keypress event
on(keyPress "<PageDown>") {
    status = "You pressed the Right soft key.";
}
```

- 9 Dans le scénario, sélectionnez l'image 1 dans le calque intitulé Actions.

- 10 Dans le panneau Actions, tapez le code suivant :

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

Les deux paramètres de la commande `SetSoftKeys`, Gauche et Droite dans le cas présent, spécifient les étiquettes affichées par Flash Lite au-dessus des touches programmables lorsque l'affichage de l'application n'est pas en mode plein écran. Dans ce cas, la commande `FullScreen` (voir `FullScreen` dans le *Guide de référence du langage ActionScript Flash Lite 1.x*) est utilisée pour obliger l'application à s'afficher en mode plein écran. Par conséquent, les valeurs choisies pour ces paramètres peuvent être des chaînes de texte ou des expressions arbitraires.

```
fscommand2("SetSoftKeys", foo, bar);
```

- 11 Sélectionnez Contrôle > Tester l'animation pour vérifier l'application dans Adobe Device Central.

Cliquez sur les touches programmables Gauche et Droite ou appuyez sur les touches Page précédente et Page suivante de votre clavier pour tester l'application.



Chapitre 5 : Utilisation du texte et des polices

Ce chapitre explique comment ajouter des champs texte statique et dynamique ou des champs texte de saisie à vos applications Macromedia Flash Lite 1.x d'Adobe.

A propos du texte dans Flash Lite

Flash Lite prend en charge trois types de champs texte : statique, dynamique et de saisie. Le contenu d'un champ texte statique ne change pas lors de la lecture. Par exemple, vous pouvez utiliser les champs texte statique pour afficher des titres de page ou des étiquettes. Pour plus d'informations sur la création d'un champ texte statique, consultez la section « [Exemple d'application de champ texte](#) » à la page 48.

Les champs texte dynamique permettent de contrôler le contenu lors de l'exécution. Vous pouvez associer un nom de variable ActionScript à un champ texte dynamique auquel vous pouvez faire référence dans le code. Par exemple, si vous concevez une calculatrice dans Flash Lite, vous pouvez utiliser un champ texte dynamique pour afficher les résultats des calculs. Pour plus d'informations sur l'utilisation des champs texte dynamique, consultez la section « [Affectation d'un nom de variable à un champ texte](#) » à la page 41.

Un champ texte de saisie ressemble à un champ texte dynamique, sauf que l'utilisateur peut interagir avec lui pour ouvrir la boîte de dialogue de saisie du périphérique et saisir du texte. Pour plus d'informations sur l'utilisation des champs texte de saisie, consultez la section « [Utilisation des champs texte de saisie](#) » à la page 41.

Présentation des méthodes de rendu des polices dans Flash Lite

Pour rendre du texte sur un écran de périphérique, Flash Lite peut soit utiliser les polices disponibles sur le périphérique, soit utiliser les données de police intégrées au fichier SWF. Les polices de périphérique ont pour avantage de réduire la taille des fichiers SWF, mais elles ne permettent qu'un contrôle limité de l'affichage. Lorsque vous intégrez des données de police au fichier SWF, vous bénéficiez d'un meilleur contrôle sur l'affichage des polices, mais la taille du fichier est plus volumineuse.

Pour les données de police intégrées, Flash Lite prend en charge le texte anti-aliasé et aliasé (bitmap), ce qui facilite la lisibilité des caractères de petite taille. Vous pouvez également utiliser des polices de pixels provenant de concepteurs tiers pour améliorer la lisibilité des petits caractères. Pour plus d'informations, consultez les sections « [Méthodes de rendu des polices dans Flash Lite](#) » à la page 45 et « [Intégration des polices vectorielles dans les fichiers SWF](#) » à la page 47.

Fonctionnalités de texte de Flash Player non prises en charge dans Flash Lite 1.x

Les champs texte de Flash Lite 1.x ne prennent pas en charge les fonctionnalités suivantes disponibles dans les versions de bureau de Flash Player :

- La technologie avancée de rendu des polices disponible dans Flash Player 8 et les versions ultérieures.
- HTML, CSS ou toute autre fonctionnalité de format de texte enrichi.
- Les objets TextField et TextFormat.

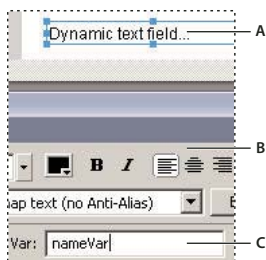
Affectation d'un nom de variable à un champ texte

Vous pouvez associer des champs texte dynamique et de saisie à un nom de variable ActionScript, ce qui vous permet de récupérer ou de définir le contenu du texte utilisant du code ActionScript.

Pour associer un nom de variable à un champ texte dynamique ou de saisie :

- 1 Dans Flash, créez un document mobile et enregistrez-le sous `dynamic_text.fla`.
- 2 Créer un champ texte dynamique ou de saisie sur la scène et sélectionnez-le.
- 3 Dans la zone de texte Var de l'inspecteur Propriétés, tapez **nameVar**.

La valeur saisie doit être un identifiant de variable valide ; ainsi, le premier caractère doit être une lettre, un trait de soulignement (`_`) ou le signe dollar (`$`), et chaque caractère suivant doit être une lettre, un chiffre, un trait de soulignement ou le signe dollar.



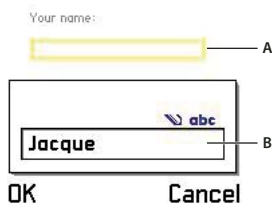
A. Champ texte dynamique sélectionné sur la scène B. Inspecteur Propriétés C. Nom de variable affecté au champ texte sélectionné

- 4 Dans le scénario, sélectionnez l'image 1 dans le calque intitulé ActionScript.
- 5 Ouvrez le panneau Actions (Fenêtre > Actions), puis tapez le code indiqué ci-dessous :
`nameVar = "George Washington"`
- 6 Testez l'application dans l'émulateur Adobe Device Central (Contrôle > Tester l'animation).

Pour obtenir un autre exemple de l'utilisation des champs texte dynamique, consultez la section « [Exemple d'application de champ texte](#) » à la page 48.

Utilisation des champs texte de saisie

Dans Flash Lite, les champs texte de saisie, tels que les champs texte dynamique, permettent de récupérer et de définir leur contenu lors de l'exécution à l'aide de code ActionScript. En outre, les champs texte de saisie permettent aux applications Flash Lite de récupérer le texte saisi par l'utilisateur à l'aide de la boîte de dialogue de texte de saisie du périphérique. (Flash Lite ne prend pas en charge la saisie de texte en ligne.) L'illustration suivante montre une boîte de dialogue de saisie sur un périphérique Symbian Series 60 :



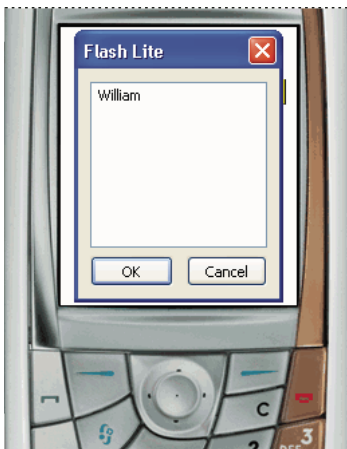
A. Champ texte de saisie Boîte de dialogue de saisie de texte du périphérique B. Boîte de dialogue de texte de saisie du périphérique

Pour ouvrir la boîte de dialogue de saisie du périphérique, l'utilisateur doit d'abord donner le focus à un champ de saisie, puis appuyer sur la touche de sélection. Par défaut, Flash Lite dessine un rectangle jaune autour du champ texte de saisie ayant le focus.

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

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

Lorsque vous testez votre application dans Adobe Flash , l'émulateur Adobe Device Central reproduit les fonctionnalités de la boîte de dialogue de saisie de texte. L'illustration suivante montre la boîte de dialogue de saisie de texte fonctionnant dans l'émulateur :

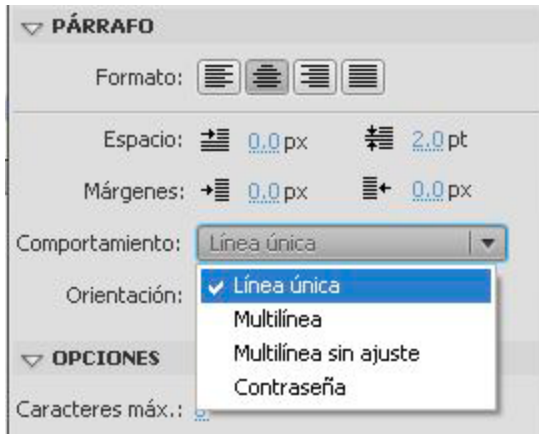


Boîte de dialogue de saisie de texte fonctionnant dans l'émulateur

Pour obtenir un exemple d'utilisation d'un champ texte de saisie dans une application, consultez la section « [Exemple d'application de champ texte](#) » à la page 48.

Types de champs texte de saisie

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



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

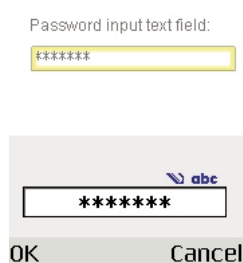
Par exemple, lorsque l'utilisateur modifie un champ texte de saisie sur une seule ligne, la boîte de dialogue de saisie du périphérique affiche une zone de texte de saisie sur une seule ligne. La zone de saisie défile horizontalement si l'utilisateur saisit plus de caractères qu'elle ne peut en afficher. L'illustration suivante montre la boîte de dialogue de saisie de texte d'un périphérique pour un champ texte de saisie sur une seule ligne :



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



Lorsqu'un utilisateur modifie un champ de saisie de mot de passe, la boîte de dialogue de saisie de texte du périphérique masque tous les caractères saisis à l'aide d'astérisques (après un bref délai).



Limitation des caractères saisis

La commande `SetInputTextType` permet de limiter les caractères que l'utilisateur peut entrer dans la boîte de dialogue de saisie. Supposons qu'une application contienne un champ texte de saisie devant permettre à l'utilisateur d'indiquer une valeur numérique, telle que son âge. Et supposons par ailleurs que ce champ texte de saisie possède le nom de variable `ageVar`.

Afin de garantir que l'utilisateur ne saisit que des valeurs numériques dans la boîte de dialogue de saisie, vous pouvez ajouter le code suivant à votre application :

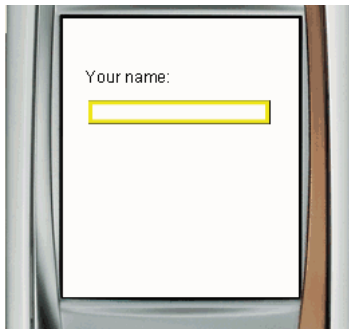
```
fscommand2("SetInputTextType", "ageVar", "Numeric");
```

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

Pour plus d'informations, consultez la rubrique consacrée à la commande `SetInputTextType` dans le *Guide de référence du langage ActionScript Flash Lite 1.x*.

Champs texte de saisie et rectangle de focus

Par défaut, Flash Lite dessine un rectangle jaune autour du champ de saisie ayant le focus du clavier, comme indiqué dans l'illustration suivante :



Vous pouvez désactiver le rectangle de focus en définissant la propriété globale `_focusRect` sur `false`. Cependant, dans ce cas, l'utilisateur n'aura aucun moyen de voir que le champ texte a le focus du clavier et ne saura pas s'il doit appuyer sur la touche de sélection du périphérique. Adobe recommande de ne pas désactiver le rectangle de focus si vous utilisez des champs texte de saisie.

Pour plus d'informations sur le contrôle du comportement du rectangle de focus jaune, consultez la section `_focusrect` dans le Guide de référence du langage ActionScript Flash Lite 1.x.

Méthodes de rendu des polices dans Flash Lite

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

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

Rendre la police sous forme de bitmap Flash Lite rend du texte bitmap en alignant les polices vectorielles sur les limites des pixels, ce qui facilite la lisibilité des petits caractères (10 points ou moins). Cette option exige d'inclure les polices vectorielles dans le fichier SWF publié pour la police sélectionnée (pour plus d'informations, consultez la section « [Intégration des polices vectorielles dans les fichiers SWF](#) » à la page 47).

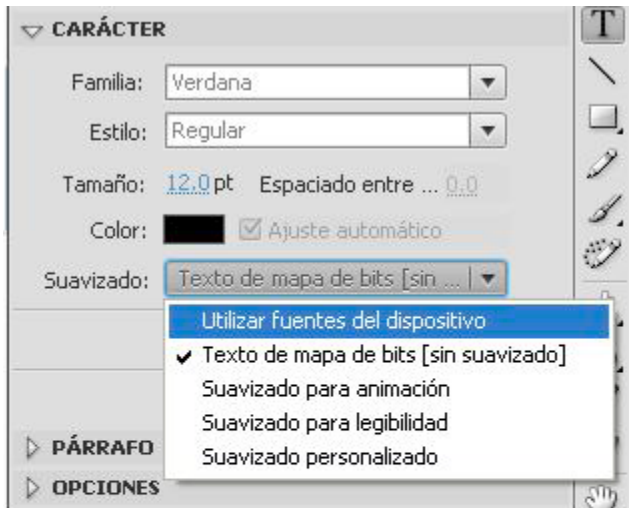
Rendre la police sous forme de vecteurs anti-aliasés Flash Lite rend du texte anti-aliasé en utilisant des représentations vectorielles des polices vectorielles intégrées au fichier SWF publié (pour plus d'informations, consultez la section « [Intégration des polices vectorielles dans les fichiers SWF](#) » à la page 47).

Le menu contextuel Méthode de rendu des polices dans l'inspecteur Propriétés permet de sélectionner une méthode de rendu des polices pour un champ texte. Ce menu contextuel contient cinq options de rendu ; cependant, trois seulement sont disponibles pour les développeurs Flash Lite 1.0 et 1.1. Les deux autres méthodes (Anti-alias pour la lisibilité et Anti-alias personnalisé) sont disponibles uniquement pour les applications ciblant Flash Player 8 ou les versions ultérieures.

Pour sélectionner une méthode de rendu des polices pour un champ texte :

- 1 Sélectionnez un champ texte sur la scène.

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



- Sélectionnez Utiliser les polices de périphérique pour que Flash Lite utilise une police disponible sur le périphérique. Aucune donnée de police n'est intégrée au fichier SWF publié.
- Sélectionnez Texte bitmap (sans anti-alias) pour que Flash Lite aligne les polices vectorielles sur les limites des pixels, ce qui permet un affichage clair et net des petits caractères. Cette option exige que Flash intègre les polices vectorielles dans le fichier SWF publié (pour plus d'informations, consultez la section « [Intégration des polices vectorielles dans les fichiers SWF](#) » à la page 47).
- Sélectionnez Anti-alias pour l'animation afin que Flash Lite utilise l'anticrénelage sur le champ de texte en fonction du paramètre actuel de qualité de rendu (pour de plus amples informations, consultez « [Qualité de rendu et texte anti-aliasé de Flash Lite](#) » à la page 47-). Cette option nécessite que Flash intègre les polices vectorielles dans le fichier SWF publié (pour de plus amples informations consultez « [Qualité de rendu et texte anti-aliasé de Flash Lite](#) » à la page 47-).

Flash Lite 1.0 et 1.1 ne prennent pas en charge les options de rendu des polices Anti-alias pour la lisibilité et Anti-alias personnalisé. Ces options de rendu sont uniquement disponibles à partir de Flash Player 8 sur les ordinateurs de bureau.

Qualité de rendu et texte anti-aliasé de Flash Lite

Flash Lite propose trois paramètres de qualité de rendu : low (faible), medium (moyenne) et high (haute). Plus la qualité du rendu est haute, plus le rendu des contours vectoriels dans Flash Lite est lisse et précis ; si la qualité est plus faible, les contours dessinés sont moins lisses. Par défaut, Flash Lite utilise une qualité moyenne pour le rendu des polices vectorielles. Vous pouvez contrôler la qualité de rendu à l'aide de la commande `SetQuality` (pour plus d'informations, consultez la rubrique consacrée à `SetQuality` dans le *Guide de référence du langage ActionScript Flash Lite 1.x*).

Flash Lite rend le texte anti-aliasé en utilisant des représentations vectorielles des polices vectorielles. Pour obtenir un affichage aussi lisse que possible du texte anti-aliasé, vous devez définir la qualité de rendu du lecteur sur « high » (élevée). La qualité de rendu affecte tout le contenu vectoriel à l'écran, pas uniquement le texte anti-aliasé. Les figures suivantes montrent les qualités de rendu élevée, moyenne et faible pour un champ texte anti-aliasé (Arial, 24 points) :

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

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

Intégration des polices vectorielles dans les fichiers SWF

Pour rendre une police de champ texte, Flash Lite peut utiliser les polices disponibles sur le périphérique ou les polices vectorielles intégrées au fichier SWF publié (pour plus d'informations, consultez la section « [Méthodes de rendu des polices dans Flash Lite](#) » à la page 45). L'intégration des polices vectorielles dans le fichier SWF permet de garantir un affichage identique de la police du champ texte sur toutes les plates-formes cibles, mais implique une taille de fichier plus volumineuse. Flash Lite oblige les polices vectorielles à rendre soit du texte bitmap (sans anti-alias) soit du texte anti-aliasé.

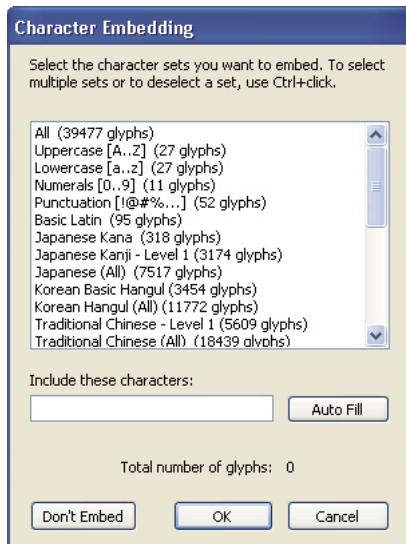
Pour les champs texte statique qui utilisent les méthodes de rendu des polices anti-alias ou bitmap, Flash intègre automatiquement les polices vectorielles requises pour afficher le contenu du champ texte. Par exemple, si un champ texte statique contient le mot *Envoyer*, Flash intègre automatiquement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

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

Pour intégrer des polices vectorielles pour un champ texte dynamique ou de saisie :

- 1 Sélectionnez le champ texte dynamique ou de saisie sur la scène.

- 2 Dans l'inspecteur Propriétés, sélectionnez Bitmap (sans anti-aliasing) ou Anti-alias pour l'animation dans le menu contextuel Méthode de rendu des polices.
- 3 Cliquez sur le bouton Intégrer qui se trouve en regard du menu Méthode de rendu des polices pour ouvrir la boîte de dialogue Intégration de caractères.



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

Exemple d'application de champ texte

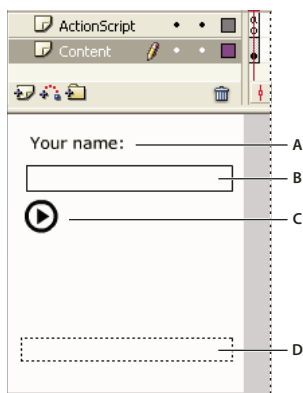
Cette section indique comment créer une application simple qui utilise les champs texte statique, dynamique et de saisie. L'application demande à l'utilisateur (à l'aide de texte dans un champ texte statique) d'indiquer son nom (à l'aide d'un champ texte de saisie). Une fois que l'utilisateur a saisi son nom et sélectionné un bouton, l'application affiche un message (à l'aide d'un champ texte dynamique) à son attention.

Pour créer un exemple d'application de champ texte :

- 1 Dans Flash, créez un document mobile.
- 2 Dans le scénario (Fenêtre > Scénario), sélectionnez la première image dans le calque intitulé ActionScript.
- 3 Ouvrez le panneau Actions (Fenêtre > Actions).
- 4 Tapez **stop()**; dans le panneau Actions pour arrêter la tête de lecture sur cette image.
- 5 Sélectionnez l'outil Texte dans le panneau Outils pour créer un champ texte contenant le texte « Votre nom : » sur la scène.
- 6 Avec le nouveau champ texte sélectionné, dans le menu contextuel Méthode de rendu des polices de l'inspecteur Propriétés, sélectionnez Utiliser les polices de périphérique.

- 7 Créez un nouveau champ texte sous le premier et, dans l'inspecteur Propriétés, sélectionnez Texte de saisie dans le menu contextuel Type de texte. Sélectionnez Utiliser les polices de périphérique dans le menu contextuel Méthode de rendu des polices, tapez **nameVar** dans la zone de texte Var, puis choisissez l'option Afficher la bordure autour du texte.
- 8 Sélectionnez Fenêtre > Bibliothèques communes > Boutons pour ouvrir une bibliothèque de symboles de bouton prédéfinis.
- 9 Dans la bibliothèque de boutons, cliquez deux fois sur le dossier Circle Buttons pour l'ouvrir.
- 10 Faites glisser une occurrence du symbole « circle button – next » sur la scène.
- 11 En utilisant l'outil Texte, créez un autre champ texte dans la partie inférieure de la scène.
 Ce champ texte affichera un message contenant le nom saisi par l'utilisateur dans le champ texte de saisie.
- 12 Le nouveau champ texte étant sélectionné, tapez **message** dans la zone de texte Var de l'inspecteur Propriétés. Sélectionnez Dynamique dans le menu contextuel Type de texte, puis choisissez Utiliser les polices de périphérique dans le menu contextuel Méthode de rendu des polices.

La scène de votre application doit ressembler à l'illustration suivante :



A. Champ texte statique B. Champ texte de saisie (nameVar) C. Bouton D. Champ texte dynamique (message)

- 13 Sélectionnez le bouton situé sur la scène et ouvrez le panneau Actions (Fenêtre > Actions).
- 14 Entrez le code suivant dans le panneau Actions :

```
on (press) {
    message = "Hello, " + nameVar;
}
```

15 Sélectionnez Contrôle > Tester l'animation pour vérifier l'application dans l'émulateur Adobe Device Central.

- a Appuyez sur la flèche Bas du clavier de l'émulateur pour donner le focus au champ texte de saisie.



- b Appuyez sur la touche de sélection de l'émulateur pour ouvrir sa boîte de dialogue de texte de saisie, puis saisissez votre nom en utilisant le clavier de votre ordinateur.



- c Appuyez sur OK pour fermer la boîte de dialogue de texte de saisie.

- d Appuyez à nouveau sur la flèche Bas du clavier de l'émulateur pour donner le focus au bouton, puis appuyez sur la touche de sélection.



Création de texte défilant

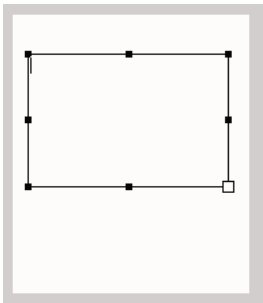
Flash Lite 1.1 prend en charge les propriétés de champ texte `scroll` et `maxscroll`, ce qui permet de créer des champs texte défilants. La propriété `scroll` désigne la première ligne visible d'un bloc de texte ; vous pouvez lire et définir sa valeur. Par exemple, le code suivant fait défiler de cinq lignes vers le bas le champ texte dont le nom de variable est `story_text` :

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

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

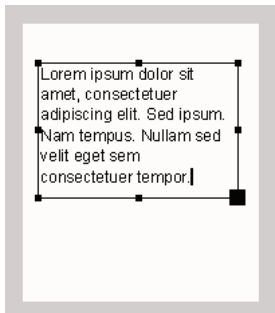
Pour créer un champ texte défilant et le contrôler à l'aide de code ActionScript :

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



- 3 Dans le menu contextuel Type de ligne de l'inspecteur Propriétés, sélectionnez Multiligne.
- 4 Dans l'inspecteur Propriétés, choisissez Texte dynamique dans le menu contextuel Type de texte.

- 5 Dans le menu contextuel Méthode de rendu des polices de l'inspecteur Propriétés, sélectionnez Utiliser les polices de périphérique.
- 6 Sélectionnez Texte > Défilant pour faire défiler le champ texte.
- 7 Dans la zone de texte Var de l'inspecteur Propriétés, tapez **story**. Cette action associe la variable ActionScript `story` au champ texte.
- 8 Double-cliquez dans le champ texte et tapez un texte suffisamment long pour qu'une ou plusieurs lignes dépassent de sa bordure inférieure.



- 9 Créez un nouveau symbole de bouton, puis ajoutez-en une occurrence sur la scène ou dans la zone en dehors de la scène.

Ce bouton agit comme un bouton de *détecteur de touches* et il n'est pas nécessaire que l'utilisateur le voit. Pour plus d'informations sur la création de boutons de détecteur de touches, consultez la section « [Création d'un bouton de détecteur de touches](#) » à la page 25.

- 10 Sélectionnez le bouton et ouvrez le panneau Actions (Fenêtre > Actions).

- 11 Entrez le code suivant dans le panneau Actions :

```
on(keyPress "<Down>") {
    story.scroll++;
}
on(keyPress "<Up>") {
    story.scroll--;
}
```

- 12 Sélectionnez Contrôle > Tester l'animation pour vérifier l'application dans l'émulateur Adobe Device Central.

Appuyez sur les flèches Haut et Bas de votre clavier (ou les boutons Haut et Bas du clavier de l'émulateur) pour faire défiler le texte vers le haut ou le bas.

Pour plus de simplicité, cet exemple permet de saisir le contenu du champ texte dans l'outil de programmation. Vous pouvez néanmoins facilement modifier l'exemple pour que le contenu du champ texte soit mis à jour à l'aide de code ActionScript. Pour ce faire, vous pouvez rédiger du code ActionScript qui affecte le texte souhaité au nom de variable que vous avez associé au champ texte multiligne (`story`, dans cet exemple).

```
story = "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed ipsum. Nam tempus. Nullam sed velit eget sem consectetur tempor. Morbi eleifend venenatis pede. Cras ac lorem eget massa tincidunt iaculis...etc."
```


Chapitre 6 : Utilisation du son

Macromedia Flash Lite 1.1 d'Adobe prend en charge les sons de périphérique et les sons Flash standard ou natifs. Ce chapitre explique comment incorporer du son dans vos applications sur des périphériques mobiles.

A propos du son dans Flash Lite

Flash Lite 1.1 prend en charge deux types de son : le son de périphérique et le son Flash standard (*natif*). Les sons de périphérique sont stockés dans le fichier SWF publié au format audio natif du périphérique, tel que MIDI ou MFi. Pour lire le son de périphérique, Flash Lite transmet les données audio au périphérique qui les décode et les lit. Flash Lite ne peut pas synchroniser les sons de périphérique avec une animation. Flash Lite 1.0 prend en charge uniquement les sons de périphérique.

Outre les formats audio de périphérique pris en charge par Flash Lite 1.0, Flash Lite 1.1 prend en charge le format audio Flash natif, autrement dit le type de son pris en charge par la version pour ordinateurs de bureau d'Adobe Flash Player. Contrairement aux sons de périphérique, les sons natifs peuvent être synchronisés sur une animation dans le scénario.

Flash Lite 1.0 prend en charge les formats audio de périphérique MIDI et MFi. Outre ces formats, Flash Lite 1.1 prend en charge le format audio de périphérique SMAF (Synthetic Music Mobile Application), ainsi que le son Flash natif compressé au format PCM (ou WAV), ADPCM et MP3.

Sons d'événement et sons en flux continu (synchronisés)

Flash Lite 1.1 prend en charge les sons d'événement et les sons en flux continu (*synchronisés*). Les sons d'événement sont lus indépendamment du scénario et leur lecture se poursuit jusqu'à la fin du tampon audio ou l'arrêt à l'aide de code ActionScript. Les sons d'événement doivent être totalement téléchargés avant de commencer leur lecture.

Les sons en flux continu sont synchronisés avec le scénario sur lequel ils se trouvent et sont souvent utilisés pour synchroniser des fichiers audio avec des animations. Les sons en flux continu s'arrêtent lorsque la tête de lecture du scénario est arrêtée. Lors de la lecture, Flash Lite omet des images de l'animation, si nécessaire, pour maintenir la synchronisation avec le son.

Seul le son Flash natif peut être synchronisé avec le scénario ; vous ne pouvez utiliser les sons de périphérique que sous forme de sons d'événement. Flash Lite 1.0 prend en charge uniquement les sons d'événement.

Utilisation des sons de périphérique

Un *son de périphérique* est un son codé au format audio natif du périphérique, tel que MIDI ou MFi. Flash Lite ne permet pas d'importer directement des fichiers audio de périphérique dans un document Flash. Vous devez d'abord importer un fichier audio de proxy dans un format pris en charge comme MP3, WAV ou AIFF. Vous devez ensuite lier le fichier audio de proxy à un son de périphérique portable externe (un fichier MIDI, par exemple). Pendant le processus de publication du document, le fichier audio de proxy est remplacé par ce son. Le fichier SWF généré contient le son externe et l'utilise pour la lecture sur un périphérique portable.

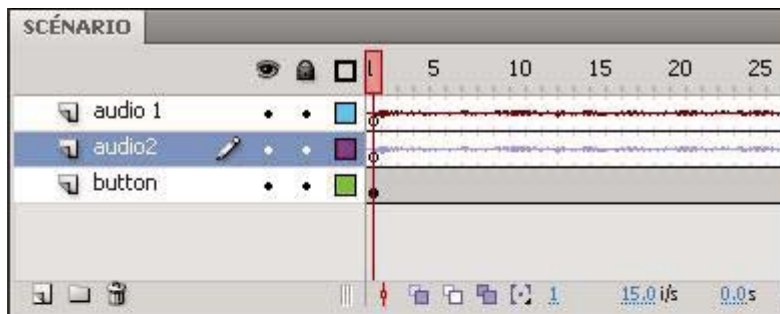
Lorsque vous utilisez des sons de périphérique dans Flash Lite, gardez à l'esprit les consignes suivantes :

- Les sons de périphérique ne peuvent être utilisés que sous forme de sons d'événement ; vous ne pouvez pas synchroniser des sons de périphérique dans un scénario.
- Flash Lite ne prend pas en charge les options Effet, Sync et Modifier pour les sons de périphérique.
- Vous devez spécifier un fichier audio de périphérique externe pour chaque son d'un document.
- Comme avec tous les fichiers externes, le fichier audio de périphérique ou le fichier de regroupement audio doit être disponible lors de la publication de votre fichier SWF, mais il n'est pas requis par ce dernier pour la lecture.

Vous pouvez également regrouper plusieurs sons de périphérique sur un même fichier. Cette opération s'avère utile si vous créez le même contenu pour plusieurs périphériques prenant en charge différents formats audio de périphérique. Pour plus d'informations, consultez la section « [Utilisation des sons composés](#) » à la page 59.

Dans Flash Lite 1.1, un son de périphérique peut être lu à tout moment. Dans Flash Lite 1.0, un son de périphérique ne peut être lu qu'en réponse à la pression d'une touche par l'utilisateur sur son périphérique. Pour plus d'informations, consultez la section « [Déclenchement des sons de périphérique dans Flash Lite 1.0](#) » à la page 58.

Dans l'outil de programmation de Flash, le scénario affiche les formes d'onde comme indiqué dans l'illustration suivante. Les formes d'onde des sons liés à des sons de périphérique externes sont de couleur verte ; les formes d'onde des sons qui ne sont pas liés à des sons de périphérique externes sont de couleur bleue.



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

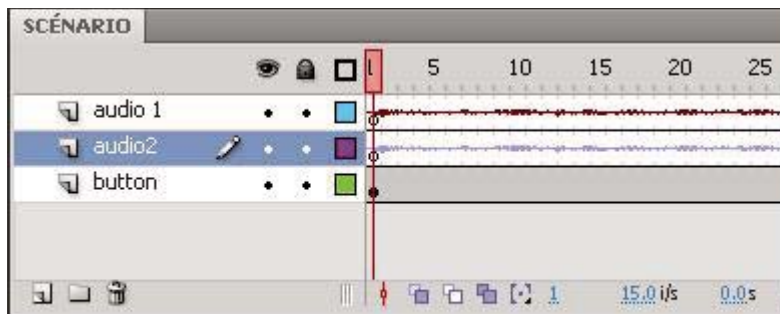
Ajout d'un son de périphérique à un bouton

La procédure suivante explique comment ajouter un son de périphérique au scénario d'un symbole de bouton pour que le son soit lu lorsque l'utilisateur « clique » sur ce bouton (autrement dit, lorsqu'il appuie sur la touche de sélection de son périphérique quand le bouton a le focus). Pour ce faire, vous devez associer le son de proxy à l'image Abaissé dans le scénario d'un symbole de bouton. Vous devez ensuite associer au son de proxy le son de périphérique à lire.

Une version finale de l'application `button_sound_complete fla` se trouve à l'adresse www.adobe.com/go/learn_ftl_samples_and_tutorials. Cette page Web regroupe des didacticiels et des exemples : repérez, téléchargez et décompressez le fichier .zip correspondant à votre version de Flash Lite, puis affichez le dossier Samples qu'il contient et sélectionnez l'exemple qui vous intéresse.

Pour ajouter un son de périphérique à un document Flash :

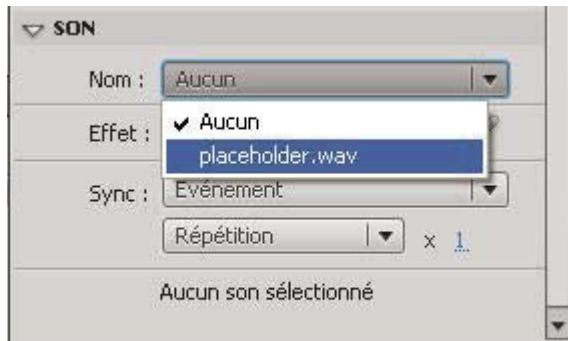
- 1 Dans Flash, créez un document mobile et enregistrez-le sous `device_sound fla`.
- 2 Sélectionnez Fichier > Importer > Importer dans la bibliothèque. Dans la page regroupant didacticiels et exemples à l'adresse www.adobe.com/go/learn_ftl_samples_and_tutorials, repérez, téléchargez et décompressez le fichier .zip correspondant à votre version de Flash Lite, puis naviguez jusqu'au dossier Samples. Sélectionnez le fichier `proxy.wav` et cliquez sur OK.
Ce fichier audio agit comme le son de périphérique pour le son de périphérique à inclure.
- 3 Sélectionnez Fenêtre > Bibliothèques communes > Boutons.
Cette action ouvre une bibliothèque externe contenant des symboles de bouton prédéfinis.
- 4 Dans la bibliothèque de boutons, cliquez deux fois sur le dossier Circle Buttons pour l'ouvrir.
- 5 Dans le scénario, sélectionnez le calque Content.
- 6 Faites glisser le symbole de bouton Lecture depuis la bibliothèque de boutons jusqu'à la scène.
- 7 Double-cliquez sur le nouveau bouton pour l'ouvrir en mode d'édition.
Le scénario change et affiche les images Haut, Dessus, Abaissé et Cliqué du bouton.
- 8 Sélectionnez Insertion > Scénario > Calque pour créer un nouveau calque.
- 9 Sélectionnez Modification > Scénario > Propriétés du calque, puis redéfinissez le nom du calque sur Son.
- 10 Sélectionnez l'image Abaissé dans le calque Son et appuyez sur la touche de fonction F6 pour insérer une image-clé.



Nouvelle image-clé

- 11 Avec la nouvelle image-clé sélectionnée dans le scénario, dans l'inspecteur Propriétés choisissez proxy.wav dans le menu contextuel Son.

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



Inspecteur Propriétés

12 Pour lier le son de proxy au fichier audio du périphérique courant, procédez comme suit :

- a** Dans la bibliothèque, double-cliquez sur le symbole proxy.wav pour ouvrir la boîte de dialogue Propriétés audio.
- b** Dans cette boîte de dialogue, cliquez sur l'icône de dossier située à droite du champ texte Son de périphérique pour ouvrir la boîte de dialogue Sélectionner le son de périphérique.

- c Consultez la page Web à l'adresse http://www.adobe.com/go/learn_flt_samples_and_tutorials_fr. Elle regroupe des didacticiels et des exemples : repérez, téléchargez et décompressez le fichier .zip correspondant à votre version de Flash Lite, puis affichez le dossier Samples qu'il contient et sélectionnez le fichier nommé hi_hat.mid.



- d Cliquez sur OK.

- 13 Sélectionnez Contrôle > Tester l'animation pour lancer l'émulateur Adobe Device Central et tester votre fichier SWF. Dans l'émulateur, appuyez sur la flèche Bas du clavier pour donner le focus au bouton Lecture, puis appuyez sur la touche de sélection pour lire le son.

Déclenchement des sons de périphérique dans Flash Lite 1.0

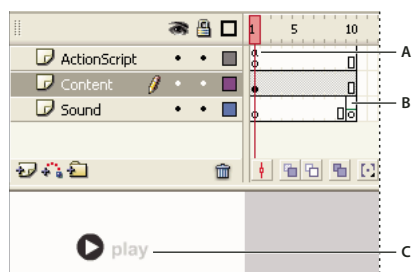
Dans Flash Lite 1.0, un son de périphérique ne peut être lu qu'en réponse à la pression d'une touche par l'utilisateur sur son périphérique. Il existe deux méthodes pour répondre à cette demande. La première consiste à associer le son à l'image Abaissé dans le scénario d'un symbole de bouton. Lorsque le bouton a le focus et que l'utilisateur appuie sur la touche de sélection de son périphérique, Flash Lite lit le son dans l'image Abaissé du bouton. Vous trouverez un exemple de cette technique à la section « [Ajout d'un son de périphérique à un bouton](#) » à la page 54.

La deuxième méthode pour déclencher un son de périphérique dans Flash Lite 1.0 consiste à ce que l'utilisateur appuie sur une touche qui positionne la tête de lecture sur une image dans le scénario contenant un son de périphérique. Tout son de périphérique au niveau de cette image est lu lorsque la tête de lecture entre dans l'image.

Par exemple, supposons que votre application comporte un bouton à l'image 1 de la scène, et un son de périphérique associé à l'image 10. Vous devez lier le gestionnaire `on (press)` suivant à l'occurrence de bouton :

```
on (press) {
    gotoAndStop (10);
}
```

L'illustration suivante montre à quoi votre application peut ressembler dans Flash :



A. Son associé à cette image B. Action stop() C. Appel de la fonction gotoAndStop(10) associé au bouton

Lorsque l'utilisateur sélectionne le bouton, le son de l'image 10 est lu. Cette technique implique que le son de périphérique soit associé à la même image que celle spécifiée dans la fonction `gotoAndPlay()`. Par exemple, dans le cas précédent, si le son avait été associé à l'image 11 au lieu de l'image 10, Flash Lite ne l'aurait pas lu lorsque la tête de lecture aurait atteint l'image 11.

Utilisation des sons composés

Flash Lite 1.1 permet d'encapsuler plusieurs formats audio spécifiques des périphériques dans un seul bloc de données balisées. Cela permet aux développeurs de créer un seul élément de contenu compatible avec plusieurs périphériques. Par exemple, une application Flash simple peut contenir le même son représenté aux formats MIDI et MFi. Vous pouvez lire cette application Flash sur un périphérique ne prenant en charge que le format MIDI ou sur un périphérique ne prenant en charge que le format MFi, chaque périphérique lisant le format audio qu'il prend en charge de façon native.

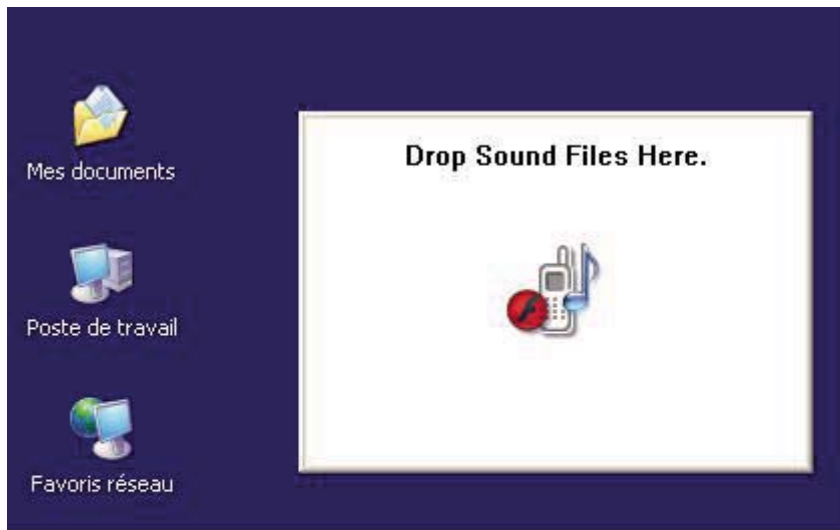
Vous devez utiliser un utilitaire appelé Système de regroupement audio de Flash Lite pour regrouper plusieurs sons de périphérique dans un seul fichier de regroupement audio (FLS). Vous pouvez ensuite ajouter ce fichier FLS à votre document Flash Lite de la même manière que vous ajoutez un son de périphérique standard : importez d'abord un fichier audio de proxy dans votre document Flash, puis spécifiez le fichier de regroupement audio devant remplacer le fichier audio de proxy au moment de la publication. Pour plus d'informations sur l'ajout de sons de périphérique à vos applications Flash Lite, consultez la section « [Ajout d'un son de périphérique à un bouton](#) » à la page 54.

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

Pour créer un fichier de regroupement audio :

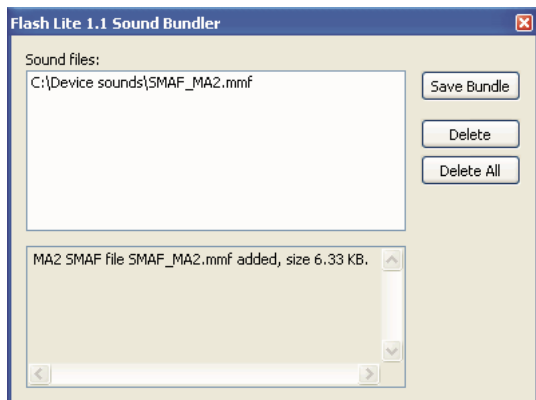
- 1 Repérez, téléchargez puis extrayez les fichiers du kit de développement de contenu (CDK) de Macromedia Flash Lite 1.1 à l'adresse www.adobe.com/go/developer_flashlite_fr. (Le CDK contient l'application Système de regroupement audio.)
- 2 Accédez au dossier dans laquelle vous avez extrait les fichiers du CDK et démarrez l'application Système de regroupement audio de Flash Lite (FlashLiteBundler.exe).

Le Système de regroupement audio s'affiche sous la forme d'une fenêtre flottante.



- 3 Faites glisser le premier fichier audio à regrouper depuis le bureau vers la fenêtre flottante.

La fenêtre Système de regroupement audio de Flash Lite 1.1 apparaît. La partie supérieure de la fenêtre Système de regroupement audio est une liste des fichiers ajoutés au regroupement audio. La partie inférieure de la fenêtre contient des informations relatives aux sons du regroupement audio, y compris le format audio, la taille des données audio et le nom du fichier.

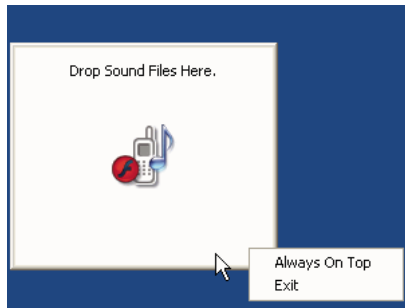


A. Liste des fichiers du regroupement audio B. Informations relatives aux fichiers du regroupement audio

- 4 Faites glisser les autres fichiers audio à regrouper dans la fenêtre.

Vous ne pouvez pas regrouper plusieurs fichiers dans un format audio donné. Par exemple, vous ne pouvez pas regrouper deux fichiers MIDI dans un même fichier FLS.

- 5 Pour supprimer un fichier du regroupement audio, sélectionnez-le dans la liste des fichiers audio et cliquez sur Supprimer. Pour supprimer tous les fichiers du groupe audio, cliquez sur Tout supprimer.
- 6 Cliquez sur Enregistrer le groupe pour enregistrer le fichier FLS.
- 7 Pour quitter le Système de regroupement audio, cliquez avec le bouton droit sur la fenêtre Système de regroupement audio et sélectionnez Quitter.



L'étape suivante consiste à ajouter le fichier de regroupement audio (FLS) à votre document Flash. La procédure est la même que l'ajout de sons de périphérique standard aux documents Flash, excepté qu'au lieu de spécifier un fichier de son de périphérique simple pour remplacer le fichier audio de proxy, vous devez spécifier le fichier FLS que vous avez créé. Pour plus d'informations, consultez la section « [Utilisation des sons de périphérique](#) » à la page 53.

Utilisation de sons Flash natifs

Outre les formats audio de périphérique pris en charge par Flash Lite 1.0, Flash Lite 1.1 prend en charge les sons Flash standard ou *natifs*. Dans Flash Lite 1.1, les sons natifs peuvent être lus en tant que sons d'événement ou synchronisés (consultez la section « [Sons d'événement et sons en flux continu \(synchronisés\)](#) » à la page 53).

Le processus général d'ajout de sons natifs à votre contenu Flash 1.1 Lite est identique à celui des applications pour ordinateurs de bureau Flash, avec les exceptions suivantes :

- Flash Lite ne prend pas en charge le chargement de fichiers MP3 externes.
- Flash Lite ne prend pas en charge l'objet Sound.
- Flash Lite ne prend pas en charge l'option de compression audio Speech (consultez la section « Compression des sons pour l'exportation » dans le manuel *Utilisation de Flash*).

Pour plus d'informations sur l'utilisation du son natif dans Flash, consultez les sections suivantes dans le manuel *Utilisation de Flash Lite* :

- « Importation de sons » dans le manuel *Utilisation de Flash*.
- « Ajout de sons dans un document » dans le manuel *Utilisation de Flash*.
- « Démarrage et arrêt de sons dans des images-clés » dans le manuel *Utilisation de Flash*.
- « Compression des sons pour l'exportation » dans le manuel *Utilisation de Flash*.

Chapitre 7 : Optimisation du contenu pour de meilleures performances et une taille de fichier réduite

Pour optimiser du contenu Flash, il est nécessaire de respecter quelques principes de base.

Optimisation des performances

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

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

Dans les versions antérieures de Flash, l'émulateur prévu pour Flash Lite 1.x était intégré à l'environnement de programmation de Flash. Dans les versions récentes de Flash, l'émulateur fait partie d'Adobe® Device Central. Device Central permet de simuler l'exécution de vos projets Flash Lite sur divers appareils en émulant l'affichage du périphérique, l'utilisation de la mémoire et les performances de certains périphériques particuliers. Pour toute information sur l'utilisation de l'émulateur afin d'optimiser votre contenu Flash Lite sur des périphériques portables, consultez la section « Meilleures pratiques relatives au contenu de périphériques mobiles » dans la documentation Device Central.

Chapitre 8 : Test du contenu Flash Lite

Adobe Flash CS4 Professional comprend un émulateur Adobe Flash Lite qui est intégré à Adobe Device Central CS4 : ce composant permet de tester directement dans l'outil de programmation l'affichage et l'exécution de votre application. Lorsque l'application s'exécute dans l'émulateur vous convient, vous pouvez la tester sur un périphérique réel. Ce chapitre décrit les fonctionnalités Flash Lite de test et de débogage qui sont disponibles dans Flash.

Présentation des fonctionnalités de test Flash Lite

Les fonctionnalités de test Flash Lite de Flash CS4 Professional font partie d'Adobe Device Central et comprennent une vaste base de données de profils de périphériques ainsi qu'un émulateur de périphérique. Device Central fonctionne également avec de nombreux autres produits Adobe, tels que Adobe Creative Suite® et Adobe Dreamweaver®.

L'émulateur Adobe Device Central permet de prévisualiser votre contenu Flash Lite dans l'outil de programmation de Flash. L'émulateur est configuré pour reproduire le comportement et l'aspect du périphérique cible. L'interface Device Central permet de sélectionner et de gérer les périphériques cibles et de spécifier le type de contenu Flash Lite cible de votre application, par exemple une sonnerie personnalisée, un navigateur ou une application autonome. Dans Device Central, chaque combinaison de périphérique test et de type de contenu Flash Lite définit une configuration de périphérique présentant les fonctionnalités disponibles pour votre application, notamment les formats audio pris en charge, la possibilité d'établir des connexions réseau ainsi que d'autres fonctionnalités. Pour plus d'informations sur les types de contenus Flash Lite, consultez la section « [Types de contenus Flash Lite](#) » à la page 65.

Pour toute information complémentaire, y compris des précisions sur les interactions avec l'émulateur, consultez l'aide de Device Central.

Fonctionnalités de test non prises en charge par l'émulateur

L'émulateur Adobe Device Central ne prend pas en charge toutes les fonctionnalités disponibles dans la fenêtre de test standard (ordinateur de bureau). Vous trouverez ci-dessous la liste des fonctionnalités qui ne sont pas disponibles dans l'émulateur Adobe Device Central, mais qui le sont dans la fenêtre de test de la version de bureau de Flash :

- Les fonctionnalités Lister les variables (Débogage > Lister les variables) et Lister les objets (Débogage > Lister les objets) ;
- Les fonctionnalités Testeur de bande passante, Graphique de lecture en continu et Graphique image par image ;
- La commande de menu Affichage > Simuler le téléchargement ;
- Le débogueur ActionScript ;
- La commande de menu Affichage > Afficher les zones redessinées ;
- La barre d'outils Contrôleur (Fenêtre > Barre d'outils > Contrôleur).

Utilisation de l'émulateur

Pour démarrer l'émulateur, sélectionnez Contrôle > Tester l'animation dans l'outil de programmation Flash, de la même façon que pour obtenir un aperçu de votre contenu Flash sur un ordinateur de bureau. Cependant, Adobe Device Central présente un aspect et des fonctionnalités qui diffèrent par rapport à la fenêtre de test du contenu de la version de bureau de Flash.

Appel d'Adobe Device Central

Sélectionnez Contrôle > Tester l'animation ou appuyez sur Ctrl+Entrée pour exécuter Device Central. Une barre de progression s'affiche pour indiquer que Flash exporte votre fichier SWF vers Device Central. Lorsque l'exportation est terminée, Device Central s'exécute, active l'émulateur et charge votre fichier SWF :

Dans Device Central, la liste Ensemb. de périph. regroupe tous les appareils enregistrés par Flash avec votre application. Par défaut, le premier périphérique du groupe est sélectionné pour l'émulation. Le type de contenu sélectionné est celui qui a été enregistré avec l'application au moment de sa création (pour en savoir plus sur les types de contenus Flash Lite, consultez la section « [Types de contenus Flash Lite](#) » à la page 65).

Vous pouvez tester l'application en fonction de différents types de contenus et de différents périphériques. Le changement de type de contenu et l'ajout ou la suppression de périphériques dans la liste modifie automatiquement les paramètres définis dans Flash.

Pour tester le fonctionnement du contenu avec un autre périphérique, double-cliquez sur le périphérique de votre choix dans le haut ou le bas de la liste. Une icône rotative apparaît en regard du nouveau périphérique à tester, et l'interface de l'émulateur change pour représenter l'exécution de votre application dans l'appareil sélectionné.

Définition des options de débogage de l'émulateur

Pendant l'exécution du contenu, l'émulateur Adobe Device Central peut transmettre des messages de débogage au panneau Sortie de Flash et afficher ces mêmes messages dans une représentation locale de ce panneau.

Les informations transmises sont les suivantes :

Messages de suivi générés par l'appel d'une fonction `trace()` dans votre application Flash Lite. Pour plus d'informations sur l'utilisation de la fonction `trace()`, consultez le *Guide de référence du langage ActionScript Flash Lite 1.x*.

Messages d'informations contenant des informations générales relatives au périphérique de test sélectionné, à la taille du fichier SWF ou autres. Ces messages apparaissent dans le panneau d'alerte de l'émulateur.

Messages d'avertissement contenant des informations relatives aux problèmes de problèmes votre contenu Flash Lite susceptibles d'affecter la lecture.

Pour filtrer le type d'informations transmises par l'émulateur, procédez comme indiqué ci-dessous.

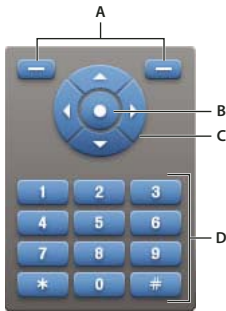
Pour définir les options de sortie de Flash Lite :

- 1 Sélectionnez Contrôle > Tester l'animation. Flash exporte votre application vers Adobe Device Central et l'affiche dans l'émulateur.
- 2 Sélectionnez Affichage > Sortie Flash > Afficher dans Device Central.
 - Sélectionnez ou désélectionnez l'option Suivi.
 - Sélectionnez ou désélectionnez l'option Informations.
 - Sélectionnez ou désélectionnez l'option Avertissements.

Interactions avec l'émulateur

Vous pouvez interagir avec le clavier de l'émulateur en utilisant la souris de votre ordinateur ou des raccourcis clavier. Vous pouvez interagir avec les touches suivantes du clavier de l'émulateur :

- Les touches numériques (0 à 9) et les touches astérisque (*) et dièse (#)
- Le pavé de navigation (gauche, droite, bas, haut et sélection)
- Touches programmables Gauche et Droite



A. Touches programmables B. Touche de sélection C. Autres touches prises en charge D. Pavé de navigation

Vous pouvez cliquer directement sur le clavier de l'émulateur avec la souris ou bien utiliser les raccourcis clavier équivalents suivants :

- Les touches fléchées du clavier (gauche, droite, haut, bas) correspondent aux touches du pavé de navigation de l'émulateur.
- Les touches Entrée et Retour correspondent à la touche de sélection de l'émulateur.
- Les touches Page précédente et Page suivante correspondent respectivement aux touches programmables Gauche et Droite de l'émulateur.
- Les touches numériques de votre clavier correspondent aux touches numériques équivalentes du clavier de l'émulateur.

Pour toute information complémentaire sur les interactions possibles avec l'émulateur pour tester votre application, consultez l'aide en ligne d'Adobe Device Central.

Types de contenus Flash Lite

Comme indiqué dans la section « [A propos des types de contenus Flash Lite](#) » à la page 4, Flash Lite est installé sur de nombreux périphériques. Toutes les installations Flash Lite prennent en charge un ou plusieurs modes d'application ou *types de contenus*. Par exemple, certains périphériques ont recours à Flash Lite pour activer les fonctionnalités d'économiseurs d'écran et de sonneries animées Flash. D'autres périphériques exploitent Flash Lite pour afficher du contenu Flash intégré dans des pages Web mobiles.

Le tableau suivant répertorie et décrit tous les types de contenus Flash Lite disponibles à l'heure actuelle. Comme indiqué dans la section « [Disponibilité de Flash Lite 1.x](#) » à la page 3, la plupart des types de contenus sont disponibles uniquement dans une région géographique ou auprès d'opérateurs de téléphonie mobile spécifiques. La troisième colonne du tableau présente la disponibilité de chaque type de contenu par région et par opérateur. Pour obtenir des informations supplémentaires et actualisées sur la disponibilité des types de contenus Flash Lite, consultez la page Flash Enabled Mobile Device (Périphériques mobiles équipés de Flash), à l'adresse www.adobe.com/go/mobile_supported_devices_fr.

Flash Lite prend en charge les types de contenus suivants :

Type de contenu Flash Lite	Description	Disponibilité
Carnet d'adresses	Utilise Flash Lite pour permettre aux utilisateurs d'associer un fichier SWF à une entrée dans l'application du carnet d'adresses de leur périphérique.	NTT DoCoMo et Vodafone (Japon uniquement)
Alarme	Utilise Flash Lite pour permettre à l'utilisateur de sélectionner un fichier SWF devant être lu pour l'alarme de son périphérique.	KDDI et Vodafone (Japon uniquement)
Navigateur	Utilise Flash Lite pour rendre le contenu Flash intégré aux pages Web mobiles et affiché dans le navigateur Web du périphérique.	NTT DoCoMo, KDDI et Vodafone (Japon uniquement)
Historique des appels	Utilise Flash Lite pour afficher une image ou une animation associée à chaque entrée du carnet d'adresses de l'utilisateur, ainsi que le nom et le numéro de téléphone correspondants.	KDDI (téléphones Casio uniquement)
Ecran d'appel	Utilise Flash Lite pour afficher une animation lorsque l'utilisateur reçoit un appel ou en passe un.	NTT DoCoMo et KDDI (Japon uniquement)
Chaku Flash	Utilise Flash Lite pour permettre à l'utilisateur de sélectionner un fichier SWF devant être lu comme sonnerie pour les appels entrants.	KDDI (Japon uniquement)
Boîte de données	Utilise Flash Lite pour rendre du contenu Flash dans l'application Boîte de données du périphérique, ce qui permet à l'utilisateur de gérer et de prévisualiser des fichiers multimédia sur le périphérique.	NTT DoCoMo, KDDI et Vodafone (Japon uniquement)
Dossier de données	Utilise Flash Lite pour rendre du contenu Flash dans l'application Dossier de données du périphérique, ce qui permet à l'utilisateur de gérer et de prévisualiser des fichiers multimédia sur le périphérique.	KDDI (Japon uniquement)
Menu Icône	Utilise Flash Lite pour permettre à l'utilisateur de sélectionner des menus d'icônes personnalisés pour l'application de lancement du périphérique.	KDDI (téléphones Casio uniquement)
Visionneuse	Utilise l'application Visionneuse pour permettre à l'utilisateur de gérer et de prévisualiser des fichiers multimédias sur son périphérique, y compris des fichiers SWF.	NTT DoCoMo (Japon uniquement)
Appel entrant	Utilise Flash Lite pour afficher une animation lorsque l'utilisateur reçoit un appel.	NTT DoCoMo, KDDI et Vodafone (Japon uniquement)
Messagerie	Utilise Flash Lite pour afficher une animation lorsque l'utilisateur envoie ou reçoit un message électronique.	Vodafone (Japon uniquement)
Multimédia	Utilise Flash Lite pour prévisualiser des fichiers SWF (ainsi que d'autres formats multimédias).	KDDI (Japon uniquement)

Type de contenu Flash Lite	Description	Disponibilité
Mes images	Utilise l'application Mes images pour permettre à l'utilisateur de gérer et de prévisualiser des fichiers SWF sur son périphérique, ainsi que d'autres formats d'image.	NTT DoCoMo (Japon uniquement)
OpenEMIRO	Affiche du contenu Flash Lite lorsque le périphérique sort du mode d'attente. Ce type est identique au type de contenu Ecran de sortie de veille figurant sur d'autres périphériques.	KDDI (périphériques Casio uniquement)
Ecran de veille	Utilise Flash Lite pour afficher l'écran de veille du périphérique.	KDDI et Vodafone (Japon uniquement)
Lecteur SMIL	Utilise Flash Lite pour prévisualiser des fichiers SWF (ainsi que d'autres formats multimédias).	KDDI (Japon uniquement)
Lecteur autonome	Met Flash Lite à la disposition de l'utilisateur en tant qu'application autonome pour lui permettre de lancer et d'afficher des fichiers SWF arbitraires situés sur son périphérique ou arrivés dans la boîte de réception de sa messagerie.	Disponible dans le monde entier pour les périphériques Symbian Series 60 et UIQ.
Ecran d'attente	Utilise Flash Lite pour afficher l'écran d'attente du périphérique (ou papier peint).	NTT DoCoMo et KDDI (Japon uniquement)
Ecran externe	Utilise Flash Lite pour afficher du contenu sur l'écran externe ou secondaire disponible sur certains téléphones à rabat.	KDDI (Japon uniquement)
Ecran de sortie de veille	Utilise Flash Lite pour afficher une animation au moment du démarrage du téléphone.	NTT DoCoMo (Japon uniquement)

Informations spécifiques à Flash Lite dans l'émulateur

Dans l'émulateur, des panneaux affichent des informations spécifiques à l'application Flash Lite. Ces panneaux sont alignés sur le côté droit de la fenêtre et vous pouvez les développer ou les réduire tout comme dans Flash.

Pour toute information complémentaire sur ces panneaux et leur utilisation, consultez l'aide en ligne d'Adobe Device Central.

Taille d'écran et taille de scène disponible

Toutes les combinaisons de périphérique cible et de type de contenu Flash Lite déterminent, entre autres, la zone d'écran disponible qu'une application Flash Lite peut occuper. La zone de la scène disponible peut être égale ou inférieure à la taille de l'écran du périphérique.

Par exemple, la zone de scène disponible pour un fichier SWF affiché en mode plein écran dans le lecteur autonome sur un périphérique de la gamme Series 60 de Nokia est égale à la taille de l'écran (176 x 208 pixels). Sur d'autres périphériques (comme ceux vendus au Japon), la zone de scène disponible pour un fichier SWF exécuté dans l'un des types de contenus spécialisés (comme Carnet d'adresses ou Economiseur d'écran) peut être inférieure à la taille totale de l'écran. Par exemple, la taille d'écran du Fujitsu 700i est de 240 x 320 pixels ; cependant, un fichier SWF exécuté dans l'application Carnet d'adresses du périphérique dispose d'une zone de scène de 96 x 72 pixels.

Si la taille de scène d'un fichier SWF est différente de la taille de la scène disponible, le lecteur Flash Lite redimensionne le contenu (proportionnellement) pour l'ajuster à l'espace disponible. Lorsque vous testez votre contenu dans l'émulateur, ce dernier vous signale également si la taille de scène de votre application est différente de celle qui est disponible.

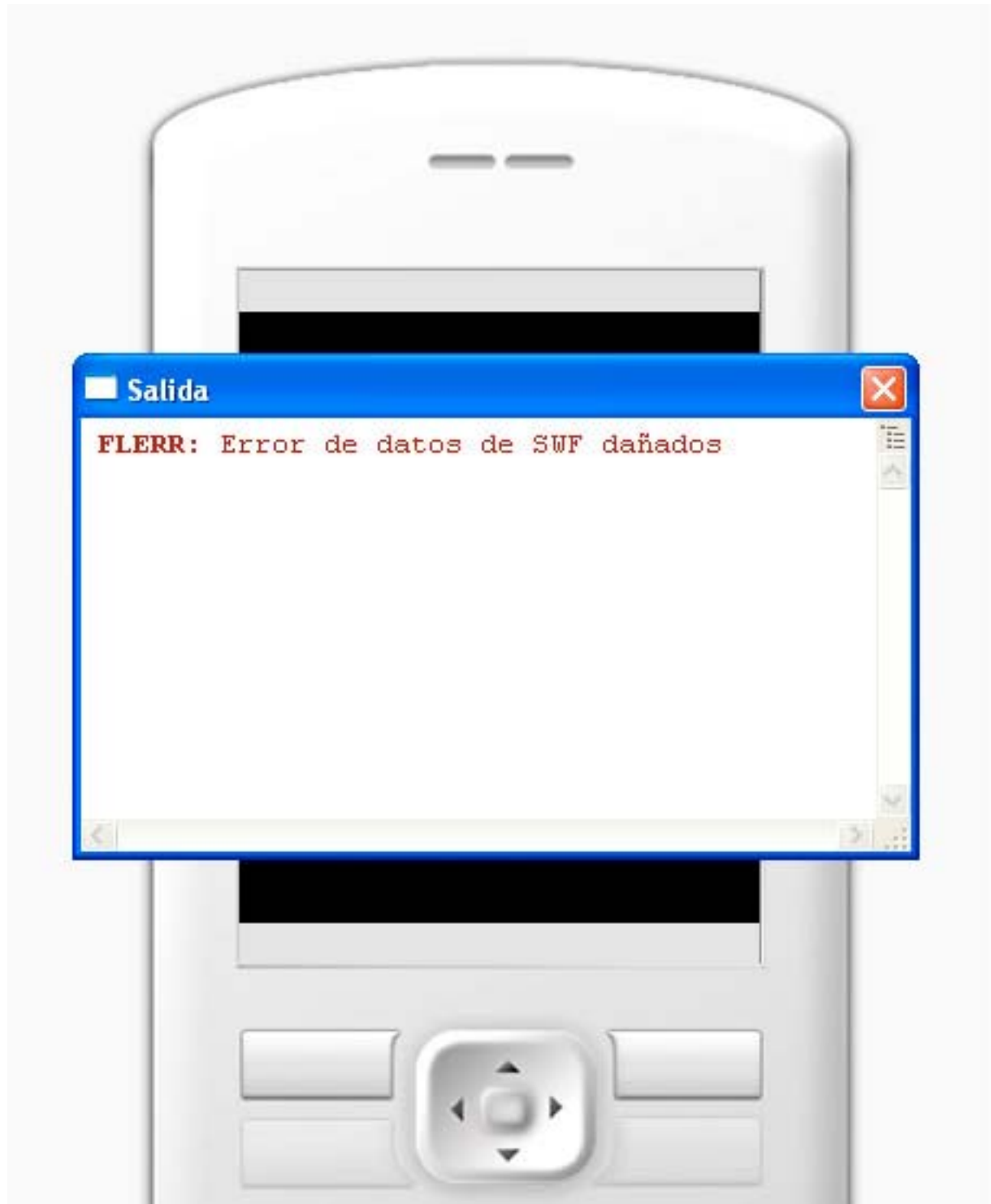
Pour éviter tout problème de redimensionnement indésirable, Adobe recommande d'adapter les dimensions de la scène de votre document Flash à celles de la zone de scène disponible pour le type de contenu et le périphérique de test sélectionnés.

Erreurs

Pendant le test du contenu, l'émulateur d'Adobe Device Central peut générer des alertes. Le premier type d'alerte s'affiche uniquement dans l'émulateur et fournit des informations sur les erreurs réelles ou potentielles ; le deuxième s'affiche à la fois dans l'émulateur et sur le périphérique réel.

Le premier type de message d'alerte fournit des informations de débogage concernant votre fichier SWF. Par exemple, si le fichier SWF contient du code ActionScript non pris en charge par Flash Lite (ou par la version de Flash Lite disponible sur le périphérique de test sélectionné), l'émulateur déclenche une alerte.

L'autre type de message peut survenir aussi bien dans l'émulateur que sur le périphérique réel. Ces types d'erreurs s'affichent dans une boîte de dialogue d'erreur que l'utilisateur doit fermer pour permettre à l'application de poursuivre. L'illustration suivante présente un exemple de boîte de dialogue d'erreur, telle qu'elle est affichée dans l'émulateur :



Sur le périphérique, la boîte de dialogue d'erreur qui s'affiche contient la chaîne « Problème de contenu » suivie d'un numéro d'erreur. Dans l'émulateur, la boîte de dialogue d'erreur contient également une courte chaîne d'erreur. Enfin, l'émulateur affiche une description plus détaillée de l'erreur dans le panneau Sortie.

Le tableau suivant répertorie toutes les erreurs survenant dans le lecteur Flash Lite, avec notamment les numéros d'erreur, les descriptions brèves qui s'affichent dans la boîte de dialogue d'erreur et les descriptions détaillées qui s'affichent dans le panneau Sortie :

Numéro de l'erreur	Chaîne d'erreurs	Description et causes possibles
1	Mémoire insuffisante.	L'émulateur ne dispose pas de suffisamment de mémoire heap. Sauf mention contraire, l'émulateur alloue au fichier SWF 1 Mo de mémoire à utiliser.
2	Limite d'empilement atteinte	L'émulateur a détecté que sa limite d'empilement a été atteinte ou dépassée. Les causes peuvent être diverses : plusieurs niveaux de clips imbriqués ou dessins vectoriels complexes.
3	Fichier SWF corrompu	L'émulateur a détecté que les données du fichier SWF sont corrompues.
4	Blocage du code ActionScript.	L'émulateur a détecté que l'exécution d'une portion de code ActionScript figurant dans le fichier SWF prenait trop de temps. Par conséquent, l'émulateur a arrêté l'exécution de ce code ActionScript.
5	Traitement du code ActionScript.	L'émulateur a détecté une erreur ActionScript, telle qu'une référence à un clip inexistant.
6	Boucle infinie ActionScript.	L'émulateur a détecté une boucle ActionScript infinie ou profondément imbriquée (par exemple, des instructions <code>if . . else</code> profondément imbriquées).
7	Mémoire vidéo non valide	L'émulateur a détecté la présence d'une mémoire vidéo non valide.
8	Rectangle d'affichage non valide	L'émulateur a détecté la présence d'un rectangle d'affichage non valide.
9	Numéro d'image incorrect	L'émulateur a détecté que le fichier SWF a tenté d'accéder ou de traduire un numéro d'image incorrect.
10	Touche non valide	L'émulateur a détecté la saisie d'une touche non valide.
11	Données JPEG incorrectes	L'émulateur a détecté que des données JPEG ou PNG contenues dans le fichier SWF sont corrompues, qu'il n'y a pas suffisamment de mémoire pour décoder les données JPEG ou que le format des données JPEG n'est pas pris en charge.
12	Données audio incorrectes	L'émulateur a détecté la présence dans le fichier SWF d'un format de données audio non pris en charge.
13	Clip racine purgé	L'émulateur a détecté que le clip racine a été purgé et n'a pas été remplacé par un autre fichier SWF.

Détermination des fonctionnalités de la plate-forme

Chaque combinaison de périphérique cible et de type de contenu Flash Lite définit un ensemble de fonctionnalités Flash Lite, telles que le type de navigation, les formats audio de périphérique pris en charge et le texte de saisie. Si l'option de débogage Informations est activée dans le panneau des paramètres de l'émulateur, ce dernier génère une liste des fonctionnalités de la plate-forme pour le périphérique et le type de contenu actuellement sélectionnés. Pour plus d'informations sur les options de débogage des paramètres, consultez la section « [Définition des options de débogage de l'émulateur](#) » à la page 64.

Le tableau suivant présente les fonctionnalités de la plate-forme Flash Lite telles qu'elles sont rapportées dans le panneau Sortie lorsque vous testez votre application dans l'émulateur :

Nom de la fonctionnalité	Description et valeurs possibles
DeviceSoundKeyOnly	Indique si le périphérique lit des sons de périphérique uniquement en réponse à la pression d'une touche par l'utilisateur sur le périphérique (Yes), ou indépendamment de toute action de l'utilisateur (No). Pour plus d'informations, consultez la section « Déclenchement des sons de périphérique dans Flash Lite 1.0 » à la page 58.
DeviceSoundsOrdered	Liste séparée par des virgules des formats audio de périphérique pris en charge par la plate-forme. L'ordre des formats audio indique le premier son lu par Flash Lite si un fichier SWF contient un fichier de regroupement audio contenant plusieurs formats. Pour plus d'informations sur les regroupements audio, consultez la section « Utilisation des sons composés » à la page 59.
FSCCommand	Indique la fréquence à laquelle Flash Lite traite les appels de la fonction <code>fscCommand()</code> ou <code>fscCommand2()</code> . Les valeurs valides sont les suivantes : OnePerKey : un seul appel de la fonction <code>fscCommand()</code> est autorisé pour chaque pression de touche. OnePerKeyPerFrame : un seul appel de la fonction <code>fscCommand()</code> est autorisé par gestionnaire d'événements ou par image. All : aucune restriction n'est appliquée quant à la fréquence des appels de la fonction <code>fscCommand()</code> . None : la fonction <code>fscCommand()</code> n'est pas prise en charge.
InputText	Indique si la plate-forme prend en charge la fonctionnalité de texte de saisie (Yes) ou non (No). Pour plus d'informations sur le texte de saisie, consultez la section « Utilisation des champs texte de saisie » à la page 41).
LoadMovie	Indique la fréquence à laquelle Flash Lite traite les appels de la fonction <code>loadMovie()</code> . Les valeurs valides sont les suivantes : OnePerKey : un seul appel de la fonction <code>loadMovie()</code> est autorisé pour chaque pression de touche. OnePerKeyPerFrame : un seul appel de la fonction <code>loadMovie()</code> est autorisé par gestionnaire d'événements ou par image. All : aucune restriction n'est appliquée quant à la fréquence des appels de la fonction <code>loadMovie()</code> . None : la fonction <code>loadMovie()</code> n'est pas prise en charge.

Nom de la fonctionnalité	Description et valeurs possibles
LoadVars	<p>Indique la fréquence à laquelle Flash Lite traite les appels de la fonction <code>loadVariables()</code>. Les valeurs valides sont les suivantes :</p> <p>OnePerKey : un seul appel de la fonction <code>loadVariables()</code> est autorisé pour chaque pression de touche.</p> <p>OnePerKeyPerFrame : un seul appel de la fonction <code>loadVariables()</code> est autorisé par gestionnaire d'événements ou par image.</p> <p>All : aucune restriction n'est appliquée quant à la fréquence des appels de la fonction <code>loadVariables()</code>.</p> <p>None : la fonction <code>loadVariables()</code> n'est pas prise en charge.</p>
Loop	<p>Indique si le contenu SWF s'exécute en boucle (retourne à la première image dans le scénario) une fois la fin du scénario atteinte (Yes) ou s'il s'arrête sur la dernière image (No).</p>
MultipleDeviceSound	<p>Indique si le périphérique prend en charge le mixage de plusieurs sons de périphérique (Yes) ou non (No).</p>
NativeSounds	<p>Liste non organisée des formats audio que le lecteur Flash Lite peut lire de façon native (par opposition aux sons de périphérique qui sont transmis depuis le lecteur Flash Lite au périphérique pour être lus). Valeurs possibles : NativeSound_PCM, NativeSound_ADPCM et NativeSound_MP3.</p>
NavigationType	<p>Indique le mode de navigation pris en charge par la plate-forme : navigation bidirectionnelle, navigation quadridirectionnelle et navigation quadridirectionnelle en boucle. Pour plus d'informations sur les modes de navigation, consultez la section « Modes de navigation par tabulation » à la page 21. Les valeurs valides sont les suivantes :</p> <p>2Way : seules les flèches Haut et Bas sont prises en charge.</p> <p>4Way : les quatre flèches de direction (Haut, Bas, Gauche et Droite) sont prises en charge pour la navigation.</p> <p>4WayWrapAround : identique à la navigation quadridirectionnelle, sauf que le focus retourne automatiquement en haut de l'affichage.</p>
SMS	<p>Indique si Flash Lite prend en charge l'envoi de SMS (Yes) ou non (No).</p>
getUrl	<p>Indique la fréquence à laquelle Flash Lite traite les appels de la fonction <code>getUrl()</code>. Les valeurs valides sont les suivantes :</p> <p>OnePerKey : un seul appel de la fonction <code>getUrl()</code> est autorisé pour chaque pression de touche.</p> <p>OnePerKeyPerFrame : un seul appel de la fonction <code>getUrl()</code> est autorisé par gestionnaire d'événements ou par image.</p> <p>All : aucune restriction n'est appliquée quant à la fréquence des appels de la fonction <code>getUrl()</code>.</p> <p>None : la fonction <code>getUrl()</code> n'est pas prise en charge.</p>

Nom de la fonctionnalité	Description et valeurs possibles
keySet	<p>Indique les événements de touche pris en charge par Flash Lite sur le périphérique. Pour plus d'informations sur la gestion des événements de touche, consultez la section « Gestion des événements de touche » à la page 24. Les valeurs valides sont les suivantes :</p> <p>All : tous les événements de touche sont traités.</p> <p>Phone : seuls les événements associés aux touches 0 à 9, #, *, à la touche de sélection, ainsi qu'aux touches de navigation quadridirectionnelle sont traités.</p>
mouseType	<p>Indique les événements de souris pris en charge par Flash Lite. Les valeurs valides sont les suivantes :</p> <p>None : aucun événement de souris n'est pris en charge.</p> <p>Partial : les événements <code>press</code>, <code>release</code>, <code>rollOver</code>, et <code>rollOut</code> sont pris en charge ; les événements <code>releaseOutside</code>, <code>dragOut</code>, et <code>dragOver</code> ne sont pas pris en charge.</p> <p>Mouse : les messages <code>Mouse Up</code>/<code>Mouse Down</code>/<code>Mouse Move</code> sont traités. Dans le cas d'un téléphone NTT DoCoMo doté de la fonctionnalité de curseur virtuel, le message <code>Mouse Move</code> (souris déplacée) doit déclencher l'événement <code>rollOver</code>/<code>rollOut</code>.</p>
soundEnabled	<p>Indique si le son est activé sur le périphérique (Yes) ou non (No).</p>

Chapitre 9 : Formation à ActionScript Flash Lite 1.x

A propos d'ActionScript Flash Lite 1.x

Vous pouvez utiliser ActionScript pour ajouter une logique de programmation et de l'interactivité à vos applications Adobe Flash Lite. La version d'ActionScript des logiciels Macromedia® Flash Lite 1.0 et Macromedia Flash Lite 1.1 d'Adobe, dont l'ensemble porte le nom d'ActionScript Flash Lite 1.x, est un code hybride de Macromedia Flash 4 ActionScript d'Adobe, doté de commandes et de propriétés supplémentaires spécifiques du lecteur Flash Lite, telles que la possibilité de lancer des appels téléphoniques ou d'envoyer des messages textuels, ou encore de récupérer les informations relatives à l'heure et à la date à partir du périphérique.

Présentation d'ActionScript Flash Lite 1.x

ActionScript Flash Lite 1.x est composé des éléments suivants :

ActionScript Flash Player 4 Cet élément inclut les opérateurs (par exemple, opérateurs de comparaison et d'affectation), les propriétés de clip (par exemple, `_height`, `_x` et `_y`), les fonctions de contrôle du scénario (par exemple, `gotoAndPlay()` et `stop()`) et les fonctions réseau, telles que `loadVariables()` et `loadMovie()` (Flash Lite 1.1 uniquement). Pour obtenir une liste du code ActionScript Flash 4 non pris en charge, consultez la section « [Code ActionScript de Flash 4 non pris en charge par le code ActionScript de Flash Lite 1.x](#) » à la page 75.

Propriétés et commandes d'intégration téléphonique Flash Lite comprend des commandes permettant, par exemple, de connaître l'heure et la date du périphérique, de lancer un appel téléphonique ou d'envoyer un SMS, ou encore de démarrer des applications externes installées sur le périphérique.

Variables des fonctionnalités de la plate-forme (Flash Lite 1.1 uniquement) Ces propriétés fournissent des informations relatives aux fonctionnalités du périphérique ou de l'environnement d'exécution Flash Lite. Par exemple, la variable `_capLoadData` indique si votre application peut charger des données via le réseau.

Fonction `fscommand2()` Comme avec la fonction `fscommand()`, vous pouvez utiliser la fonction `fscommand2()` pour communiquer avec l'environnement ou le système hôte (dans le cas présent, le téléphone ou le périphérique portable). La fonction `fscommand2()` a été améliorée par rapport à la fonction `fscommand()` ; elle est désormais en mesure de transmettre un nombre arbitraire d'arguments et de récupérer immédiatement les valeurs renvoyées (plutôt que d'avoir à attendre jusqu'à l'image suivante, comme avec `fscommand()`).

Différences entre ActionScript Flash Lite 1.0 et Flash Lite 1.1

Les fonctionnalités ActionScript suivantes de Flash Lite 1.1 ne sont pas disponibles dans Flash Lite 1.0 :

- Accès réseau ou informations relatives à l'état du réseau. Par exemple, dans Flash Lite 1.0 vous ne pouvez pas utiliser les fonctions `loadVariables()` ou `loadMovie()` pour charger des données ou fichiers SWF externes, ni les commandes `fscommand2()` pour déterminer l'intensité du signal de connexion d'un périphérique ou l'état d'une demande de connexion au réseau.
- Récupération des informations relatives à l'heure et à la date depuis le périphérique.
- Variables des fonctionnalités de la plate-forme, qui fournissent des informations sur les fonctionnalités de la plate-forme Flash Lite et du périphérique.
- La fonction `fscommand2()` et ses commandes associées, telles que `SetSoftKeys` et `FullScreen`.

- Les propriétés de champ texte `scroll` et `maxscroll`.

Code ActionScript de Flash 4 non pris en charge par le code ActionScript de Flash Lite 1.x

Les fonctionnalités ActionScript de Flash 4 suivantes ne sont pas prises en charge, ou seulement partiellement, dans le code ActionScript de Flash Lite 1.x :

- Les fonctions `startDrag()` et `stopDrag()`.
- Le code ActionScript de Flash Lite 1.x ne prend en charge qu'un sous-ensemble des événements de bouton pris en charge dans Flash Player 4. Pour plus d'informations sur la gestion des événements de bouton, consultez la section « [Création d'interactivité et d'éléments de navigation](#) » à la page 20.
- Le code ActionScript de Flash Lite 1.x ne prend en charge qu'un sous-ensemble des événements de pression de touche pris en charge dans Flash Player 4. Pour plus d'informations sur les événements de pression de touche pris en charge dans Flash Lite, consultez la section « [Création d'interactivité et d'éléments de navigation](#) » à la page 20.
- La propriété `_dropTarget`.
- La propriété `_soundBufTime`.
- La propriété `_url`.
- La fonction de conversion `String()`.

Fonctionnalités non disponibles dans le code ActionScript Flash Lite 1.x

Le lecteur Flash Lite étant basé sur une ancienne version de Flash Player, il ne prend pas en charge toutes les fonctionnalités de programmation disponibles dans les versions plus récentes de Flash Player ou d'autres langages de programmation de votre connaissance. Cette section présente les fonctionnalités de programmation non disponibles dans le code ActionScript de Flash Lite 1.x ainsi que les solutions de remplacement disponibles.

Fonctions définies par l'utilisateur Flash Lite 1.x ne prend pas en charge la fonctionnalité de définition et d'appel de fonctions personnalisées. Cependant, vous pouvez utiliser la fonction `call()` pour exécuter du code se trouvant sur une image quelconque dans le scénario. Pour plus d'informations, consultez la section « [Utilisation de la fonction call\(\) pour créer des fonctions](#) » à la page 78.

Tableaux, objets ou autres types de données complexes natifs Flash Lite 1.x ne prend pas en charge les structures de données de tableau natives ni aucun autre type de données complexes. Toutefois, vous pouvez émuler des tableaux à l'aide de pseudo-tableaux ; cette technique implique l'utilisation de la fonction `eval()` permettant d'évaluer de façon dynamique les chaînes concaténées. Pour plus d'informations, consultez la section « [Emulation des tableaux](#) » à la page 77.

Chargement à l'exécution de fichiers image et audio externes Contrairement à la version pour ordinateurs de bureau de Flash Player, ActionScript Flash Lite 1.x ne peut pas charger de fichiers JPEG ou MP3 externes. Dans Flash Lite 1.1, la fonction `loadMovie()` vous permet de charger des fichiers SWF externes. Pour plus d'informations, consultez la section « [Chargement de fichiers SWF externes](#) » à la page 85.

Guide d'introduction au code ActionScript Flash 4

Le code ActionScript de Macromedia Flash Lite 1.x d'Adobe est basé sur la version d'ActionScript, disponible pour la première fois dans Macromedia Flash Player 4 d'Adobe. En conséquence, plusieurs fonctionnalités de programmation proposées dans des versions ultérieures de Flash Player (version pour ordinateurs de bureau) ne sont pas disponibles dans les applications Flash Lite 1.x.

Si vous ne maîtrisez pas les fonctionnalités et la syntaxe du code ActionScript Flash 4 ou si vous avez oublié certaines informations depuis votre dernier projet de développement Flash, cette section propose un guide d'introduction à l'utilisation du code ActionScript Flash 4 dans vos applications Flash Lite.

Récupération et définition des propriétés de clip

Pour récupérer ou définir une propriété de clip (si définissable), vous pouvez utiliser une syntaxe à point ou bien les fonctions `setProperty()` ou `getProperty()`. Vous pouvez également utiliser la fonction `tellTarget()`.

Pour utiliser une syntaxe à point, indiquez le nom d'occurrence du clip suivi d'un point (`.`), puis du nom de la propriété. Par exemple, le code suivant renvoie la coordonnée d'écran `x` (représentée par la propriété de clip `_x`) du clip intitulé `cartoonArea` et attribue le résultat à une variable nommée `x_pos`.

```
x_pos = cartoonArea._x;
```

L'exemple suivant est identique à l'exemple précédent, mais il utilise la fonction `getProperty()` pour récupérer la position `x` du clip :

```
x_pos = getProperty(cartoonArea, _x);
```

La fonction `setProperty()` permet de définir une propriété d'occurrence de clip, comme indiqué dans l'exemple suivant :

```
setProperty(cartoonArea, _x, 100);
```

L'exemple suivant est identique au précédent, mais il utilise une syntaxe de point :

```
cartoonArea._x = 100;
```

Vous pouvez également récupérer ou définir des propriétés de clip à partir d'une instruction `tellTarget()`. Le code suivant est identique à l'exemple `setProperty()` précédent :

```
tellTarget("/cartoonArea") {  
    _x = 100;  
}
```

Pour plus d'informations sur la fonction `tellTarget()`, consultez la section « [Contrôle d'autres scénarios](#) » à la page 76.

Contrôle d'autres scénarios

Pour spécifier le chemin d'un scénario, utilisez une syntaxe à barre oblique (`/`) associée à deux points (`..`) pour créer la référence du chemin. Vous pouvez également utiliser les notations `_levelN`, `_root` ou `_parent` de Flash 5 pour faire référence, respectivement, à un niveau d'animation spécifique, au scénario racine de l'application ou au scénario parent.

Supposons par exemple que le scénario principal de votre fichier SWF contienne une occurrence de clip appelée `box`. Cette occurrence `box` contient elle-même une autre occurrence de clip appelée `cards`. Les exemples suivants ciblent le clip `cards` depuis le scénario principal :


```
tellTarget ("/box/cards")  
tellTarget ("_level0/box/cards")
```

L'exemple suivant cible le scénario principal depuis le clip `cards` :

```
tellTarget ("../cards")  
tellTarget ("_root")
```

L'exemple suivant cible le clip parent `cards` :

```
tellTarget ("../cards")  
tellTarget ("_parent/cards")
```

Utilisation des variables

Pour spécifier une variable dans un scénario, utilisez une syntaxe à barre oblique (/) associée à des points (..) et suivie de deux-points (:). Vous pouvez également utiliser la notation par point.

Le code suivant fait référence à la variable `car` dans le scénario principal :

```
/:car  
_root.car
```

L'exemple suivant illustre la variable `car` dans une occurrence de clip se trouvant dans le scénario principal :

```
/mc1/mc2/:car  
_root.mc1.mc2.car
```

L'exemple suivant illustre la variable `car` dans une occurrence de clip se trouvant dans le scénario actuel :

```
mc2/:car  
mc2.car
```

Emulation des tableaux

Les tableaux permettent de créer et de manipuler des listes d'informations triées telles que des variables et des valeurs. Toutefois, Flash Lite 1.1 ne prend pas en charge les structures de données de tableau natives. Une technique courante de programmation dans Flash Lite (et Flash 4) consiste à émuler des tableaux à l'aide du traitement de chaîne. Un tableau émulé est également appelé un pseudo-tableau. Pour traiter les pseudo-tableaux, la fonction ActionScript `eval()` permet d'accéder aux variables, propriétés ou clips en fonction de leur nom. Pour plus d'informations, consultez la section « [Utilisation de la fonction eval\(\)](#) » à la page 83.

Un pseudo-tableau comprend généralement au minimum deux variables qui partagent le même nom de base, suivi d'un suffixe numérique. Le suffixe représente l'indice de chaque élément du tableau.

Par exemple, vous pouvez créer les variables ActionScript suivantes :

```
color_1 = "orange";  
color_2 = "green";  
color_3 = "blue";  
color_4 = "red";
```

Vous pouvez alors utiliser le code suivant pour passer en boucle sur les éléments du pseudo-tableau :

```
for (i = 1; i <=4; i++) {  
    trace (eval ("color_" + i));  
}
```

En plus de référencer les variables existantes, vous pouvez également utiliser la fonction `eval()` située à gauche d'une affectation de variable pour créer des variables lors de l'exécution. Supposons par exemple que vous souhaitiez conserver une liste des meilleurs scores d'une partie jouée par un utilisateur. Chaque fois que l'utilisateur termine une partie, vous ajoutez son score à la liste :

```
eval("highScore" + add scoreIndex) = currentScore;  
scoreIndex++;
```

Chaque fois que ce code est exécuté, il ajoute un nouvel élément à la liste des meilleurs scores et incrémente la variable `scoreIndex` qui détermine l'indice de tous les éléments de la liste. Vous pouvez par exemple vous retrouver avec les variables suivantes :

```
highScore1 = 2000  
highScore2 = 1500  
highScore3 = 3000
```

Utilisation de texte et de chaînes

Flash Lite contient des propriétés et commandes ActionScript de base permettant d'utiliser du texte. Vous pouvez récupérer et définir les valeurs des champs texte, concaténer des chaînes, coder ou décoder des chaînes de texte pour l'URL et créer des champs texte défilants.

Concaténation de chaînes

Pour concaténer des chaînes dans Flash Lite, vous pouvez utiliser l'opérateur `add` indiqué dans l'exemple suivant :

```
city = "Boston";  
team = "Red Sox";  
fullName = city add " " add team;  
// Result:  
// fullName = "Boston Red Sox"
```

Texte défilant

Vous pouvez utiliser la propriété `scroll` des champs texte dynamique et de saisie pour récupérer ou définir la position de défilement courante du champ. Vous pouvez également utiliser la position `maxscroll` pour déterminer la position de défilement courante d'un champ texte par rapport à la position de défilement maximale. Pour savoir comment créer un champ texte défilant, consultez la section « [Création de texte défilant](#) » à la page 51.

Utilisation de la fonction `call()` pour créer des fonctions

Vous ne pouvez pas définir ou appeler des fonctions personnalisées dans Flash Lite comme c'est le cas dans Flash Player 5 et les versions ultérieures. Cependant, vous pouvez utiliser la fonction ActionScript `call()` pour exécuter du code se trouvant sur une image quelconque dans le scénario. Cette technique permet d'encapsuler le code le plus utilisé dans un emplacement unique pour en faciliter la gestion.

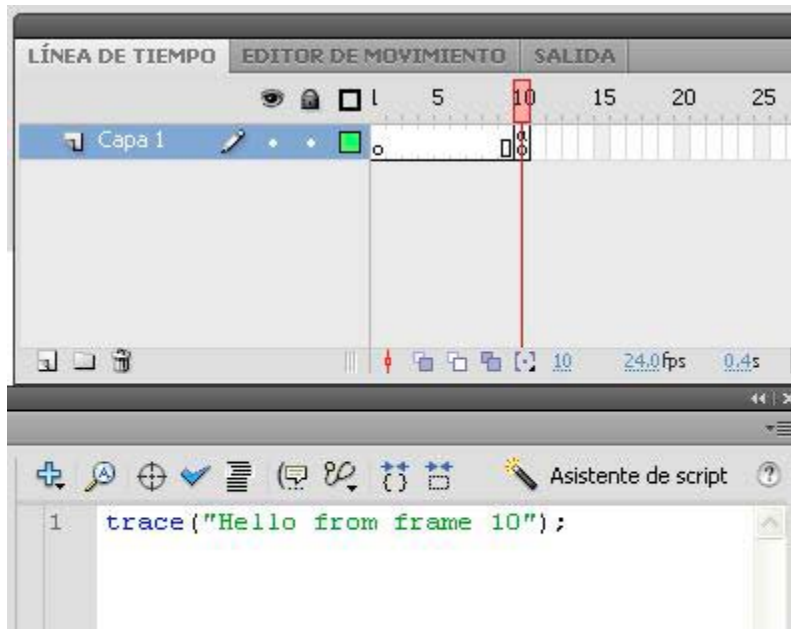
La fonction `call()` prend comme paramètre un numéro d'image ou d'étiquette. Par exemple, le code ActionScript suivant appelle le code situé sur l'image `moveUp` :

```
call("moveUp");
```

La fonction `call()` agit de manière synchronisée ; tout code ActionScript suivant une fonction `call()` ne s'exécute que lorsque l'intégralité du code ActionScript est exécutée au niveau de l'image spécifiée.

Pour appeler du code ActionScript sur une autre image :

- 1 Dans un nouveau document Flash, insérez une image-clé dans l'image 10.



- 2 Avec la nouvelle image-clé sélectionnée, ouvrez le panneau Actions (Fenêtre > Actions) et saisissez le code suivant :

```
trace("Hello from frame 10");
```

- 3 Sélectionnez l'image-clé de l'image 1, puis tapez le code suivant dans le panneau Actions :

```
stop();  
call(10);
```

Ce code arrête la tête de lecture sur l'image 1, puis appelle le code sur l'image 10.

- 4 Testez l'application dans Adobe® Device Central™.

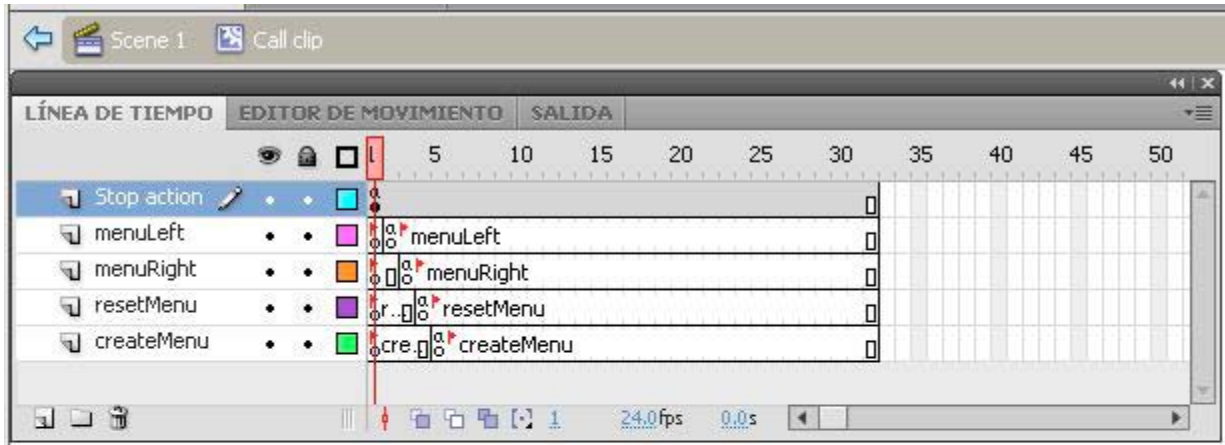
Le message « Hello from frame 10 » doit s'afficher dans l'émulateur.

Vous pouvez également appeler du code résidant dans un autre scénario, par exemple un scénario de clip. Pour exécuter le code, indiquez le nom d'occurrence du clip suivi de deux-points (:), puis du numéro d'image ou d'étiquette. Par exemple, le code ActionScript suivant appelle le code situé sur l'image `moveUp` dans l'occurrence de clip nommée `callClip` :

```
call("callClip:moveUp");
```

Cette technique est souvent utilisée pour créer des *clips d'appel* ou des *clips de fonction* ; ces clips ont pour seule fonction d'encapsuler du code fréquemment utilisé. Un clip d'appel contient une image-clé de chaque fonction que vous souhaitez créer. Vous étiquetez généralement chaque image-clé en fonction de son rôle. Adobe recommande également de créer un nouveau calque pour chaque image-clé et de donner à chaque calque le même nom qu'à l'étiquette d'image assignée à l'image-clé.

L'illustration suivante présente le scénario d'un exemple de clip d'appel. La première image-clé d'un clip d'appel contient toujours une action `stop()` permettant de garantir que la tête de lecture ne passe pas continuellement en boucle sur les images du scénario. Les images-clés suivantes contiennent du code pour chaque « fonction ». Chaque image-clé de fonction est étiquetée de manière à identifier son utilisation. Pour faciliter la modification et l'affichage du clip d'appel, chaque image-clé de fonction est généralement insérée sur un calque distinct.



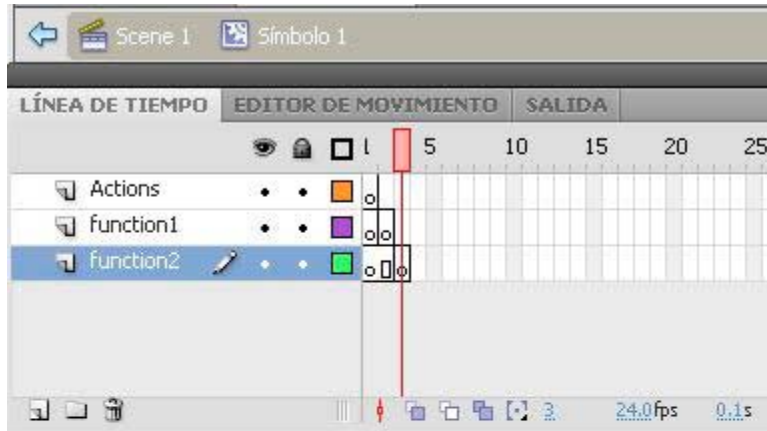
A. Etiquettes d'images B. Chaque image-clé contient du code. C. La première image-clé contient une action stop().

La procédure suivante explique comment créer et utiliser un clip d'appel.

Pour créer et utiliser un clip d'appel :

- 1 Dans Adobe Flash, créez un document mobile.
- 2 Sélectionnez Insertion > Nouveau symbole.
- 3 Dans la boîte de dialogue Créer un nouveau symbole, tapez **Call Clip** dans le champ texte Nom, puis cliquez sur OK.
Le clip s'ouvre en mode d'édition.
- 4 Cliquez deux fois sur le bouton Insérer un calque dans la fenêtre Scénario pour insérer deux nouveaux calques.
Nommez le calque supérieur **Actions**, le deuxième **function1** et le troisième **function2**.

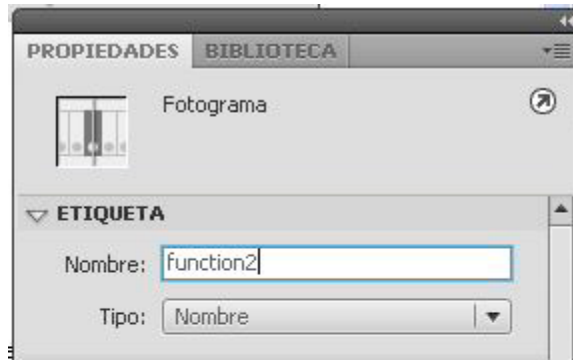
- 5 Insérez une image-clé dans l'image 2 du calque function1 et une autre image-clé dans l'image 3 du calque function2, comme indiqué ci-dessous :



- 6 Sélectionnez l'image-clé dans le calque Actions et ouvrez le panneau Actions.
- 7 Ajoutez une action **stop()** dans le panneau Actions.
- 8 Sélectionnez l'image-clé de l'image 2 dans le calque function1, puis effectuez l'une des opérations suivantes :
- a Dans l'inspecteur Propriétés, tapez **function1** dans la zone de texte Etiquette de l'image.
 - b Dans le panneau Actions (Fenêtre > Actions), tapez le code suivant :

```
trace("function1 was called.");
```

- 9 Sélectionnez l'image-clé de l'image 3 dans le calque function2, puis effectuez l'une des opérations suivantes :
 - a Dans l'inspecteur Propriétés, tapez **function2** dans la zone de texte Etiquette de l'image.



- b Dans le panneau Actions (Fenêtre > Actions), tapez le code suivant :

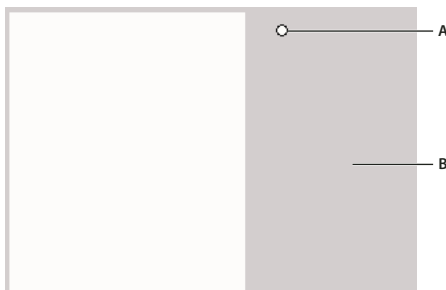
```
trace("function2 was called.");
```

- 10 Appuyez sur Ctrl+E (Windows) ou sur Commande+E (Macintosh) pour retourner dans le scénario principal.
- 11 Paramétrez l'affichage de votre document de manière à inclure l'espace de travail entourant la scène en choisissant Affichage > Zone de travail.


L'utilisateur n'ayant pas besoin de voir le clip d'appel, vous pouvez le placer dans la zone de travail.

- 12 Ouvrez le panneau Bibliothèque (Fenêtre > Bibliothèque) et faites glisser le symbole du clip d'appel dans la zone de travail entourant la scène.

Le clip d'appel ne contenant aucun élément visuel, il apparaît sur la scène sous la forme d'un petit cercle représentant le point d'alignement du clip.



A. Occurrence de clip d'appel B. Zone de travail entourant la scène

 Afin de faciliter l'identification de votre clip d'appel sur la scène, ajoutez du texte ou tout autre élément visuel à la première image-clé dans le scénario du clip.

- 1 Dans l'inspecteur Propriétés, tapez **callClip** dans la zone de texte Nom de l'occurrence.
- 2 Dans le scénario, sélectionnez l'image 1 dans le calque ActionScript.
- 3 Dans le panneau Actions, tapez le code suivant :

```
call("callClip:function1");  
call("callClip:function2");
```

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

Vous pouvez voir le texte suivant :

```
function1 was called.  
function2 was called.
```

Utilisation de la fonction eval()

La fonction `eval()` permet de référencer de façon dynamique les variables et occurrences de clip lors de l'exécution. Elle prend comme paramètre une expression de chaîne et renvoie soit la valeur de la variable représentée par cette expression, soit une référence au clip.

Par exemple, le code suivant évalue la valeur de la variable ActionScript `name` et attribue le résultat à `nameValue` :

```
name = "Jack";  
nameValue = eval("name");  
// result: nameValue = "Jack"
```

Flash Lite ne prenant pas en charge les structures de données de tableau natives, la fonction `eval()` est souvent utilisée en association avec des boucles `for()` et l'opérateur `add` (concaténation de chaîne) pour créer des tableaux basés sur des chaînes. Pour plus d'informations, consultez la section « [Emulation des tableaux](#) » à la page 77.

Vous pouvez également utiliser la fonction `eval()` pour référencer les occurrences de clip en fonction de leur nom. Supposons par exemple que vous disposiez de trois clips intitulés `clip1`, `clip2` et `clip3`. La boucle `for()` suivante incrémente la position `x` de chaque clip de 10 pixels :

```
for(index = 1; index <= 3; index++) {  
    eval("clip" + index)._x += 10  
}
```

Tâches de script courantes

Cette section décrit les tâches de script courantes de Flash Lite qui permettent d'utiliser le périphérique de l'utilisateur. Elles incluent, par exemple, la récupération des informations relatives aux fonctionnalités du périphérique, le lancement d'appels téléphoniques et l'envoi de messages texte, ainsi que l'évaluation de l'état du réseau.

Définition des fonctionnalités de la plate-forme et du périphérique

Macromedia Flash Lite 1.1 d'Adobe comporte de nombreuses variables ActionScript qui fournissent des informations sur les fonctionnalités disponibles pour les applications Flash Lite exécutées sur un périphérique spécifique. Par exemple, la variable `_capLoadData` indique si le périphérique prend en charge le chargement de données externes et la variable `_capSMS` indique s'il prend en charge l'envoi de messages SMS. Pour obtenir une liste complète des variables de fonctionnalité, consultez la section *Fonctionnalités du Guide de référence du langage ActionScript Flash Lite 1.x*.

Les variables de fonctionnalité sont généralement utilisées pour déterminer si un périphérique prend en charge une fonctionnalité spécifique avant son utilisation. Supposons par exemple que vous souhaitiez développer une application qui télécharge des données depuis un serveur Web à l'aide de la fonction `loadVariables()`. Avant de tenter de télécharger les données, vous pouvez d'abord vérifier la valeur de la variable `_capLoadData` pour déterminer si le périphérique prend en charge cette fonctionnalité en procédant comme suit :

```
if(_capLoadData == 1) {  
    loadVariables("http://foo.com/data.txt");  
} else {  
    status_message = "Sorry, unable to load external data."  
}
```

Flash Lite définit les variables de fonctionnalité dans le scénario racine du fichier SWF principal. Par conséquent, pour accéder à ces variables depuis un autre scénario (par exemple, depuis le scénario d'un clip), vous devez spécifier le chemin d'accès à la variable. L'exemple suivant utilise une barre oblique (/) pour indiquer le chemin entièrement qualifié de la variable `_capSMS`.

```
canSendSMS = /:_capSMS
```

Ouverture d'une page Web

Utilisez la commande `getURL()` pour ouvrir une page Web dans le navigateur Web du périphérique. La procédure est la même que pour ouvrir une page Web depuis une application Flash pour ordinateurs de bureau. Par exemple, la procédure suivante permet d'ouvrir la page Web d'Adobe :

```
getURL("http://www.adobe.com");
```

Flash Lite ne traite qu'une seule action `getURL()` par image ou par gestionnaire d'événements. Certains téléphones limitent l'action `getURL()` aux événements de pression de touche uniquement, auquel cas l'appel `getURL()` n'est traité que s'il est déclenché dans un gestionnaire d'événements de pression de touche. Même dans ces conditions, une seule action `getURL()` est traitée par gestionnaire d'événements de pression de touche. Le code suivant, associé à une occurrence de bouton sur la scène, ouvre une page Web lorsque l'utilisateur appuie sur le bouton de sélection du périphérique :

```
on (keyPress "<Enter>") {  
    getURL("http://www.adobe.com");  
}
```

Lancement d'un appel téléphonique

Pour lancer un appel téléphonique depuis une application Flash Lite, utilisez la fonction `getURL()`. Cette fonction permet généralement d'ouvrir une page Web, mais dans le cas présent, vous devez spécifier la fonction `tel` : comme protocole (au lieu de `http`), puis fournir le numéro de téléphone à composer. Lorsque vous appelez cette fonction, Flash Lite affiche une boîte de dialogue de confirmation demandant à l'utilisateur l'autorisation de composer le numéro spécifié.

Le code suivant tente de lancer un appel vers le 555-1212 :

```
getURL("tel:555-1212");
```

Flash Lite ne traite qu'une seule action `getURL()` par image ou par gestionnaire d'événements. Certains téléphones limitent l'action `getURL()` aux événements de pression de touche uniquement, auquel cas l'appel `getURL()` n'est traité que s'il est déclenché dans un gestionnaire d'événements de pression de touche. Même dans ces conditions, une seule action `getURL()` est traitée par gestionnaire d'événements de pression de touche. L'exemple suivant lance un appel téléphonique lorsque l'utilisateur appuie sur le bouton de sélection du périphérique :


```
on (keyPress "<Enter>") {  
    getURL("tel:555-1212");  
}
```

Envoi d'un message texte ou multimédia

Vous pouvez utiliser Flash Lite pour envoyer des messages SMS (texte) ou MMS (multimédia). Pour envoyer un message SMS ou MMS depuis une application Flash Lite, utilisez la commande `getURL()` en lui indiquant le protocole `sms:` ou `mms:` au lieu du protocole standard `http`, puis précisez le numéro de téléphone auquel envoyer le message.

```
getURL("sms:555-1212");
```

Vous pouvez également spécifier le corps du message dans la chaîne de requête de l'URL, comme le montre le code ci-dessous :

```
getURL("sms:555-1212?body=More info please");
```

Pour envoyer un message MMS, préférez le protocole `mms:` au protocole `sms:`, comme suit :

```
getURL("mms:555-1212");
```

Remarque : Vous ne pouvez pas spécifier de pièce jointe pour le message MMS depuis Flash Lite.

Flash Lite ne traite qu'une seule action `getURL()` par image ou par gestionnaire d'événements. Certains téléphones limitent l'action `getURL()` aux événements de pression de touche uniquement, auquel cas l'appel `getURL()` n'est traité que s'il est déclenché dans un gestionnaire d'événements de pression de touche. Même dans ces conditions, une seule action `getURL()` est traitée par gestionnaire d'événements de pression de touche. L'exemple suivant envoie un message SMS lorsque l'utilisateur appuie sur le bouton de sélection du périphérique :

```
on (keyPress "<Enter>") {  
    getURL("sms:555-1212");  
}
```

Envoi d'un message électronique

Vous pouvez utiliser Flash Lite pour envoyer un message électronique. Utilisez la commande `getURL()` en lui indiquant le protocole `mailto:` suivi de l'adresse électronique du destinataire. Vous pouvez également spécifier l'objet et le corps du message dans la chaîne de requête de l'URL, comme indiqué ci-dessous :

```
getURL("mailto:mobile-developer@adobe.com?subject=Flash Lite");
```

Pour spécifier simplement le corps du message dans la chaîne de requête, utilisez le code suivant :

```
getURL("mailto:mobile-developer@adobe.com?body=More+info+please");
```

Chargement de fichiers SWF externes

La fonction `loadMovie()` permet de charger des fichiers SWF à partir d'un réseau ou d'un fichier local. Cette fonctionnalité est disponible uniquement dans Flash Lite 1.1 et versions ultérieures. Les avertissements suivants s'appliquent lorsque vous chargez des fichiers SWF externes :

- Flash Lite peut charger d'autres fichiers SWF de Flash Lite 1.0 ou Flash Lite 1.1, ou bien des fichiers SWF au format Flash 4 ou antérieur. Si vous tentez de charger un fichier SWF dans un autre format (par exemple, un fichier SWF de Flash Player 6), Flash Lite génère une erreur d'exécution.

- Flash Lite ne peut pas charger directement des fichiers image externes, tels que des images JPEG ou GIF. Pour charger ces types de média, vous devez convertir les données au format de fichier SWF. Vous pouvez effectuer cette opération « manuellement » à l'aide de l'outil de programmation de Flash en important le fichier image dans un nouveau document, puis en l'exportant dans un fichier SWF de Flash Lite ou Flash 4. Il existe également des utilitaires tiers qui permettent d'effectuer ce type de conversion automatiquement.

Pour plus d'informations sur le chargement de fichiers SWF, consultez la section `loadMovie()` dans le *Guide de référence du langage ActionScript Flash Lite 1.x*.

Chargement de données externes

Pour charger des données externes dans une application Flash Lite, utilisez la fonction `loadVariables()`. Vous pouvez charger des données via le réseau (à partir d'une adresse HTTP) ou le système de fichiers local. Cette fonctionnalité est disponible uniquement dans Flash Lite 1.1 et versions ultérieures.

Cette section décrit l'utilisation de la fonction `loadVariables()` qui permet de charger des données à partir d'un fichier externe et de les afficher dans des champs texte dynamique. Vous allez d'abord créer le fichier de données, un fichier texte contenant cinq paires nom-valeur séparées par des esperluettes (&). Vous allez ensuite créer l'application Flash Lite qui charge et affiche les données contenues dans le fichier texte.

Cet exemple présume que le fichier de données et le fichier SWF se trouvent dans le même dossier, soit sur votre ordinateur (si vous testez dans l'émulateur) soit sur la carte mémoire du périphérique (si vous testez sur un véritable périphérique). Pour tester l'application sur le périphérique, vous devez effectuer l'une des opérations suivantes :

- Transférez le fichier texte sur votre périphérique et placez-le dans le même dossier que le fichier SWF.
- Publiez le fichier texte sur une URL d'un serveur Web (par exemple, `www.your-server.com/data.txt`). Modifiez ensuite l'appel `loadVariables()` dans l'exemple d'application de manière à pointer vers cette URL, comme indiqué ci-dessous :

```
loadVariables("http://www.your-server.com/data.txt", "data_clip");
```

Pour obtenir un exemple d'application chargeant des données via le réseau, consultez la section « Interface de Flash Lite pour la lecture de nouvelles » dans le guide *Exemples Flash*.

Pour créer le fichier de données :

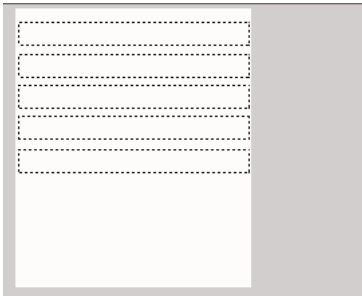
- 1 A l'aide d'un éditeur de texte (par exemple, Notepad ou SimpleText), créez un fichier contenant le texte suivant :

```
item_1=Hello&item_2=Bonjour&item_3=Hola&item_4=Buon+giorno&item_5=G'day
```
- 2 Enregistrez le fichier sous `data.txt`.

Pour créer l'application Flash Lite devant charger les données :

- 1 Dans Flash, créez un document mobile.
- 2 Enregistrez le fichier sous le nom `dataloading fla` dans le dossier contenant le fichier texte (`data.txt`) que vous avez créé précédemment.
- 3 Dans le scénario, sélectionnez l'image 1 du calque Content.

- 4 Avec l'outil Texte, créez cinq champs texte dynamique sur la scène, comme le montre l'illustration suivante :



- 5 Sélectionnez le premier champ texte (celui du haut) puis, dans l'inspecteur Propriétés, tapez `item_1` dans la zone de texte VAR.

Ce nom de variable correspond au nom de la première variable définie dans le fichier `data.txt` que vous avez précédemment créé (`item_1=Hello`).

- 6 De la même manière que celle décrite dans les deux étapes précédentes, affectez aux quatre champs texte restants les noms de variable `item_2`, `item_3`, `item_4` et `item_5`.

- 7 Appuyez sur la touche Maj et sélectionnez tous les champs, puis choisissez Modifier > Convertir en symbole.

- 8 Dans la boîte de dialogue Convertir en symbole, sélectionnez Clip comme type de symbole, puis cliquez sur OK.

- 9 Sélectionnez le clip que vous venez de créer puis, dans l'inspecteur Propriétés, tapez **data_clip** dans la zone de texte Nom de l'occurrence.

- 10 Dans le scénario, cliquez sur l'image 1 du calque Actions et ouvrez le panneau Actions (Fenêtre > Actions).

- 11 Tapez le code suivant dans le panneau Actions :

```
loadVariables("data.txt", "data_clip");
```

- 12 Enregistrez vos modifications (Fichier > Enregistrer), puis testez l'application dans l'émulateur (Contrôle > Tester l'animation).

Tous les champs texte devraient afficher les données du fichier texte, comme le montre l'illustration suivante :



Index

A

ActionScript Flash Lite 1.x
 code ActionScript de Flash 4 non pris en charge 75
 différences entre 1.0 et 1.1 74
 fonctionnalités non disponibles dans 75
 présentation 74
 animation interpolée 14
 appel téléphonique, lancement 84
 application Hello World 7

C

Café Townsend, application
 création d'éléments de navigation 16
 création de l'animation Specials 14
 création de l'écran Specials 13
 création du menu principal 12
 présentation 10
 call(), fonction
 utilisation 78
 calques de masque, utilisation 14
 chaînes, concaténation 78
 champs texte
 champs texte de saisie, utilisation 41
 création de texte défilant 51
 dynamique 16
 limitation des caractères dans les champs texte de saisie 44
 méthodes de rendu des polices, présentation 40
 propriétés des paramètres 16
 champs texte de saisie
 et rectangle de focus 45
 exemple d'application 48
 limitation des caractères dans 44
 charger des données externes 86
 charger des fichiers SWF externes 85
 clips
 récupération et définition des propriétés 76
 référencement dynamique 83
 clips de fonction, création 78
 concaténer des chaînes 78

création d'éléments de navigation
 création d'un bouton de détecteur de touches 16
 utilisation des boutons 12
 utilisation des touches programmables 16
 création Flash Lite, fonctionnalités
 émulateur 6
 paramètres du périphérique 6

D

disponibilité de Flash Lite 3

E

émulateur Adobe Device Central 6
 options de débogage 64
 émulateur Flash Lite
 fonctionnalités non prises en charge 63
 interaction 65
 messages d'avertissement et d'erreur 68
 envoyer des messages 85
 eval(), fonction
 utilisation 83
 événements de bouton
 gestion 30
 présentation 30
 événements de pression de touche
 codes de touche ActionScript 25
 création d'un bouton de détecteur de touches 25
 rédaction d'un script de gestionnaire de touches 25
 touches prises en charge 20
 traitement avec ActionScript 24

F

Flash Lite
 disponibilité 3
 fonctionnalités de création dans Flash 6
 lecteur autonome 3
 présentation de la programmation 5
 présentation de la technologie 3
 types de contenus 4
 fonctionnalités de la plate-forme, détermination 71
 fonctions, émulation avec call() 78

G

getURL(), fonction
 envoi d'un message électronique avec 85
 envoi d'un message multimédia avec 85
 envoi d'un message texte avec 85
 lancement d'un appel téléphonique avec 84
 ouverture de pages Web avec 84

I

inspecteur Propriétés, bouton Paramètres du périphérique 7
 interactivité
 création à l'aide de boutons 30
 création d'un menu à l'aide de boutons 32
 création d'un menu à l'aide de clips 26
 détection des pressions de touche 25
 gestion des événements de pression de touche 24
 navigation par tabulation 21
 utilisation des touches programmables 35

L

lancer des appels téléphoniques 84
 loadMovie(), fonction
 utilisation 85
 loadVariables(), fonction
 utilisation 86

M

menus
 création à l'aide de boutons 32
 création à l'aide de clips 26
 message électronique, envoi 85
 message multimédia, envoi 85
 message texte, envoi 85
 méthodes de rendu des polices
 appliquées aux champs texte 45
 présentation 45
 modes d'application, dans Flash Lite 65

N

navigation par tabulation
 bidirectionnelle 22
 exemple d'application utilisant 32
 modes de 21

- présentation 21
- quadridirectionnelle 22
- quadridirectionnelle en boucle 22
- recommandations pour 24
- rectangle de focus 22
- navigation. *Voir* navigation par tabulation

O

- ouvrir une page Web 84

P

- page Web, ouverture 84
- périphériques cibles 4
- périphériques de test 4
- plate-forme, présentation des variables de fonctionnalité 83
- polices vectorielles intégrées
 - fonctionnement 47
 - présentation 47
- processus de création de contenu 5

Q

- qualité de rendu de Flash Lite
 - et champs texte 47
 - qualité de rendu par défaut 47

R

- regroupement audio, utilitaire
 - utilisation pour les sons composés 59

S

- scénarios, contrôle à l'aide d'ActionScript 76
- scène, taille à l'écran et disponibilité 67
- son
 - composé 59
 - événement et lecture en flux continu 53
 - périphérique 53
 - périphérique et natif 53
 - présentation 53
 - utilitaire de regroupement audio 59
- son de périphérique
 - ajout 54
 - déclenchement dans Flash Lite 1.0 58
 - présentation 53
- son natif, présentation 61

T

- tableaux, émulation à l'aide de chaînes 77
- taille de scène disponible 7
- texte défilant, création 78

- texte dynamique, champ 16
- touches programmables
 - commande SetSoftKeys 35
 - utilisation 35
- touches programmables, utilisation 16
- types de contenus 4
- types de contenus dans Flash Lite, description 65
- Types de contenus Flash Lite 4

V

- variables
 - référencement 77
 - référencement dynamique 83
 - syntaxe de point et syntaxe à barre oblique 77