

Utilisation de ADOBE® FLASH® BUILDER 4

Dernière mise à jour le 29/4/2010

© 2010 Adobe Systems Incorporated. All rights reserved.

Utilisation d'Adobe® Flash® Builder™ 4.

If this guide is distributed with software that includes an end-user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end-user license agreement.

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

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names in sample templates are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe, the Adobe logo, ActionScript, ColdFusion, Flash, Flash Builder, Flex, and Flex Builder 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 trademark of Microsoft Corporation in the United States and/or other countries. Java is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. All other trademarks are the property of their respective owners.

This Work is licensed under the Creative Commons Attribution Non-Commercial 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes software developed by the OpenSymphony Group (<http://www.opensymphony.com/>)

This product contains either BSAFE and/or TIPEM software by RSA Data Security, Inc.

The Flash Builder 3 software contains code provided by the Eclipse Foundation ("Eclipse Code"). The source code for the Eclipse Code as contained in Flash Builder 3 software ("Eclipse Source Code") is made available under the terms of the Eclipse Public License v1.0 which is provided herein, and is also available at <http://www.eclipse.org/legal/epl-v10.html>.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, 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. 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 : A propos de Flash Builder

Possibilités d'utilisation de Flash Builder	1
Versions de Flash Builder	2
Programmes d'installation Flash Builder	2
Client de l'Aide de la communauté Adobe	3

Chapitre 2 : Workbench Flash Builder

Notions de base du workbench Flash Builder	4
Navigaton et personnalisation du workbench Flash Builder	23

Chapitre 3 : Utilisation des projets

A propos des projets Flash Builder	35
Création de projets Flex	41
Gestion des projets	49
Exportation et importation de projets	53
Exportation d'une application vers une version validée	59
Gestion des ressources du projet	62
Projets ActionScript	66
Projets de bibliothèque	69
Génération de projets	73
Génération par ligne de commande de Flash Builder	86
Exécution des applications	90
Création de modules	96

Chapitre 4 : Modification du code dans Flash Builder

A propos de la modification du code dans Flash Builder	103
Aide au codage dans Flash Builder	105
Navigaton et organisation du code	111
Formatage et modification du code	116
Recherche de références et restructuration de code	118
Icônes représentant les éléments de langage	120
A propos des marqueurs	122
A propos de la vérification de la syntaxe	124
Raccourcis clavier d'édition de code	126
Personnalisation des modèles de fichier	127

Chapitre 5 : Création de composants MXML personnalisés

A propos des composants personnalisés	134
Création de composants MXML avec Flash Builder	134
Création de composants visuels	135
Définition de commentaires ASDoc pour les composants personnalisés	136
Modification et distribution des composants MXML personnalisés	136

Chapitre 6 : Développement d'applications AIR avec Flash Builder

Création de projets AIR avec Flash Builder	137
Conversion de projets Flex en projets Adobe AIR	137
Débogage des applications AIR avec Flash Builder	137
Création de packages d'application AIR avec Flash Builder	138
Création d'un projet de bibliothèque AIR	140

Chapitre 7 : Débogage, test et surveillance des applications

Débogage des applications	141
Environnement de test FlexUnit	150
Surveillance des applications accédant aux services de données	155

Chapitre 8 : Profilage des applications Flex

A propos du profilage	159
Fonctionnement du profileur Flex	160
Utilisation du profileur	162
A propos des vues du profileur	168
A propos de la récupération de place	182
Identification des zones à problèmes	182
A propos des filtres du profileur	186

Chapitre 9 : Création d'une interface utilisateur avec Flash Builder

A propos de la structure des interfaces utilisateur dans Flex	188
Ajout et modification des composants	191
Utilisation visuelle des composants	196
Utilisation de présentations basées sur des contraintes	205
Création de questionnaires d'événement	208
Application de thèmes	210
Application de styles	213
Modification de l'interface utilisateur à l'aide des habillages	218
Génération de rendus d'élément personnalisés	224
Actualisation du mode Création pour un rendu correct	227
Ajout d'états d'affichage et de transitions	227
Liaison des contrôles à des données	233
Ajout de composants de représentation graphique	234
Ajout d'interactivité à l'aide d'effets	235

Chapitre 10 : Utilisation de données dans Flash Builder

A propos de l'utilisation des données dans Flash Builder	237
Génération automatique du code Flex Ajax Bridge	241
Gestion de la sécurité de Flash Player	245

Chapitre 11 : Personnalisation de Flash Builder

Préférences Adobe	248
Préférences de Flash Builder	249
Extension de Flash Builder	255

Chapitre 1 : A propos de Flash Builder

Adobe® Flash® Builder™ est un environnement de développement intégré permettant la génération d'applications Internet riches (Rich Internet Applications, RIA) multiplateformes. Flash Builder vous permet de générer des applications utilisant la structure Adobe Flex®, MXML, Adobe Flash Player, Adobe AIR®, ActionScript 3.0, Adobe® LiveCycle® Data Services ES et les composants de représentation graphique Adobe® Flex®. Il comporte également des outils de test, de débogage et de profilage favorisant productivité et efficacité.

Flash Builder est basé sur Eclipse, un environnement IDE de type « open source », et fournit tous les outils requis pour développer des applications qui utilisent la structure Flex et ActionScript 3.0. Flash Builder s'exécute sous Microsoft Windows et Apple Macintosh OS X et est disponible en plusieurs versions. Les options de configuration d'installation vous permettent d'installer Flash Builder en ensemble de plug-ins dans une installation de workbench Eclipse existante ou de créer une installation incluant le workbench Eclipse.

Possibilités d'utilisation de Flash Builder

Flash Builder vous permet de développer des applications dans un environnement IDE complet et de réaliser les tâches suivantes :

- Création de projets Flex compatibles avec toute technologie de serveur back-end, dont Adobe ColdFusion® et Adobe LiveCycle® Data Services, avec ou sans serveur Flex. Voir « [Création de projets Flex](#) » à la page 41.
- Création d'applications accédant aux services de données à distance. Flash Builder fournit des outils et des assistants facilitant l'accès aux services de données. Voir Présentation de l'accès aux services de données.
- Création de projet ActionScript. Voir « [Projets ActionScript](#) » à la page 66.
- Rédaction et modification du code source des applications à l'aide d'éditeurs fournissant des fonctions telles que la restructuration de code, les indicateurs de code, la navigation de code rationalisée et la vérification automatique des erreurs de syntaxe. Voir « [A propos de la modification du code dans Flash Builder](#) » à la page 103.
- Utilisation de l'éditeur MXML en mode Création entre autres pour générer des applications à l'aide des options de présentation, pour faire glisser des composants vers la zone de création avant de les repositionner et de les redimensionner et pour simplifier l'utilisation par des états d'affichage. Voir « [Création d'une interface utilisateur avec Flash Builder](#) » à la page 188.
- Création de fonctions ActionScript dans le code MXML ou dans des fichiers distincts de fonctions, classes et interfaces ActionScript.
- Création de composants personnalisés et accès rapide à partir de la vue Composants. Voir « [Création de composants MXML personnalisés](#) » à la page 134.
- Gestion des projets d'application grâce aux nombreuses fonctions fournies par l'environnement IDE Eclipse sous-jacent. Par exemple, ajout et suppression de projets et de ressources, établissement de liens à des ressources en dehors du projet, etc. Voir « [Gestion des projets](#) » à la page 49 et « [Création de dossiers et de fichiers dans un projet](#) » à la page 63.
- Multiples méthodes de génération des applications à l'aide des générateurs intégrés ou création de générateurs personnalisés à l'aide d'Apache Ant. Voir « [Génération de projets](#) » à la page 73.
- Exécution des applications dans un navigateur Web ou dans la version autonome de Flash Player. Création de configurations de lancement personnalisées pour contrôler l'exécution des applications. Voir « [Exécution des applications](#) » à la page 92 et « [Gestion des configurations de lancement](#) » à la page 94.

- Débogage des applications grâce aux outils de débogage intégrés de Flash Builder. Voir « [Débogage des applications](#) » à la page 141.
- Utilisation du Moniteur de réseau pour générer une piste d'audit détaillée de toutes les données transférées entre l'application Flex locale et le serveur back-end. Voir « [Surveillance des applications accédant aux services de données](#) » à la page 155.
- Publication du code source des applications afin que les utilisateurs et les autres développeurs puissent le visionner. Voir « [Publication du code source](#) » à la page 85.
- Création de projets de bibliothèque générant des fichiers de bibliothèque de composants partagés (SWC) pour la réutilisation et la distribution du code. Voir « [Projets de bibliothèque](#) » à la page 69.
- Personnalisation de l'IDE. Par exemple, réorganisation de l'interface afin d'inclure vos outils favoris dans la présentation spécifique. Voir « [Navigation et personnalisation du workbench Flash Builder](#) » à la page 23.

Versions de Flash Builder

Flash Builder est disponible en deux versions : Standard et Premium.

Flash Builder version Standard Cette version fournit un environnement IDE complet qui vous permet de créer des applications à l'aide de la structure Flex et de l'API Flash. Flash Builder version Standard inclut également des éditeurs MXML, ActionScript et CSS, ainsi que des outils de débogage. Il fournit une bibliothèque de diagrammes et de graphiques interactifs qui vous permettent de créer des tableaux de bord riches, des analyses de données interactives et des composants de visualisation des données.

Flash Builder version Premium Outre les fonctionnalités de la version Standard, Flash Builder Premium comporte des outils de profilage de la mémoire et des performances ainsi que des outils de tests automatisés. Utilisez le Moniteur de réseau pour consulter les données qui sont transmises entre une application client et un service de données. L'environnement de test FlexUnit permet de générer et de modifier des tests reproductibles pouvant être exécutés à partir de scripts ou directement à l'intérieur ou à l'extérieur de l'environnement Flash Builder. Les fonctionnalités de génération par ligne de commande vous permettent de synchroniser les paramètres de génération d'un développeur avec une génération nocturne.

Programmes d'installation Flash Builder

Flash Builder comporte deux programmes d'installation.

Programme d'installation plug-in Ce programme d'installation est destiné aux utilisateurs ayant déjà recours au workbench Eclipse et souhaitant ajouter les plug-ins Flash Builder à leur ensemble de plug-ins Eclipse. Cela peut être le cas par exemple des développeurs d'applications Java. Etant donné qu'Eclipse est une plateforme extensible et ouverte, des centaines de plug-ins sont disponibles à différentes fins de développement.

Programme d'installation autonome Ce programme d'installation est un package personnalisé de plug-ins Eclipse et Flash Builder, créé spécialement pour développer des applications qui utilisent la structure Flex et ActionScript 3.0. L'interface utilisateur de l'installation autonome est plus étroitement intégrée que l'installation plug-in. L'installation autonome permet d'éviter en grande partie les confusions auxquelles l'installation plug-in, multi-usage et ouverte, risque de prêter. Elle est idéale pour les nouveaux utilisateurs ainsi que pour ceux qui souhaitent développer uniquement des applications reposant sur la structure Flex et ActionScript 3.0.

Si vous ne savez pas quel programme d'installation utiliser, suivez les recommandations suivantes :

- Si Eclipse version 3.4.2 (ou ultérieure) est déjà installé, utilisez le programme d'installation plug-in pour ajouter les fonctionnalités Flash Builder à la copie existante d'Eclipse.
- Si aucune version d'Eclipse n'est installée et que votre objectif principal est de développer des applications Flex et ActionScript, utilisez le programme d'installation autonome de Flash Builder. Ce programme d'installation vous permet également d'installer d'autres plug-ins Eclipse afin d'étendre la portée de vos futurs travaux de développement.

Les programmes d'installation plug-in et autonome de Flash Builder fournissent les mêmes fonctionnalités.

Client de l'Aide de la communauté Adobe

Cette version de Flash Builder comporte le client de l'Aide de la communauté Adobe (Adobe Community Help Client, CHC). Le client CHC est une application basée sur AIR qui remplace le moteur d'aide Eclipse pour Flash Builder et constitue la plateforme pour la nouvelle génération d'aide Adobe. Le client CHC se caractérise par les propriétés suivantes :

- Toujours en ligne

Si vous disposez d'une connexion réseau, le client CHC accède au contenu situé sur le Web pour vous fournir en permanence les informations les plus récentes. En l'absence de connexion Internet, il est également opérationnel en mode local.

- Centré sur la recherche

Vous pouvez utiliser les fonctions de recherche de l'Aide de la communauté (adobe.com) ou procéder à une recherche locale. La recherche de l'Aide de la communauté regroupe les ressources, dont celles des sites tiers. La recherche adobe.com comporte des critères vous permettant d'en restreindre la portée.

- Navigation en contexte

La navigation en contexte fournit un ensemble généré dynamiquement de liens associés dirigeant vers des pages clés.

Chapitre 2 : Workbench Flash Builder

Notions de base du workbench Flash Builder

Adobe® Flash® Builder™ repose sur Eclipse, un environnement de développement intégré de type « open source ». Il vous permet de développer des applications Flex® et ActionScript® 3.0 à l'aide de puissants outils de codage, de mise en forme visuelle, de création et de débogage.

A propos du workbench

Le workbench Flash Builder est un environnement de développement entièrement intégré conçu pour vous aider à développer des applications pour la structure Adobe Flex. Vous pouvez développer des applications à déployer sur Adobe Flash Player et des applications de bureau à déployer sur Adobe AIR®. Flash Builder repose sur Eclipse, un environnement IDE de type « open source ». Flash Builder est un ensemble de plug-ins Eclipse vous permettant de créer des applications Flex et ActionScript 3.0. La plupart des fonctionnalités de base de l'environnement IDE de Flash Builder provient d'Eclipse. Cela inclut, par exemple, les fonctions de gestion, de recherche et de navigation dans les ressources. Les plug-ins Flash Builder ajoutent les fonctions et fonctionnalités permettant de créer des applications Flex et ActionScript 3.0. Ils modifient également l'interface utilisateur IDE et certaines fonctionnalités pour prendre en charge ces tâches.

Les informations nécessaires à l'utilisation de Flash Builder sont présentées dans la documentation de Flash Builder. A moins d'utiliser d'autres plug-ins Eclipse (CVS ou Java, par exemple) avec Flash Builder, ou si vous souhaitez étendre les plug-ins Flash Builder (voir « [Extension du workbench Flash Builder](#) » à la page 23), vous n'avez pas à vous soucier de l'infrastructure Eclipse sous-jacente.

Workbench Le terme *workbench* désigne l'environnement de développement Flash Builder. Le workbench contient trois éléments principaux : *les perspectives*, *les éditeurs* et *les vues*. Vous utilisez ces trois composants dans différentes combinaisons à différents stades du processus de développement de l'application. Le workbench constitue le conteneur de tous les outils de développement des applications. Vous pouvez le comparer à Microsoft Visual Studio, qui offre un cadre et les principales fonctionnalités d'une large gamme d'outils de développement.

Perspective Une perspective représente un groupe de vues et d'éditeurs du workbench. Il s'agit essentiellement d'un environnement de travail spécial qui vous aide à réaliser un type de tâche spécifique. Par exemple, Flash Builder contient deux perspectives. La perspective Développement Flash sert au développement d'applications et la perspective Débogage Flash au débogage des applications. Flash Builder Premium contient également la perspective Profil Flash.

Si vous utilisez la configuration plug-in de Flash Builder (voir « [Programmes d'installation Flash Builder](#) » à la page 2), le workbench peut contenir d'autres perspectives telles qu'une perspective Java contenant des éditeurs et des vues permettant de développer des applications Java.

Pour plus d'informations sur les perspectives, voir « [A propos des perspectives de Flash Builder](#) » à la page 7.

Editeur Un éditeur vous permet d'éditer différents types de fichier. Les éditeurs dont vous disposez varient selon le nombre et le type de plug-ins Eclipse installés. Flash Builder contient des éditeurs permettant de rédiger du code MXML, ActionScript 3.0 et CSS (Cascading Style Sheets, feuilles de style en cascade). Pour plus d'informations sur la modification du code dans Flash Builder, voir « [A propos de la modification du code dans Flash Builder](#) » à la page 103.

Vues Une vue comporte généralement un éditeur. Par exemple, lors de l'édition MXML, les vues Composants et Propriétés apparaissent également dans la perspective Développement Flash. Ces vues prennent en charge le développement d'applications et s'affichent par conséquent lorsqu'un fichier MXML est ouvert en édition.

Certaines vues prennent en charge les principales fonctionnalités du workbench. Par exemple, la vue Explorateur de packages vous permet de gérer les fichiers et dossiers au sein du workbench et la vue Tâches affiche toutes les tâches automatiquement générées par le workbench ou ajoutées manuellement.

Le terme *vue* est synonyme de *panneau*, ce dernier étant utilisé dans les versions précédentes de Flex Builder, dans Adobe Dreamweaver® et dans d'autres outils de développement Adobe.

Espace de travail A ne pas confondre avec le *workbench*, un *espace de travail* représente une zone définie d'un système de fichiers contenant les ressources (fichiers et dossiers) constituant vos projets d'application. Vous ne pouvez travailler que dans un seul espace de travail à la fois, mais vous pouvez cependant sélectionner un espace de travail différent à chaque lancement de Flash Builder. Pour plus d'informations, voir « [Gestion des projets](#) » à la page 49.

Ressource Le terme *ressource* est utilisé de façon générique pour désigner les fichiers et dossiers des projets d'un espace de travail. Pour plus d'informations, voir « [Création de dossiers et de fichiers dans un projet](#) » à la page 63.

Projet Toutes les ressources constituant vos applications figurent dans des projets. Vous ne pouvez pas créer une application dans Flash Builder sans créer au préalable un projet. Vous pouvez créer trois types de projet dans Flash Builder : les projets Flex, les projets ActionScript et les projets de bibliothèque. Pour plus d'informations, voir « [Utilisation des projets](#) » à la page 35.

Configuration de lancement Une *configuration de lancement* est créée pour chacun de vos projets. Elle définit les paramètres de projet utilisés lors de l'exécution et du débogage de vos applications. Par exemple, les noms et emplacements des fichiers SWF d'application compilés sont contenus dans la configuration de lancement, et vous pouvez modifier ces paramètres. Pour plus d'informations, voir « [Exécution des applications](#) » à la page 92.

A propos des éditeurs Flash Builder

Flash Builder contient des éditeurs vous permettant d'éditer du code MXML, ActionScript 3.0 et CSS. Les éditeurs constituent la pierre angulaire du workbench et des vues. Les perspectives dans lesquelles ils figurent prennent en charge leurs fonctionnalités.

Les éditeurs sont associés à des types de ressource. Ainsi, lorsque vous ouvrez des ressources dans le workbench, l'éditeur correspondant s'affiche. Le workbench représente un environnement orienté document (et orienté projet) destiné au développement d'applications.

Le développement d'applications Flex dans Flash Builder repose sur l'utilisation des éditeurs MXML, ActionScript 3.0 et CSS. Chaque éditeur intègre les fonctionnalités nécessaires au type de ressource donné. Flash Builder contient les éditeurs suivants.

Editeur MXML Utilisez l'éditeur MXML pour éditer le code MXML et pour incorporer un code ActionScript et CSS dans les balises `<fx:script>` et `<fx:style>`. L'éditeur MXML propose deux modes : Source et Création. Le mode Source sert à l'édition du code. Le mode Création permet la présentation et la conception visuelles des applications. Les deux modes sont synchronisés et les modifications effectuées dans un mode sont aussitôt répercutées dans l'autre mode. Pour plus d'informations, voir « [A propos de la modification du code dans Flash Builder](#) » à la page 103.

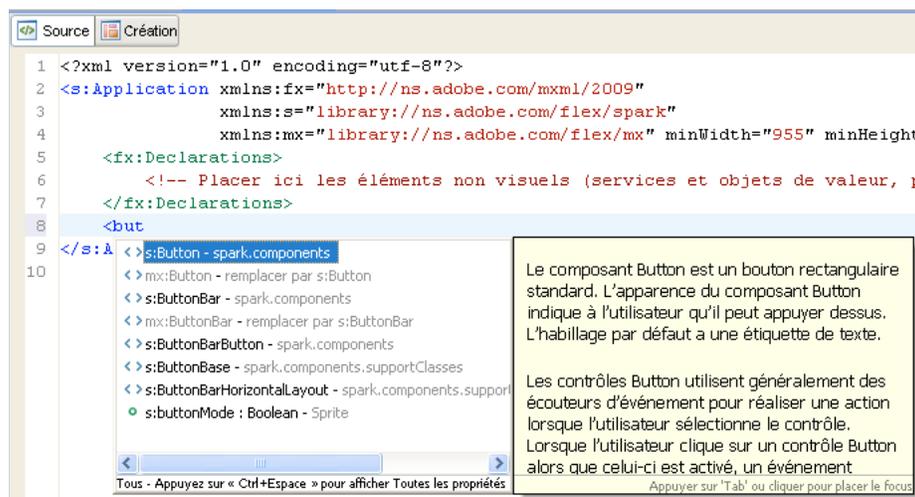
Editeur ActionScript Utilisez l'éditeur ActionScript pour éditer des fichiers de classe et d'interface ActionScript. Bien que vous puissiez incorporer des fonctions ActionScript dans un fichier MXML à l'aide de la balise `<fx:script>`, les classes sont généralement définies dans des fichiers ActionScript, puis importées dans des fichiers MXML. Cette méthode vous permet de définir la plupart de vos applications Flex dans ActionScript.

Editeur CSS Utilisez l'éditeur CSS pour afficher et éditer des feuilles de style en cascade. Vous pouvez ensuite appliquer des styles aux éléments visuels de vos applications. Pour plus d'informations, voir « [Utilisation visuelle des composants](#) » à la page 196 et Using Styles and Themes.

Indicateurs de code

Les éditeurs contiennent de nombreuses fonctionnalités qui simplifient et rationalisent le développement de code. Ils proposent en particulier l'option Assistant de contenu, qui affiche des indicateurs de code et du contenu ASDoc lorsque vous saisissez du code MXML, ActionScript et CSS. Le contenu ASDoc s'affiche également lorsque vous passez le curseur de la souris sur une classe de l'éditeur.

Les indicateurs de code apparaissent automatiquement lorsque vous saisissez du code (vous pouvez également afficher les indicateurs de code en appuyant sur Ctrl+Espace). L'exemple suivant illustre les indicateurs de code dans l'éditeur MXML :



Les indicateurs de code apparaissent lorsque vous commencez à saisir une expression de code reconnue par Flex ou le langage (MXML, ActionScript et CSS). Par exemple, si vous entrez le nom d'un composant, une liste de toutes les propriétés de ce composant apparaît.

Les indicateurs de code ActionScript sont également pris en charge. Les indicateurs de code ActionScript s'affichent dans les balises `<fx:Script>` incorporées dans un document MXML et dans des fichiers ActionScript autonomes. L'Assistant de contenu affiche des indicateurs de code pour tous les éléments de langage ActionScript : interfaces, classes, variables, fonctions, types de renvoi, etc. L'assistant de contenu affiche également le contenu ASDoc pour les classes et interfaces ActionScript.

Il propose également des indicateurs de code pour les composants MXML personnalisés ou les classes ActionScript que vous créez vous-même. Si vous définissez par exemple un composant MXML personnalisé et l'ajoutez à votre projet, les indicateurs de code apparaissent lorsque vous référez le composant dans le fichier d'application MXML.

Pour plus d'informations, voir « [A propos de l'assistant de contenu](#) » à la page 106.

Génération de gestionnaires d'événement

Flash Builder permet la génération automatique de code de gestionnaires d'événement afin de simplifier l'association d'événements aux composants de votre application. En mode Création, sélectionnez un composant. Dans la catégorie Événements de la vue Propriétés, cliquez ensuite sur le bouton Générer un gestionnaire d'événement d'un événement répertorié. L'éditeur affiche le mode Source et génère un gestionnaire d'événement dans la section Script du code source. L'éditeur place également une propriété d'événement dans la balise de composant qui référence le gestionnaire d'événement généré.

Vous pouvez également générer des gestionnaires d'événement en mode Source en faisant appel à l'assistant de contenu.

Pour plus d'informations, voir « [Création de gestionnaires d'événement](#) » à la page 208 et « [Création de gestionnaires d'événement pour les composants](#) » à la page 110.

Navigation par code

La navigation par code simplifie la saisie du code, en particulier avec les projets de grande taille comportant de nombreuses ressources. La navigation par code offre la possibilité de sélectionner un élément de code (une référence à un composant personnalisé d'un fichier d'application MXML, par exemple) et d'aller à la source de la définition du code, quel que soit son emplacement dans le projet, espace de travail ou chemin.

D'autres fonctionnalités de navigation par code incluent la réduction de code qui vous permet de réduire et de développer des instructions de code à plusieurs lignes. Une autre fonctionnalité est la vue Structure, qui offre une présentation hiérarchique et vous permet de parcourir tous les éléments de l'interface utilisateur et de code d'un fichier. Pour plus d'informations, voir « [Navigation et organisation du code](#) » à la page 111.

Formatage de code

Lorsque vous saisissez du code, Flash Builder insère automatiquement des lignes de code pour améliorer la lisibilité, ajoute une couleur distinctive aux éléments de code et propose de nombreuses commandes de formatage rapide du code que vous saisissez (ajout d'un commentaire de bloc, par exemple).

Lorsque vous collez du code MXML ou ActionScript dans l'éditeur de code, Flash Builder met automatiquement en retrait le code en fonction de vos préférences. Vous pouvez également spécifier une mise en retrait pour un bloc de code sélectionné.

Pour plus d'informations, voir « [Formatage et modification du code](#) » à la page 116.

Recherche de références et restructuration du code

Flash Builder peut rechercher toutes les références et les déclarations correspondant aux identifiants d'un fichier, d'un projet ou d'un espace de travail donné. La recherche porte également sur les éléments reliés à partir de fichiers SWC ainsi que les entrées des chemins de bibliothèque (classes, interfaces, fonctions, variables et quelques métadonnées, par exemple). La restructuration vous permet de renommer les identificateurs de votre code tout en mettant à jour toutes les références correspondantes dans la totalité de la base de code. Pour plus d'informations, voir « [Recherche de références et restructuration de code](#) » à la page 118.

A propos des perspectives de Flash Builder

Pour prendre en charge une tâche particulière ou un groupe de tâches, les éditeurs et les vues correspondantes sont combinés dans une *perspective*. Flash Builder contient deux perspectives : Débogage Flash et Développement Flash. Flash Builder Premium contient une perspective supplémentaire, Profil Flash.

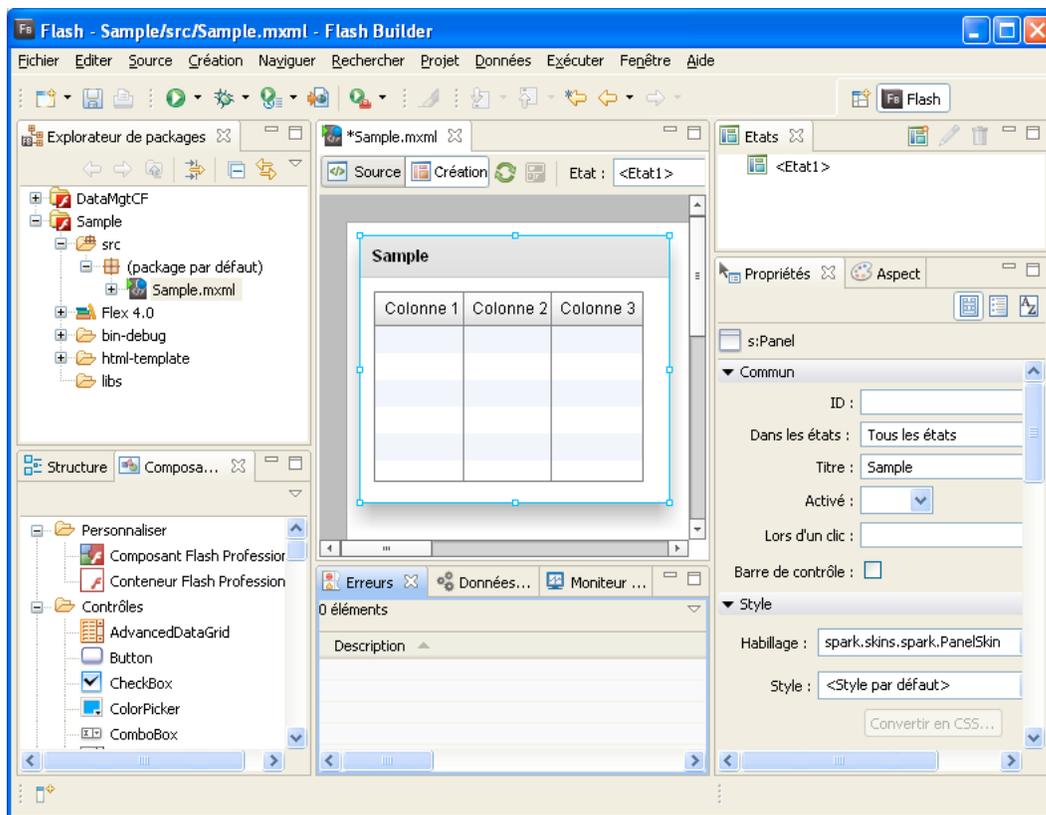
Les perspectives changent automatiquement selon la tâche en cours. Par exemple, si vous créez un projet Flex, le workbench affiche la perspective Développement ; si vous lancez une session de débogage, la perspective Débogage Flash s'affiche lorsque le premier point d'arrêt est atteint. Vous pouvez changer manuellement de perspective en sélectionnant Fenêtre > Perspective dans le menu principal (Fenêtre > Ouvrir la perspective dans la version plug-in). Vous pouvez également utiliser la *barre des perspectives*, située dans la barre d'outils principale du workbench.



Si vous utilisez la configuration plug-in de Flash Builder et que vous avez installé d'autres plug-ins Eclipse, de nombreuses autres perspectives peuvent s'afficher. Chaque plug-in Eclipse propose des perspectives prédéfinies, mais vous pouvez les personnaliser à votre guise ou créer vos propres perspectives. La personnalisation ou la création d'une perspective requiert la sélection, le placement et le redimensionnement des éditeurs et des vues nécessaires à la réalisation de vos tâches de développement. Pour plus d'informations sur l'utilisation et la personnalisation des perspectives, voir « [Utilisation des perspectives](#) » à la page 23.

Perspective Développement Flash

La perspective Développement Flash comporte les éditeurs et les vues dont vous avez besoin pour créer des applications pour la structure Flex. Lorsque vous créez un projet, Flash Builder affiche la perspective Développement pour que vous puissiez commencer à développer une application. L'exemple suivant illustre les vues Explorateur de packages, Structure et Erreurs :



Le point central de la perspective (et du workbench en général) est la zone de l'éditeur. Elle affiche tous les documents actuellement ouverts dans une interface à plusieurs onglets et est entourée des vues correspondantes. Les perspectives prédéfinissent la présentation de tous les éléments qu'elles contiennent. Vous pouvez toutefois les réorganiser comme vous le souhaitez. Pour plus d'informations, voir « [Navigation et personnalisation du workbench Flash Builder](#) » à la page 23.

En mode Source (édition de code), la perspective Développement contient les éléments suivants.

Editeurs Flash Builder

La zone de l'éditeur affiche tous les documents ouverts. Lorsque vous créez un projet Flex, le fichier d'application MXML principal s'ouvre dans la zone de l'éditeur. Vous pouvez ensuite ouvrir et parcourir les documents MXML, ActionScript et CSS que vous utilisez.

L'éditeur MXML comporte deux modes (Source et Création). Le passage de l'un à l'autre adapte la perspective Développement en fonction des tâches qui y seront effectuées.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
3     xmlns:s="library://ns.adobe.com/flex/spark"
4     xmlns:mx="library://ns.adobe.com/flex/mx"
5     minWidth="955" minHeight="600"
6     creationComplete="initApp()" >
7     <fx:Declarations>
8         <!-- Placer ici les éléments non visuels (services
9     </fx:Declarations>
10    <fx:Script>
11        <![CDATA[
12            private function initApp():void
13            {
14                trace("Hello World!");
15            }
16        ]]>
17    </fx:Script>
18    <s:Panel x="10" y="10" width="234" height="200" title="
19    <mx:DataGrid x="10" y="10" width="203">
```

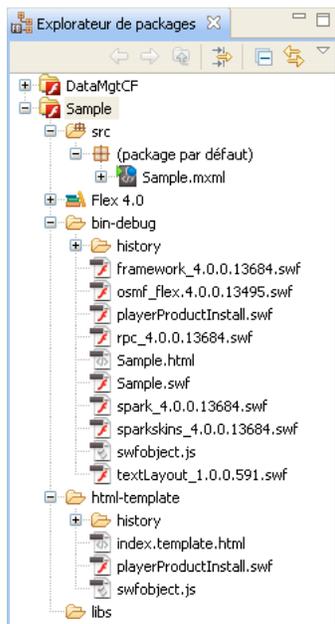
L'éditeur ActionScript est un éditeur à fonction unique permettant la création de fichiers ActionScript.

L'éditeur CSS est un éditeur à fonction unique pour les projets basés sur le SDK Flex 4. Pour les projets basés sur le SDK Flex 3, l'éditeur CSS comporte toutefois deux modes (Source et Création).

Pour plus d'informations sur l'utilisation de l'éditeur MXML, voir « [A propos de la modification du code dans Flash Builder](#) » à la page 103 et « [Utilisation visuelle des composants](#) » à la page 196.

Vue Explorateur de packages

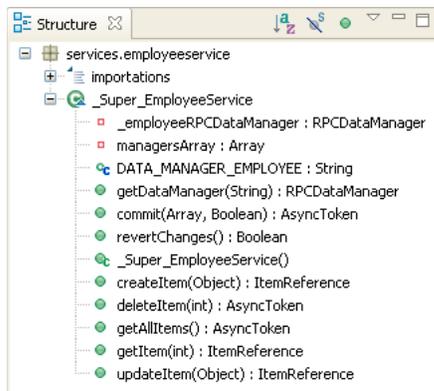
La vue Explorateur de packages contient l'ensemble des projets et des ressources de l'espace de travail et constitue par conséquent un élément essentiel du workbench Flash Builder. Elle apparaît toujours dans les perspectives Développement et Débogage.



Pour plus d'informations sur l'Explorateur de packages et l'utilisation de projets, voir « [Utilisation des projets](#) » à la page 35.

Vue Structure

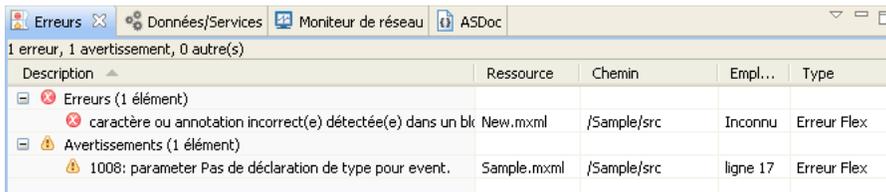
En mode Source, la vue Structure offre une présentation hiérarchique de la structure du code du document MXML ou ActionScript sélectionné, ce qui vous permet d'inspecter et de parcourir les sections ou lignes de code du document. La vue Structure affiche également les alertes d'erreur de syntaxe générées par le compilateur. Cette vue apparaît également lorsque vous utilisez l'éditeur ActionScript.



Pour plus d'informations sur l'utilisation de la vue Structure en mode Source, voir « [Utilisation de la vue Structure pour parcourir et examiner le code](#) » à la page 111.

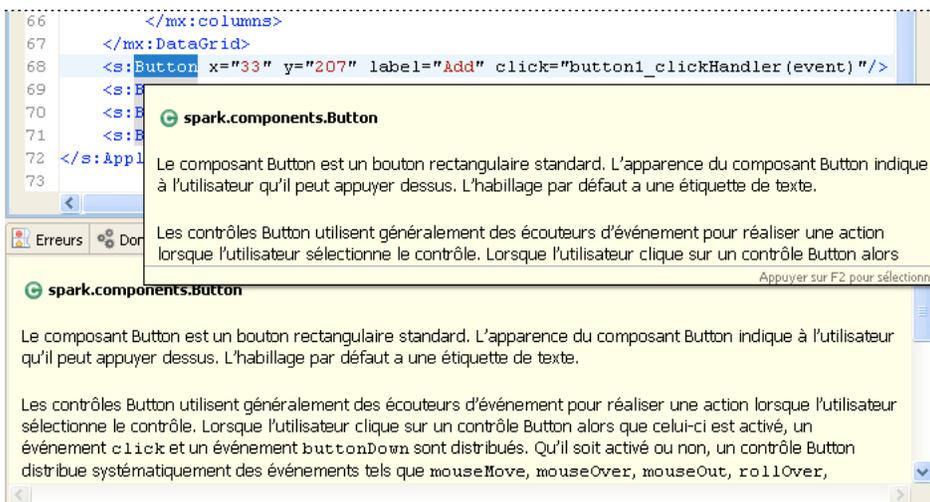
Vue Erreurs

Lorsque vous saisissez du code, le compilateur Flash Builder détecte les erreurs de syntaxe et d'autres erreurs de compilation et les affiche dans la vue Erreurs. Lorsque vous déboguez vos applications, les erreurs, avertissements et autres informations apparaissent dans la vue Erreurs.



Remarque : vous pouvez également ajouter les vues *Tâches* et *Signets*. Ces vues offrent des raccourcis supplémentaires vous permettant de gérer et de parcourir le code. Pour plus d'informations sur ces vues, voir « [A propos des marqueurs](#) » à la page 122. Pour une présentation des vues optionnelles disponibles dans Flash Builder, voir « [Autres vues utiles du workbench](#) » à la page 20.

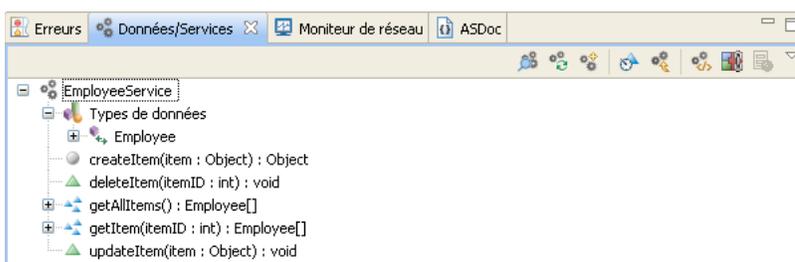
Vue ASDoc



Flash Builder affiche le contenu ASDoc correspondant au code que vous saisissez ou aux éléments de code sur lesquels le curseur de la souris est positionné. Il affiche en outre dans la vue ASDoc le contenu ASDoc correspondant au code sélectionné.

Pour plus d'informations, voir « [Aide au codage dans Flash Builder](#) » à la page 105.

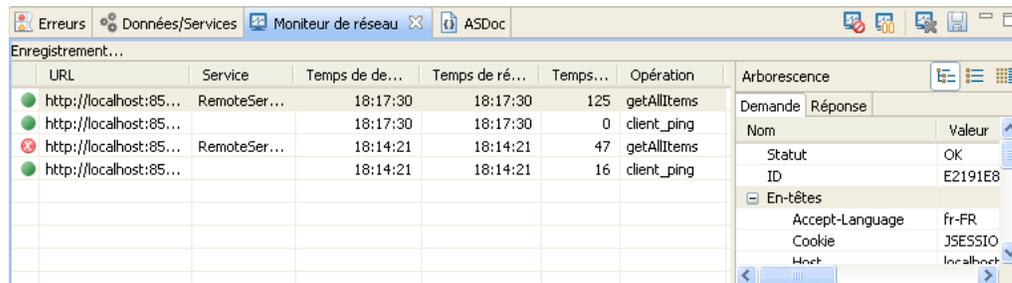
Vue Données/Services



Flash Builder propose des assistants et des outils qui permettent de se connecter aux services de données. Utilisez la vue Données/Services pour vous connecter aux services de données. Une fois établie la connexion à un service de données, la vue Données/Services affiche le service, les types des données retournées et les opérations disponibles pour le service. Elle vous permet en outre de configurer l'accès au service et de lier les données retournées aux composants de l'interface utilisateur.

Pour plus d'informations, voir Construction d'applications centrées sur les données avec Flash Builder.

Vue Moniteur de réseau



URL	Service	Temps de de...	Temps de ré...	Temps...	Opération
http://localhost:85...	RemoteSer...	18:17:30	18:17:30	125	getAllItems
http://localhost:85...		18:17:30	18:17:30	0	client_ping
http://localhost:85...	RemoteSer...	18:14:21	18:14:21	47	getAllItems
http://localhost:85...		18:14:21	18:14:21	16	client_ping

The screenshot also shows a detailed view of a request on the right side of the window, including fields like Statut (OK), ID (E2191E8), and various headers (Accept-Language: fr-FR, Cookie: JSESSIONID, Host: localhost).

Le Moniteur de réseau permet d'analyser les données transmises entre une application et le ou les services de données. Il est disponible avec Flash Builder Premium.

Pour plus d'informations, voir « [Surveillance des applications accédant aux services de données](#) » à la page 155.

Perspective Développement Flash en mode Création

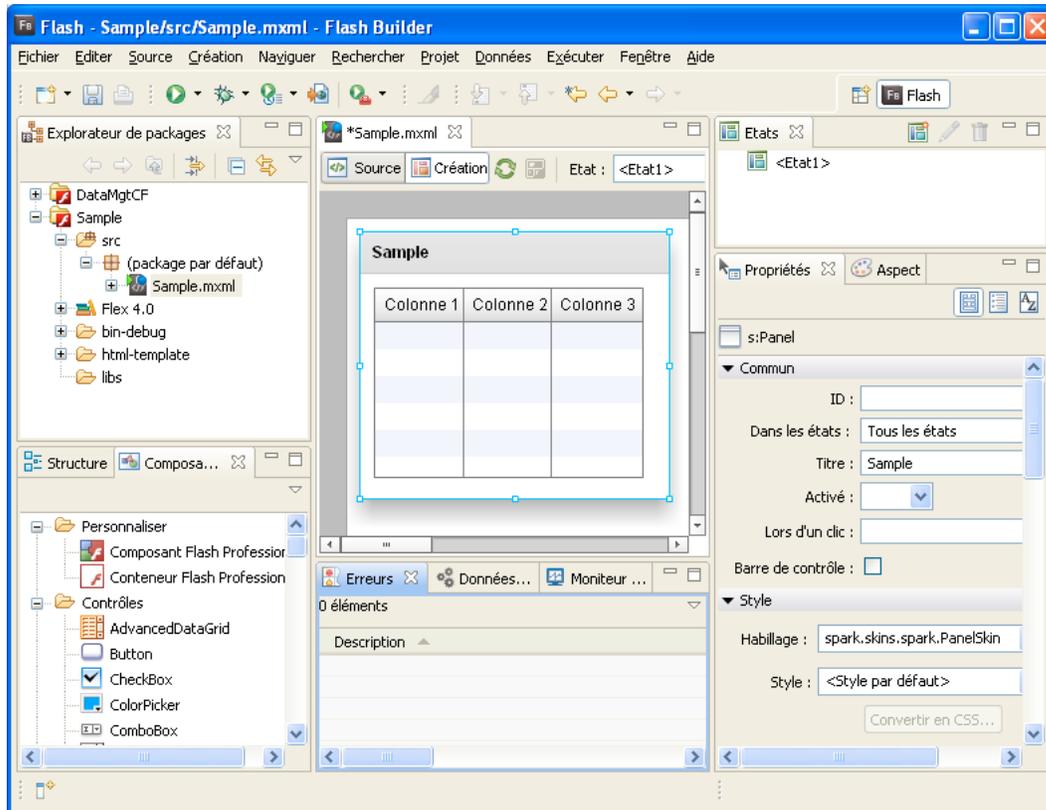
La présentation et la création visuelles des applications ont lieu en mode Création de l'éditeur MXML. Ce mode représente visuellement le code édité en mode Source. Il contient cependant, par rapport à ce dernier, des vues supplémentaires prenant en charge les tâches de création. Il s'agit des vues Composants, Propriétés, Aspect et Etats. Le mode Création comporte en outre la vue Structure qui affiche la structure MXML des applications. Pour plus d'informations sur la création d'applications Flex dans Flash Builder, voir « [Création d'une interface utilisateur avec Flash Builder](#) » à la page 188.

Remarque : le mode Création n'est pas disponible avec les projets ActionScript. Pour prévisualiser l'interface utilisateur des applications ActionScript, vous devez les créer et les exécuter. Pour plus d'informations sur l'utilisation d'ActionScript, voir « [Projets ActionScript](#) » à la page 66 et « [Exécution des applications](#) » à la page 90.

En mode Création, la perspective Développement contient l'éditeur MXML et les vues Composants, Etats, Propriétés, Aspect et Structure.

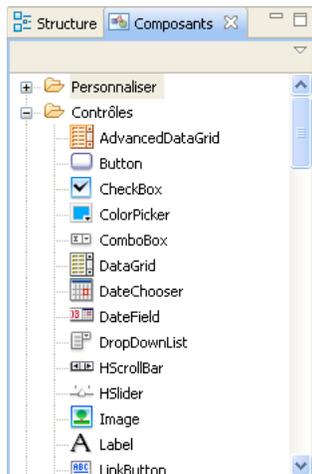
Editeur MXML

Le mode Création MXML vous permet d'interagir visuellement avec les applications. Vous pouvez par exemple faire glisser des composants vers la zone de création, sélectionner et redimensionner des composants et ainsi de suite. Toujours en mode Création, vous pouvez développer l'éditeur MXML de manière telle à pouvoir visualiser et sélectionner chaque composant individuellement. Les modes Panoramique et Zoom permettent en outre d'obtenir une vue plus détaillée des éléments, option utile notamment en présence de composants conteneur incorporés. Pour plus d'informations sur l'utilisation du mode Création, voir « [Création d'une interface utilisateur avec Flash Builder](#) » à la page 188.



Vue Composants

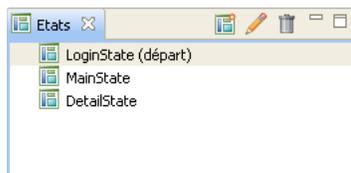
La vue Composants contient tous les composants Flex standard. Vous pouvez les sélectionner et les ajouter à la zone de création. Cette vue contient également les composants personnalisés que vous créez.



Pour plus d'informations, voir « [Ajout et modification des composants](#) » à la page 191.

Vue Etats

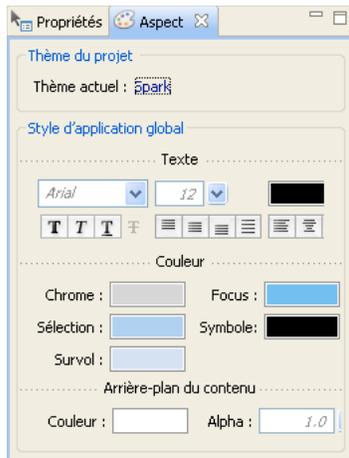
Flex vous permet de créer des applications dont l'apparence varie selon les événements déclenchés directement par l'utilisateur ou générés de façon programmée. Ces changements d'interface utilisateur sont appelés *états d'affichage*. Vous créez et gérez les états d'affichage dans la vue Etats.



Pour plus d'informations sur les états d'affichage, voir « [Ajout d'états d'affichage et de transitions](#) » à la page 227.

Vue Aspect

La vue Aspect vous permet de définir le style global de l'application pour le texte et les couleurs. Vous pouvez également y définir le thème des applications du projet.

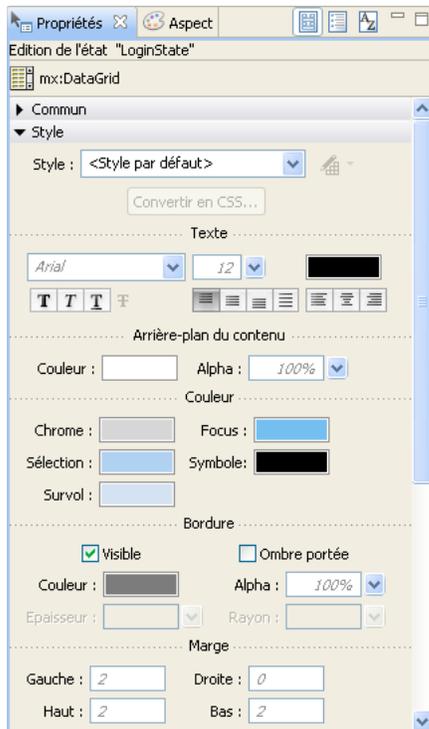


Panneau Aspect

Pour plus d'informations, voir « [Application de styles](#) » à la page 214.

Vue Style

La vue Style vous permet de définir le style des différents composants. Les styles définissables varient en fonction du composant.

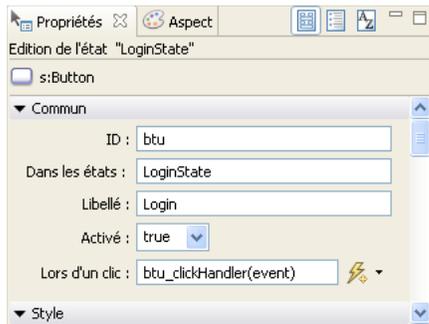


Panneau Style

Pour plus d'informations, voir « [Application de styles](#) » à la page 214.

Vue Propriétés Flex

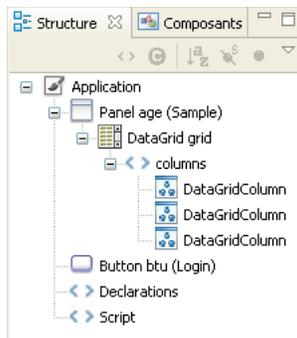
Lorsqu'un composant est sélectionné, ses propriétés s'affichent dans la vue Propriétés Flex. Vous pouvez définir et éditer ces propriétés en fonction de vos besoins. Vous pouvez afficher les propriétés d'un composant de façon graphique (comme l'illustre l'exemple suivant), par catégories ou sous forme d'une liste alphabétique.



Pour plus d'informations, voir « [Définition des propriétés des composants](#) » à la page 201.

Vue Structure

En mode Création, la vue Structure présente une vue hiérarchique de la structure MXML de vos applications. Vous pouvez facilement parcourir la structure d'une application en sélectionnant des instructions de balises MXML et des composants individuels. L'élément sélectionné dans la vue Structure est mis en surbrillance dans la vue Création.

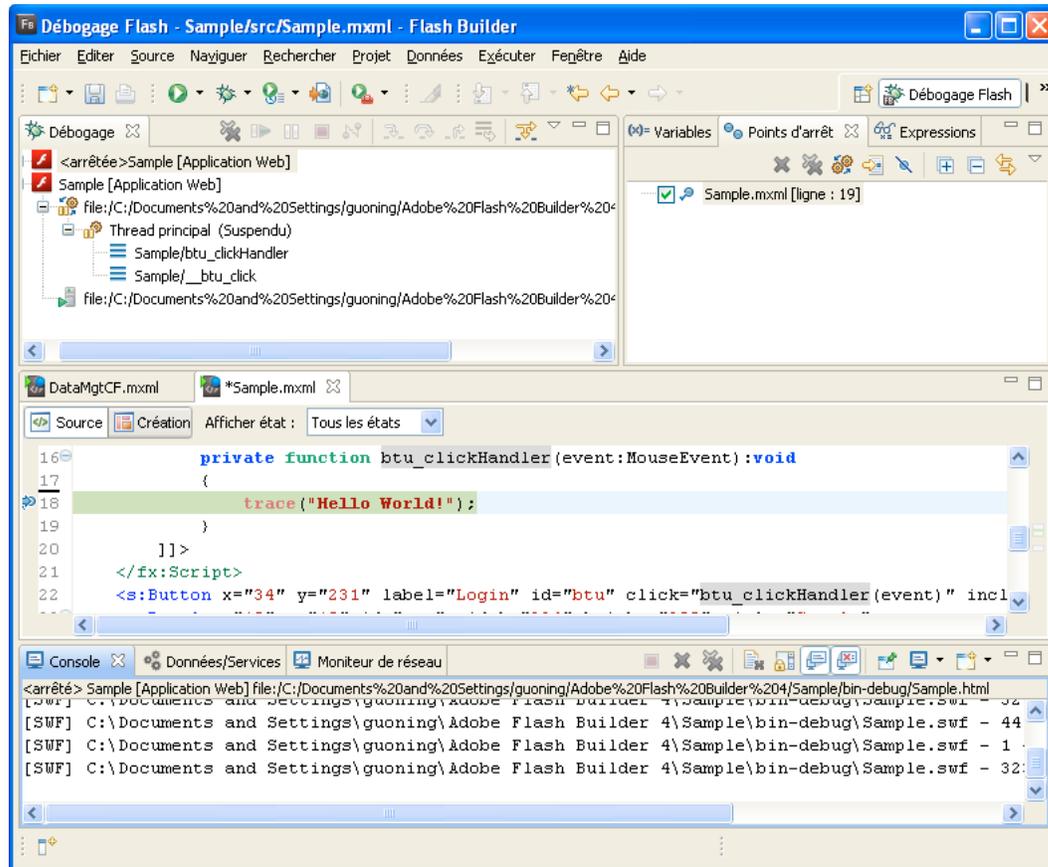


Pour plus d'informations sur l'utilisation de la vue Structure en mode Création, voir « [Examen de la structure du code MXML](#) » à la page 204.

Perspective Débogage Flash

La perspective Débogage Flash contient les outils nécessaires au débogage de vos applications. Tout comme la perspective Développement, l'éditeur en est le principal outil. Pour le débogage des applications, l'éditeur, associé aux outils correspondants, recherche et met en surbrillance les lignes de code qui doivent être corrigées avant que vous puissiez poursuivre le test de l'application.

Vous pouvez par exemple définir des points d'arrêt dans le code pour interrompre l'exécution et examiner les valeurs des variables et d'autres informations qui précèdent ce point. Vous pouvez également passer au point d'arrêt suivant ou à l'étape suivante d'un appel de fonction pour visualiser les changements de valeur des variables.

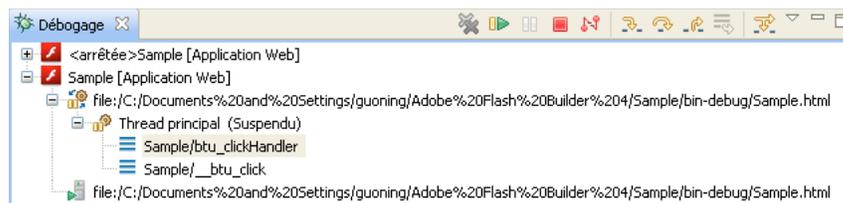


La perspective Débogage apparaît automatiquement lorsque le premier point d'arrêt est atteint. Vous pouvez également passer manuellement à la perspective Débogage en la sélectionnant dans la barre des perspectives située à l'extrémité droite de la barre d'outils principale du workbench.

La perspective Débogage contient les vues Débogage, Points d'arrêt, Console, Variables et Expressions.

Vue Débogage

La vue Débogage (dans d'autres débogueurs, cette option porte le nom *pile des appels*) affiche le frame de pile du thread suspendu de l'application que vous déboguez. Utilisez la vue Débogage pour gérer le processus de débogage. Cette vue vous permet par exemple de reprendre ou d'interrompre le thread ou une étape dans les instructions de code.



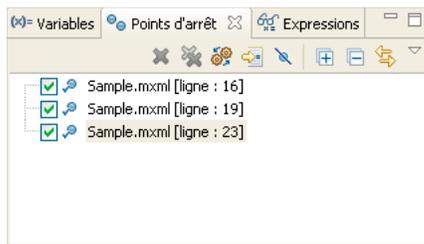
Pour plus d'informations sur l'utilisation de la vue Débogage, voir « [Gestion de la session de débogage dans la vue Débogage](#) » à la page 144.

Les applications créées avec Flex ne comportent qu'un seul thread (ce qui n'est pas le cas de Java, par exemple). Vous ne pouvez déboguer qu'une seule application à la fois. Au cours du débogage d'une application, vous ne pouvez donc afficher les processus et la vue Débogage que d'un seul thread d'exécution.

La vue Débogage affiche la liste des fonctions appelées jusqu'à ce point, dans l'ordre de leur appel. Par exemple, la première fonction appelée apparaît en bas de la liste. Vous pouvez double-cliquer sur une fonction pour la déplacer dans le script ; Flash Builder met à jour les informations de la vue Variables conformément au nouvel emplacement de la fonction dans le script.

Vue Points d'arrêt

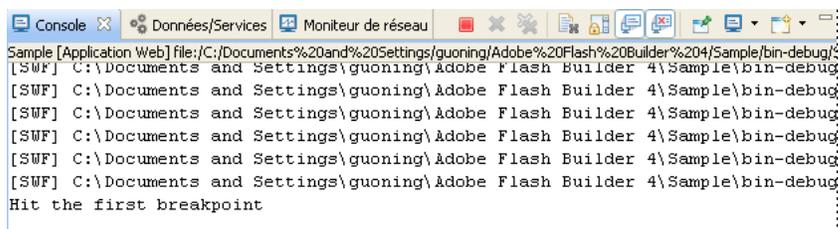
La vue Points d'arrêt répertorie tous les points d'arrêt définis dans le projet. Double-cliquez sur un point d'arrêt pour afficher son emplacement dans l'éditeur. Vous pouvez également désactiver, ignorer et supprimer des points d'arrêt.



Pour plus d'informations, voir « [Gestion des points d'arrêt dans la vue Points d'arrêt](#) » à la page 143.

Vue Console

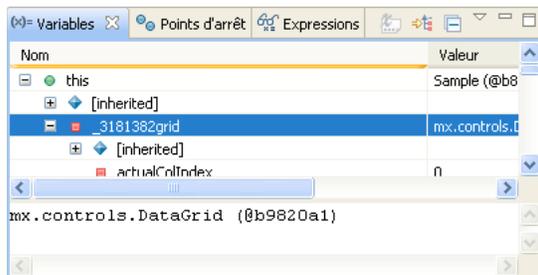
La vue Console affiche le résultat des instructions trace placées dans le code ActionScript ainsi que les informations provenant du débogueur lui-même (statut, avertissements et erreurs).



Pour plus d'informations, voir « [Utilisation de la vue Console](#) » à la page 145.

Vue Variables

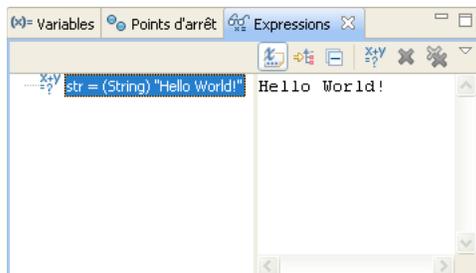
La vue Variables affiche des informations sur les variables dans un cadre de pile sélectionné. Vous pouvez sélectionner les variables à surveiller (dans la vue Expressions) et modifier leurs valeurs dans la session de débogage. Lors de la session de débogage, vous pouvez afficher les modifications effectuées dans le fichier SWF en cours d'exécution et tester des solutions pour résoudre le problème.



Pour plus d'informations, voir « [Gestion des variables dans la vue Variables](#) » à la page 145.

Vue Expressions

La vue Expressions permet de surveiller un ensemble de variables critiques. Vous pouvez choisir les variables que vous jugez critiques dans la vue Variables et les ajouter à la vue Expressions afin de les surveiller. Vous pouvez également ajouter et évaluer des expressions de contrôle.



Lorsque vous déboguez l'application, vous pouvez surveiller les variables et, si nécessaire, modifier les valeurs. Vous pouvez ajouter et supprimer des variables dans la vue Expressions. Les expressions de contrôle sont des expressions de code évaluées à chaque interruption du débogage. Pour plus d'informations, voir « [Utilisation de la vue Expressions](#) » à la page 146.

Pour plus d'informations sur le débogage des applications Flex et ActionScript, voir « [Débogage des applications](#) » à la page 141.

Perspective Profil Flash

Flash Builder Premium propose une perspective supplémentaire. Le profileur Adobe Flex permet d'identifier les problèmes de performance et les fuites de mémoire des applications. La perspective Profil affiche plusieurs panneaux (ou vues) présentant les données de profilage sous différentes formes. Lorsque vous interagissez avec l'application, le profileur enregistre les données concernant le statut de l'application, notamment le nombre d'objets, leur taille, le nombre d'appels de la méthode et le temps consacré à ces appels. Pour plus d'informations sur le profileur, voir « [A propos du profilage](#) » à la page 159.

Autres vues utiles du workbench

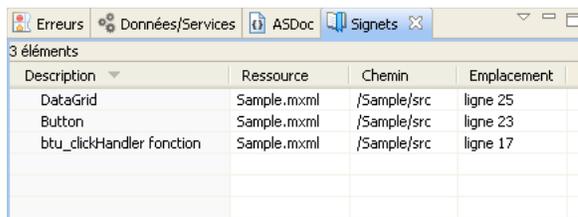
Outre les éditeurs et les vues associés aux perspectives par défaut de Flash Builder (développement, débogage et profilage), le workbench contient d'autres vues qui simplifient le processus de développement d'une application.

Vous pouvez accéder aux vues qui n'apparaissent pas déjà dans une perspective et les ajouter au workbench en sélectionnant Fenêtre > Autres vues > Généralités (Fenêtre > Afficher la vue > Autre dans la version plug-in). Ces vues optionnelles sont classées par type et associées à des fonctionnalités distinctes du workbench ou à des plug-ins Eclipse spécifiques. Pour plus d'informations sur l'utilisation des vues, voir « [Utilisation des éditeurs et des vues](#) » à la page 25.

Certaines vues du workbench en particulier vous seront très précieuses lors du développement d'applications dans Flash Builder. Il s'agit des vues Tâches, Signets et Recherche.

Vue Signets

La vue Signets sert à gérer les signets que vous ajoutez à des lignes de code spécifiques ou à des ressources. Tout comme dans un navigateur Web, les signets permettent de garder la trace d'éléments importants. La sélection d'un signet recherche et affiche ce signet dans le workbench.

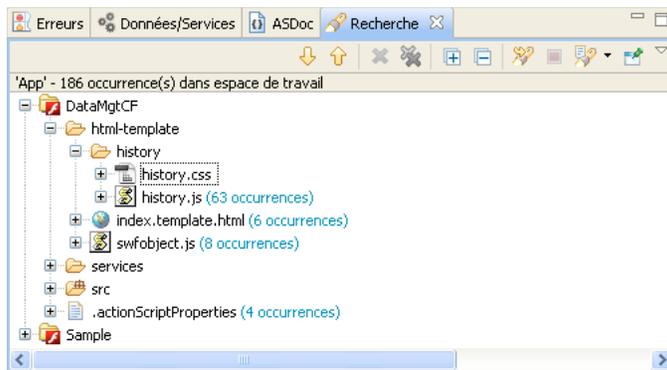


Description	Ressource	Chemin	Emplacement
DataGrid	Sample.mxml	/Sample/src	ligne 25
Button	Sample.mxml	/Sample/src	ligne 23
btu_clickHandler fonction	Sample.mxml	/Sample/src	ligne 17

Pour plus d'informations sur la vue Signets, voir « [A propos des marqueurs](#) » à la page 122.

Vue Recherche

La vue Recherche s'affiche automatiquement lorsque vous recherchez des ressources dans l'espace de travail. Elle permet de définir des recherches, accéder à des recherches précédentes et filtrer la liste des résultats de la recherche.



Pour plus d'informations sur la vue Recherche, voir « [Recherche dans le workbench](#) » à la page 32.

Menus, barres d'outils et raccourcis du workbench

Toutes les commandes du workbench apparaissent dans le système de menus, les menus contextuels, les barres d'outils et les raccourcis clavier.

Barre d'outils du workbench

Les boutons de la barre d'outils du workbench correspondent aux principales commandes ainsi qu'aux commandes les plus utilisées. Ces commandes sont également disponibles dans les différents menus de Flash Builder.



Les boutons suivants apparaissent dans la barre d'outils du workbench (de gauche à droite).

Nouveau Affiche un menu déroulant contenant tous les types de projet et tous les documents que vous pouvez créer.

Enregistrer Enregistre le document ouvert dans l'éditeur et actuellement sélectionné.

Imprimer la source Imprime le document ouvert dans l'éditeur et actuellement sélectionné.

Générer tout Apparaît lorsque l'option « Générer automatiquement » est désactivée dans le menu Projet.

Exécuter Ouvre le fichier SWF de l'application principale dans le navigateur Web par défaut ou directement dans l'application Flash Player autonome. Vous pouvez également sélectionner d'autres fichiers d'application dans le projet à partir du menu contextuel associé. Pour plus d'informations, voir « [Exécution des applications](#) » à la page 92.

Débuguer Utilise le fichier d'application principal du projet actuel pour ouvrir une session de débogage. Vous pouvez également sélectionner d'autres fichiers d'application dans le projet à partir du menu contextuel associé. Pour plus d'informations, voir « [Lancement d'une session de débogage](#) » à la page 141.

Profiler Crée, gère et exécute des configurations. Pour plus d'informations, voir « [A propos du profilage](#) » à la page 159.

Exporter vers une version validée Lance un assistant qui vous aide à sélectionner l'application pour laquelle vous souhaitez exporter une version optimisée de qualité finale.

Outils externes Sélectionne une configuration de lancement personnalisée.

Marquer les occurrences Permet de sélectionner et de marquer les occurrences de code en mode Source.

Annotation suivante Permet de sélectionner et de passer aux annotations de code suivantes en mode Source.

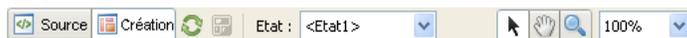
Annotation précédente Permet de sélectionner et de passer aux annotations de code précédentes en mode Source.

Emplacement de la dernière édition Renvoie à l'emplacement de la dernière modification apportée à une ressource (la ligne de code exacte ou, en mode Création, l'élément de l'interface utilisateur auquel vous avez ajouté un contrôle ou pour lequel vous avez défini une propriété, par exemple).

Précédent et Suivant Permettent d'accéder aux documents précédemment sélectionnés. Vous pouvez également sélectionner les documents actuellement ouverts dans la liste fournie par le menu contextuel associé à ces boutons.

Barre d'outils de l'éditeur MXML

Les boutons de la barre d'outils de l'éditeur MXML permettent de contrôler l'éditeur en modes Source et Création. Pour afficher la barre d'outils, ouvrez un fichier MXML en mode Création.



Les boutons suivants apparaissent dans la barre d'outils de l'éditeur MXML (de gauche à droite).

Source Affiche l'éditeur en mode Source, c'est-à-dire la zone d'édition du code.

Création Affiche l'éditeur en mode Création, c'est-à-dire la zone de présentation et de conception visuelles des applications Flex.

Actualiser Recharge les éléments visuels (images, fichiers SWF ou fichiers de classes contenant les méthodes API de dessin) définissant l'apparence de l'application. Ces éléments sont désignés collectivement par le terme *habillage*. Pour plus d'informations, voir Creating Skins.

Etat Liste déroulante répertoriant tous les états d'affichage définis. La sélection d'un état d'affichage entraîne la mise à jour de l'affichage de la création visuelle. Pour plus d'informations, voir « [Ajout d'états d'affichage et de transitions](#) » à la page 227.

Mode Sélection Activé par défaut à l'ouverture d'un fichier, ce mode vous permet de sélectionner, déplacer et redimensionner des éléments.

Mode Panoramique Permet d'effectuer un panoramique et de parcourir la zone de création. En revanche, ne permet pas de sélectionner ou de déplacer des éléments.

Mode Zoom Affiche par défaut des valeurs d'agrandissement prédéfinies. Pour effectuer un zoom arrière, appuyez sur Alt+clic (Opt+clic sous Mac). Pour revenir au mode d'affichage 100%, double-cliquez sur le bouton Mode Zoom.

Agrandissement Liste déroulante affichant des pourcentages d'agrandissement. Cette liste est également disponible dans le menu Création > Agrandissement. La liste affiche par défaut la valeur 100%.

Barre d'outils de l'éditeur CSS (uniquement Flex 3)

Si vous modifiez un fichier CSS pour un projet d'application basé sur le SDK Flex 3, l'éditeur CSS affiche le mode Création. L'éditeur CSS comporte plusieurs boutons vous permettant de contrôler l'éditeur en modes Source et Création. Pour afficher la barre d'outils de l'éditeur CSS, créez un projet basé sur le SDK Flex 3 et ouvrez un fichier CSS en mode Création.



Les boutons suivants apparaissent dans la barre d'outils CSS (de gauche à droite).

Source Affiche l'éditeur en mode Source, c'est-à-dire la zone d'édition du code.

Création Affiche l'éditeur en mode Création, c'est-à-dire la zone de présentation et de conception visuelles des applications Flex.

Actualiser Recharge les éléments visuels (images, fichiers SWF ou fichiers de classes contenant les méthodes API de dessin) définissant l'apparence de l'application. Ces éléments sont désignés collectivement par le terme *habillage*. Pour plus d'informations, voir Creating Skins..

Style Répertorie les styles contenus dans le fichier CSS.

Nouveau style Affiche la boîte de dialogue Nouveau style permettant de choisir les types de sélecteur et les composants à appliquer au nouveau style.

Supprimer le style Supprime le style sélectionné du fichier CSS.

Mode Sélection Activé par défaut à l'ouverture d'un fichier. Permet de sélectionner, de déplacer et de redimensionner des éléments.

Mode Panoramique Permet d'effectuer un panoramique et de parcourir la zone de création. En mode Panoramique, les éléments ne peuvent être ni sélectionnés ni déplacés.

Mode Zoom Affiche par défaut des valeurs d'agrandissement prédéfinies. Pour effectuer un zoom arrière, appuyez sur Alt+clic (Opt+clic sous Mac). Pour revenir au mode d'affichage 100%, double-cliquez sur le bouton Mode Zoom.

Agrandissement Affiche les pourcentages d'agrandissement. Cette liste est également disponible dans le menu Création > Agrandissement. La liste affiche par défaut la valeur 100%.

Arrière-plan Lance un sélecteur de couleurs permettant de sélectionner une couleur d'arrière-plan pour la zone d'aperçu. Ce changement de couleur ne modifie pas le fichier CSS et n'a aucune incidence sur l'application Flex lorsque vous l'exécutez.

Aperçu en tant que Ne s'affiche que lorsque la règle de style n'est associée à aucun composant MXML en particulier.

Editer les grilles d'échelle (pas affiché) Ne s'affiche que lorsque la règle de style utilise des habillages de fichier d'image.

Utilisation des raccourcis clavier

De nombreuses opérations accessibles à partir du système de menus de Flash Builder sont également disponibles sous forme de raccourcis clavier.

Affichage de la liste de tous les raccourcis clavier de Flash Builder

❖ Sélectionnez Aide > Assistant de touches.

Vous pouvez utiliser l'option Assistant de touches comme référence pour tous les raccourcis clavier Flash Builder, ou exécuter ces commandes directement à partir du panneau Assistant de touches en double-cliquant sur les commandes. Vous pouvez également modifier les raccourcis clavier ou créer vos propres raccourcis. Pour plus d'informations, voir « [Modification des raccourcis clavier](#) » à la page 30.

Extension du workbench Flash Builder

Flash Builder se structure autour d'un ensemble de plug-ins Eclipse fournissant les outils nécessaires à la création d'applications Flex et ActionScript 3.0. Cette structure permet aux plug-ins de proposer des points d'extension pouvant être utilisés pour étendre les fonctionnalités et les capacités de l'outil. Pour plus d'informations, voir [Référence d'extensibilité d'Adobe Flash Builder](#).

Navigation et personnalisation du workbench Flash Builder

Le terme *workbench* désigne l'environnement de développement Flash Builder. Le workbench contient trois éléments principaux : les perspectives, les éditeurs et les vues. L'utilisation associée de ces trois composants varie en fonction du stade de développement de l'application. Le workbench constitue le conteneur de tous les outils de développement des applications.

Remarque : pour plus d'informations sur certaines fonctionnalités du workbench Eclipse, voir le guide de l'utilisateur du workbench Eclipse à l'adresse <http://help.eclipse.org/help31/index.jsp>.

Utilisation des perspectives

Les perspectives comportent des combinaisons de vues et d'éditeurs adaptées à l'exécution de certains groupes de tâches. Vous utiliserez ainsi généralement la perspective Débogage Flash pour déboguer une application.

Pour une présentation des perspectives, voir « [A propos des perspectives de Flash Builder](#) » à la page 7.

Ouverture et changement de perspectives

L'ouverture d'un fichier associé à une perspective particulière conduit à l'affichage automatique de cette perspective. La configuration autonome de Flash Builder propose trois perspectives :

- Développement Flash
- Débogage Flash
- Profil Flash

La perspective Profil Flash est disponible dans Flash Builder Premium.

❖ Sélectionnez Fenêtre > Perspective ou choisissez Autre pour afficher toutes les autres perspectives Eclipse (dans la configuration plug-in de Flash Builder, sélectionnez Fenêtre > Ouvrir la perspective).

Vous pouvez également cliquer sur le bouton Ouvrir la perspective dans l'angle supérieur droit de la fenêtre du workbench afin de sélectionner une perspective dans le menu contextuel.

Pour afficher la liste complète des perspectives, sélectionnez Autre dans le menu contextuel Ouvrir la perspective.

Le titre de la perspective que vous ouvrez remplace celui de la perspective précédemment affichée. Une icône apparaît en regard du titre. Elle permet de parcourir rapidement les perspectives dans la même fenêtre. Par défaut, les perspectives s'affichent dans la même fenêtre.

Définition de la perspective par défaut

La perspective par défaut est signalée par le terme *par défaut* affiché entre parenthèses après le nom de la perspective.

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Général > Perspectives.
- 2 Sous Perspectives disponibles, sélectionnez la perspective à définir par défaut, puis cliquez sur Perspective par défaut.
- 3 Cliquez sur OK.

Ouverture des perspectives dans une nouvelle fenêtre

Vous pouvez ouvrir les perspectives dans une nouvelle fenêtre.

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Général > Perspectives.
- 2 Sous Ouverture d'une nouvelle perspective, sélectionnez Dans une nouvelle fenêtre.
Pour rétablir les valeurs par défaut, sélectionnez Dans la même fenêtre.
- 3 Cliquez sur OK.

Personnalisation d'une perspective

Pour modifier la présentation d'une perspective, changez les éditeurs et les vues visibles dans une perspective donnée. Par exemple, vous pouvez afficher la vue Signets dans une perspective et la masquer dans une autre.

Vous pouvez également configurer d'autres aspects d'une perspective, notamment les sous-menus Fichier > Nouveau, Fenêtre > Perspective > Autre, Fenêtre > Autres vues ainsi que les jeux d'actions (boutons et options de menu) qui apparaissent dans la barre d'outils et dans les éléments du menu principal (les noms des menus varient légèrement dans la configuration plug-in de Flash Builder).

Création d'une perspective

- 1 Ouvrez une perspective existante.
- 2 Affichez les vues et les éditeurs de votre choix.

Pour plus d'informations, voir « [Ouverture des vues](#) » à la page 26 et « [Ouverture de fichiers à éditer](#) » à la page 28.

- 3 Sélectionnez Fenêtre > Perspective > Sauvegarder la perspective sous (Fenêtre > Sauvegarder la perspective sous dans la configuration plug-in de Flash Builder).
- 4 Dans la boîte de dialogue Sauvegarder la perspective sous, entrez le nouveau nom de la perspective, puis cliquez sur OK.

Configuration d'une perspective

- 1 Ouvrez la perspective à configurer.
- 2 Sélectionnez Fenêtre > Perspective > Personnaliser la perspective (Fenêtre > Personnaliser la perspective dans la configuration plug-in de Flash Builder).
- 3 Sélectionnez les onglets Raccourcis ou Commandes en fonction des éléments que vous souhaitez ajouter à la perspective personnalisée.
- 4 Cochez les cases correspondant aux éléments que vous souhaitez afficher dans les menus et dans les barres d'outils de la perspective sélectionnée.
- 5 Cliquez sur OK.
- 6 Sélectionnez Fenêtre > Perspective > Sauvegarder la perspective sous (Fenêtre > Sauvegarder la perspective sous dans la configuration plug-in de Flash Builder).
- 7 Dans la boîte de dialogue Sauvegarder la perspective sous, entrez le nouveau nom de la perspective, puis cliquez sur OK.
Flash Builder ajoute le nom de la perspective enregistrée au menu Fenêtre > Perspective (Fenêtre > Ouvrir la perspective dans la configuration plug-in de Flash Builder).

Suppression d'une perspective personnalisée

Vous pouvez supprimer des perspectives définies précédemment. Vous ne pouvez pas supprimer une perspective que vous n'avez pas créée.

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Général > Perspectives.
- 2 Sous Perspectives disponibles, sélectionnez la perspective que vous souhaitez supprimer.
- 3 Cliquez sur Supprimer, puis sur OK.

Réinitialisation des perspectives

Vous pouvez rétablir la présentation d'origine d'une perspective que vous avez modifiée.

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Général > Perspectives.
- 2 Sous Perspectives disponibles, sélectionnez la perspective à réinitialiser.
- 3 Cliquez sur Réinitialiser, puis sur OK.

Utilisation des éditeurs et des vues

La plupart des perspectives du workbench se composent d'un éditeur ainsi que d'une ou de plusieurs vues. Un éditeur est un composant visuel du workbench, généralement utilisé pour éditer ou explorer une ressource. Les vues sont également des composants visuels de l'espace de travail. Elles prennent en charge les éditeurs, offrent des alternatives de présentation des éléments sélectionnés dans l'éditeur et permettent de parcourir les informations du workbench.

Pour une présentation des éditeurs et des vues, voir « [A propos du workbench](#) » à la page 4.

Ouverture des vues

Les perspectives contiennent des combinaisons prédéfinies de vues et d'éditeurs. Vous pouvez également ouvrir des vues qui ne figurent pas dans la perspective active.

- ❖ Sélectionnez Fenêtre et choisissez une vue Flash Builder ou choisissez Fenêtre > Autres vues pour choisir d'autres vues du workbench Eclipse (dans la configuration plug-in de Flash Builder, sélectionnez Fenêtre > Affichage d'une vue).

Après avoir ajouté une vue à la perspective active, vous pouvez l'enregistrer en tant que partie de la perspective. Pour plus d'informations, voir « [Personnalisation d'une perspective](#) » à la page 24.

Vous pouvez également créer des vues rapides qui facilitent l'accès aux vues fréquemment utilisées. Pour plus d'informations, voir « [Création et utilisation de vues rapides](#) » à la page 27.

Déplacement et ancrage des vues

Vous pouvez déplacer les vues à l'intérieur du workbench, les ancrer à différents emplacements ou en annuler l'ancrage, en fonction de vos besoins.

- 1 Déplacez la vue vers l'emplacement souhaité en faisant glisser sa barre de titre.

Lorsque vous déplacez la vue dans le workbench, le curseur prend la forme d'une flèche noire pointant vers l'emplacement où la vue sera ancrée une fois le bouton de la souris relâché.

 Vous pouvez faire glisser un groupe de vues empilées en cliquant sur l'espace vide situé à droite des onglets des vues et en déplaçant la souris.

Vous pouvez également déplacer une vue à l'aide de son menu contextuel. Ouvrez le menu contextuel à partir de l'onglet de la vue, sélectionnez Déplacer > Vue, déplacez la vue vers l'emplacement souhaité, puis cliquez de nouveau avec le bouton de la souris.

- 2 (Facultatif) Enregistrez vos modifications en sélectionnant Fenêtre > Perspective > Sauvegarder la perspective sous (Fenêtre > Sauvegarder la perspective sous dans la configuration plug-in de Flash Builder).

Réorganisation des onglets des vues

Outre l'ancrage des vues à différents emplacements du workbench, vous pouvez réorganiser les onglets des vues à l'intérieur d'un groupe de vues.

- ❖ Cliquez sur l'onglet de la vue à déplacer, faites glisser la vue vers l'emplacement souhaité, puis relâchez le bouton de la souris. Un symbole représentant une pile apparaît lorsque vous faites glisser l'onglet d'une vue sur d'autres onglets.

Changement de vues

Le passage d'une vue à l'autre peut se faire de différentes façons :

- Cliquez sur l'onglet de la vue à laquelle vous souhaitez accéder.
- Sélectionnez une vue dans le menu Fenêtre de Flash Builder.
- Utilisez un raccourci clavier.

Utilisez Ctrl+F7 sous Windows ou Commande+F7 sous Macintosh. Appuyez sur la touche F7 pour sélectionner une vue.

 Pour obtenir une liste de tous les raccourcis clavier, cliquez sur Aide > Assistant de touches.

Création et utilisation de vues rapides

Les vues rapides sont des vues masquées que vous pouvez ouvrir et fermer rapidement. Elles sont identiques aux autres vues à ceci près qu'elles n'encombrent pas le workbench.

Cliquez sur l'icône de vue rapide dans la barre de raccourcis pour afficher la vue. Cliquez à l'extérieur de la vue rapide (ou sur le bouton Réduire de sa barre d'outils) pour la masquer à nouveau.

Remarque : si vous convertissez la vue Explorateur de packages en vue rapide, puis ouvrez un fichier à partir de la vue rapide Explorateur de packages, la vue est automatiquement masquée pour vous permettre d'utiliser ce fichier.

Création d'une vue rapide

- ❖ Faites glisser la vue que vous souhaitez convertir en vue rapide vers la barre de raccourcis située dans l'angle inférieur gauche de la fenêtre du workbench.

L'icône de la vue que vous avez déplacée apparaît dans la barre de raccourcis. Cliquez sur cette icône pour ouvrir la vue. Cliquez à l'extérieur de la vue pour la masquer.

Restauration d'une vue rapide en vue normale

- ❖ Dans le menu contextuel de la vue, désélectionnez Vue rapide.

Filtrage des vues Tâches et Erreurs

Vous pouvez filtrer les tâches ou les problèmes qui s'affichent dans les vues Tâches ou Erreurs. Vous pouvez par exemple limiter l'affichage aux problèmes consignés par le workbench ou aux tâches que vous avez consignées comme rappels. Vous pouvez filtrer les éléments en fonction de la ressource ou du groupe de ressources auquel ils sont associés, de la chaîne de texte affichée dans le champ Description, de la gravité du problème, de la priorité de la tâche ou du statut de la tâche.

- 1 Dans la barre des tâches de la vue Tâches ou Erreurs, cliquez sur Filtrer.
- 2 Renseignez la boîte de dialogue Filtres, puis cliquez sur OK.

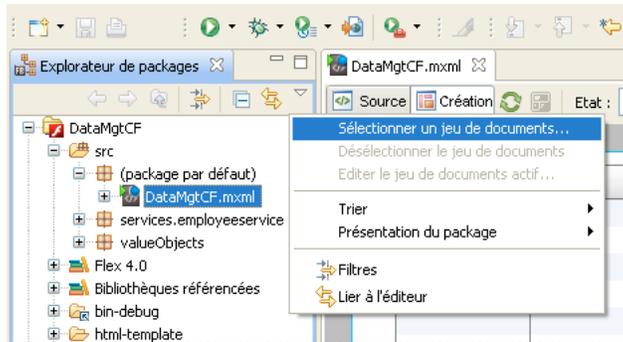
Pour plus d'informations sur les vues, voir « [Notions de base du workbench Flash Builder](#) » à la page 4.

Création de jeux de documents

Si votre espace de travail contient de nombreux projets, vous pouvez créer un jeu de documents afin de regrouper un certain nombre de projets. Vous pouvez alors afficher des jeux de documents différents dans les vues Explorateur de packages et Tâches ainsi que limiter la portée de vos recherches à des jeux de documents (et éviter donc les recherches portant sur l'ensemble de l'espace de travail).

Création d'un jeu de documents

- 1 Dans la vue Explorateur de packages, ouvrez le menu de la barre d'outils et choisissez Sélectionner un jeu de documents.



- 2 Sélectionnez Nouveau.

Flash Builder propose deux types de jeu : les points d'arrêt (utilisés lors du débogage) et les ressources.

- 3 Sélectionnez le type Ressource, puis cliquez sur Suivant.

- 4 Entrez le nom du jeu de documents, puis choisissez les projets de l'espace de travail que vous souhaitez inclure dans le jeu.

- 5 Cliquez sur Terminer.

Le jeu de documents est aussitôt appliqué à la vue Explorateur de packages et seuls les projets et les ressources qui figurent dans le jeu s'affichent.

Affichage de tous les projets dans l'espace de travail

- ❖ Dans la vue Explorateur de packages, ouvrez le menu de la barre d'outils et choisissez Désélectionner le jeu de documents.

Ouverture de fichiers à éditer

L'ouverture d'un fichier lance un éditeur dans lequel il peut être modifié.

- ❖ Effectuez l'une des opérations suivantes :

- Dans le menu contextuel du fichier (quelle que soit la vue de navigation dans laquelle il se trouve), sélectionnez Ouvrir.
- Double-cliquez sur le fichier dans l'une des vues de navigation.
- Double-cliquez sur le signet associé au fichier dans la vue Signets.
- Double-cliquez sur un avertissement d'erreur ou un enregistrement de tâche associé au fichier dans la vue Erreurs.

Le fichier s'ouvre dans l'éditeur défini par défaut pour ce type de fichier. Pour ouvrir le fichier dans un autre éditeur, sélectionnez Ouvrir avec dans le menu contextuel du fichier. Sélectionnez l'éditeur que vous souhaitez utiliser.

Association d'éditeurs à des types de fichier

Le workbench permet d'associer les éditeurs à différents types de fichiers.

- 1 Sélectionnez Fenêtre > Préférences.
- 2 Cliquez sur le signe plus pour développer la catégorie Général.

- 3 Cliquez sur le signe plus pour développer la catégorie Editeurs, puis sélectionnez Associations de fichiers.
- 4 Sélectionnez un type de fichier dans la liste Types de fichier.
Pour ajouter un type de fichier à la liste, cliquez sur Ajouter, entrez le nouveau type de fichier dans la boîte de dialogue Ajouter un type de fichier, puis cliquez sur OK.
- 5 Dans la liste Editeurs associés, sélectionnez l'éditeur à associer à ce type de fichier.
Pour ajouter à la liste un éditeur interne ou externe, cliquez sur Ajouter et renseignez la boîte de dialogue.
- 6 Cliquez sur OK.

 Vous pouvez modifier les préférences par défaut de l'éditeur à partir du menu contextuel de toute ressource se trouvant dans l'une des vues de navigation. Dans le menu contextuel, sélectionnez Ouvrir avec.

Edition de fichiers en dehors du workbench

Vous pouvez éditer un fichier MXML ou ActionScript dans un éditeur externe puis l'utiliser dans Flash Builder. Le workbench effectue les opérations de création ou d'actualisation nécessaires afin d'appliquer les modifications que vous avez apportées au fichier en dehors du workbench.

Actualisation d'un fichier MXML ou ActionScript édité en dehors du workbench

- 1 Editez le fichier MXML ou ActionScript dans l'éditeur externe de votre choix.
- 2 Enregistrez et fermez le fichier.
- 3 Démarrez Flash Builder.
- 4 Dans l'une des vues de navigation du workbench, ouvrez le menu contextuel, puis sélectionnez Actualiser.

 Si vous utilisez régulièrement des éditeurs externes, vous pouvez activer l'actualisation automatique. Pour ce faire, sélectionnez Fenêtre > Préférences, développez la catégorie Général, sélectionnez Espace de travail, puis cochez la case Actualiser automatiquement. Lorsque vous activez cette option, le workbench consigne toutes les modifications externes apportées au fichier. La vitesse de l'opération dépend de votre plateforme.

Mosaïque d'éditeurs

Le workbench vous permet d'ouvrir plusieurs fichiers dans différents éditeurs. Mais contrairement aux vues, les éditeurs ne peuvent pas être déplacés à l'extérieur du workbench pour créer des fenêtres. Vous pouvez cependant disposer les éditeurs en mosaïque dans la zone d'édition afin d'afficher les fichiers source côte à côte.

- 1 Ouvrez deux ou plusieurs fichiers dans la zone d'édition.
- 2 Cliquez sur l'onglet de l'un des éditeurs.
- 3 Faites-le glisser vers le bord gauche, droit, supérieur ou inférieur de la zone d'édition.

Le pointeur prend la forme d'une flèche noire pointant vers l'emplacement où la vue sera ancrée une fois le bouton de la souris relâché.

- 4 (Facultatif) Faites glisser les bords des éditeurs pour les redimensionner, si besoin est.

Agrandissement d'une vue ou d'un éditeur

Vous pouvez procéder de différentes façons pour agrandir une vue ou un éditeur de manière à occuper entièrement la fenêtre du workbench.

Agrandissement (restauration) d'une vue ou d'un éditeur

- Accédez au menu contextuel de la barre de titre de la vue ou de l'éditeur et sélectionnez Agrandir (Restaurer).

- Double-cliquez sur l'onglet de la vue.
- Dans le menu Fenêtre de Flash Builder, sélectionnez Agrandir/Rétablir l'éditeur.
- Cliquez sur les icônes Agrandir/Restaurer situées dans l'angle supérieur droit de la vue ou de l'éditeur.

Changement d'espace de travail

Vous ne pouvez travailler que dans un seul espace de travail à la fois. Lorsque vous installez et exécutez Flash Builder pour la première fois, un message vous invite à créer un espace de travail qui devient l'espace de travail par défaut. Vous pouvez créer d'autres espaces de travail et passer de l'un à l'autre en sélectionnant l'espace de travail voulu au lancement de Flash Builder ou en choisissant Fichier > Commuter l'espace de travail.

Personnalisation du workbench

Vous pouvez personnaliser le workbench en fonction de vos besoins. Vous pouvez par exemple définir l'affichage des éléments dans la barre d'outils principale, créer des raccourcis clavier ou modifier les polices et les couleurs de l'interface utilisateur.

Réorganisation de la barre d'outils principale

Flash Builder vous permet de réorganiser les sections de la barre d'outils principale. Les sections de la barre d'outils principale sont séparées par un espace.

- 1 Vérifiez que la barre d'outils est déverrouillée. Dans le menu contextuel de la barre d'outils, désélectionnez l'option Verrouiller les barres d'outils.
- 2 Positionnez le pointeur de la souris sur la ligne verticale qui délimite sur la gauche la section à déplacer.
- 3 Cliquez sur cette ligne et déplacez la section vers la gauche, la droite, le haut ou le bas. Relâchez le bouton de la souris pour positionner la section à son nouvel emplacement.

 *Afin d'éviter tout déplacement non intentionnel des barres d'outils, verrouillez-les de nouveau en activant l'option correspondante de leur menu contextuel.*

Modification des raccourcis clavier

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Général > Touches.
- 2 Dans l'écran Vue de la boîte de dialogue Touches, sélectionnez la commande à modifier.
- 3 Dans le champ Liaison, entrez le nouveau raccourci clavier à lier à la commande.
- 4 Dans la liste déroulante Quand, sélectionnez à quel moment le raccourci clavier doit être activé.
- 5 Cliquez sur Appliquer ou sur OK.

 *Pour obtenir une liste de tous les raccourcis clavier, cliquez sur Aide > Assistant de touches.*

Modification des polices et des couleurs

Par défaut, le workbench utilise les polices et les couleurs du système d'exploitation de l'ordinateur. Vous pouvez cependant personnaliser les polices et les couleurs de différentes manières. Le workbench permet de configurer les polices suivantes.

Police de bannière Apparaît dans la barre de titre de nombreux assistants. Par exemple, l'assistant Nouveau projet Flex utilise la police de la bannière pour le titre supérieur.

Police de boîte de dialogue Apparaît dans les widgets et les boîtes de dialogue.

Police d'en-tête Apparaît dans l'en-tête de section.

Police de texte Apparaît dans les éditeurs de texte.

Police de console CVS Apparaît dans la console CVS.

Police des ressources ignorées Affiche les ressources ignorées par la console CVS.

Police des modifications sortantes Affiche les modifications sortantes dans CVS.

Police de la console (Police du texte, par défaut) Apparaît dans la console Déboguer.

Police du texte des volets Détails (Police du texte, par défaut) Apparaît dans les volets Détails des vues Débogage.

Police de la table de la vue Mémoire (Police du texte, par défaut) Apparaît dans le tableau de la vue Mémoire.

Police de texte de l'éditeur Java (Police du texte, par défaut) Apparaît dans les éditeurs Java.

Police du texte de l'éditeur de fichier de propriétés (Police du texte, par défaut) Apparaît dans les éditeurs de fichier de propriétés.

Police du texte de comparaison (Police du texte, par défaut) Apparaît dans les outils de comparaison textuelle ou de fusion.

Police du texte de la comparaison Java (Police du texte, par défaut) Apparaît dans les outils de comparaison ou de fusion Java.

Police du texte de comparaison de fichiers de propriétés Java (Police du texte de l'éditeur de fichier de propriétés, par défaut) Apparaît dans les outils de comparaison ou de fusion de fichier de propriétés Java.

Police du titre de partie (Police du texte de l'éditeur de fichier de propriétés, par défaut) Apparaît dans les titres des vues et des éditeurs.

Police du message de la vue (Police du texte de l'éditeur du fichier des propriétés, par défaut) Affiche les messages dans la barre de titre de la vue (le cas échéant).

Les plug-ins qui utilisent d'autres polices peuvent également comporter des préférences personnalisables. Le plug-in Java Development Tools permet par exemple de définir la police utilisée par l'éditeur Java (dans Préférences, sélectionnez > Général > Apparence > Couleurs et polices > Java > Police de texte de l'éditeur Java).

Certaines zones (l'arborescence de l'Explorateur de packages, par exemple) sont toujours affichées dans la police du système d'exploitation. Pour modifier la police de ces zones, utilisez les outils de configuration fournis par le système d'exploitation (les propriétés d'affichage du Panneau de configuration de Windows, par exemple).

Modification des polices et des couleurs

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Général > Apparence > Couleurs et polices.
- 2 Développez les catégories Comparaison de texte, Couleurs et polices de base, CVS, Déboguer ainsi que Dossiers des éditeurs et des vues pour rechercher et sélectionner la police et les couleurs à modifier.

Remarque : vous pouvez également cliquer sur Utiliser la police du système au lieu de Modifier pour laisser le système d'exploitation choisir une valeur de police appropriée. Sous Windows par exemple, si vous sélectionnez cette option, Flash Builder utilise la police sélectionnée dans les propriétés d'affichage du Panneau de configuration de Windows.

- 3 Définissez vos préférences en matière de police et de couleur.

Modification des couleurs

Le workbench fait appel à différentes couleurs pour distinguer les éléments, par exemple les textes d'erreur et les liens hypertexte. Il utilise les mêmes couleurs que le système d'exploitation. Pour les modifier, utilisez dans ce cas aussi les outils de configuration fournis par le système d'exploitation (les propriétés d'affichage du Panneau de configuration de Windows, par exemple).

Modification des couleurs

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Général > Apparence > Couleurs et polices.
- 2 Développez les catégories Comparaison de texte, Couleurs et polices de base, CVS, Déboguer ainsi que Dossiers des éditeurs et des vues pour rechercher et sélectionner la couleur à modifier.
- 3 Cliquez sur la barre de couleurs à droite pour ouvrir le sélecteur de couleurs.
- 4 Sélectionnez une nouvelle couleur.

Contrôle du clic et du double-clic

Vous pouvez définir la réaction du workbench à un clic ou à un double-clic.

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Général.
- 2 Dans la section Mode d'ouverture, sélectionnez les options voulues et cliquez sur OK.

Recherche dans le workbench

Flash Builder intègre un outil de recherche qui vous permet de localiser rapidement des ressources. Pour plus d'informations sur la recherche de texte dans un fichier particulier, voir « [Recherche et remplacement de texte dans l'éditeur](#) » à la page 118.

Recherche de fichiers

Flash Builder vous permet d'effectuer des recherches complexes de fichiers.

- ❖ Dans la version plug-in de Flash Builder, sélectionnez Rechercher > Rechercher ou Rechercher > Fichier.

Dans la version autonome de Flash Builder, sélectionnez Editer > Rechercher dans les fichiers.

Remarque : cliquez sur Personnaliser pour définir les types d'onglet de recherche disponibles dans la boîte de dialogue Rechercher.

Recherche de références et de déclarations

Flash Builder présente des fonctions de recherche avancées plus puissantes que les fonctions de recherche/remplacement. Afin de faciliter la compréhension de la manière dont les fonctions, variables et autres identifiants sont utilisés, Flash Builder permet de rechercher et de repérer les références ou les déclarations vers les identifiants dans les fichiers ActionScript ou MXML, les projets ou les espaces de travail. Pour plus d'informations, voir « [Recherche de références et restructuration de code](#) » à la page 118.

Utilisation de la vue Recherche

La vue Recherche affiche les résultats de votre recherche.

Ouverture d'un fichier de la liste

- ❖ Double-cliquez sur le fichier.

Suppression d'un fichier de la liste

- ❖ Sélectionnez le fichier à supprimer, puis cliquez sur Supprimer les occurrences sélectionnées.

Suppression de tous les fichiers de la liste

- ❖ Cliquez sur Supprimer toutes les occurrences.

Navigation entre les fichiers correspondants

- ❖ Cliquez sur Afficher l'occurrence suivante ou Afficher l'occurrence précédente.

Affichage des recherches précédentes

- ❖ Cliquez sur la flèche en regard de l'option Afficher les recherches précédentes et sélectionnez une recherche dans la liste déroulante.

Retour à la vue Recherche après sa fermeture

- 1 Sélectionnez Fenêtre > Autres vues > Généralités (Fenêtre > Afficher la vue > Autre dans la configuration plug-in de Flash Builder).
- 2 Développez la catégorie Généralités, sélectionnez Recherche, puis cliquez sur OK.

Utilisation des modes Source et Création de l'éditeur

L'éditeur MXML de Flash Builder vous permet d'utiliser les modes Source ou Création. Flash Builder permet également de créer une vue partagée dans laquelle vous pourrez afficher les deux modes simultanément.

Affichage du fichier en mode Création

- ❖ Cliquez sur Création en haut de la zone de l'éditeur.

Affichage du fichier en mode Source

- ❖ Cliquez sur Source en haut de la zone de l'éditeur.

Utilisation simultanée des modes Source et Création

- 1 Dans le menu contextuel de l'onglet de l'éditeur, sélectionnez Nouvel éditeur.
Vous disposez à présent de deux onglets d'édition pour le même fichier.
- 2 Faites glisser l'un des onglets vers la droite pour le juxtaposer à l'autre onglet.
- 3 Configurez l'un des éditeurs en mode Création et l'autre éditeur en mode Source.

Commutation des modes Source et Création

- ❖ Appuyez sur Ctrl+' (guillemet ouvrant).

Accès aux raccourcis clavier

Les raccourcis clavier disponibles dans Flash Builder dépendent de nombreux facteurs, y compris de la vue ou de l'éditeur sélectionné, de l'ouverture ou non d'une boîte de dialogue, des plug-ins installés et du système d'exploitation. Vous pouvez obtenir à tout moment la liste des raccourcis clavier disponibles à l'aide de la fonction Assistant de touches.

- ❖ Sélectionnez Aide > Assistant de touches.

Définition des préférences du workbench

Vous pouvez définir les préférences de nombreux aspects du workbench. Vous pouvez par exemple configurer Flash Builder de manière telle à ce qu'il vous invite à sélectionner l'espace de travail au démarrage, définir l'éditeur à utiliser à l'ouverture de certains types de ressource et paramétrer différentes options pour l'exécution et le débogage de vos applications.

Vos préférences Flash Builder s'appliquent uniquement à l'espace de travail actuel. Vous pouvez cependant exporter les préférences du workbench et les importer dans un autre espace de travail. L'exportation des préférences est utile si vous gérez plusieurs espaces de travail ou si vous souhaitez partager les préférences du workbench avec d'autres membres de votre équipe de développement.

Vous pouvez également définir des préférences individuelles pour chaque projet d'un espace de travail, par exemple définir des options de compilation ou de débogage distinctes pour chacun de vos projets Flex.

Définition des préférences du workbench Flash Builder

- 1 Ouvrez la fenêtre Préférences.
- 2 Développez Général, choisissez l'une des catégories de préférences du workbench et modifiez-la si nécessaire.
- 3 Cliquez sur OK.

Chapitre 3 : Utilisation des projets

Adobe® Flash® Builder™ vous permet de créer, gérer, assembler et distribuer des projets pour la création d'applications pour le Web et de bureau. Lorsque vous générez des fichiers SWC (shared component library), vous pouvez partager des composants et d'autres ressources entre diverses applications ou avec d'autres développeurs. Vous pouvez également utiliser différentes versions du SDK Flex d'Adobe directement dans Flash Builder.

A propos des projets Flash Builder

Flash Builder adopte une approche classique du développement de logiciels : le regroupement des ressources (dossiers et fichiers) qui constituent une application au sein d'un conteneur appelé *projet*. Un projet contient un ensemble de propriétés qui contrôlent la façon dont l'application est générée, l'endroit où l'application construite est hébergée, la façon dont le débogage est géré et les relations avec d'autres projets dans l'espace de travail.

La vue Explorateur de packages vous permet de gérer les projets, en ajoutant, modifiant et supprimant des ressources. Vous pouvez également fermer les projets à l'intérieur d'un espace de travail, importer des ressources et créer des liens vers des ressources externes.

Outre les projets Flex, Flash Builder offre un type de projet de base appelé projet *ActionScript*. Grâce à un projet ActionScript, vous pouvez coder et déboguer des applications ActionScript qui accèdent directement aux API d'Adobe Flash Player et sont compilées dans des fichiers SWF. Les projets ActionScript ne font pas appel à la structure Flex ni au langage MXML.

Applications déployées dans Flash Player

Faites appel à l'assistant Nouveau projet Flex pour créer des applications pouvant être déployées dans Flash Player. A la création du projet, sélectionnez l'option Application Web (exécutée dans Adobe Flash Player). Ce type d'application est compilé dans des fichiers SWF autonomes. Pour plus d'informations, voir « [Utilisation des projets](#) » à la page 35 et « [Projets ActionScript](#) » à la page 66.

Applications déployées dans Adobe AIR

Faites appel à l'assistant de création d'un projet Flex pour créer des applications pouvant être déployées dans Adobe® AIR®. Sous Type d'applications, sélectionnez l'option Application de bureau (exécution dans Adobe AIR). Utilisez la fonction Exporter vers une version validée pour générer un package AIR de qualité finale prêt à être installé. Pour plus d'informations, voir « [Développement d'applications AIR avec Flash Builder](#) » à la page 137.

Flash Builder vous permet de déboguer, assembler et gérer des projets AIR. Il vous permet également d'exécuter des applications dans AIR.

Adobe AIR Marketplace est un endroit où les développeurs d'AIR publient des applications AIR que les utilisateurs peuvent télécharger. Pour trouver Marketplace, aller à www.adobe.com/go/marketplace. Si vous avez des questions sur Adobe AIR Marketplace, aller à www.adobe.com/go/marketplace_faq.

Projets de bibliothèque Flex

Flash Builder vous permet également de créer des bibliothèques de code personnalisé que vous partagez entre vos applications ou que vous distribuez à d'autres développeurs. Un projet de bibliothèque génère un fichier SWC, c'est-à-dire un fichier archive pour les composants Flex ainsi que pour d'autres ressources. Pour plus d'informations, voir « [Projets de bibliothèque](#) » à la page 69.

Applications contenues dans les projets

Pour pouvoir commencer à créer une application dans Flash Builder, vous devez d'abord créer un projet. Spécifiez s'il s'agit d'une application Web (exécutée dans Adobe Flash Player) ou d'une application de bureau (exécutée dans Adobe AIR). Lorsque vous créez un projet Flex, un fichier d'application principale est créé. Vous ajoutez ensuite toutes les autres ressources, dont les fichiers d'application MXML, les fichiers de composants MXML personnalisés et les fichiers ActionScript, qui formeront votre application. Lorsque vous créez un projet ActionScript, un fichier ActionScript principal est créé. Vous pouvez ensuite construire une application grâce à ActionScript et à l'API de Flash Player. Pour plus d'informations, voir « [Création de projets Flex](#) » à la page 41 et « [Gestion des projets](#) » à la page 49.

Projets gérés dans les espaces de travail

Les projets sont gérés depuis un *espace de travail*, c'est-à-dire une zone définie du système de fichiers contenant les ressources (fichiers et dossiers) composant les applications. Vos projets résident par défaut dans l'espace de travail. Vous pouvez toutefois créer des projets en dehors de l'espace de travail. Flash Builder les relie automatiquement à l'espace de travail. Le passage d'un espace de travail à l'autre entraîne le redémarrage de Flash Builder.

Multiplicité des projets dans chaque espace de travail

Vous pouvez ajouter autant de projets que vous le souhaitez à un espace de travail. Tous les projets s'affichent dans l'Explorateur de packages, dans lequel vous pouvez les gérer à votre guise. Vous pouvez ajouter des ressources, classer les projets dans des dossiers et générer des projets dans l'espace de travail. Pour plus d'informations, voir « [Gestion des projets](#) » à la page 49 et « [Création de dossiers et de fichiers dans un projet](#) » à la page 63.

Ressources externes liées

Outre les ressources internes des projets, vous pouvez créer des liens avec des ressources situées en dehors d'un projet et d'un espace de travail. Les ressources externes liées apparaissent en tant que partie du projet, mais elles se trouvent en dehors de l'emplacement du projet. Pour plus d'informations, voir « [Liaison à des ressources situées en dehors de l'espace de travail du projet](#) » à la page 64.

Multiplicité des applications dans un projet

Flash Builder vous permet de définir plusieurs fichiers dans votre projet pour former une application. Lorsque vous créez un projet, Flash Builder génère un fichier d'application principale qui sert de point d'accès à l'application. Le compilateur fait appel à ce fichier pour générer le fichier SWF de l'application. Vous pouvez cependant créer des fichiers d'application supplémentaires en fonction de la complexité du projet. Tous les fichiers d'application doivent se trouver dans le dossier src sous le dossier racine du projet. Pour plus d'informations, voir « [Gestion des fichiers d'application du projet](#) » à la page 52.

Prise en charge de plusieurs SDK Flex

Que vous ayez des projets en cours de développement ou que vous deviez gérer une base de code d'un ancien projet, Flash Builder vous permet d'utiliser différentes versions du SDK Flex. Pour spécifier les SDK installés, vous devez configurer l'espace de travail de Flash Builder, qui propose un SDK par défaut pour tous les projets. Une fois un projet configuré, vous pouvez ajouter, supprimer ou modifier les configurations des SDK dans la boîte de dialogue Préférences en sélectionnant Flex > SDK installés. Vous pouvez aussi modifier les configurations des SDK en sélectionnant Projet > Propriétés > Compilateur Flex. Pour plus d'informations, voir « [Utilisation de plusieurs SDK dans Flash Builder](#) » à la page 83.

Génération automatique de projets

Par défaut, votre projet est automatiquement généré à chaque fois que vous enregistrez des changements dans un fichier. Vous maîtrisez complètement la façon dont vos applications sont générées ainsi que la fréquence à laquelle elles le sont. Si vous n'avez pas de besoins spécifiques en termes de personnalisation de la génération, celle-ci se fait en toute transparence et les fichiers SWF de l'application sont générés automatiquement. Pour plus d'informations, voir « [Génération de projets](#) » à la page 73.

Exportation vers une version validée

Une fois que l'application est prête à être déployée, l'assistant Exporter vers une version validée vous permet de créer une version *non déboguée* de qualité finale de l'application. L'assistant copie les ressources requises dans un dossier bin-release distinct du dossier de la version de débogage. Vous pouvez définir si le code source de l'application doit être ou non inclus. L'application exportée est une version de production optimisée qui peut être affichée par les utilisateurs finaux. Pour les projets Adobe AIR, les applications AIR sont exportées dans un fichier AIR. La fonctionnalité Exporter vers une version validée permet de créer un fichier AIR signé numériquement que les utilisateurs doivent installer avant d'exécuter une application.

Scripts Ant personnalisés

Apache Ant est un outil de génération Java qui permet de créer des scripts personnalisés pour la création d'applications dans Flash Builder. Ant vous permet de modifier et d'étendre le processus de génération standard. Pour plus d'informations, voir « [Personnalisation des générations avec Apache Ant](#) » à la page 83.

Génération par ligne de commande

Flash Builder Premium vous permet d'implémenter des générations par ligne de commande. Faites appel aux générations par ligne de commande pour synchroniser les paramètres de génération d'un développeur avec la génération nocturne. Pour plus d'informations, voir « [Génération par ligne de commande de Flash Builder](#) » à la page 86.

Types de projets

Flash Builder permet de créer des types de projets dans les configurations suivantes.

Projets Flex

Les options de configuration des projets dépendent de la façon dont l'application accède aux données et de l'installation d'Adobe® LiveCycle® Data Services ES ou d'Adobe BlazeDS. Vous pouvez créer des projets pour des applications Web (exécutées dans Flash Player) ou de bureau (exécutées dans Adobe AIR). Les options sont les suivantes.

Aucun En l'absence de serveur d'applications, cette configuration de base vous permet de spécifier le dossier de sortie de l'application Flex compilée. Vous pouvez également définir les chemins de génération du nouveau projet.

ASP.NET Si Microsoft Windows et Microsoft Visual Web Developer sont installés, vous pouvez créer des projets qui font appel à ASP.NET Development Server pour le déploiement. De plus, si vous avez accès à Internet Information Service (IIS), vous pouvez créer des projets avec un dossier de sortie Flex sous IIS.

ColdFusion Cette configuration de projet vous permet de créer des projets utilisant ColdFusion avec LiveCycle Data Services, BlazeDS ou ColdFusion Flash Remoting. Si aucune de ces options n'est sélectionnée, le dossier de sortie du projet ColdFusion créé sera situé sous la racine Web (ou dossier virtuel). Flash Builder propose des outils pour accéder aux données distantes des sources ColdFusion. Pour plus d'informations, voir Construction d'applications centrées sur les données avec Flash Builder.

J2EE Cette configuration de projet vous permet de créer des projets dont le déploiement est basé sur J2EE. Vous pouvez spécifier si vous souhaitez utiliser le service d'accès aux objets distants et choisir entre LiveCycle Data Services et BlazeDS. En l'absence de sélection d'un service d'accès aux objets distants, un dossier de sortie est créé sous la racine du serveur d'applications Java. Après avoir activé l'option Utiliser le service d'accès aux objets distants, vous pouvez choisir entre LiveCycle Data Services et BlazeDS. En fonction de votre choix, le projet est déployé sur un serveur LiveCycle Data Services ou BlazeDS. Si le plug-in Eclipse Web Tools Project (WTP) est installé, sélectionnez l'option Créer un projet mixte Java/Flex à l'aide de WTP pour créer des projets mixtes Java/Flex avec ou sans service d'accès aux objets distants. Pour les projets compilés localement avec WTP, les projets sont déployés sur votre serveur J2EE.

Vous pouvez utiliser LiveCycle Data Services avec ou sans WTP. Si vous l'utilisez avec WTP, le projet n'est pas déployé sur le serveur LiveCycle Data Services local, mais il est déployé à l'aide des fonctions WTP.

PHP Cette configuration de projet vous permet de créer des projets Flex dotés d'un dossier de sortie Flex sous la racine Web Apache/IIS (ou dossier virtuel). Flash Builder propose des outils pour accéder aux données distantes des serveurs PHP. Pour plus d'informations, voir Construction d'applications centrées sur les données avec Flash Builder.

Autre Si le serveur d'applications dont vous disposez n'est pas répertorié, cette option permet de spécifier le dossier de sortie de l'application compilée. Vous pouvez également définir les chemins de génération du nouveau projet.

Services Web et HTTP Pour chaque type d'application, vous pouvez créer des applications pouvant accéder à des services Web et HTTP. Flash Builder propose des outils pour l'accès aux services Web et HTTP. Pour plus d'informations, voir Construction d'applications centrées sur les données avec Flash Builder.

Projets ActionScript

En s'appuyant sur l'API de Flash plutôt que la structure Flex, les projets ActionScript permettent aux développeurs d'ActionScript d'utiliser Flash Builder pour coder, générer et déboguer les applications basées uniquement sur ActionScript. Ces projets ne faisant pas appel à MXML pour définir une interface utilisateur, vous ne pouvez pas afficher la présentation et la conception de l'application en mode Création. Vous utilisez exclusivement l'éditeur de source et les outils de débogage le cas échéant, puis vous générez le projet dans des fichiers d'application SWF afin de prévisualiser l'application et de la tester dans un navigateur Web ou un Flash Player autonome. Pour plus d'informations sur les projets ActionScript, voir « [Projets ActionScript](#) » à la page 66.

Projets de bibliothèque

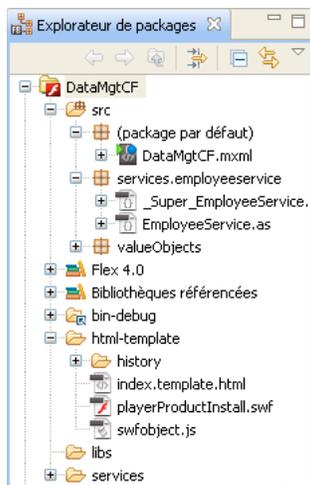
Les projets de bibliothèque permettent d'assembler et de distribuer les composants et autres ressources. Ils génèrent des fichiers SWC que vous ajoutez à d'autres projets ou que vous distribuez à d'autres développeurs. Pour plus d'informations, voir « [Projets de bibliothèque](#) » à la page 69.

Projets Flash Professional

Utilisez les projets Flash Professional pour modifier, générer ou déboguer les fichiers FLA ou XFL créés avec Adobe Flash Professional CS5. Les projets Flash Professional nécessitent l'installation de Flash Professional CS5. Pour plus d'informations, voir « [Création de projets Flash Professional](#) » à la page 48.

Projets dans l'Explorateur de packages

Tous les projets d'un espace de travail s'affichent dans l'Explorateur de packages, comme l'illustre l'exemple suivant. L'Explorateur de packages présente une arborescence des projets dans une vue à la fois physique et logique (plate). Cette vue vous permet de gérer les projets en ajoutant et en supprimant des ressources (dossier et fichiers), en important des ressources externes et en créant des liens vers ces ressources ainsi qu'en déplaçant les ressources vers d'autres projets dans l'espace de travail.



Principales caractéristiques de l'Explorateur de packages :

- L'affichage des packages ActionScript peut se présenter de manière hiérarchique ou plate.
Utilisez le menu de l'Explorateur de packages pour spécifier la présentation des packages.
- Les bibliothèques de projet sont représentées dans deux nœuds de niveau supérieur : un nœud pour le SDK Flex et l'autre pour les bibliothèques référencées.
Vous pouvez développer le contenu d'une bibliothèque et ouvrir les éditeurs pour afficher les pièces jointes.
- Les symboles d'erreur et d'avertissement apparaissant au niveau des nœuds de l'Explorateur de packages signalent d'éventuels problèmes dans un package.
- Vous pouvez restreindre les projets et ressources visibles.
Vous pouvez créer un jeu de documents (collection de ressources), définir des filtres d'affichage et trier les ressources par nom et type. Ces options sont disponibles dans les menus de l'Explorateur de packages. Pour plus d'informations sur la modification des vues, voir « [Navigation et personnalisation du workbench Flash Builder](#) » à la page 23.
- Vous pouvez développer les fichiers ActionScript, MXML et CSS ainsi qu'afficher leur contenu sous forme d'arborescence.

Dans l'Explorateur de packages, vous pouvez ouvrir les ressources du projet pour les modifier. Vous pouvez par exemple modifier les codes MXML et ActionScript dans les blocs `<fx:Script>` et le code CSS dans les blocs `<fx:Style>`. Vous pouvez également passer en mode Création et agencer visuellement les composants et les contrôles pour définir la présentation de l'application et son comportement. Pour plus d'informations sur l'utilisation des éditeurs de Flash Builder, voir « [A propos de la modification du code dans Flash Builder](#) » à la page 103 et « [Création d'une interface utilisateur avec Flash Builder](#) » à la page 188.

Vous pouvez ensuite ajouter des projets, fichiers et dossiers et les organiser et les gérer selon vos besoins (voir « [Création de dossiers et de fichiers dans un projet](#) » à la page 63).

La plupart des commandes de menu que vous utilisez dans la vue Explorateur de packages sont également disponibles dans le menu contextuel de la vue.

Pour plus d'informations sur l'utilisation de projets dans l'Explorateur de packages, voir « [Gestion des projets](#) » à la page 49 et « [Création de dossiers et de fichiers dans un projet](#) » à la page 63.

Création de projets et ouverture de ressources

Flash Builder comporte des assistants pour la création de projets Flex, ActionScript et de bibliothèque Flex. Le tableau suivant décrit chacun de ces projets. Pour créer un projet, sélectionnez Fichier > Nouveau.

Type de projet	Description
Projet ActionScript	Projet ActionScript basé sur l'API Flash, non sur la structure Flex. Les projets ActionScript permettent aux développeurs d'ActionScript d'utiliser Flash Builder pour coder, générer et déboguer les applications basées uniquement sur ActionScript. Pour plus d'informations, voir « Création de projets ActionScript » à la page 67.
Projet Flex	Un projet Flex contient un ensemble de propriétés qui contrôlent la façon dont l'application est générée, l'endroit où l'application construite est hébergée, la manière dont le débogage est géré et les relations avec les autres projets de l'espace de travail. Il vous permet de créer des applications qui seront exécutées dans Flash Player (applications Web) ou dans Adobe AIR (applications de bureau). Pour plus d'informations, voir « Définition des propriétés des projets Flex » à la page 49.
Projet de bibliothèque Flex	Les projets de bibliothèque Flex permettent d'assembler et de distribuer les composants ainsi que d'autres ressources. Ils génèrent des fichiers SWC que vous ajoutez à d'autres projets ou que vous distribuez à d'autres développeurs. Pour plus d'informations, voir « Projets de bibliothèque » à la page 69.
Projet Flash Professional	Utilisez des projets Flash Professional pour accéder aux fichiers FLA ou XFL créés avec Flash Professional CS5. Cette fonctionnalité permet aux développeurs Flash Professional de profiter de l'environnement de modification et de débogage que fournit Flash Builder. Pour accéder aux fonctions de Flash Professional dans Flash Builder, vous devez installer Flash Professional CS5.

Ressources des projets

Les applications Flex et ActionScript prennent en charge plusieurs types de ressources standard (MXML, ActionScript et CSS). Le tableau suivant répertorie les types de ressources que vous pouvez ajouter à vos projets. Pour ajouter ces ressources, sélectionnez Fichier > Nouveau.

Type de ressource	Description
Classe ActionScript	Fichier de classe ActionScript. Lorsque vous ajoutez ce type de ressource, l'assistant Nouvelle classe ActionScript vous invite à préciser des éléments de définition de la classe, tels que la super-classe, les interfaces, etc. Pour plus d'informations sur l'utilisation d'ActionScript dans Flash Builder, voir « Création d'une classe ActionScript » à la page 67.
Fichier ActionScript	Modèle de fichier de texte pour la création de fonctions ActionScript.
Interface ActionScript	Fichier d'interface ActionScript. Lorsque vous ajoutez ce type de ressource, l'assistant Nouvelle interface ActionScript vous invite à préciser des éléments de définition de l'interface tels que les interfaces étendues et le package dans lequel elles résident. Pour plus d'informations sur l'utilisation d'ActionScript dans Flash Builder, voir « Création d'une interface ActionScript » à la page 68.
Fichier CSS	Modèle de fichier de texte pour la création d'un fichier Cascading Style Sheets.
Fichier	Fichier de texte non formaté. Pour plus d'informations, voir « Création de dossiers et de fichiers dans un projet » à la page 63.
Dossier	Dossier standard de système de fichiers permettant de classer le contenu des projets. Pour plus d'informations, voir « Création de dossiers et de fichiers dans un projet » à la page 63.

Type de ressource	Description
Application MXML	Fichier d'application standard dont l'élément MXML racine est la balise <code><:Application></code> (pour les projets Flex 4). Vous pouvez définir la présentation à utiliser pour l'application. Un projet Flex peut contenir plusieurs fichiers d'application. Pour plus d'informations, voir « Gestion des fichiers d'application du projet » à la page 52.
Composant MXML	Fichier de composant standard dont l'élément MXML racine est la balise <code><:Group></code> . L'assistant vous permet de spécifier un élément MXML racine alternatif. Pour plus d'informations, voir « Création de composants MXML avec Flash Builder » à la page 134.
Rendu d'élément MXML	Les rendus d'élément contrôlent l'aspect d'un élément de données dans un conteneur DataGroup, SkinnableDataContainer ou dans une sous-classe de ces conteneurs. L'aspect peut inclure la police, la couleur d'arrière-plan, la bordure et tout autre aspect visuel de l'élément de données. Pour plus d'informations, voir « Génération de rendus d'élément personnalisés » à la page 224.
Habillage MXML	Les classes d'habillage modifient l'aspect des contrôles dans une interface utilisateur. La création, la modification et l'importation d'habillages varient en fonction des composants (Spark ou MX). Pour plus d'informations, voir « Modification de l'interface utilisateur à l'aide des habillages » à la page 218.
Module MXML	Ressource pouvant être ajoutée à un projet d'application existant ou créée séparément, mais toujours associée à une application. Pour plus d'informations, voir « Création de modules dans Flash Builder » à la page 96.
Classe de script de tests Classe de suite de tests	Scripts et suites de tests FlexUnit. Vous pouvez générer et modifier des tests pouvant être répétés et exécutables à partir de scripts ou directement dans Flash Builder. Pour plus d'informations, voir « Environnement de test FlexUnit » à la page 150.
Package	Crée un nouveau package pour les fichiers source d'un projet. L'emplacement par défaut est le répertoire <code>src</code> du dossier du projet.
Autre	Autres types de fichiers enregistrés dans Flash Builder. Sélectionnez Fichier > Nouveau > Autre pour ajouter d'autres types de fichiers. Si un plug-in Java est installé dans Flash Builder, vous pouvez par exemple ajouter de nouvelles classes, de nouvelles interfaces et de nouveaux packages Java. Lorsqu'un type de fichier est enregistré dans Flash Builder, un éditeur correspondant est également disponible dans le workbench. Pour plus d'informations, voir « Association d'éditeurs à des types de fichier » à la page 28. Vous pouvez en outre ajouter aux projets des types de fichiers non enregistrés en les important. Pour plus d'informations, voir « Importation de projets » à la page 55.

Pour plus d'informations sur l'ajout de ressources aux projets, voir « [Création de dossiers et de fichiers dans un projet](#) » à la page 63.

Création de projets Flex

Lorsque vous créez un projet, l'assistant Nouveau projet Flex vous guide à travers les différentes étapes et vous demande de préciser le type de projet à créer, le nom du projet, son emplacement ainsi que d'autres options.

Pour plus d'informations sur la création d'un projet ActionScript, voir « [Création de projets ActionScript](#) » à la page 67. Pour plus d'informations sur la création de projets de bibliothèque, voir « [Projets de bibliothèque](#) » à la page 69.

Création d'un projet Flex sans serveur

En l'absence de serveur, cette configuration de base vous permet de spécifier le dossier de sortie des applications compilées. Vous pouvez également, en option, définir les chemins de génération du nouveau projet.

- 1 Sélectionnez Fichier > Nouveau > Projet Flex.

- 2 Entrez un nom de projet.
- 3 Sélectionnez un emplacement pour le projet. L'emplacement par défaut est l'espace de travail actuel. Pour choisir un emplacement de projet différent, désactivez l'option Utiliser l'emplacement par défaut.
- 4 Choisissez le type d'application (Web ou de bureau).
- 5 Sélectionnez Aucun comme type de serveurs d'applications.
- 6 Cliquez sur Terminer ou sur Suivant pour accéder à d'autres options de configuration.
Voir « [Options supplémentaires pour la configuration d'un projet](#) » à la page 46.

Création d'un projet Flex avec ASP.NET

Si Microsoft Windows et Microsoft Visual Web Developer sont installés, vous pouvez créer des projets Flex déployés avec ASP.NET. De plus, si vous avez accès à un serveur de développement Internet Information Service (IIS), vous pouvez créer des projets Flex avec un dossier de sortie Flex sous IIS.

- 1 Sélectionnez Fichier > Nouveau > Projet Flex.
- 2 Entrez un nom de projet.
- 3 Spécifiez l'emplacement du projet. L'emplacement par défaut est l'espace de travail actuel. Sous Windows, l'espace de travail par défaut est C:\Documents and Settings\Développeur Flex\Adobe Flash Builder\. Pour choisir un emplacement de projet différent, désactivez l'option Utiliser l'emplacement par défaut.
- 4 Choisissez le type d'application (Web ou de bureau).
- 5 Sélectionnez ASP.NET comme type de serveurs d'applications.
- 6 Cliquez sur Suivant.
- 7 Sélectionnez le serveur ASP.NET :
 - Si vous utilisez un serveur de développement ASP.NET, vous n'avez pas besoin de préciser l'emplacement du serveur.
 - Si vous utilisez IIS, entrez la racine de l'application Web et l'URL de l'application Web.
 - Spécifiez le dossier de sortie de l'application Flex.
- 8 Cliquez sur Terminer ou sur Suivant pour accéder à d'autres options de configuration.
Voir « [Options supplémentaires pour la configuration d'un projet](#) » à la page 46.

Création d'un projet Flex avec J2EE

Cette configuration de projet vous permet de créer des projets Flex utilisant un servlet J2EE avec l'option de service d'accès aux objets distants. Si aucune option n'est sélectionnée et qu'un serveur Java est utilisé, un dossier de sortie est créé sous la racine du serveur. Si le plug-in Eclipse Web Tools Project (WTP) est installé, vous pouvez créer des projets mixtes Java et Flex avec ou sans service d'accès aux objets distants.

***Remarque :** LiveCycle Data Services ES et BlazeDS prennent en charge des versions spécifiques du SDK Flex. Consultez le [Tableau de compatibilité LiveCycle Data Services](#) pour découvrir les versions du SDK Flex prises en charge par votre version de LiveCycle DataService ES. Ce tableau de compatibilité répertorie également les versions du SDK Flex prises en charge par BlazeDS.*

Vous disposez de deux options de compilation d'un projet Flex utilisant J2EE. L'option recommandée compile l'application localement, puis enregistre les fichiers (notamment le fichier SWF et l'enveloppe HTML) sur le serveur. La seconde option compile le fichier source de l'application directement sur le serveur.

- 1 Sélectionnez Fichier > Nouveau > Projet Flex.
- 2 Entrez un nom de projet.
- 3 Spécifiez l'emplacement du projet. L'emplacement par défaut est l'espace de travail actuel. Pour choisir un emplacement de projet différent, désactivez l'option Utiliser l'emplacement par défaut.
- 4 Choisissez le type d'application (Web ou de bureau).
- 5 Sélectionnez J2EE comme type de serveurs d'applications.
- 6 Cochez l'option Utiliser le service d'accès aux objets distants. LiveCycle Data Services ES est sélectionné par défaut. Vous pouvez sélectionner BlazeDS. Si vous avez installé WTP, vous pouvez aussi choisir de créer un projet mixte Java et Flex faisant appel à WTP (le dossier source Java est sélectionné pour vous).
- 7 Cliquez sur Suivant.
- 8 Configurez le serveur J2EE.

- Si vous sélectionnez les options Utiliser le service d'accès aux objets distants et LiveCycle Data Services ou BlazeDS, précisez les paramètres racine :

Dossier racine Serveur Flex (application Web) qui sert votre application (C:\fds2\jrun4\servers\default\flex, par exemple). Si vous n'utilisez pas l'option de serveur de développement par défaut pour Flex, vous pouvez spécifier un nouvel emplacement pour le dossier racine. Il doit toutefois s'agir d'un dossier valide mappé à l'URL racine spécifiée. Si vous utilisez un serveur distant, précisez son emplacement (*monServeur\MesApplications\jrun4\servers\default\flex*, par exemple).

URL racine URL racine valide du serveur Flex (application Web) qui sert votre application. Pour LCDS, l'URL racine par défaut des occurrences du serveur local est `http://localhost:8400/lcds/`. Si vous utilisez un serveur distant, l'URL peut avoir l'aspect suivant : `http://monserveur.com:8400/lcds/`.

Racine du contexte La racine du contexte correspond généralement au dernier segment du chemin racine de l'URL.

- Si vous avez sélectionné l'option Créer un projet mixte Java/Flex à l'aide de WTP (avec ou sans LiveCycle Data Services) :
 - Indiquez le nom des dossiers source et des exécutions cible Java et Flex.

Lorsque vous créez un projet Flex avec LiveCycle Data Services ES, Flash Builder crée un répertoire portant le même nom que le projet ou utilise un répertoire existant portant ce nom. Ce répertoire est un sous-répertoire du dossier racine que vous avez spécifié pour le projet.
 - Avec LiveCycle Data Services ES, précisez un fichier flex.war qui se trouve dans le dossier d'installation du serveur.

Remarque : quelle que soit l'option choisie pour un projet LiveCycle Data Services ES dans Flash Builder, vous devez spécifier un dossier racine LiveCycle Data Services ES et une URL racine. Ces valeurs mappent la racine d'une application Web LiveCycle Data Services ES. Si vous désélectionnez les options, vous devez saisir uniquement la racine Web et l'URL racine.

- 9 Spécifiez l'emplacement auquel le projet sera compilé.
 - Pour les applications compilées localement, Flash Builder crée un dossier de débogage *nom_du_projet* dans lequel les fichiers SWF et les enveloppes HTML sont enregistrés.
 - Pour les applications compilées sur le serveur, l'emplacement du projet doit se trouver sur le serveur.

10 Cliquez sur Terminer ou sur Suivant pour accéder à d'autres options de configuration.

Voir « [Options supplémentaires pour la configuration d'un projet](#) » à la page 46.

Création d'un projet Flex accédant aux services ColdFusion

Pour accéder aux données qui utilisent ColdFusion, vous devez disposer d'Adobe ColdFusion® 8 ou d'Adobe ColdFusion 9. Pour plus d'informations, consulter la [page produit de ColdFusion](#).

1 Sélectionnez Fichier > Nouveau > Projet Flex.

2 Entrez un nom de projet.

3 Spécifiez l'emplacement du projet.

Vous pouvez également accepter l'emplacement par défaut, qui est l'espace de travail actuel.

4 Spécifiez le type d'application (Web ou de bureau).

5 Pour le type de serveurs d'applications, sélectionnez ColdFusion, puis choisissez l'une des options suivantes.

Utiliser le service d'accès aux objets distants Si vous désactivez l'option Utiliser le service d'accès aux objets distants, spécifiez la racine Web et l'URL racine à l'étape suivante.

Si vous activez l'option Utiliser le service d'accès aux objets distants, vous pouvez effectuer les choix suivants.

- ColdFusion Flash Remoting

Activez cette option si vous prévoyez d'utiliser les outils de développement centrés sur les données disponibles avec Flash Builder. Cette option s'applique également si vous utilisez Flash Remoting pour invoquer des méthodes dans les composants ColdFusion (CFC). Voir [Création d'applications centrées sur les données avec Flash Builder](#).

- LiveCycle Data Services

Spécifiez LiveCycle Data Services comme type d'application ColdFusion uniquement si votre installation ColdFusion 8 est configurée pour LiveCycle Data Services 2.6.1. Voir [Integrating Adobe LiveCycle Data Services ES 2.6 with Adobe ColdFusion 8](#).

Généralement, le type de serveurs d'applications pour LiveCycle Data Services n'est pas ColdFusion, mais J2EE. Voir « [Création d'un projet Flex avec J2EE](#) » à la page 42.

- BlazeDS

Spécifiez BlazeDS comme type d'application ColdFusion uniquement si votre installation ColdFusion 8 est configurée pour Adobe BlazeDS 3.1. Voir [Integrating BlazeDS with a ColdFusion 8 Installation](#).

Généralement, le type de serveurs d'applications pour BlazeDS n'est pas ColdFusion, mais J2EE. Voir « [Création d'un projet Flex avec J2EE](#) » à la page 42.

6 Cliquez sur Suivant. Spécifiez un emplacement de serveur, une racine Web, une URL racine et une racine du contexte.

Pour l'accès à un service d'objets distants, vous pouvez définir une configuration ColdFusion autonome ou une configuration ColdFusion déployée sur un serveur J2EE.

- Autonome

Activez l'option Autonome si l'installation ColdFusion utilise la configuration du serveur.

Spécifiez l'emplacement du serveur ColdFusion, de la racine Web et de l'URL racine.

- Déployée sur un serveur J2EE

Activez cette option si l'installation ColdFusion utilise aussi bien la configuration multiserveurs que la configuration J2EE.

Spécifiez la racine Web, l'URL racine et la racine du contexte. Si vous faites appel à la configuration multiserveurs ColdFusion, vous ne devez généralement pas spécifier la racine du contexte.

La racine du contexte correspond en règle générale au dernier segment du chemin racine de l'URL lorsque vous déployez ColdFusion en tant qu'application Web dans la configuration J2EE ColdFusion.

Pour spécifier l'emplacement du serveur ou de la racine Web, recherchez un répertoire local ou saisissez le chemin vers un répertoire situé sur un serveur réseau. Vérifiez que le répertoire est partagé et que le compte sous lequel Flash Builder est exécuté possède des droits d'accès en écriture.

Vérifiez que vous avez mappé ou monté un lecteur réseau pour le serveur réseau. Le chemin au serveur réseau est fonction de la plateforme. Par exemple :

(Windows) \\10.192.18.12\serveur\racine Web

(Windows) Z:\racine Web

(Mac) /Volumes/racine Web

- 7 Cliquez sur Valider la configuration pour vérifier l'exactitude des paramètres spécifiés.

Si le répertoire de la racine Web n'est pas accessible en écriture, Flash Builder affiche un avertissement.

- 8 Sélectionnez un dossier de sortie pour l'application compilée.

- 9 Cliquez sur Terminer ou sur Suivant pour accéder à d'autres options de configuration.

Voir « [Options supplémentaires pour la configuration d'un projet](#) » à la page 46.

Création d'un projet Flex pour accéder aux services PHP

L'accès aux données de services PHP présuppose l'existence d'un serveur hébergeant les services. Il peut s'agir d'un serveur local ou d'un serveur appartenant à un réseau local.

- 1 Sélectionnez Fichier > Nouveau > Projet Flex.
- 2 Entrez un nom de projet.
- 3 Sélectionnez un emplacement pour le projet. L'emplacement par défaut est l'espace de travail actuel. Pour choisir un emplacement de projet différent, désactivez l'option Utiliser l'emplacement par défaut.
- 4 Choisissez le type d'application (Web ou de bureau).
- 5 Sélectionnez PHP comme type de serveurs d'applications. Cliquez sur Suivant.
- 6 Spécifiez la racine Web et l'URL racine pour le service. Cliquez sur Valider la configuration.

La racine Web et l'URL racine spécifiées appartiennent généralement à votre environnement local. Vous pouvez toutefois également accéder à des serveurs réseau. Vérifiez que le répertoire est partagé et que le compte sous lequel Flash Builder est exécuté possède des droits d'accès en écriture.

Vérifiez que vous avez mappé ou monté un lecteur pour le serveur réseau. Spécifiez ensuite le chemin vers le serveur. Ce chemin est fonction de la plateforme. Par exemple :

(Windows) \\10.192.18.12\serveur\racine Web

(Windows) Z:\racine Web

(Mac) /Volumes/racine Web

- 7 (Facultatif) Spécifiez le dossier de sortie de l'application.
- 8 Cliquez sur Terminer ou sur Suivant pour accéder à d'autres options de configuration.
Voir « [Options supplémentaires pour la configuration d'un projet](#) » à la page 46.

Options supplémentaires pour la configuration d'un projet

Lorsque vous créez un projet Flex, vous pouvez en personnaliser la configuration. Toutes les étapes de configuration supplémentaires sont facultatives.

Remarque : vous pouvez modifier la configuration d'un projet après sa création. Pour ce faire, accédez au mode Source de l'éditeur de Flash Builder, puis sélectionnez *Projet > Propriétés*.

- Dossier source principal, Fichier de l'application principale et URL du dossier de sortie

Par défaut, Flash Builder place les fichiers source dans le dossier src du projet. Le nom par défaut du fichier de l'application MXML principale est le nom du projet. Ces valeurs par défaut peuvent être modifiées à la création du projet.

Lorsque vous créez un projet, Flash Builder exécute les fichiers d'application à partir d'une URL par défaut, en fonction des paramètres du projet. Spécifiez une URL de dossier de sortie pour remplacer les paramètres par défaut.

Voir « [Définition d'un dossier de sortie de projet](#) » à la page 79 et « [Exécution des applications](#) » à la page 90.

- Jeu de composants

En règle générale, tous les composants sont disponibles. Dans certains cas, vous pouvez ne spécifier que les composants MX. Voir « [Jeu de composants \(MX + Spark ou MX uniquement\)](#) » à la page 78.

- Liaison de la structure

Par défaut, les classes d'application de la structure Flex 4 utilisent la liaison dynamique. Les options suivantes sont également activées par défaut :

- Vérifier les condensés RSL
- Utiliser les bibliothèques RSL locales de débogage lors du débogage
- Déterminer automatiquement l'ordre de la bibliothèque, en fonction des dépendances

Voir « [Liaison de la structure des applications](#) » à la page 78.

- Bibliothèques du chemin de génération

Vous pouvez ajouter au chemin de génération des bibliothèques de projet, des dossiers de bibliothèque SWC ou des fichiers SWC ; vous pouvez également les en supprimer. Vous pouvez en outre modifier l'ordre du chemin de génération.

Cliquez sur le bouton Editer pour modifier l'emplacement des bibliothèques ou des dossiers ajoutés.

Utilisez le bouton Ajouter un SDK Flex pour rétablir le SDK par défaut d'un projet dont vous avez supprimé le SDK Flex du chemin de génération.

- Dossiers source supplémentaires

Cliquez sur l'onglet Chemin source pour ajouter des dossiers source supplémentaires à un projet. Vous pouvez réorganiser les dossiers source, en modifier l'emplacement ou les supprimer du chemin source.

Modification des options de serveur des projets existants

Vous serez probablement amené dans certains cas à modifier la configuration d'origine du serveur d'un projet. Pour une application Web ou de bureau, accédez à la fenêtre des propriétés du projet concerné.

Sélectionnez l'option Serveur Flex pour ajouter ou modifier les options du serveur du projet.

- Sélectionnez Aucun pour supprimer la configuration serveur d'un projet.

La suppression de la configuration serveur d'un projet conduit également à la suppression de tout fichier SWC ajouté au chemin d'accès à la bibliothèque pour ce type de serveur.

- Sélectionnez un type de serveur à modifier ou ajoutez la configuration serveur d'un projet.

Toutes les options de serveur pour la configuration serveur sélectionnée sont disponibles. Pour en savoir plus sur les paramètres de configuration du serveur, voir « [Création de projets Flex](#) » à la page 41.

La modification du type de serveur d'un projet peut provoquer des erreurs dans le code existant reposant sur le type de serveur d'origine. Vous devrez examiner et corriger chaque erreur de code.

Création d'un projet Flex 4 utilisant uniquement des composants MX

Vous pouvez créer un projet Flex compatible avec les composants MX mis à disposition par Flex 3. L'option MX uniquement intervient lors de la création d'applications dont la conception est similaire aux applications créées avec une version antérieure de Flex, mais ayant malgré tout accès aux fonctions de Flex 4 et de Flash Builder 4 (telles que la nouvelle syntaxe d'états, les fonctionnalités CSS avancées, des fonctions de compilation améliorées ainsi que d'autres fonctionnalités de langage).

L'option MX uniquement ne met pas à la disposition des applications du projet les composants Spark fournis avec Flex 4.

Vous pouvez convertir un projet Flex 4 en projet MX uniquement. Gardez cependant à l'esprit que Flash Builder ne réécrit pas le code du projet. Vous devrez mettre à jour le code manuellement afin d'en supprimer toute référence aux composants Spark.

Création d'un projet Flex MX uniquement

- 1 Sélectionnez Fichier > Nouveau > Projet Flex.
- 2 Entrez un nom de projet.
- 3 Sélectionnez un emplacement pour le projet. L'emplacement par défaut est l'espace de travail actuel. Pour choisir un emplacement de projet différent, désactivez l'option Utiliser l'emplacement par défaut.
- 4 Choisissez le type d'application (Web ou de bureau).
- 5 Spécifiez un type de serveurs d'applications (ou n'en spécifiez aucun, s'il n'est pas requis). Cliquez sur Suivant.
- 6 (Facultatif) Spécifiez le dossier de sortie de l'application.
- 7 Cliquez sur Terminer ou sur Suivant pour accéder à d'autres options de configuration.

Voir « [Options supplémentaires pour la configuration d'un projet](#) » à la page 46.

Conversion d'un projet Flex en projet Flex MX uniquement

- 1 Dans Flash Builder, activez le projet à convertir.
Pour ce faire, vous devez généralement ouvrir un fichier source du projet.
- 2 Sélectionnez Projet > Propriétés > Chemin de génération Flex.

3 Pour le paramètre Jeu de composants, sélectionnez l'option MX uniquement. Cliquez sur OK.

4 Modifiez tout code d'application du projet accédant à des composants Spark.

Il ne peut y avoir de références à des composants Spark dans un projet MX uniquement.

Création de projets Flash Professional

Utilisez des projets Flash Professional pour accéder aux fichiers FLA ou XFL créés avec Flash Professional CS5. Cette fonctionnalité permet aux développeurs Flash Professional de profiter de l'environnement de modification et de débogage que fournit Flash Builder. Pour accéder aux fonctions de Flash Professional dans Flash Builder, vous devez installer Flash Professional CS5.

Vous commencez généralement par créer un projet et des fichiers dans Flash Professional. Vous créez ensuite un projet correspondant dans Flash Builder pour la modification et le débogage des fichiers. Lors de la modification des fichiers dans Flash Builder, vous pouvez définir des points d'arrêt dans les fichiers ActionScript du projet. Les points d'arrêt définis dans des fichiers appartenant au projet Flash Professional sont reconnus par le débogueur Flash Professional lors de l'appel de Déboguer l'animation.

Vous pouvez lancer Flash Professional à partir de Flash Builder pour publier et exécuter les fichiers. Vous pouvez également lancer le débogueur Flash Professional à partir de Flash Builder.

Création d'un projet Flash Professional

1 Sélectionnez Fichier > Nouveau > Projet Flash Professional.

2 Recherchez le fichier FLA ou XFL pour le projet.

Le nom du fichier est utilisé comme nom du projet.

3 Spécifiez l'emplacement du projet.

Vous pouvez utiliser l'emplacement par défaut dans l'espace de travail ou rechercher un nouvel emplacement.

4 Cliquez sur Terminer.

Flash Builder ouvre le nouveau projet dans l'Explorateur de packages. Le dossier contenant le fichier FLA cible est accessible. Le fichier FLA sélectionné devient le fichier cible du projet. Les fichiers ActionScript qui dépendent des fichiers cible peuvent être modifiés.

S'il n'est pas déjà en cours d'exécution, Flash Professional démarre.

Utilisation de projets Flash Professional dans Flash Builder

Vous pouvez appliquer les opérations suivantes aux fichiers source d'un projet Flash Professional :

- Modifier les fichiers ActionScript qui dépendent du fichier FLA cible.
- Déboguer le fichier dans le débogueur Flash Builder ou Flash Professional.

Pour déboguer le fichier dans Flash Builder, sélectionnez Exécuter > Déboguer *fichier* ou cliquez sur le bouton Déboguer dans la barre d'outils.

Pour déboguer le fichier dans Flash Professional, sélectionnez Exécuter > Déboguer l'animation ou cliquez sur le bouton Déboguer l'animation dans Flash Builder. Les points d'arrêt définis dans Flash Builder sont reconnus par le débogueur Flash Professional.

- Publier le fichier dans Flash Professional CS5.

Sélectionnez Projet > Publier la séquence ou cliquez sur Publier dans le bouton Flash Professional de la barre d'outils.

- Exécuter le fichier dans Flash Builder ou Flash Professional.

Pour exécuter le fichier dans Flash Builder, sélectionnez Exécuter > Exécuter *fichier* ou cliquez sur le bouton Exécuter de la barre d'outils.

Pour exécuter le fichier dans Flash Professional, sélectionnez Exécuter > Tester l'animation ou cliquez sur le bouton Tester l'animation de la barre d'outils.

Définition des propriétés des projets Flash Professional

1 Sélectionnez Projet > Propriétés du projet > Flash Professional.

2 Sélectionnez Ajouter pour ajouter des fichiers supplémentaires au projet.

Un projet ne peut comporter qu'un seul fichier FLA ou XFL cible comme fichier cible par défaut. Utilisez le bouton Définir par défaut pour spécifier le fichier cible par défaut du projet.

3 Cliquez sur OK.

Gestion des projets

L'Explorateur de packages permet d'ajouter des ressources à un projet, d'importer des ressources, d'exporter des projets ainsi que de déplacer et de supprimer des ressources.

Définition des propriétés des projets Flex

Chaque projet Flex possède son propre jeu de propriétés. Pour définir ces propriétés, sélectionnez le projet dans la vue Explorateur de packages. Dans le menu principal, sélectionnez ensuite Projet > Propriétés. Vous pouvez également accéder aux propriétés à partir du menu contextuel du projet.

Vous pouvez définir les préférences de projet suivantes.

Ressource Présente des informations générales concernant le projet, les paramètres pour le codage de texte et les délimiteurs de lignes du système d'exploitation.

Générateurs Précise l'outil de génération à utiliser. Flash Builder contient un outil de génération standard. Vous pouvez faire appel à Apache Ant (outil de génération « open source ») pour créer des scripts de génération ou importer des scripts de génération Ant existants (voir « [Personnalisation des générations avec Apache Ant](#) » à la page 83).

Modèle de données La configuration du modèle de données est uniquement disponible avec LiveCycle Data Services version 3 ou supérieure. Elle spécifie l'emplacement du fichier du modèle de données contenant les informations sur le type de service et de données pour LiveCycle Data Services.

Données/Services Pour les projets accédant à des services de données, cette page définit si le générateur de code par défaut doit être utilisé pour accéder aux services. Vous pouvez également indiquer si vous souhaitez utiliser une seule occurrence de serveur pour accéder aux services (pour plus d'informations sur l'extension de Flash Builder de manière à prendre en charge la génération de code personnalisé, voir [Extension de la prise en charge des services dans Flash Builder](#)). pour plus d'informations sur l'utilisation d'une seule occurrence de serveur lors de l'accès aux services, voir Utilisation d'une occurrence de serveur unique).

Applications Flex Affiche le nom des fichiers de projet définis comme fichiers d'application pouvant être compilés, débogués et exécutés en tant qu'applications distinctes (voir « [Gestion des fichiers d'application du projet](#) » à la page 52).

Chemin de génération Flex Spécifie le chemin d'accès à la génération, qui précise l'emplacement des fichiers source externes et des fichiers de bibliothèque. Vous pouvez modifier ce chemin et changer le nom du dossier de sortie (voir « [Définition d'un dossier de sortie de projet](#) » à la page 79 et « [Génération manuelle de projets](#) » à la page 81).

Compilateur Flex Spécifie d'une part les préférences optionnelles pour le compilateur (génération d'un fichier SWF accessible, activation des avertissements de compilation et de la vérification des types, spécification d'arguments de compilation supplémentaires et définition de la version du SDK Flex) et définit d'autre part les paramètres de l'enveloppe HTML (voir « [Options de génération avancées](#) » à la page 81).

Modules Flex Précise les modules à générer et à optimiser pour le projet. Pour plus d'informations sur l'utilisation des modules dans Flash Builder, voir « [Création de modules dans Flash Builder](#) » à la page 96.

Serveur Flex Spécifie le type de serveurs d'applications pour le projet. A la création d'un projet, vous avez la possibilité de spécifier un type de serveurs d'applications. Vous pouvez modifier ce type ici. Si vous modifiez le type de serveurs d'applications d'un projet, vous risquez de ne plus pouvoir accéder aux services de données précédemment configurés. Voir « [Création de projets Flex](#) » à la page 41 et Création d'un projet Flex pour l'accès à des services de données.

Thème Flex Spécifie le thème à utiliser pour toutes les applications du projet. Vous pouvez spécifier l'un des thèmes disponibles dans Flash Builder ou en importer un. Pour plus d'informations, voir « [Application de thèmes](#) » à la page 210.

Références de projet Répertorie les projets auxquels le projet en cours fait référence.

Exécuter/déboguer les paramètres Gère les paramètres de configuration de lancement. Voir « [Gestion des configurations de lancement](#) » à la page 94.

Conversion de projets Flex en projets Adobe AIR

Vous pouvez convertir le type d'applications d'un projet Flex d'application Web (exécutée dans Flash Player) en application de bureau (exécutée dans Adobe AIR). La conversion entraîne les modifications suivantes :

- Un fichier descripteur AIR est créé pour chaque application du projet.
- Les configurations de lancement du projet sont mises à jour de manière telle à être lancées correctement au cours de l'exécution AIR.
- Les paramètres des enveloppes HTML sont supprimés.
- Les paramètres personnalisés de Flash Player sont supprimés.
- Le chemin d'accès à la bibliothèque est modifié pour inclure le fichier airglobal.swc au lieu du fichier playerglobal.swc.

Au cours de la conversion, vous pouvez définir pour chaque application du projet si vous souhaitez modifier les balises Application de base en balises WindowedApplication. Si vous décidez de convertir ces balises, cette modification sera la seule à intervenir dans le code d'application au cours de la conversion. Examinez les attributs des balises de base après la conversion pour vérifier que l'application est exécutée correctement dans Adobe AIR.

Modification d'un projet d'application Web en projet d'application de bureau

Remarque : cette procédure ne peut pas être annulée.

1 Sélectionnez le projet que vous souhaitez convertir.

Il doit s'agir d'un projet Flex avec type d'application Web (exécutée dans Flash Player).

2 Dans le menu contextuel du projet, sélectionnez :

Ajouter/Modifier le type de projet > Convertir en projet de bureau (exécution dans Adobe AIR).

3 Dans la boîte de dialogue Convertir en projet Bureau/Adobe AIR, spécifiez si vous souhaitez apporter des modifications au code :

- Convertir les balises Application en WindowedApplication

Les balises Application de toutes les applications existantes du projet sont converties en balises WindowedApplication. Aucune autre modification ne sera apportée au code. Examinez les attributs des balises de base pour vérifier que l'application est exécutée correctement dans Adobe AIR.

Les nouvelles applications créées dans le projet sont des applications de bureau pouvant être exécutées dans Adobe AIR.

- Ne réécrire aucun code

Aucune modification n'est apportée au code. Vous devez éditer les applications dans le projet avant de pouvoir les exécuter dans Adobe AIR.

Les nouvelles applications créées dans le projet sont des applications de bureau pouvant être exécutées dans Adobe AIR.

Déplacement d'un projet d'un espace de travail vers un autre

Vous pouvez faire appel à une association d'opérations de suppression et d'importation pour déplacer un projet d'un espace de travail vers un autre. Lorsque vous supprimez un projet d'un espace de travail, vous pouvez l'effacer de l'espace de travail mais le laisser dans le système de fichiers (voir « [Suppression de projets](#) » à la page 51). Après avoir supprimé un projet d'un espace de travail, vous pouvez l'importer dans un autre.

Spécification du SDK d'un projet

Lorsque vous créez un projet Flex, vous pouvez spécifier quel SDK Flex vous souhaitez utiliser. Les paramètres du SDK pourront être modifiés ultérieurement en cliquant sur Projet > Propriétés > Compilateur Flex > Utiliser un SDK spécifique.

Vous pouvez compiler un projet en utilisant un SDK Flex qui n'est pas disponible dans votre installation Flash Builder en le téléchargeant et en l'y ajoutant. Par exemple, pour que le SDK utilisé corresponde à celui du serveur, procédez à l'extraction du SDK du serveur, puis ajoutez-le à Flash Builder en cliquant sur Projet > Propriétés > Compilateur Flex > Configurer les SDK Flex.

Importation de projets utilisant une compilation de serveur distant

L'importation de projets utilisant une compilation de serveur n'est pas prise en charge. Vous pouvez importer un projet spécifiant une compilation de serveur. L'importation comportera toutefois des erreurs qui s'afficheront dans la vue Erreurs. Les erreurs sont accompagnées d'un lien fournissant des informations sur la conversion d'un projet de compilation de serveur en projet de compilation d'outils.

Suppression de projets

Lorsque vous supprimez un projet, vous le retirez de l'espace de travail en cours. Vous pouvez également le supprimer simultanément du système de fichiers.

Au lieu de supprimer le projet de l'espace de travail, vous pouvez le fermer. La fermeture du projet vous permet de conserver une référence à ce projet dans l'espace de travail tout en libérant des ressources système. Pour plus d'informations, voir « [Fermeture et ouverture de projets](#) » à la page 52.

1 Dans l'Explorateur de packages, sélectionnez le projet à supprimer.

2 Dans le menu principal, sélectionnez Editer > Supprimer.

3 Sélectionnez une option.

Supprimer également le contenu sous le répertoire Supprime définitivement le projet de l'espace de travail et du système de fichiers.

Ne pas supprimer le contenu Supprime le projet de l'espace de travail sans le supprimer du système de fichiers.

Fermeture et ouverture de projets

Pour économiser de la mémoire et accélérer la génération sans supprimer un projet, vous pouvez le fermer. Lorsque vous fermez un projet, vous réduisez le projet et ses ressources, mais son nom reste visible dans l'Explorateur de packages. Un projet fermé nécessite moins de mémoire qu'un projet ouvert et est exclu des générations. Vous pouvez facilement rouvrir un projet fermé.

1 Dans l'Explorateur de packages, sélectionnez le projet à fermer ou rouvrir.

2 Dans le menu contextuel de l'Explorateur de packages, sélectionnez Fermer le projet ou Ouvrir le projet.

Changement du fichier d'application principale

Lorsque vous créez un projet Flex, un fichier d'application principale est créé. Par défaut, il est nommé d'après le projet. Le fichier d'application principale est le point d'accès à vos applications et est la base du fichier SWF de l'application. Cependant, à mesure que vous ajoutez des fichiers à l'application, vous pouvez désigner un fichier différent comme fichier d'application principale.

Si vous préférez définir plusieurs fichiers d'application de manière à ce que chaque fichier d'application soit généré dans un fichier SWF distinct, consultez « [Gestion des fichiers d'application du projet](#) » à la page 52.

1 Dans l'Explorateur de packages, sélectionnez le fichier d'application MXML que vous voulez définir comme fichier d'application principale.

2 Dans le menu contextuel de l'Explorateur de packages, sélectionnez Définir comme application par défaut.

Vous pouvez gérer les fichiers d'application du projet en sélectionnant Projet > Propriétés > Applications Flex (ou Applications ActionScript si vous utilisez un projet ActionScript).

Gestion des fichiers d'application du projet

Un projet possède généralement un seul fichier d'application principale, qui sert de point d'accès à l'application. Le compilateur Flash Builder utilise ce fichier pour générer le fichier d'application SWF.

Une application Flex complexe peut par exemple être dotée de nombreux composants MXML personnalisés qui représentent des éléments d'application distincts mais interdépendants. Vous pouvez créer un fichier d'application contenant un composant personnalisé et ensuite le générer, l'exécuter et le tester séparément.

Par défaut, vous pouvez exécuter l'application à chaque ajout d'un fichier d'application MXML au projet Flex. Ce fichier est alors ajouté à la liste des fichiers d'application du projet. Tous les fichiers définis comme fichiers d'application doivent se trouver dans le dossier source du projet.

Vous pouvez gérer la liste des fichiers d'application en sélectionnant un projet et en affichant ses propriétés.

1 Dans l'Explorateur de packages, sélectionnez un projet.

2 Sélectionnez Projet > Propriétés dans le menu principal ou sélectionnez Propriétés dans le menu contextuel.

3 Dans la boîte de dialogue Propriétés du projet, sélectionnez Applications Flex (ou Applications ActionScript si vous utilisez un projet ActionScript).

- 4 Ajoutez et supprimez les fichiers d'applications selon vos besoins. Cliquez sur OK.

Exportation et importation de projets

Flash Builder exporte les projets Flex et les projets de bibliothèques Flex au format FXP. Les projets ActionScript peuvent être exportés uniquement vers des fichiers archive, généralement au format ZIP.

Le format FXP est un format d'archive contenant les dossiers, les fichiers et les métadonnées d'un projet. Un projet exporté contient tous les projets de bibliothèque Flex dépendants.

Remarque : vous pouvez également faire appel à l'assistant d'exportation Eclipse pour exporter des projets Flex et des projets de bibliothèque Flex au format ZIP ou vers un autre format d'archive.

Lors de l'exportation ou de l'importation d'un projet Flex, certains contenus du projet nécessiteront un traitement spécial.

- Versions différentes du SDK Flex

Vous pouvez importer un projet Flex faisant référence à une version du SDK Flex qui n'a pas été installée avec Flash Builder. Pour plus d'informations sur le téléchargement et l'installation de versions supplémentaires du SDK Flex, voir « [SDK Flex installés](#) » à la page 253.

Si Flash Builder ne trouve pas une version spécifique du SDK Flex, une boîte de dialogue s'ouvre vous permettant de la rechercher.

Après avoir installé la version correcte du SDK Flex, accédez à son emplacement.

- Fichiers de services

Les projets Flex qui se connectent à des services de données comportent des dossiers `services` contenant des liens vers des fichiers de services déployés. À l'exportation du projet, Flash Builder exporte les liens, mais pas les services déployés. À l'importation, vous devrez déployer les fichiers de services manuellement et mettre à jour les liens, si besoin est.

Pour les projets se connectant aux services via LiveCycle Data Service ou BlazeDS, vérifiez que les destinations de service sont disponibles sur le serveur cible.

Pour les projets qui référencent des fichiers locaux, déployez ces derniers en utilisant le même chemin que celui du projet d'origine à l'importation. Cette manière de procéder s'applique aux projets accédant à des fichiers de service XML statiques ou à des fichiers locaux pour des services HTTP ou Web.

- Structure Zend Framework

Les projets Flex qui se connectent à des services de données en utilisant PHP et la structure Zend Framework contiennent deux fichiers de configuration. À l'importation, examinez ces fichiers afin de vérifier qu'ils ont été correctement configurés pour votre système :

```
amf-config.ini
```

```
gateway.php
```

Pour plus d'informations sur l'installation, la configuration et le dépannage de votre installation Zend Framework, voir Installation de Zend Framework.

- Fichiers de modèles de données (LiveCycle Data Services)

Un projet Flex utilisant LiveCycle Data Services (LCDS) est relié à un fichier de modèle de données.

Après exportation puis importation, Flash Builder référence le fichier de modèle de données actuel et non le lien qui y renvoie. Si vous souhaitez utiliser un fichier lié (et non celui inclus dans le projet exporté), modifiez le fichier de modèle de données en accédant aux propriétés du projet. Sélectionnez Projet > Propriétés > Modèle de données pour apporter les modifications.

Exportation d'un projet Flex ou d'un projet de bibliothèque Flex au format FXP

Les menus des configurations autonome et plug-in de Flash Builder correspondant à cette procédure diffèrent légèrement.

Certains projets Flex devront faire l'objet d'un traitement spécial à l'importation. Voir « [Exportation et importation de projets](#) » à la page 53.

- 1 Dans Flash Builder, sélectionnez Fichier > Exporter un projet Flex (FXP).

Dans la version plug-in de Flash Builder, sélectionnez Fichier > Exporter > Flash Builder > Projet Flash Builder.

Vous pouvez également faire appel au menu contextuel du projet dans l'Explorateur de packages. Sélectionnez Exporter > Flash Builder > Projet Flash Builder.

- 2 Dans l'assistant Exportation d'un projet Flex, sélectionnez le projet à exporter.

La liste déroulante Projet répertorie tous les projets disponibles.

- 3 (Facultatif) Activez l'option Valider la compilation du projet.

Sélectionnez cette option pour obtenir confirmation que le projet sera compilé sans erreurs. La présence d'erreurs ne vous empêchera pas de poursuivre l'exportation.

- 4 Cliquez sur Terminer.

Pour les projets sur serveur, les chemins absolus vers les ressources du serveur sont enregistrés sous forme de variables de chemin. Lorsque vous importerez ultérieurement le projet, vous devrez spécifier des valeurs pour les variables de chemin.

Exportation d'un projet ActionScript au format ZIP (ou vers tout autre format d'archive)

- 1 Dans Flash Builder, sélectionnez Fichier > Exporter > Autre.

- 2 Dans l'assistant d'exportation, sélectionnez Général > Fichier archive. Cliquez sur Suivant.

- 3 Sélectionnez le projet et les fichiers à exporter :

- Dans le volet de gauche, développez le projet pour en sélectionner les dossiers à inclure dans l'exportation.
- Dans le volet de droite, indiquez pour chaque dossier sélectionné les fichiers à exporter.

- 4 Recherchez un emplacement dans lequel enregistrer le projet exporté et attribuez un nom au fichier.

- 5 Définissez les options pour le fichier archive, puis cliquez sur Terminer.

Importation de projets

Flash Builder peut importer des projets Flex, des projets de bibliothèque Flex et des projets ActionScript. Les projets peuvent être importés à partir de dossiers de projet existants ou à partir de fichiers issus d'une préalable exportation à partir de Flash Builder. Vous pouvez importer plusieurs versions d'un même projet Flex ou d'un même projet de bibliothèque Flex. Une fois l'importation de plusieurs versions effectuée, vous pouvez les comparer, puis copier ou fusionner leurs différences.

Projets nécessitant un traitement spécial

Certains projets Flex devront faire l'objet d'un traitement spécial à l'importation. Cela est le cas par exemple lorsque :

- le projet fait référence à une version antérieure du SDK Flex ;
- le projet fait référence à des fichiers de service pour l'accès aux données ;
- la configuration Zend Framework pour l'accès aux services PHP doit être mise à jour ;
- le projet utilise des liens LiveCycle Data Services (LCDS) vers un fichier de modèle de données.

Pour plus d'informations sur les projets nécessitant un traitement spécial, voir « [Exportation et importation de projets](#) » à la page 53.

Prise en charge des projets Catalyst

Flash Builder prend en charge le développement des applications conçues dans Adobe® Flash® Catalyst™. Catalyst exporte un projet au format FXP et des composants au format FXPL. Le fichier FXPL est un package de bibliothèque. Les fichiers FXP et FXPL peuvent être ensuite importés dans Flash Builder à des fins de développement. Pour les fichiers FXP, il en résulte un projet Web Flex exécuté dans Adobe Flash Player. Un fichier FXPL contient un fichier de bibliothèque. Vous pouvez importer des fichiers FXPL sous la forme d'un projet de bibliothèque Flex ou importer le contenu dans un projet Flex existant.

Vous pouvez créer un projet Adobe AIR à partir d'un projet Catalyst. Importez le fichier FXP du projet Catalyst dans Flash Builder. Convertissez le type d'application du projet d'application Web (exécutée dans Flash Player) en application de bureau (exécutée dans Adobe AIR). Voir « [Conversion de projets Flex en projets Adobe AIR](#) » à la page 50.

Importation d'un projet Flex ou d'un projet de bibliothèque Flex

Vous pouvez importer un projet à partir d'un fichier FXP exporté ou en accédant au dossier contenant le projet.

Remarque : pour plus d'informations sur l'importation de contenu d'un projet de bibliothèque dans un autre projet Flex, voir « [Importation d'un projet FXPL Catalyst](#) » à la page 56. Un projet FXPL est un projet de bibliothèque créé par Adobe Catalyst.

Les menus des configurations autonome et plug-in de Flash Builder correspondant à cette procédure diffèrent légèrement.

- 1 Dans le menu de Flash Builder, sélectionnez Fichier > Importer un projet FXP.

Dans la version plug-in de Flash Builder, sélectionnez Fichier > Importer > Flash Builder > Projet Flash Builder.

Les fonctions d'importation d'un projet sont également disponibles dans le menu contextuel de l'Explorateur de packages.

- 2 (Dossier du projet) Si l'importation est effectuée à partir d'un dossier de projet existant, activez l'option Dossier du projet et recherchez le dossier contenant le projet.
- 3 (Fichier FXP) Si vous effectuez une importation à partir d'un fichier FXP, sélectionnez Fichier et naviguez jusqu'à l'emplacement du fichier.

Si le fichier FXP contient plusieurs projets, vous pouvez sélectionner individuellement les projets à importer.

4 (Projet de bibliothèque ou projet FXPL) Si vous importez un projet de bibliothèque ou un projet FXPL Catalyst, vous avez la possibilité d'importer le contenu dans un projet existant. Voir « [Importation d'un projet FXPL Catalyst](#) » à la page 56.

5 (Fichier FXP) Si un projet du même nom existe dans l'espace de travail, spécifiez la méthode d'importation.

- Importer en tant que nouveau projet : Flash Builder identifie le projet en ajoutant un chiffre à son nom. Les versions antérieures du projet sont conservées.

Dans le champ Extraire le nouveau projet vers, spécifiez l'emplacement dans lequel vous souhaitez extraire le fichier. Cet emplacement est généralement un répertoire de l'espace de travail de Flash Builder représentant un dossier du projet. Vous pouvez spécifier un nouveau dossier de projet ou remplacer un projet existant.

- Remplacer un projet existant : sélectionnez le projet que vous souhaitez remplacer. La version antérieure du projet est définitivement supprimée.

6 (Variables de chemin) Si vous importez un projet définissant des variables de chemin, mettez à jour les variables de chemin pour le projet.

Les projets compilés pour ColdFusion, PHP, LiveCycle Data Services ou d'autres technologies de serveur font appel aux variables de chemin pour accéder au serveur Web et aux ressources du serveur. D'autres projets peuvent présenter des variables de chemin personnalisées.

Sélectionnez chaque variable de chemin et attribuez-lui une valeur valide.

7 (Références aux polices) Si vous importez un fichier FXP exporté par Catalyst, le projet peut contenir des références aux polices. Vous avez la possibilité de résoudre ces références.

Voir « [Résolution des références aux polices lors de l'importation de projets Catalyst](#) » à la page 57.

8 Cliquez sur Terminer.

9 (Projets pour serveurs PHP) Si vous importez un projet pour serveur d'applications de type PHP, installez ou mettez à jour l'installation Zend.

La boîte de dialogue Zend vous guide au long du processus.

Remarque : si vous annulez le processus dans la boîte de dialogue Zend, installez ou mettez à jour manuellement Zend Framework. L'accès aux services PHP est impossible tant que Zend Framework n'est pas installé et configuré correctement. Pour plus d'informations sur l'installation, la configuration et le dépannage de l'installation Zend Framework, voir *Installation de Zend Framework*.

10 (Projets pour serveurs) Déployez des services.

- a Placez manuellement des services dans la racine Web du serveur. Utilisez la structure de répertoire utilisée dans le projet d'origine.
- b Dans la vue Données/Services, sélectionnez Actualiser dans le menu contextuel d'un service.

Importation d'un projet FXPL Catalyst

Un projet FXPL Catalyst est un projet de bibliothèque créé par Adobe Catalyst. Lors de l'importation d'un projet FXPL, vous avez la possibilité d'importer le contenu dans un autre projet Flex ou dans un projet de bibliothèque Flex.

Cette fonctionnalité est conçue pour permettre aux développeurs de collaborer avec les concepteurs d'applications Catalyst. Vous pouvez toutefois l'utiliser pour importer le contenu d'un projet de bibliothèque dans un autre projet Flex ou dans un projet de bibliothèque Flex.

Les menus des configurations autonome et plug-in de Flash Builder correspondant à cette procédure diffèrent légèrement. Cette procédure suppose que vous importez un projet de bibliothèque.

1 Dans le menu de Flash Builder, sélectionnez Fichier > Importer un projet FXP.

Dans la version plug-in de Flash Builder, sélectionnez Fichier > Importer > Flash Builder > Projet Flash Builder.

Les fonctions d'importation d'un projet sont également disponibles dans le menu contextuel de l'Explorateur de packages.

2 Activez l'option Fichier et recherchez l'emplacement du fichier.

3 Spécifiez la méthode d'importation.

- Importer nouvelle copie du projet : Flash Builder identifie le projet en ajoutant un chiffre à son nom. Les versions antérieures du projet sont conservées.

Dans le champ Extraire le nouveau projet vers, spécifiez l'emplacement dans lequel vous souhaitez extraire le fichier. Cet emplacement est généralement un répertoire de l'espace de travail de Flash Builder représentant un dossier du projet. Vous pouvez spécifier un nouveau dossier de projet ou remplacer un projet existant.

- Importez le contenu dans un projet existant.

Pour Dossier source, accédez au dossier `src` d'un projet existant. Pour Package, accédez à un package existant ou spécifiez un nouveau nom de package pour le contenu.

- Remplacer un projet existant : si un projet du même nom existe dans l'espace de travail, vous pouvez remplacer le projet existant.

Sélectionnez le projet à remplacer. La version antérieure du projet est définitivement supprimée.

4 Cliquez sur Terminer.

Lors de l'importation de fichiers FXPL, Flash Builder tente de résoudre les références aux polices du fichier FXPL. Voir « [Résolution des références aux polices lors de l'importation de projets Catalyst](#) » à la page 57.

Résolution des références aux polices lors de l'importation de projets Catalyst

Lors de l'importation d'un projet FXP créé avec Adobe Catalyst, ce projet peut contenir des références à des polices qui ne sont pas disponibles sur votre système.

L'assistant d'importation permet de corriger les références aux polices à l'aide de CSS. Si vous sélectionnez l'option correspondante, Flash Builder importe la feuille de style Catalyst `Main.css`. `Main.css` contient les références aux polices utilisées dans le projet.

Si vous obtenez des erreurs de compilation provenant des polices référencées dans la feuille de style, corrigez les références de la feuille de style avec les polices disponibles sur votre système.

Les projets FXPL Catalyst ne contiennent pas de feuilles de style. Flash Builder tente de corriger les références aux polices lors de l'importation d'un fichier FXPL. S'il ne trouve pas de police correspondante sur le système cible, les références aux polices d'origine sont conservées. Pour les projets FXPL, les références aux polices que Flash Builder ne parvient pas à résoudre sont détectées à l'exécution. Les références aux polices non résolues entraînent une substitution de police ou une erreur d'exécution.

Remarque : pour les fichiers FXPL, Flash Builder modifie l'attribut `fontFamily` dans les fichiers MXML lorsqu'il tente de résoudre les références aux polices.

Importation d'un projet Flex 3

Vous pouvez importer un projet Flex 3 dans Flash Builder en utilisant le mode de compatibilité Flex 3. Dans ce cas, les espaces de noms et les composants Flex 3 demeureront inchangés. Vous pourrez cependant profiter du compilateur disponible avec Flex 4.

Les documents créés en mode de compatibilité Flex 3 utilisent les composants MX et les espaces de noms suivants :

```
mx="http://www.adobe.com/2006/mxml"
```

- 1 Dans Flash Builder, sélectionnez Fichier > Importer un projet Flex.
- 2 Recherchez le fichier ZIP du projet Flex 3 précédemment exporté ou le dossier du projet Flex 3.
- 3 Cliquez sur Terminer.
- 4 Dans la boîte de dialogue Choisir la version du SDK Flex, vérifiez que le SDK Flex 4 a été activé. Sélectionnez Utiliser le mode de compatibilité Flex 3.
- 5 Cliquez sur OK.

Comparaison des modifications apportées à un projet

Si vous importez plusieurs versions d'un projet, vous pouvez comparer, copier ou fusionner les contenus des versions. Vous pouvez uniquement comparer des versions différentes d'un même projet.

- 1 Dans l'Explorateur de packages, sélectionnez l'un des projets que vous souhaitez comparer.
- 2 Ouvrez le menu contextuel et sélectionnez Comparer le projet avec la version.
L'affichage de comparaison s'ouvre, vous permettant de comparer le projet à d'autres versions de ce même projet.
- 3 Sélectionnez la version avec laquelle vous souhaitez effectuer la comparaison. L'éditeur de comparaison Eclipse s'ouvre.
- 4 Dans l'éditeur de comparaison, accédez au fichier que vous souhaitez comparer et sélectionnez Afficher la comparaison du contenu dans le menu contextuel.
L'éditeur de comparaison affiche les deux versions du fichier et en met en évidence les différences.
Vous pouvez utiliser les options de l'éditeur de comparaison pour copier ou fusionner les différences dans le fichier.
Pour plus d'informations, voir la documentation Eclipse sur l'éditeur de comparaison.

Importation d'un projet ActionScript

Les projets ActionScript sont exportés au format d'archive ZIP. Utilisez l'assistant d'importation Eclipse pour importer les projets ActionScript.

- 1 Dans Flash Builder, sélectionnez Fichier > Importer > Autre > Général > Fichier archive.
Vous pouvez également faire appel au menu contextuel de l'Explorateur de packages.
- 2 Dans la boîte de dialogue d'importation d'un projet Flex, sélectionnez le fichier ZIP à importer.
- 3 Cliquez sur Terminer.

Importation de projets exportés avec l'assistant d'exportation Eclipse

Un projet exporté à l'aide de l'assistant d'exportation Eclipse devra être importé avec l'assistant d'importation Eclipse. Sélectionnez Fichier > Importer > Général. Recherchez le format correspondant à votre projet.

Pour plus d'informations, voir la documentation Eclipse sur l'importation de projets. Cette documentation est disponible dans l'aide des assistants d'importation et d'exportation Eclipse.

Les services d'un projet créés avec les outils Flash Builder d'accès aux services de données devront être ajoutés manuellement. Copiez les fichiers de serveur dans le dossier services du serveur approprié. Dans la vue Données/Services, utilisez les propriétés du service pour un serveur afin de déterminer l'emplacement du service.

Si vous avez exporté un projet PHP basé sur la structure Zend Framework, cette dernière doit être installée sur le serveur cible. Modifiez le fichier `amf-config.ini` qui configure la structure Zend Framework. Pour `zend_path`, spécifiez le chemin absolu du répertoire d'installation de Zend Framework. Pour plus d'informations sur l'installation, la configuration et le dépannage de l'installation Zend Framework, voir Installation de Zend Framework.

Importation d'un projet dans plusieurs espaces de travail

Flash Builder importe les projets dans ses espaces de travail. Un projet peut être importé dans plusieurs espaces de travail. Dans ce cas, les fichiers du projet se trouvent dans un seul emplacement sur le disque, mais sont référencés par chaque espace de travail. Les modifications apportées au projet se répercutent dans tous les espaces de travail.

Importation de fichiers source dans un nouveau projet

Vous pouvez créer un projet dans lequel importer les fichiers source et les ressources situés sur votre système de fichiers mais n'appartenant à aucun projet.

- 1 Dans le menu Flash Builder, sélectionnez Fichier > Nouveau > *Projet*.

Il peut s'agir d'un projet Flex, d'un projet de bibliothèque Flex ou d'un projet ActionScript.

- 2 Dans l'assistant Nouveau projet Flex, spécifiez la source et le dossier de sortie.

Remarque : vous pouvez également accepter les emplacements par défaut proposés par l'assistant et y déplacer les fichiers source.

Exportation d'un programme d'installation d'application Adobe AIR

Pour les projets AIR, une génération de production crée un fichier AIR signé numériquement que l'utilisateur doit installer préalablement à l'exécution de l'application. Ce processus s'apparente à la création d'un fichier `.exe` d'un programme d'installation pour une application native standard. Vous pouvez créer un package intermédiaire non signé que vous pourrez signer ultérieurement avant la publication.

Pour plus d'informations sur l'exportation d'une version de production et d'un programme d'installation AIR, voir « [Exportation vers une version validée \(application de bureau, exécutée dans Adobe AIR\)](#) » à la page 60.

Exportation d'une application vers une version validée

Vous pouvez exporter une version optimisée de qualité finale (fichier SWF ou AIR non débogué) de l'application en utilisant l'assistant Exporter vers une version validée. Les ressources requises sont copiées dans un dossier distinct de celui de la version de débogage. Une fois l'assistant exécuté, vous devrez procéder à des opérations supplémentaires pour déployer l'application sur un serveur.

Exportation vers une version validée (application Web, exécutée dans Adobe Flash Player)

- 1 Sélectionnez Projet > Exporter vers une version validée pour ouvrir l'assistant correspondant.
- 2 Sélectionnez le projet et l'application que vous souhaitez exporter.

- 3 (Facultatif) Sélectionnez Activer l'affichage de la source pour permettre l'accès aux fichiers source à partir de l'application exportée.

Cliquez sur Choisir des fichiers source pour spécifier les fichiers source à inclure. Outre les fichiers source spécifiés, l'assistant génère un fichier archive ZIP contenant les fichiers source.

Important : pour les projets pour serveurs, vous pouvez sélectionner le dossier des services à l'exportation des fichiers source. L'exportation de fichiers implémentant des services a des implications en termes de sécurité. Ces fichiers peuvent donner accès à votre base de données, noms d'utilisateurs et mots de passe inclus. Voir *Exportation des fichiers source vers la version validée d'une application*.

- 4 Cliquez sur Terminer.
- 5 Copiez le dossier contenant la version exportée vers la racine Web du serveur hébergeant l'application.
- 6 (Projets pour serveurs) Si vous exportez la version d'un projet spécifiant un type de serveurs d'applications, déployez les services à la racine Web du serveur cible.

Conservez la structure de répertoire utilisé lors du développement.

Cette étape s'applique aux services ColdFusion, PHP, BlazeDS et LCDS. Le type de serveurs d'applications est défini à la création du projet dans Flash Builder.

Pour des fichiers locaux déployés sur le serveur, un fichier de régulation interdomaines est nécessaire pour accéder à ces services. Cette manière de procéder s'applique aux projets accédant à des fichiers de service XML statiques ou à des fichiers locaux pour des services HTTP ou Web. Voir [Utilisation des fichiers de régulation interdomaines](#).

- 7 (Projets pour serveurs PHP uniquement) Pour les projets PHP, exécutez les étapes supplémentaires suivantes :
 - a Installez la structure Zend Framework sur le serveur. Voir *Installation de Zend Framework*.
 - b Modifiez le fichier `amf-config.ini` qui se trouve dans le dossier de sortie de la version exportée.

Pour `zend_path`, spécifiez le chemin absolu du répertoire d'installation de Zend Framework.

Définissez `amf.production` sur `true`.

Saisissez pour `webroot` le chemin absolu vers la racine Web du serveur.

Exportation vers une version validée (application de bureau, exécutée dans Adobe AIR)

Pour les projets AIR, une version de production crée un fichier AIR signé numériquement que l'utilisateur devra installer préalablement à l'exécution de l'application. Ce processus s'apparente à la création d'un fichier .exe de programme d'installation pour une application native standard. Vous pouvez également créer un package intermédiaire non signé que vous pourrez signer ultérieurement avant la publication. Avant d'entamer la procédure d'exportation vers une version validée, vous devez décider de la manière dont vous souhaitez signer numériquement l'application AIR :

- Signer l'application en ayant recours à un certificat numérique VeriSign ou Thawte
- Créer et utiliser un certificat numérique autosigné
- Assembler l'application et la signer ultérieurement

Les certificats numériques délivrés par VeriSign et Thawte confirment votre identité d'éditeur et indiquent qu'aucune modification n'a été apportée au fichier d'installation depuis sa signature. Les certificats numériques autosignés sont utilisés dans le même but, mais se distinguent par le fait qu'ils ne sont pas validés par un tiers.

Vous pouvez également assembler l'application AIR sans signature numérique en créant un fichier AIR intermédiaire (.airi). La validité de ce type de fichier est limitée. Il ne peut en effet pas être installé. Il est utilisé dans les procédures de test (effectuées par les développeurs) et peut être lancé avec l'outil de ligne de commande AIR ADT. La mise à disposition de cette fonctionnalité est liée au fait que dans certains environnements de développement, la signature est prise en charge par un développeur ou une équipe spécifique, garantissant ainsi un niveau de sécurité supérieur.

Exportation d'une application et d'un programme d'installation AIR

- 1 (Facultatif) Modifiez les paramètres du serveur dans les propriétés du projet.

Une application de bureau exportée peut accéder uniquement aux services utilisés au cours du développement. Si vous souhaitez modifier le serveur pour l'application de bureau exportée, modifiez les paramètres du projet.

- 2 Sélectionnez **Projet > Exporter vers une version validée** pour ouvrir l'assistant correspondant.

- 3 Sélectionnez le projet et l'application que vous souhaitez exporter.

- 4 (Facultatif) Sélectionnez **Activer l'affichage de la source** pour permettre l'accès aux fichiers source à partir de l'application exportée.

Cliquez sur **Choisir des fichiers source** pour spécifier les fichiers source à inclure. Outre les fichiers source spécifiés, l'assistant génère un fichier archive ZIP contenant les fichiers source.

Important : *pour les projets pour serveurs, vous pouvez sélectionner le dossier des services à l'exportation des fichiers source. L'exportation de fichiers implémentant des services a des implications en termes de sécurité. Ces fichiers peuvent donner accès à votre base de données, noms d'utilisateurs et mots de passe inclus. Voir Exportation des fichiers source vers la version validée d'une application.*

- 5 Sélectionnez l'option **Exporter vers un fichier AIR** ou l'option **Exporter vers le programme d'installation natif**. Cliquez sur **Suivant**.

L'option **Exporter vers un fichier AIR** crée un fichier de programme d'installation générique permettant l'installation de l'application sur les plates-formes Windows ou Mac OS X.

L'option **Exporter vers le programme d'installation natif** crée un programme d'installation pour la plate-forme cible (Windows ou Mac OS X).

- 6 A la page **Signature numérique**, procédez comme suit :

Spécifiez le certificat numérique représentant l'identité de l'éditeur de l'application. Pour générer un certificat autosigné, cliquez sur **Créer** et renseignez les champs requis.

Pour exporter un fichier dont la signature peut être effectuée ultérieurement, exportez un fichier intermédiaire AIRI.

- 7 Dans la page **Contenu du fichier AIR**, sélectionnez les fichiers de sortie à inclure dans le fichier AIR ou AIRI.

- 8 Cliquez sur **Terminer**.

- 9 Copiez le projet .air vers le bureau cible.

- 10 (Projets pour serveurs) Si vous exportez la version d'un projet spécifiant un type de serveurs d'applications, déployez les services sur le serveur cible.

Cette étape s'applique aux services ColdFusion, PHP, BlazeDS et LCDS. Le type de serveurs d'applications est défini à la création du projet dans Flash Builder.

- 11 (Projets pour serveurs PHP uniquement) Exécutez les étapes supplémentaires suivantes :

- a Installez la structure Zend Framework sur le serveur. Voir **Installation de Zend Framework**.

- b Modifiez le fichier `amf-config.ini` qui se trouve dans le dossier de sortie de la version exportée.

Pour `zend_path`, spécifiez le chemin absolu du répertoire d'installation de Zend Framework.

Définissez `amf.production` sur **true**.

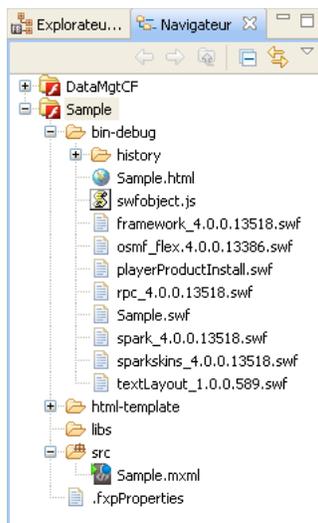
Saisissez pour `webroot` le chemin absolu vers la racine Web du serveur.

Version déboguée

La version déboguée de l'application contient des informations de débogage. Elle intervient au cours du débogage de l'application. La version validée résultant de l'exportation ne contient aucune information de débogage. La taille de son fichier est donc inférieure à celle du fichier de la version déboguée. Un fichier d'enveloppe HTML contient un lien vers le fichier SWF de l'application. Il intervient dans l'exécution ou le débogage de l'application dans un navigateur Web.

Remarque : les commandes *Exécuter* et *Débuguer* lancent la version de développement dans le dossier *bin-debug* (et non dans le dossier de version validée, *bin-release*).

Dans une application standard, le dossier de sortie ressemble à l'exemple suivant.



Vous pouvez exécuter ou déboguer vos applications Flex et ActionScript dans un navigateur Web ou dans l'application Flash Player autonome. Pour définir la manière dont les applications sont exécutées ou déboguées, modifiez la configuration de lancement du projet (voir « [Exécution des applications](#) » à la page 92). Pour plus d'informations sur l'exécution ou le débogage des applications, voir « [Exécution des applications](#) » à la page 90 et « [Débogage des applications](#) » à la page 141.

L'utilisation de LiveCycle Data Services ES conduit à la création d'applications tirant parti des technologies de serveur Flex. Lorsque vous générez des applications LiveCycle Data Services ES, vous pouvez opter entre la compilation locale des fichiers de sortie avec Flash Builder ou la compilation sur le serveur intervenant au moment d'accéder à l'application pour la première fois.

Gestion des ressources du projet

Les projets sont constitués de ressources (dossiers et fichiers) que vous pouvez gérer depuis l'Explorateur de packages. Les projets sont contenus dans un espace de travail. L'Explorateur de packages propose une représentation logique de l'espace de travail dans le système de fichiers. Il est actualisé à chaque fois que vous ajoutez, supprimez ou modifiez une ressource.

Vous pouvez également modifier les ressources du projet directement dans le système de fichiers, sans passer par Flash Builder ni par l'Explorateur de packages.

Création de dossiers et de fichiers dans un projet

Vous pouvez ajouter des dossiers et des fichiers à un projet selon vos besoins. Vous pouvez par exemple créer un dossier destiné à stocker tous les modèles de données ou à organiser toutes les ressources qui composent l'aspect visuel de l'application.

Création d'un dossier

1 Dans l'Explorateur de packages, sélectionnez Fichier > Nouveau > Dossier.

2 Si l'espace de travail contient plusieurs projets, sélectionnez le projet à ajouter au dossier autonome.

Si vous créez le dossier dans le dossier du chemin d'accès source, il est traité comme un nom de package. Vous pouvez y placer des fichiers source qui seront reconnus par le compilateur.

Si vous le créez en dehors du dossier du chemin d'accès source, vous pourrez l'ajouter ultérieurement à ce chemin afin de le transformer en racine d'une structure de packages. Une fois que vous avez terminé cette procédure, sélectionnez Projet > Propriétés, puis Chemin de génération Flex. Cliquez sur Ajouter un dossier et accédez au dossier que vous venez de créer.

3 Entrez le nom du dossier, puis cliquez sur Terminer.

Création d'un fichier

1 Dans l'Explorateur de packages, sélectionnez Fichier > Nouveau > Fichier.

2 Si l'espace de travail contient plusieurs projets, sélectionnez le projet auquel vous voulez ajouter le fichier.

3 Entrez le nom du fichier, puis cliquez sur Terminer.

Vous pouvez également ajouter des dossiers et des fichiers se trouvant en dehors du projet en cours. Pour plus d'informations, voir « [Liaison à des ressources situées en dehors de l'espace de travail du projet](#) » à la page 64.

Suppression de dossiers et de fichiers

La suppression de dossiers et de fichiers du projet les élimine de l'espace de travail et, par conséquent, du système de fichiers.

Remarque : la suppression d'une ressource liée ne supprime du projet que le lien, pas la ressource en soi (voir « [Liaison à des ressources situées en dehors de l'espace de travail du projet](#) » à la page 64). Cependant, si vous avez créé un lien à un dossier et que vous supprimez un ou plusieurs de ses fichiers, ceux-ci sont supprimés du système de fichiers.

1 Dans l'Explorateur de packages, sélectionnez la ressource à supprimer.

2 Sélectionnez Edition > Supprimer ou appuyez sur la touche Suppr, puis cliquez sur Oui.

La ressource est supprimée du système de fichiers.

Déplacement de ressources entre les projets d'un espace de travail

Lorsque vous travaillez avec plusieurs projets dans un espace de travail, vous pouvez déplacer des ressources d'un projet vers un autre.

1 Dans l'Explorateur de packages, sélectionnez la ressource à déplacer.

2 Effectuez l'une des opérations suivantes :

- Faites glisser la ressource vers un nouveau projet.
- Coupez et collez la ressource dans un autre projet.

Remarque : vous pouvez déplacer des ressources normales et des ressources liées. Pour plus d'informations sur la liaison de ressources, voir « [Liaison à des ressources situées en dehors de l'espace de travail du projet](#) » à la page 64.

Actualisation des ressources dans l'espace de travail

A mesure que vous modifiez, ajoutez ou supprimez les ressources d'un projet, le workbench actualise automatiquement les différentes vues qui affichent ces ressources. La suppression d'un fichier du projet par exemple est immédiatement répercutée dans l'Explorateur de packages.

Vous pouvez également modifier des ressources situées en dehors de Flash Builder, directement dans le système de fichiers. Ces modifications ne sont visibles dans Flash Builder qu'après actualisation de l'espace de travail.

Par défaut, l'espace de travail de la configuration autonome de Flash Builder est actualisé automatiquement. Cette option peut être modifiée dans les préférences de Flash Builder. Ouvrez la boîte de dialogue Préférences et sélectionnez Général > Espace de travail. Vous pouvez également modifier le comportement par défaut de Flash Builder et désactiver l'actualisation automatique de l'espace de travail.

Actualisation manuelle de l'espace de travail

- ❖ Dans le menu contextuel de l'Explorateur de packages, sélectionnez Actualiser. Toutes les ressources de projet dans l'espace de travail sont actualisées.

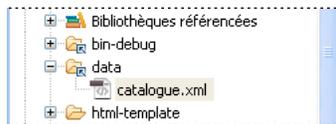
Désactivation de l'actualisation automatique

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Général > Espace de travail.
- 2 Désélectionnez l'option Actualiser automatiquement.

Liaison à des ressources situées en dehors de l'espace de travail du projet

Vous pouvez créer des liens avec des ressources situées en dehors du projet et de l'emplacement de l'espace de travail ainsi qu'avec des dossiers et des fichiers situés n'importe où dans le système de fichiers. Cette option est utile en cas de partage des ressources entre différents projets. Vous pouvez par exemple partager une bibliothèque de composants Flex ou ActionScript personnalisés entre plusieurs projets Flex.

Les dossiers contenant des ressources liées sont signalés dans l'Explorateur de packages (comme l'illustre l'exemple suivant). Vous pouvez ainsi distinguer les ressources normales des ressources liées.



Parmi d'autres exemples de liaison de ressources, citons les dossiers de fichiers d'images ou encore le cas des dossiers de sortie ne se trouvant pas dans le dossier racine du projet.

Lorsque les ressources sont partagées, les modifications que vous apportez aux dossiers et fichiers source affectent tous les projets qui leur sont liés. Soyez prudent lorsque vous supprimez de vos projets des ressources liées : dans certains cas, vous supprimez simplement la référence au lien ; dans d'autres, vous supprimez la source même. Pour plus d'informations, voir « [Suppression de dossiers et de fichiers](#) » à la page 63.

Remarque : il est recommandé de lier d'autres projets aux projets de bibliothèque. La liaison des ressources ne doit être utilisée que pour des bibliothèques tierces dotées d'un fichier SWC.

Liaison à des ressources situées en dehors de l'espace de travail du projet

- 1 Dans l'Explorateur de packages, sélectionnez le projet auquel vous souhaitez ajouter des ressources liées.
- 2 Sélectionnez Fichier > Nouveau > Dossier (ou Fichier).
- 3 Sélectionnez le projet ou le dossier du projet auquel vous souhaitez ajouter les ressources liées.
- 4 Saisissez le nom du dossier ou du fichier. Il peut être différent du nom du dossier ou du fichier avec lequel vous créez un lien.
- 5 Cliquez sur le bouton Options avancées.
- 6 Sélectionnez Lier au dossier dans le système de fichiers. Saisissez ou recherchez l'emplacement de la ressource.
- 7 Cliquez sur Terminer pour lier la ressource au projet.

Utilisation d'une variable de chemin à relier aux ressources

Vous pouvez établir des liens vers des ressources en définissant des variables de chemin. Vous évitez ainsi de devoir saisir le chemin complet d'un dossier local ou d'un dossier réseau dans lequel vous enregistrez les fichiers. Pour plus d'informations, voir « [Création d'une variable de chemin](#) » à la page 80.

- 1 Dans l'Explorateur de packages, sélectionnez le projet auquel vous souhaitez ajouter des ressources liées.

 *Les variables de chemin peuvent également intervenir dans certains paramètres de projet, tels que le chemin d'accès à la bibliothèque et le chemin d'accès à la source.*

- 2 Sélectionnez Fichier > Nouveau > Dossier (ou Fichier, si vous voulez ajouter des fichiers).
- 3 Sélectionnez le projet ou le dossier du projet auquel vous souhaitez ajouter les ressources liées.
- 4 Cliquez sur le bouton Options avancées.
- 5 Sélectionnez Lier au dossier dans le système de fichiers. Cliquez sur le bouton Variables.
- 6 Sélectionnez une variable de chemin définie ou cliquez sur Nouveau pour créer une variable de chemin.
Si vous avez sélectionné une variable de chemin définie, passez à l'étape 9. Si vous avez cliqué sur Nouveau, la boîte de dialogue Nouvelle variable s'affiche.
- 7 Saisissez le nom de la variable de chemin et saisissez ou recherchez l'emplacement du fichier ou du dossier.
Cliquez sur OK pour créer la variable de chemin.
- 8 Sélectionnez la nouvelle variable de chemin dans la boîte de dialogue Sélection d'une variable de chemin, puis cliquez sur OK.
- 9 Cliquez sur Terminer pour terminer la création du lien vers la ressource.

 *Vous pouvez également définir et gérer des variables de chemin dans les préférences du workbench de Flash Builder (dans la boîte de dialogue Préférences, sélectionnez Général > Espace de travail > Ressources liées).*

Ajout de dossiers de ressources au chemin source du projet

Pour partager des ressources entre plusieurs projets, placez-les dans des dossiers qui pourront ensuite être reliés aux différents projets à l'aide du chemin d'accès source de ces derniers. Il s'agit de la meilleure méthode de partage de ressources, telles que les classes, les composants MXML et les images. Les mises à jour de ces ressources sont immédiatement disponibles pour tous les projets qui les utilisent. Lorsque les projets sont compilés, les ressources partagées sont ajoutées au fichier SWC.

Ajout d'un dossier de ressources externes au chemin d'accès source

- 1 Sélectionnez un projet dans l'Explorateur de packages.
- 2 Sélectionnez Projet > Propriétés > Chemin de génération Flex (ou Chemin de génération ActionScript si vous utilisez un projet ActionScript).
- 3 Sur la page des propriétés du chemin de génération, cliquez sur l'onglet Chemin source.
- 4 Cliquez sur le bouton Ajouter un dossier.
- 5 Saisissez ou recherchez le chemin du dossier, puis cliquez sur OK.

Le dossier est ajouté au chemin source.

L'onglet de propriétés Chemin source vous permet également de modifier, supprimer ou changer l'ordre des éléments du chemin source.

Les dossiers qui sont ajoutés au chemin source sont signalés dans l'Explorateur de packages.

Alternatives à l'utilisation de références de projets

Les références de projet pouvant avoir une incidence sur l'ordre de génération, Flash Builder propose une alternative à leur utilisation.

Projets de bibliothèque Flex Méthode privilégiée de création d'une bibliothèque réutilisable. Flash Builder crée une référence de projet pour garantir que le projet SWC est généré avant le projet principal qui le contient sur le chemin d'accès à la bibliothèque. En outre, puisque Flash Builder l'ajoute au chemin de la bibliothèque, des indicateurs de code apparaissent dans le projet principal pour les classes du projet SWC.

Chemin source Méthode recommandée pour inclure du code dans le projet ne se trouvant pas sous la même structure de dossier. Elle active les indicateurs de code dans les fichiers du projet et dans les classes situées dans des fichiers liés. Le compilateur sait où trouver le code source. Vous pouvez ajouter au projet un nombre quelconque de chemins source. Ils s'affichent en tant que dossiers liés dans l'Explorateur de packages.

Affichage des propriétés des ressources

Dans l'Explorateur de packages, vous pouvez sélectionner une ressource et en afficher les propriétés.

- 1 Dans l'Explorateur de packages, sélectionnez une ressource.
- 2 Sélectionnez Fichier > Propriétés.

Projets ActionScript

Flash Builder vous permet de créer des projets ActionScript qui font appel à l'API Flash (et non à la structure Flex). Les outils du workbench de Flash Builder et l'éditeur ActionScript s'associent ainsi pour former un environnement de développement d'applications ActionScript, intégré et riche en fonctionnalités.

Les projets ActionScript ne possèdent pas de représentation visuelle dans Flash Builder. En d'autres termes, il n'existe pas de mode Création pour les applications ActionScript. Vous affichez les applications ActionScript en les compilant dans Flash Builder, puis en les exécutant dans Flash Player. Vous pouvez utiliser tous les outils de débogage.

Lorsque vous créez un projet ActionScript ou un fichier ActionScript autonome destiné à contenir des fonctions, une classe ou une interface, la perspective de développement Flex change pour prendre en charge l'éditeur ActionScript. Les principales vues associées à l'éditeur ActionScript sont les vues Structure et Erreurs.

Création de projets ActionScript

Lorsque vous créez un projet, l'assistant Nouveau projet ActionScript vous guide tout au long des différentes étapes et vous demande de préciser le type de projet à créer, le nom du projet, son emplacement, ainsi que d'autres options avancées.

1 Sélectionnez Fichier > Nouveau > Projet ActionScript.

2 Saisissez le nom du projet, puis spécifiez les éléments suivants.

Emplacement du projet L'emplacement par défaut est l'espace de travail actuel. Sous Windows, l'emplacement de l'espace de travail par défaut est C:\Documents and Settings\Développeur Flex\Adobe Flash Builder\. Sous Macintosh, l'emplacement de l'espace de travail par défaut est /Users/Développeur Flex/Adobe Flash Builder/. Pour choisir un emplacement de projet différent, désactivez l'option Utiliser l'emplacement par défaut.

Versión SDK Flex Choisissez la version par défaut ou spécifiez-en une. Vous pouvez aussi cliquer sur le lien Configurer les SDK Flex pour ajouter, modifier ou supprimer des SDK en accédant à la page principale des Préférences.

3 Cliquez sur Suivant pour définir les options avancées (sinon, cliquez sur Terminer).

Chemin source Précise le chemin de liaison des ressources externes à l'application. En présence d'un dossier de classes partagées ActionScript par exemple, vous pouvez créer un lien vers ce dossier en l'ajoutant au chemin source.

Chemin d'accès à la bibliothèque Précise le chemin d'accès de liaison des bibliothèques de ressources externes (fichiers SWC). Par défaut, le chemin d'accès à la bibliothèque des nouveaux projets ActionScript contient les fichiers playerglobal.swc et utilities.swc.

Dossier source principal Précise, par défaut, la racine du projet. Vous pouvez toutefois sélectionner un dossier différent au sein du projet. Vous pouvez parcourir la structure des dossiers du projet et, le cas échéant, créer un dossier pour la source.

Fichier de l'application principale Précise le nom du fichier ActionScript qui est le fichier de l'application principale. Par défaut, Flash Builder utilise le nom du projet comme nom de fichier de l'application principale. Vous pouvez le modifier.

Dossier de sortie Précise l'emplacement des fichiers de l'application compilée. Par défaut, il s'agit du dossier *bin*. Vous pouvez en sélectionner un autre.

URL du dossier de sortie Précise l'emplacement du serveur des fichiers de l'application compilée. Cette information est facultative.

4 Une fois les paramètres du projet ActionScript saisis, cliquez sur Terminer.

Création d'une classe ActionScript

Vous pouvez faire appel à un assistant Flash Builder pour créer rapidement des classes ActionScript pour les projets Flex et ActionScript. Cet assistant vous permet également de générer facilement des stubs pour des fonctions devant être implémentées.

1 Sélectionnez Fichier > Nouveau > Classe ActionScript.

2 Dans la boîte de dialogue, spécifiez les propriétés de base de la nouvelle classe, puis cliquez sur Terminer.

Une fois que vous avez cliqué sur Terminer, Flash Builder enregistre le fichier dans le package spécifié et le rouvre dans l'éditeur de code.

Si vous avez enregistré le fichier dans le projet actif ou dans le chemin source correspondant, Flash Builder affiche également le composant dans la vue Composants pour permettre son insertion rapide dans les applications. Pour plus d'informations, voir « [Ajout de composants en mode Création MXML](#) » à la page 192.

3 Rédigez la définition de la classe `ActionScript`.

Pour plus d'informations, voir *Simple Visual Components in ActionScript*.

Création d'une interface `ActionScript`

Vous pouvez faire appel à un assistant Flash Builder pour créer rapidement des interfaces `ActionScript` pour les projets Flex et `ActionScript`. Une *interface* est un ensemble de constantes et de méthodes que différentes classes peuvent partager.

- 1 Sélectionnez Fichier > Nouveau > Interface `ActionScript`.
- 2 Spécifiez les propriétés de base de la nouvelle interface dans la boîte de dialogue, puis cliquez sur Terminer.
- 3 Ajoutez à l'interface `ActionScript` les éventuelles constantes ou méthodes que les différentes classes partagent.

Génération des fonctions d'accessueur

Les fonctions des accesseurs `Get` et `Set` permettent de faire en sorte que les propriétés d'une classe restent privées pour cette classe. Les utilisateurs peuvent accéder à ces propriétés de la même manière qu'ils accèdent à une variable de classe (au lieu d'appeler une méthode de classe).

Flash Builder peut générer des fonctions d'accessueur `Get` et `Set` `ActionScript` pour les variables de classe. Lorsque vous générez des accesseurs `Get` et `Set`, Flash Builder propose les options suivantes :

- Rendre la variable de classe privée.
En règle générale, les variables de classe ont un accès privé.
- Renommer la variable de classe, en suggérant un trait de soulignement au début du nom.
Par convention, les variables de classe privée ont un trait de soulignement au début du nom.
- Renommer les fonctions d'accessueur.
- Préciser si vous voulez générer des fonctions d'accessueur `Get` et `Set`.
- Spécifier le placement de la fonction d'accessueur dans l'un des emplacements suivants :
 - Avant la première méthode
 - Après la dernière méthode
 - Avant les déclarations de variable
- Afficher un aperçu du code qui sera généré.

Génération de fonctions d'accessueur `Get` ou `Set`

- 1 Ouvrez un fichier `ActionScript` dans l'éditeur de source, puis placez le curseur sur une variable de classe.
- 2 Sélectionnez Source > Générer les méthodes `Get/Set` dans le menu Flash Builder ou le menu contextuel.
- 3 Dans la boîte de dialogue Générer les méthodes `Get/Set`, spécifiez les informations concernant les fonctions d'accessueur puis cliquez sur OK.

Remarque : pour afficher le code qui sera généré, sélectionnez *Aperçu* avant de cliquer sur OK.

Projets de bibliothèque

Les projets de bibliothèque vous permettent de créer des bibliothèques de code personnalisé que vous pouvez partager entre vos applications ou distribuer à d'autres développeurs. Un projet de bibliothèque génère un fichier SWC, c'est-à-dire un fichier archive pour les composants Flex ainsi que pour d'autres ressources. Par exemple, la structure Flex est contenue dans des fichiers SWC. Lorsque vous créez un projet Flex, les fichiers SWC de la structure Flex sont ajoutés au chemin de la bibliothèque du projet. Vous pouvez afficher et modifier le chemin de la bibliothèque dans la page de propriétés du chemin de génération du projet (pour les projets Flex, sélectionnez Projet > Propriétés > Chemin de génération Flex).

Un fichier SWF est archivé dans un fichier SWC contenant des composants, des ressources et un fichier catalog.xml, fichier manifeste des éléments contenus dans le fichier SWF. Le fichier SWF contient généralement un ou plusieurs composants et toute autre ressource requise. L'ajout de la bibliothèque à un projet vous permet d'utiliser ces composants et en active les indicateurs de code.

Les bibliothèques SWC constituent une méthode pratique d'assemblage et de distribution des composants, mais servent également de thèmes, l'aspect visuel des applications Flex. Un fichier de thème SWC contient un fichier CSS ainsi que toutes les ressources graphiques associées. Pour plus d'informations sur la création et l'utilisation des thèmes, voir About themes.

Les bibliothèques sont utiles si vous créez des composants intégralement dans ActionScript et que vous les utilisez en mode Création dans Flash Builder. Les composants ActionScript ne sont pas rendus visuellement en mode Création tant qu'ils ne sont pas compilés dans un fichier SWF. En ajoutant des composants ActionScript dans un projet de bibliothèque, vous créez un fichier SWF qui est contenu dans un fichier SWC. Vous pouvez ajouter la bibliothèque à un chemin d'accès à la bibliothèque d'un projet et les composants ActionScript sont rendus visuellement en mode Création lorsque vous les ajoutez à l'application.

Configuration de bibliothèques pour vos applications

Vous pouvez utiliser les bibliothèques SWC dans vos projets des façons suivantes.

Fusionnées dans l'application Lorsque vous ajoutez un fichier SWC au chemin d'accès à la bibliothèque du projet, les composants contenus dans la bibliothèque sont disponibles pour être utilisés dans l'application. Lorsque vous générez l'application, seuls les composants de bibliothèque que vous avez utilisés dans les faits sont compilés dans le fichier SWF de l'application. En d'autres termes, tout le code de votre application est fusionné dans un seul fichier SWF. C'est la façon la plus courante et la plus simple d'utiliser les composants de bibliothèques.

Externes à l'application Vous pouvez conserver les composants de bibliothèque séparément du fichier SWF compilé pour qu'ils ne soient pas fusionnés dans le fichier. Le compilateur résout tout le code contenu dans la bibliothèque utilisé par l'application, mais ne le fusionne pas dans le fichier SWF de l'application. L'avantage de cette méthode est de réduire la taille du fichier SWF de l'application. Les composants contenus dans le fichier SWC sont extraits et chargés dans la mémoire selon les besoins, au moment de l'exécution.

Bibliothèque partagée à l'exécution Dans les projets Flex uniquement, vous pouvez également utiliser des fichiers SWC en tant que bibliothèque partagée à l'exécution (Runtime Shared Library, RSL), qui s'apparente à une bibliothèque à liaison dynamique sur d'autres plateformes. Vous pouvez utiliser des fichiers SWC en tant que bibliothèque RSL si vous disposez d'un ensemble de composants qui sont utilisés par plusieurs applications.

Le partage de composants entre applications à l'aide de ce type de bibliothèque présente plusieurs avantages. Dans un premier temps, la bibliothèque est chargée en mémoire une fois, placée dans le cache, puis mise à disposition de toutes les applications faisant appel à ces composants. Dans un second temps, les composants se trouvant dans la bibliothèque sont chargés uniquement selon les besoins, ce qui réduit le temps de démarrage de l'application, car la taille de l'application est moindre. Le problème potentiel de cette méthode est que la bibliothèque partagée à l'exécution est

chargée en mémoire dans son intégralité, plutôt que les composants individuels que les applications utilisent. Pour plus d'informations sur l'utilisation des fichiers SWC en tant que RSL, voir Using Runtime Shared Libraries.

Création de projets de bibliothèque Flex

Lorsque vous créez un projet de bibliothèque, l'assistant Nouveau projet de bibliothèque Flex vous guide à travers les différentes étapes et vous demande d'indiquer le nom du projet, son emplacement, ainsi que le chemin de génération. Une fois le projet de bibliothèque créé, vous ajoutez des composants, spécifiez les éléments du projet de bibliothèque à inclure dans le fichier SWC, puis générez le projet afin de créer le fichier SWC. La première étape de la création d'un fichier SWC dans Flash Builder est la création d'un projet de bibliothèque Flex.

- 1 Sélectionnez Fichier > Nouveau > Projet de bibliothèque Flex.
- 2 Saisissez un nom de projet, puis précisez les éléments suivants.

Emplacement du projet L'emplacement par défaut est l'espace de travail actuel. Sous Windows, l'emplacement de l'espace de travail par défaut est C:\Documents and Settings\Développeur Flex\Adobe Flash Builder\. Sous Macintosh, l'emplacement de l'espace de travail par défaut est /Users/Développeur Flex/Adobe Flash Builder/. Pour choisir un emplacement de projet différent, désactivez l'option Utiliser l'emplacement par défaut.

Version SDK Flex Choisissez la version par défaut ou spécifiez-en une. Vous pouvez aussi cliquer sur le lien Configurer les SDK Flex pour ajouter, modifier ou supprimer des SDK sur la page principale Préférences.

Inclure les bibliothèques Adobe AIR Sélectionnez cette option si votre bibliothèque doit utiliser des fonctions AIR, telles que l'accès aux API AIR. Flash Builder change alors le chemin d'accès à la bibliothèque de ce nouveau projet de bibliothèque Flex de manière qu'il contienne les fichiers airglobal.swc et airframework.swc. Les projets Web Flex ne peuvent pas utiliser cette bibliothèque.

Ne sélectionnez pas cette option si vous créez une bibliothèque générique destinée à une utilisation exclusive dans une application Web Flex, ou dans une application Web ou AIR.

- 3 Cliquez sur Suivant.
- 4 (Facultatif) Définissez les informations concernant le chemin de génération. Vous pouvez, par exemple, ajouter des dossiers au chemin source du projet qui contient les composants à inclure dans le fichier SWC. Vous pouvez également ajouter d'autres projets, dossiers ou fichiers SWC de bibliothèque à inclure dans votre projet de bibliothèque. Voir « [Utilisation des fichiers SWC dans le cadre des projets](#) » à la page 71.
- 5 Une fois les paramètres du projet saisis, cliquez sur Terminer.

Ajout de composants à un projet de bibliothèque

Il existe plusieurs façons d'ajouter des composants à un projet de bibliothèque :

- Vous pouvez ajouter des composants, nouveaux ou existants personnalisés, des classes ActionScript et d'autres ressources au projet.
- Vous pouvez créer des liens avec des composants existant dans d'autres projets de l'espace de travail (voir « [Liaison à des ressources situées en dehors de l'espace de travail du projet](#) » à la page 64).
- Vous pouvez ajouter un dossier lié qui contient des composants au chemin source du projet de bibliothèque (voir « [Ajout de dossiers de ressources au chemin source du projet](#) » à la page 65).

Remarque : tous les composants que vous incluez dans le projet de bibliothèque doivent être associés au projet de bibliothèque (directement ou en tant que ressources liées).

Sélection d'éléments du projet de bibliothèque à inclure dans le fichier SWC

L'étape suivante de la création d'un fichier SWC de bibliothèque est la sélection des éléments (composants et ressources) à inclure dans le fichier SWC lors de sa création par le compilateur.

- 1 Sélectionnez Projet > Propriétés > Chemin de génération de bibliothèque Flex.

Les composants que vous avez ajoutés au projet (soit directement soit en les reliant) s'affichent dans l'onglet Classes.

- 2 Sélectionnez les classes de composants à inclure dans le fichier SWC.
- 3 (Facultatif) Sous l'onglet Ressources, sélectionnez les ressources à inclure dans le fichier SWC.
- 4 Une fois que vous avez effectué vos sélections, cliquez sur OK.

Création de projets de bibliothèque

Une fois que vous avez sélectionné les éléments à inclure dans le fichier SWC et si vous avez sélectionné l'option de génération automatique, le fichier SWC est compilé immédiatement et généré dans le dossier de sortie du projet. Si vous générez vos projets manuellement, vous pouvez créer le projet de bibliothèque quand vous le souhaitez en sélectionnant Projet > Générer le projet ou Générer tout.

La création du projet de bibliothèque génère un fichier SWC que vous pouvez partager avec d'autres applications ou utilisateurs.

Un fichier SWC est un fichier archive. Vous pouvez l'ouvrir grâce à n'importe quel utilitaire d'archivage, tel que WinZip. Les fichiers `library.swf` et `catalog.xml` sont contenus dans le fichier SWC, ainsi que les fichiers de propriétés et autre ressources incorporées.

Vous pouvez exporter la bibliothèque sous la forme d'un répertoire ouvert plutôt qu'en tant que fichier SWC. Vous exportez généralement une bibliothèque en tant que répertoire ouvert si vous prévoyez de faire appel au fichier `library.swf` dans le fichier SWC en tant que bibliothèque RSL.

Pour ce faire, vous devez définir les options `directory` et `output` du compilateur. Définissez l'option `output` comme le nom du répertoire à créer et l'option `directory` sur `true` pour indiquer que vous voulez ouvrir le répertoire et non un fichier SWC lorsque vous générez la bibliothèque. Pour modifier les options du compilateur, sélectionnez Projet > Propriétés > Compilateur de bibliothèque Flex et ajoutez les options au champ Arguments de compilateur supplémentaires, par exemple :

```
-directory=true -output=myOpenDir
```

Flash Builder crée un répertoire dans le projet nommé `myOpenDir` et y enregistre le contenu du fichier SWC.

Utilisation des fichiers SWC dans le cadre des projets

Pour pouvoir utiliser des fichiers SWC dans vos projets Flex, vous devez les ajouter au chemin d'accès à la bibliothèque du projet. Les fichiers SWC peuvent se trouver dans le projet, dans un projet de bibliothèque Flex, dans un dossier partagé dans l'espace de travail ou dans tout autre emplacement lié au projet (en utilisant un dossier partagé qui a été ajouté au chemin source du projet, par exemple).

Lorsque vous utilisez des fichiers SWC dans des applications, certaines options de configuration déterminent s'ils sont liés de manière statique ou dynamique à l'application, s'ils sont fusionnés dans le fichier SWF de l'application ou s'ils sont externes et si leur accès se fait séparément au moment de l'exécution.

Ajout d'un fichier SWC au chemin d'accès à la bibliothèque

- 1 Sélectionnez un projet dans l'Explorateur de packages, puis cliquez sur Projet > Propriétés > Chemin de génération Flex.

- 2 Cliquez sur l'onglet Chemin d'accès à la bibliothèque.
- 3 Sélectionnez l'une des options suivantes pour ajouter des fichiers SWC.
 - Ajouter un projet** Ajoute un projet de bibliothèque Flex.
 - Ajouter un dossier SWC** Permet d'ajouter un dossier contenant des fichiers SWC.
 - Ajouter un fichier SWC** Ajoute un fichier SWC compilé.
 - Ajouter un SDK Flex** Permet d'ajouter d'autres SDK Flex. Ce bouton est désactivé si le chemin d'accès à la bibliothèque du projet contient déjà un SDK Flex. Si vous supprimez le SDK Flex existant du chemin d'accès à la bibliothèque, le bouton est activé. Lorsque vous cliquez sur ce bouton, un nœud SDK Flex est ajouté, mais aucune invite ne précise lequel. Pour préciser quel SDK Flex utiliser, sélectionnez Projet > Propriétés > Compilateur Flex.
- 4 Saisissez ou recherchez l'emplacement du fichier SWC, du projet ou du dossier, puis sélectionnez-le. Cliquez sur OK.
Le fichier SWC, projet de bibliothèque ou dossier est ajouté au chemin d'accès à la bibliothèque.

Fusion du fichier SWC dans le fichier SWF de l'application au moment de la compilation

- 1 Sélectionnez un projet dans l'Explorateur de packages, puis cliquez sur Projet > Propriétés > Chemin de génération Flex.
- 2 Sous l'onglet Chemin d'accès à la bibliothèque, sélectionnez et développez l'entrée du fichier SWC pour en afficher les options.
- 3 Double-cliquez sur l'option Type de lien. La boîte de dialogue Options de l'élément de chemin d'accès à la bibliothèque apparaît.
- 4 Sélectionnez l'option Fusionné dans le code, puis cliquez sur OK.

Cette procédure équivaut à utiliser l'option `library-path` du compilateur.

Définition du fichier SWC en tant que fichier de bibliothèque externe

- 1 Sélectionnez un projet dans l'Explorateur de packages, puis cliquez sur Projet > Propriétés > Chemin de génération Flex.
- 2 Sous l'onglet Chemin d'accès à la bibliothèque, sélectionnez et développez l'entrée du fichier SWC pour en afficher les options.
- 3 Double-cliquez sur l'option Type de lien. La boîte de dialogue Options de l'élément de chemin d'accès à la bibliothèque apparaît.
- 4 Sélectionnez l'option Externe, puis cliquez sur OK.

Cette procédure équivaut à utiliser l'option `external-library-path` du compilateur.

Utilisation du fichier SWC en tant que bibliothèque RSL

- 1 Sélectionnez un projet dans l'Explorateur de packages, puis cliquez sur Projet > Propriétés > Chemin de génération Flex.
- 2 Sous l'onglet Chemin d'accès à la bibliothèque, sélectionnez et développez l'entrée du fichier SWC pour en afficher les options.
- 3 Double-cliquez sur l'option Type de lien. La boîte de dialogue Options de l'élément de chemin d'accès à la bibliothèque apparaît.
- 4 Sélectionnez l'option Bibliothèque RSL (Runtime Shared Library).
- 5 Entrez l'URL où la bibliothèque SWC doit résider lors du déploiement de l'application.

6 (Facultatif) Pour extraire le fichier SWF dans le fichier SWC lorsqu'il se trouve dans l'emplacement de déploiement, sélectionnez l'option Extraire automatiquement le fichier SWF dans le chemin de déploiement.

7 Cliquez sur OK.

L'utilisation des fichiers SWC en tant que bibliothèque RSL simplifie le processus d'utilisation manuelle de bibliothèques RSL. Pour ce faire, extrayez le fichier SWF du fichier SWC et définissez les valeurs de l'option `runtime-shared-library-path` du compilateur.

Pour plus d'informations sur l'utilisation des fichiers SWC en tant que bibliothèque RSL, voir Using Runtime Shared Libraries dans *Utilisation d'Adobe Flex 4*.

Génération de projets

Adobe® Flash® Builder™ génère et exporte automatiquement les projets vers des applications. Cette opération comporte la création de fichiers d'application et de bibliothèque, le placement des fichiers de sortie dans les emplacements appropriés et la signalisation de toute erreur survenue en cours de compilation.

Différentes options permettent d'ajuster les paramètres de génération. Vous pouvez ainsi déterminer la manière dont les projets sont intégrés aux applications. Vous pouvez par exemple définir des préférences pour la génération de projets individuels ou pour l'ensemble des projets de l'espace de travail, modifier le chemin de sortie ou l'ordre de génération et ainsi de suite. Vous pouvez en outre créer des instructions de génération personnalisées à l'aide d'outils tiers tels qu'Apache Ant.

Lorsque les applications sont prêtes à être publiées, vous pouvez en publier l'ensemble du code source ou uniquement certaines parties. A l'instar du code source HTML, le code source de l'application peut être visualisé dans un navigateur Web.

Présentation de la génération et de l'exportation des projets

Un déroulement de travail se compose de la génération de projets Flex et ActionScript, l'option Générer automatiquement étant activée. Au cours du développement, Flash Builder affiche les erreurs et les avertissements dans la vue Erreurs. A l'exécution de l'application, une version de débogage du fichier SWF est placée dans le dossier de sortie (bin) du projet avec les ressources requises et une enveloppe HTML. Cette génération contient des informations de débogage et n'est destinée qu'à l'usage des développeurs. Pour plus d'informations sur l'exportation de projets, voir « [Exportation et importation de projets](#) » à la page 53.

Une fois l'application prête à être déployée, l'assistant Exporter vers une version validée vous permet de créer une version optimisée de qualité finale de l'application. Cette opération enregistre le fichier SWF dans le dossier bin-release. Les informations de débogage étant supprimées, la taille du fichier diminue. S'agissant d'une version de production, elle peut être visionnée par les utilisateurs finaux. Pour les projets Adobe AIR, les applications AIR sont exportées dans un fichier AIR. La fonction Exporter vers une version validée permet de créer un fichier AIR signé numériquement que les utilisateurs doivent installer avant d'exécuter une application (à l'instar d'un fichier install.exe).

Les projets Adobe LiveCycle Data Services ES peuvent en revanche être compilés sur le serveur en y accédant. Pour plus d'informations, voir « [Gestion des projets](#) » à la page 49.

Aucune exportation n'est requise pour les projets de bibliothèque. Le fichier SWC créé par un projet de bibliothèque Flex se prête à la fois au développement et à la production. Pour plus d'informations, voir « [Projets de bibliothèque](#) » à la page 69.

Notions de base sur la génération

MXML et ActionScript 3.0 sont des langages *compilés*. Contrairement aux langages *interprétés* (JavaScript, par exemple) immédiatement exécutables par leurs environnements d'exécution, MXML et ActionScript 3.0 doivent être convertis dans un format compilé avant de pouvoir être exécutés dans Flash Player. Ce processus, ainsi que la création des fichiers de sortie correspondants, est désigné par le terme de *génération*.

Flash Builder génère automatiquement un projet à chaque modification ou enregistrement de l'un de ses fichiers. La possibilité de générer des applications manuellement ne signifie pas qu'il est nécessaire d'y recourir. Toutefois, la compréhension du processus de génération et des fichiers de sortie créés facilitera le diagnostic et la résolution des éventuels problèmes de configuration du projet.

Projets Flex Les fichiers source et les ressources intégrées (les images, par exemple) sont compilés dans un seul et unique format de sortie désigné par SWF. Les fichiers ayant ce format peuvent être exécutés directement dans l'application Flash Player autonome ou dans un navigateur Web par le biais d'un fichier d'*enveloppe HTML*, également créé par la génération. Ces fichiers sont créés dans le dossier de sortie du projet (ce dossier, nommé par défaut « bin », peut être renommé).

Projets LiveCycle Data Services ES LiveCycle Data Services ES offre la possibilité de créer des projets qui seront compilés sur le serveur. Le fichier d'application MXML, auquel il est accédé pour la première fois (par l'intermédiaire d'un navigateur Web), est compilé dans un fichier SWF.

Remarque : les projets LiveCycle Data Services ES peuvent être définis de manière telle à être compilés sur le serveur. Flash Builder les compile néanmoins au fur et à mesure du développement de vos applications, afin de permettre au compilateur de valider la syntaxe du code et d'afficher les messages d'erreur. Ces projets ne disposent pas d'option de dossier de sortie et Flash Builder ne génère aucun fichier de sortie.

Projets ActionScript 3.0 A l'instar des projets Flex, les projets ActionScript 3.0 compilent les fichiers source et les ressources intégrées dans un fichier SWF.

Projets de bibliothèque Flex Pour les projets de bibliothèque, les fichiers source sont des composants et des ressources apparentées. Lors de la génération de projets de bibliothèque, un fichier SWC est créé dans le dossier de sortie. Un fichier SWF est archivé dans un fichier SWC contenant des composants, des ressources et le fichier catalog.xml, fichier manifeste des éléments contenus dans le fichier SWF.

Génération automatique

En configuration autonome, Flash Builder génère vos applications automatiquement. En configuration plug-in, vous devez sélectionner l'option Générer automatiquement. Malgré la possibilité, décrite ci-dessus, de générer des applications manuellement, il ne devrait pas être nécessaire d'y recourir. La désactivation de la génération automatique empêche en outre le compilateur de détecter les erreurs de syntaxe et d'afficher les avertissements et les messages d'erreur au fur et à mesure de la saisie du code. La vue Erreurs ne vous fournira donc aucune information jusqu'à l'achèvement de la compilation du projet. Il est donc conseillé d'activer le mode de génération automatique de Flash Builder.

Options avancées pour la génération de projets

Les options avancées permettent de définir la durée et l'étendue des générations. Vous pouvez par exemple générer un seul projet ou un ensemble de projets dans l'espace de travail ou encore définir un jeu de documents (une collection) de projets à générer. Toutes les commandes de génération se situent dans le menu Projet, ainsi que l'illustre l'exemple suivant. Pour plus d'informations, voir « [Options de génération avancées](#) » à la page 81.



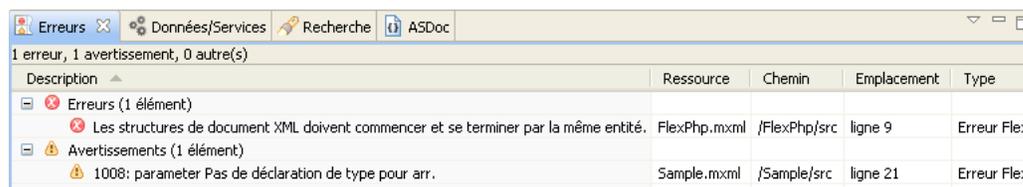
Le compilateur Flash Builder est incrémentiel. Il ne génère que les ressources qui ont été ajoutées ou concernées par une mise à jour et ignore toutes les autres, ce qui se traduit par un gain de temps et une utilisation parcimonieuse des ressources du système. Vous disposez toutefois de la possibilité de régénérer l'ensemble des ressources du projet en procédant à une *génération nettoyée*. Ce type de génération est indiqué lorsque l'application présente un comportement imprévisible au cours des tests effectués. Vous pouvez dans ce cas éliminer toute source potentielle de problèmes en ignorant et régénérant tous les fichiers du projet. Pour plus d'informations, voir « [Options de génération avancées](#) » à la page 81.

En présence de dépendances entre des projets distincts de l'espace de travail, le compilateur détermine automatiquement l'ordre dans lequel les projets ont été générés afin de résoudre correctement les dépendances. Vous pouvez toutefois remplacer l'ordre de génération par défaut en définissant manuellement l'ordre dans lequel les projets de l'espace de travail sont générés. Pour plus d'informations, voir « [Génération manuelle de projets](#) » à la page 81.

Vous pouvez en outre modifier le chemin d'accès à la génération, la liste des applications et les paramètres du compilateur pour chaque projet de l'espace de travail. Pour plus d'informations, voir « [Génération manuelle de projets](#) » à la page 81, « [Gestion des fichiers d'application du projet](#) » à la page 52 et « [Options de génération avancées](#) » à la page 81.

Affichage des erreurs de génération dans la vue Erreurs

Les erreurs détectées par le compilateur au cours de la génération s'affichent dans la vue Erreurs des perspectives Développement et Débogage, ainsi que dans l'éditeur de code, dans lequel les lignes contenant des erreurs sont signalées par un x, comme l'illustre l'exemple suivant.



Pour plus d'informations sur l'utilisation de la vue Erreurs, voir « [Utilisation de la vue Erreurs](#) » à la page 125.

Erreurs d'environnement Eclipse dans le fichier journal

Des erreurs générées par l'environnement Eclipse sont susceptibles de se produire. Elles surviennent généralement lorsque des ressources (des fichiers SWC, par exemple) ne sont pas trouvées à l'exécution. Dans ce cas, les messages d'erreur apparaissent dans le fichier journal des erreurs Eclipse. L'emplacement par défaut de ce fichier journal sous Windows est le suivant : \Documents and Settings*nom_utilisateur*\workspace\metadata\log. Sous Mac, l'emplacement par défaut se trouve également dans le répertoire de l'espace de travail. Les fichiers et dossiers commençant par un point y sont toutefois masqués par défaut.

Personnalisation des scripts de génération avec Apache Ant

Vous pouvez modifier et étendre le processus de génération standard en faisant appel à Apache Ant, outil de génération Java à code source libre. Pour plus d'informations sur la création de générateurs personnalisés, voir « [Personnalisation des générations avec Apache Ant](#) » à la page 83.

Options du compilateur Flex

Vous pouvez modifier les paramètres par défaut du compilateur Flex utilisé par Flash Builder. Pour afficher les paramètres par défaut et en modifier les valeurs, ouvrez la page des propriétés du compilateur Flex. Dans le menu Flash Builder, sélectionnez Projet > Propriétés > Compilateur Flex.

Version SDK Flex

Le SDK par défaut pour Flash Builder 4 est Flex 4.0. Cependant, si votre projet utilise une version spécifique, telle que Flex 3.4, Flash Builder compile les applications du projet à l'aide du SDK Flex spécifié.

Vous pouvez définir comme paramètre par défaut un SDK Flex spécifique ou vous pouvez définir que la compilation soit effectuée en utilisant la compatibilité avec Flex 3. La spécification de la compatibilité ascendante affecte certains comportements, tels que les règles de présentation, les marges intérieures et les espaces, les habillages ainsi que d'autres paramètres de style. Elle a également une incidence sur les règles d'analyse des fichiers de propriétés. La définition de la version de compatibilité n'applique pas toutes les différences existant entre les versions.

Pour plus d'informations, voir Backward compatibility.

Options d'Adobe Flash Player

La version par défaut de Flash Player utilisée par le compilateur est la version minimale requise par le SDK Flex utilisé pour la compilation.

Vous pouvez définir une version spécifique de Flash Player pour l'application. Les fonctionnalités nécessitant une version ultérieure de Flash Player ne seront pas compilées dans l'application.

Options du compilateur

Flash Builder fournit des cases à cocher pour les options suivantes du compilateur :

- Utiliser le moteur de texte Flash dans les composants MX

Le moteur de texte Flash (Flash Text Engine, FTE) est une bibliothèque qui fournit des contrôles de texte dotés d'un ensemble complet d'options de formatage. Tous les composants Spark du package `spark.components` prennent en charge FTE. Voir `Using embedded fonts`.

Certains contrôles MX fournissent une prise en charge de FTE. Les contrôles MX prenant en charge FTE utilisent les mêmes polices intégrées que les composants Spark utilisant FTE. Voir `Using FTE in MX controls`.

- Copier les fichiers non imbriqués dans le dossier de sortie
- Générer un fichier SWF accessible

Cette option active les fonctionnalités d'accessibilité lors de la compilation de l'application ou du fichier SWC. Pour plus d'informations sur l'utilisation des fonctionnalités d'accessibilité avec Flex, voir Accessible applications.

- Activer la vérification stricte

Lorsque la vérification stricte est activée, le compilateur imprime des appels de propriété et de fonction non définis. Il effectue également une vérification de type au moment de la compilation pour les affectations et les options fournies aux appels de méthode.

- Activer les avertissements

Cette option active des avertissements spécifiques. Pour plus d'informations, voir Viewing warnings and errors.

Vous pouvez également spécifier des arguments de compilateur disponibles avec la ligne de commande mxmcl compiler. Vous pouvez définir les valeurs de la plupart des options du champ Arguments de compilateur supplémentaires en utilisant la syntaxe de la ligne de commande. Pour plus d'informations sur la syntaxe pour la définition des options dans la boîte de dialogue Compilateur Flex, voir About the command-line compilers.

Dans le champ Arguments de compilateur supplémentaires, vous pouvez remplacer un chemin vers le répertoire du SDK par le jeton `{flexlib}`, comme l'illustre l'exemple suivant :

```
-include-libraries "${flexlib}/libs/automation.swc" "${flexlib}/libs/automation_agent.swc"
```

Enveloppe HTML

Outre les fichiers SWF, le compilateur Flash Builder génère également une enveloppe HTML que vous pouvez utiliser lors du déploiement de l'application. Les options suivantes sont disponibles :

- Générer le fichier d'enveloppe HTML
- Vérifier la version du lecteur cible

Lorsque cette option est activée, l'application compilée vérifie que la version de Flash Player est correcte.

Si l'option d'utilisation de l'installation rapide est activée, l'application exécute un fichier SWF dans le lecteur Flash Player existant pour mettre à niveau les utilisateurs vers la dernière version du lecteur.

- Activer l'intégration avec le navigateur

Cette option active la liaison profonde. La liaison profonde permet aux utilisateurs de parcourir leurs interactions avec l'application en utilisant les boutons Précédent et Suivant de leur navigateur.

Accès par ligne de commande aux compilateurs de structure Flex

Vous pouvez accéder directement par ligne de commande aux compilateurs de structure Flex (mxmcl et compc). Lorsque vous installez Flash Builder, l'invite de structure Flex se trouve dans le menu de démarrage Windows (Programmes > Adobe). Pour plus d'informations, voir About the command-line compilers dans *Utilisation d'Adobe Flex 4*.

Personnalisation des générations de projet

Flash Builder vous permet de générer des applications automatiquement en utilisant les paramètres de projet par défaut. Cette méthode est recommandée pour la génération d'applications. Vous pouvez toutefois personnaliser la génération de projets en fonction de vos besoins. Vous avez ainsi la possibilité par exemple de sélectionner un autre dossier de sortie par défaut ou de modifier les options du compilateur.

Jeu de composants (MX + Spark ou MX uniquement)

Par défaut, les projets Flex mettent à la disposition des applications l'ensemble des composants, c'est-à-dire les composants Spark fournis avec Flex 4 et les composants MX livrés avec Flex 3.

Il est possible toutefois que dans certains cas, vous ne souhaitiez faire appel qu'aux composants MX fournis avec Flex 3. Vous pouvez par exemple disposer d'un projet Flex 3 dans lequel vous ne voulez pas introduire les nouveaux composants Spark mais pour lequel vous souhaitez toutefois bénéficier des avantages des fonctionnalités introduites avec Flex 4 et Flash Builder 4 (nouvelle syntaxe d'états, fonctions de compilation améliorées et autres fonctionnalités de langage). Dans ce cas, activez l'option MX uniquement pour le jeu de composants.

L'activation de cette option supprime du chemin de génération toutes les bibliothèques associées à Spark. Lors de la conversion d'un projet Flex 4 en projet MX uniquement, Flash Builder n'apporte aucune modification au code du projet. Vous devrez mettre à jour le code manuellement afin d'en supprimer toute référence aux composants et aux bibliothèques Spark.

Liaison de la structure des applications

Par défaut, les classes d'application de la structure Flex 4 utilisent la liaison dynamique. Au lieu de compiler toutes les classes dans le fichier SWF de l'application (liaison statique), certaines classes sont chargées à partir de la bibliothèque partagée à l'exécution (Runtime Shared Library, RSL). Les applications créées avec une liaison dynamique sont dotées de fichiers SWF plus petits, ce qui permet de les télécharger plus rapidement. En revanche, ces applications utilisent plus de mémoire, toutes les classes de la structure étant chargées, y compris celles dont vous n'avez pas besoin. Pour plus d'informations, voir Bibliothèque partagée à l'exécution.

Vous pouvez modifier les propriétés d'un projet afin de personnaliser ce comportement pour toutes les applications d'un projet. Après avoir sélectionné un projet, dans le menu de Flash Builder, choisissez Projet > Propriétés > Chemin de génération Flex > Chemin d'accès à la bibliothèque.

Par défaut, Flash Builder utilise le comportement par défaut du SDK pour la liaison des structures. Pour Flex 4, il s'agit de la liaison dynamique des bibliothèques partagées à l'exécution (RSL) ; pour Flex 3, de la liaison statique. Utilisez le menu déroulant Liaison de structure pour remplacer le comportement par défaut.

Pour le SDK Flex 4, les options suivantes sont activées par défaut :

- Vérifier les condensés RSL

Vérifie que le condensé de la bibliothèque RSL correspond à celui stocké dans l'application au moment de la compilation lorsque l'application a été liée à la bibliothèque RSL interdomaines. Pour plus d'informations, voir About RSL digests.

- Utiliser les bibliothèques RSL locales de débogage lors du débogage

Utilise les bibliothèques RSL locales lors du débogage de l'application. L'utilisation des bibliothèques RSL locales vous permet d'intervenir dans les fichiers RSL de débogage. Cette option est ignorée lors de l'exportation d'une version validée.

- Déterminer automatiquement l'ordre de la bibliothèque, en fonction des dépendances

Si cette option est activée, Flash Builder détermine l'ordre des bibliothèques en fonction des dépendances au sein des bibliothèques. Pour personnaliser l'ordre des bibliothèques, désactivez cette option et indiquez l'ordre voulu à l'aide des boutons Haut et Bas.

Activation et désactivation des générations automatiques

En configuration autonome, Flash Builder génère vos projets automatiquement. En configuration plug-in, vous devez activer cette option. Flash Builder est conçu pour générer les projets automatiquement. La désactivation de cette option empêche le compilateur de détecter les erreurs de syntaxe et d'afficher les avertissements et les messages d'erreur au fur et à mesure de la saisie du code. Pour plus d'informations sur la génération manuelle de projets, voir « [Génération manuelle de projets](#) » à la page 81.

Effectuez l'une des opérations suivantes :

- Sélectionnez **Projet > Générer automatiquement**.
- Ouvrez la boîte de dialogue **Préférences** et sélectionnez **Général > Espace de travail**. Sélectionnez ou désélectionnez l'option **Générer automatiquement**.

Cette option a une incidence sur tous les projets de l'espace de travail.

Définition d'un dossier de sortie de projet

Lorsque vous créez un projet dans Flash Builder, le résultat de la création est généré par défaut dans le dossier de sortie. Les projets LiveCycle Data Services ES faisant appel à l'option de compilation sur le serveur dérogent à cette règle. A son exécution, l'application est en effet compilée sur le serveur.

Vous pouvez modifier le nom de ce dossier à la création du projet ou une fois le projet créé. Vous avez en outre la possibilité de créer un dossier ou de sélectionner un dossier existant de l'espace de travail.

- 1 Dans l'Explorateur de packages, sélectionnez un projet.
- 2 Cliquez avec le bouton droit de la souris (Windows) ou en maintenant la touche Ctrl enfoncée (Macintosh), puis sélectionnez **Propriétés** dans le menu contextuel.

La boîte de dialogue des propriétés du projet apparaît.

- 3 Sélectionnez la page **Chemin de génération Flex**.
- 4 Modifiez le dossier de sortie spécifié en saisissant un nouveau nom ou en recherchant et sélectionnant le dossier existant du projet.

***Remarque :** le dossier de sortie d'une application LiveCycle Data Services ES ne peut pas être modifié de cette manière. Son emplacement est en effet contrôlé par le serveur Flex et est accessible uniquement par le biais du fichier `Flex-config.xml` du projet.*

- 5 Cliquez sur **OK**.

Le dossier de sortie existant est remplacé par le nouveau dossier choisi.

***Important :** la modification du nom du dossier de sortie entraîne la suppression du dossier de sortie d'origine ainsi que de l'ensemble de son contenu. Vous devrez régénérer le projet afin de recréer les fichiers d'application SWF et d'enveloppe HTML.*

Modification d'un chemin de génération de projet

Chaque projet possède son propre chemin de génération composé du chemin source et du chemin de bibliothèque (les chemins de génération de projet de bibliothèque sont légèrement plus complexes. Pour plus d'informations, voir « [Projets de bibliothèque](#) » à la page 69). Le chemin source correspond à l'emplacement des fichiers source MXML et ActionScript du projet. Le chemin de bibliothèque correspond à l'emplacement des classes de la structure Flex de base ainsi que des composants Flex personnalisés que vous avez créés (sous la forme de fichiers SWC).

Modification du chemin source

- 1 Sélectionnez un projet dans l'Explorateur de packages.

- 2 Cliquez avec le bouton droit de la souris (Windows) ou en maintenant la touche Ctrl enfoncée (Macintosh), puis sélectionnez Propriétés dans le menu contextuel. La boîte de dialogue des propriétés du projet apparaît.
- 3 Sélectionnez la page Chemin de génération Flex (dans le cadre d'un projet ActionScript, sélectionnez la page Chemin de génération ActionScript).
- 4 Cliquez sur le bouton Ajouter un dossier pour ajouter un dossier au chemin source.
- 5 Saisissez le nom du dossier ou cliquez sur le bouton Parcourir pour rechercher l'emplacement des classes personnalisées.

Vous pouvez également utiliser des variables de chemin afin d'éviter de saisir le chemin complet du système de fichiers. Vous pouvez saisir le nom d'une variable existante ou en créer une. Pour plus d'informations, voir « [Création d'une variable de chemin](#) » à la page 80.

- 6 Apportez les modifications nécessaires au chemin source, puis cliquez sur OK.

Modification du chemin de la bibliothèque

- 1 Répétez les étapes 1 à 3 de la procédure précédente afin d'accéder à la page des propriétés Chemin de génération Flex.
- 2 Cliquez sur l'onglet Chemin d'accès à la bibliothèque.

Le chemin de la bibliothèque contient des références aux classes de la structure Flex contenues dans des fichiers SWC. Un fichier SWC est un fichier d'archivage pour les composants Flex et d'autres ressources (pour plus d'informations, voir « [Utilisation des fichiers SWC dans le cadre des projets](#) » à la page 71).

Vous pouvez modifier le chemin vers la structure ou, si vous avez créé des composants Flex personnalisés, ajouter de nouveaux dossiers ou fichiers SWC au chemin de la bibliothèque. Vous avez également la possibilité de supprimer des éléments du chemin.

- 3 Apportez les modifications nécessaires au chemin de la bibliothèque, puis cliquez sur OK.

Création d'une variable de chemin

Vous pouvez établir des liens vers des ressources en définissant des variables de chemin. Vous évitez ainsi de devoir saisir le chemin complet d'un dossier local ou d'un dossier réseau dans lequel vous enregistrez les fichiers. Définissez par exemple une variable de chemin nommée Classes, puis spécifiez le chemin vers un dossier du système de fichiers. La variable Classes peut ensuite être sélectionnée comme étant l'emplacement du nouveau dossier auquel la relation a été établie. Si le dossier est déplacé, il vous suffit de mettre à jour l'emplacement dans la variable de chemin définie pour que tous les projets reliés à la variable Classes puissent continuer à accéder aux ressources.

Définition ou création d'une variable de chemin

- 1 Sélectionnez un projet dans l'Explorateur de packages Flex.
- 2 Cliquez avec le bouton droit de la souris (Windows) ou en maintenant la touche Ctrl enfoncée (Macintosh), puis sélectionnez Propriétés dans le menu contextuel. La boîte de dialogue des propriétés du projet apparaît.
- 3 Sélectionnez la page Chemin de génération Flex (dans le cadre d'un projet ActionScript, sélectionnez la page Chemin de génération ActionScript).
- 4 Vous pouvez créer une variable pour chaque élément du chemin (dont les dossiers du chemin source et les dossiers SWC, les projets et les fichiers SWC du chemin de la bibliothèque). A titre d'exemple, cliquez sur le bouton Ajouter un dossier dans l'onglet Chemin source. La boîte de dialogue d'ajout d'un dossier apparaît.
- 5 Saisissez une variable de chemin conformément au format suivant : `${ nom_de_la_variable_de_chemin }`.

Remarque : la variable doit exister, sinon la saisie du chemin échoue. Pour obtenir la liste des variables de ressource existantes, sélectionnez Fenêtre > Préférences dans le menu principal, puis Général > Espace de travail > Ressources liées. Les variables de ressources liées peuvent également être gérées à partir de cette page de propriétés.

- 6 Cliquez sur OK pour ajouter la variable au chemin.

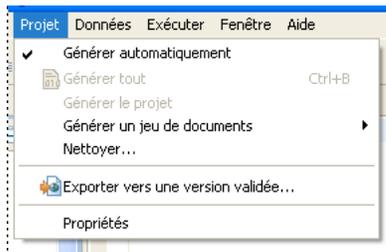
Options de génération avancées

Flash Builder offre des options avancées pour la personnalisation des générations de projet. Vous pouvez par exemple générer des projets manuellement, modifier l'ordre par défaut de génération des projets dans l'espace de travail et créer des générateurs personnalisés avec l'utilitaire Apache Ant.

Génération manuelle de projets

La génération manuelle de projets vous permet de contrôler la durée et l'étendue de la génération. Vous pouvez par exemple générer un seul projet ou un ensemble de projets dans l'espace de travail. Vous pouvez également définir un jeu de documents de projets ou de ressources de projet et ne générer que ces projets et ces ressources. Un jeu de documents est une collection de ressources d'espace de travail (projets, fichiers et dossiers) pouvant être sélectionnées et groupées en fonction de l'utilisation qui en est faite. Pour plus d'informations sur les jeux de documents, voir « [Création de jeux de documents](#) » à la page 27.

Les options de génération se situent dans le menu Projet, ainsi que l'illustre l'exemple suivant.



Génération d'un projet unique

- 1 Dans l'Explorateur de packages, sélectionnez le projet que vous souhaitez générer.
- 2 Sélectionnez Projet > Générer le projet dans le menu principal.

Le projet sélectionné est généré. Des fichiers d'application validés ou débogués (nouveaux ou mis à jour) sont ajoutés au dossier de sortie du projet.

Remarque : le programme vous invite à enregistrer les fichiers de projet qui ne l'ont pas encore été avant le début de la génération. Pour éviter l'affichage de cette invite, activez l'enregistrement automatique des fichiers avant le début de la génération dans les préférences de l'espace de travail.

Génération de tous les projets dans l'espace de travail

- ❖ Sélectionnez Projet > Générer tout dans le menu principal.

Tous les projets de l'espace de travail sont générés. Les fichiers d'application sont ajoutés aux dossiers de sortie des projets respectifs. Dans la mesure où vous n'avez pas défini l'enregistrement automatique des fichiers avant le début de la génération, le programme vous invite à les enregistrer.

Génération d'un jeu de documents

Utilisez l'une des méthodes suivantes :

- Sélectionnez **Projet > Générer un jeu de documents > Sélectionner un jeu de documents** dans le menu principal. Cliquez sur **Nouveau** pour créer un jeu de documents. Pour plus d'informations sur la création d'un jeu de documents, voir « [Création de jeux de documents](#) » à la page 27.
- Choisissez un jeu de documents existant en sélectionnant **Projet > Générer un jeu de documents > Sélectionner un jeu de documents** dans le menu principal.

Tous les projets du jeu de documents sont générés. Les fichiers d'application sont ajoutés au dossier de sortie du projet.

Enregistrement automatique des ressources du projet

Lorsque vous générez des projets manuellement, le programme vous invite à enregistrer toutes les ressources avant de débiter la génération. Pour éviter l'affichage de cette invite, activez l'enregistrement automatique des ressources du projet dans les préférences de l'espace de travail.

- 1 Ouvrez la boîte de dialogue **Préférences** et sélectionnez **Général > Espace de travail**.
- 2 Sélectionnez l'option **Sauvegarder automatiquement avant la génération**.
- 3 (Facultatif) Vous pouvez modifier la fréquence d'enregistrement des ressources en saisissant une valeur (en minutes) dans la zone de texte **Intervalle de sauvegarde du plan de travail**.

Exécution d'une génération nettoyée

Une fois un projet créé, les générations ultérieures n'impliqueront que les ressources ajoutées ou modifiées. Une génération nettoyée permet de forcer le compilateur Flash Builder à régénérer toutes les ressources d'un projet. Vous pouvez y recourir par exemple pour éliminer toutes les sources potentielles d'un problème survenu au cours du test de l'application.

- 1 Sélectionnez **Projet > Nettoyer** dans le menu principal.
- 2 Choisissez le ou les projets dont vous souhaitez éliminer les fichiers de génération pour les régénérer.
- 3 Cliquez sur **OK**.

Modification de l'ordre de génération d'un projet

Flash Builder permet de créer des relations entre les projets ouverts dans l'espace de travail. Vous pouvez par exemple importer les classes **ActionScript** d'un projet à l'autre. La création de relations entre les projets a une incidence sur l'ordre dans lequel les projets sont générés.

Par défaut, le compilateur génère des projets reliés dans l'ordre garantissant la génération correcte de tous les projets. Prenons l'exemple de deux projets, dont le premier fait référence à des classes contenues dans le second : le second projet est généré en premier. Dans la plupart des cas, le compilateur génère les projets dans l'ordre requis et les applications sont créées correctement sans qu'aucune intervention ne soit nécessaire.

Vous avez cependant la possibilité de modifier l'ordre de génération. Cette modification peut s'avérer nécessaire par exemple si vous avez créé un générateur personnalisé Ant et que vous l'avez associé à un projet de l'espace de travail que vous devez générer avant les autres projets. Pour plus d'informations sur la création de générateurs personnalisés, voir « [Personnalisation des générations avec Apache Ant](#) » à la page 83.

- 1 Ouvrez la boîte de dialogue **Préférences** et sélectionnez **Général > Espace de travail > Ordre de génération**.

La boîte de dialogue **Ordre de génération** présente les options suivantes.

Utiliser l'ordre de génération par défaut L'ordre de génération par défaut dépend des dépendances entre les projets. Il est géré par le compilateur.

Ordre de génération du projet Vous pouvez définir manuellement l'ordre dans lequel tous les projets de l'espace de travail doivent être générés. Il est également possible de supprimer un projet de la liste définissant l'ordre de génération. Dans ce cas, le projet sera généré après tous les autres projets de la liste.

Itérations maximales lors de la génération avec des cycles En présence de références cycliques dans un projet (ce que nous vous conseillons d'éviter), vous pouvez définir le nombre de tentatives de génération, afin de permettre au compilateur de générer tous les projets correctement. Par défaut, le nombre maximal d'itérations est 10.

- 2 Modifiez si nécessaire l'ordre de génération, puis cliquez sur OK.

Personnalisation des générations avec Apache Ant

La création d'un générateur personnalisé vous permet de modifier et d'étendre le processus de génération standard. Flash Builder comporte un script de génération standard intervenant dans la compilation des applications. Vous pouvez, le cas échéant, créer des scripts de génération personnalisés grâce à Apache Ant, outil de génération Java à code source libre.

Le développement de scripts de génération Ant dépasse le cadre de ce guide. Cette rubrique illustre toutefois comment créer un générateur personnalisé et l'appliquer à vos projets.

Vous pouvez appliquer les générateurs personnalisés à tous les types de projet Flash Builder.

Création d'un générateur

- 1 Dans l'Explorateur de packages, sélectionnez un projet. Cliquez avec le bouton droit de la souris (Windows) ou en maintenant la touche Ctrl enfoncée (Macintosh) pour afficher le menu contextuel et sélectionnez Propriétés.
- 2 Sélectionnez la page de propriétés Générateurs. La liste sera plus ou moins longue en fonction du nombre de plug-ins que vous utilisez. Flash Builder fournit un générateur nommé Flex qui ne peut pas être modifié.
- 3 Sélectionnez Nouveau.
- 4 Dans la boîte de dialogue Sélection d'un type de configuration, choisissez le type de configuration approprié. Flash Builder prend en charge le type Programme. Sélectionnez-le, puis cliquez sur OK pour poursuivre. Définissez les propriétés du nouveau générateur dans la page des propriétés et référencez le script Ant (fichier XML).
- 5 Cliquez sur OK pour appliquer les valeurs définies au projet.

Pour plus d'informations sur l'utilisation des scripts de génération Ant, voir la documentation Eclipse disponible à l'adresse help.eclipse.org/help31/index.jsp.

Utilisation de plusieurs SDK dans Flash Builder

Flash Builder permet de modifier la version du SDK utilisé pour la compilation des projets. Vous pouvez sélectionner le SDK à la création ou en cours de développement du projet.

Le SDK se compose d'une structure et du compilateur. Le SDK Flex 2.0.1 signifie que vous utilisez la version 2.0.1 des fichiers SWC de la structure Flex et la version 2.0.1 du compilateur Flex. Vous ne pouvez pas utiliser le compilateur Flex 3 avec les fichiers SWC de la structure Flex 2.0.1, par exemple.

La possibilité de modifier le SDK est utile en présence d'un projet développé avec Flex Builder 2.0.1 (qui utilise le SDK Flex 2.0.1) alors que vous exécutez la version 4 de Flash Builder (qui utilise par défaut le SDK Flex 4). En sélectionnant un SDK antérieur pour la génération, vous pouvez préserver la compatibilité avec la dernière version du SDK des projets qui n'ont pas été mis à jour. En outre, si vous travaillez sur un projet utilisant le SDK Flex 2.0.1 mais vous souhaitez faire appel à des fonctions de Flash Builder 4 (restructuration du code, par exemple), vous pouvez mettre à niveau Flash Builder, puis sélectionner un SDK antérieur comme SDK par défaut.

La modification du SDK après le développement d'un projet entraîne une régénération complète, et non une génération incrémentielle. Flash Builder signale alors toute différence qui aurait généré des erreurs de compilateur, tout comme si le projet avait été développé avec le SDK d'origine. Si vous utilisez par exemple un composant AdvancedDataGrid (introduit dans le SDK Flex 3), puis modifiez le SDK du projet en sélectionnant la version 2.0.1, Flash Builder signale que la classe AdvancedDataGrid est inconnue.

Flash Builder régénère également tous les fichiers de prise en charge des projets, dont les fichiers de gestion de l'historique et de liaisons profondes utilisés par l'enveloppe HTML. Pour les projets SDK Flex 2.0.1, Flash Builder crée les fichiers compatibles de gestion des historiques history.swf, history.html et history.js dans le répertoire des modèles HTML. Pour les projets SDK Flex 3, Flash Builder crée les fichiers compatibles de liaisons profondes history.htm et history.js et historyFrame.html dans le répertoire html-templates/history.

Les options présentes dans Flash Builder varient en outre en fonction du SDK sélectionné. Si vous ajoutez par exemple un module à un projet utilisant le SDK Flex 2.0.1, Flash Builder ne vous propose pas d'optimiser le module. Cette opération devra être effectuée manuellement.

Pour plus d'informations sur les différences entre les SDK Flex 3 et Flex 2.0.1, voir Backward compatibility dans *Utilisation d'Adobe Flex 4*.

Flash Builder utilise le SDK par défaut pour les nouveaux projets créés. Le SDK par défaut est la dernière version fournie avec Flash Builder. Vous pouvez toutefois sélectionner tout SDK répertorié dans la liste des SDK disponibles dans Flash Builder.

Lorsque vous créez un projet de bibliothèque Flex ou un projet ActionScript, vous pouvez sélectionner le SDK à utiliser dans les boîtes de dialogue Nouveau projet de bibliothèque Flex et Nouveau projet ActionScript.

Ajout d'un nouveau SDK Flex à la liste des SDK disponibles

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Flash Builder > SDK Flex installés.
Elle répertorie les SDK installés. Le nom du SDK par défaut est coché.
- 2 Cliquez sur Ajouter.
- 3 Saisissez l'emplacement du SDK dans le champ Emplacement du SDK Flex.
- 4 Attribuez-lui un nom dans le champ Nom du SDK Flex. Ne saisissez pas le nom d'un SDK existant.
- 5 Cliquez sur OK pour enregistrer vos modifications.
- 6 Cliquez de nouveau sur OK pour ajouter le nouveau SDK à la liste des SDK disponibles. Cette liste est gérée dans l'espace de travail Flash Builder pour tous les projets Flex. Au prochain projet créé, la liste des SDK disponibles comportera également ce nouveau SDK.

Modification de la version de SDK du projet en cours

- 1 Sélectionnez Projets > Propriétés.
- 2 Choisissez Compilateur Flex.
- 3 Cliquez sur Utiliser un SDK spécifique.
- 4 Sélectionnez un SDK dans la liste déroulante. Si elle ne contient pas le SDK recherché, cliquez sur le lien Configurer les SDK Flex.
- 5 Cliquez sur OK.

Flash Builder applique le nouveau SDK au projet en cours. Des erreurs et des avertissements peuvent survenir si le projet utilise un code qui n'est pas compatible avec le nouveau SDK.

Sélection d'un nouveau SDK par défaut

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Flash Builder > SDK Flex installés.
Le nom du SDK par défaut est coché.
- 2 Cochez la case correspondant à un autre SDK pour le définir comme SDK par défaut. Ce SDK sera appliqué à tout projet, y compris celui en cours, pour lequel l'option Utiliser le SDK par défaut a été sélectionnée dans la boîte de dialogue Compilateur Flex. Un projet paramétré pour utiliser un SDK spécifique ne sera pas concerné par cette modification, et ce même si vous modifiez le SDK par défaut de l'espace de travail.
- 3 Cliquez sur OK pour enregistrer vos modifications.
- 4 Cliquez à nouveau sur le bouton OK.

Publication du code source

Une fois qu'une application est prête à être publiée, Flash Builder vous permet de définir si les utilisateurs peuvent visionner le code source ainsi que les ressources de l'application. A l'instar du langage HTML, les utilisateurs peuvent accéder à la source et la visionner dans un navigateur Web en sélectionnant Afficher la source dans le menu contextuel. Le visualiseur de source met en forme et colore le code afin d'en faciliter la lecture. Il constitue également un moyen pratique de partager le code avec d'autres développeurs Flex et ActionScript 3.0.

Activation de l'option d'affichage de la source

- 1 Ouvrez l'ensemble du projet d'application dans l'éditeur, puis sélectionnez Projet > Exporter vers une version validée.
- 2 Sélectionnez Activer l'affichage de la source ou, pour les projets ActionScript, Inclure la source.
- 3 Cliquez sur Choisir des fichiers source.
- 4 Dans la boîte de dialogue Publier la source de l'application, sélectionnez le ou les fichiers de l'application à inclure dans le menu Afficher la source. Par défaut, le fichier principal de l'application est sélectionné.
***Important :** pour les projets pour serveurs, vous pouvez sélectionner le dossier des services à l'exportation des fichiers source. L'exportation de fichiers implémentant des services a des implications en termes de sécurité. Ces fichiers peuvent donner accès à votre base de données, noms d'utilisateurs et mots de passe inclus. Voir Exportation des fichiers source vers la version validée d'une application.*
- 5 (Facultatif) Modifiez le dossier de sortie de la source. Par défaut, un dossier d'affichage de la source est ajouté au dossier de sortie du projet.
- 6 Cliquez sur OK.

Les utilisateurs exécutant l'application peuvent accéder au code source en sélectionnant Afficher la source dans le menu contextuel. Le code source est affiché dans le navigateur Web par défaut sous forme d'arborescence reflétant la structure des ressources (package, dossiers et fichiers) contenues dans l'application (que vous avez décidé de publier). Sélectionnez un élément source pour en afficher le code dans le navigateur. Les utilisateurs peuvent également télécharger tous les fichiers source en cliquant sur le lien vers le fichier Download.zip.

***Remarque :** les restrictions de sécurité d'Internet Explorer risquent d'entraver l'affichage du code source sur votre ordinateur local. Vous devrez dans ce cas déployer l'application sur un serveur Web.*

Ajout du menu d'affichage de la source aux projets ActionScript

Dans les projets Flex, l'ajout de l'élément de menu Afficher la source à l'application s'effectue par l'intermédiaire de l'assistant Exporter vers une version validée. Dans les applications ActionScript, cette option doit être ajoutée manuellement.

La fonction suivante, contenue dans la structure Flex, peut être utilisée dans un constructeur d'application ActionScript pour activer le menu d'affichage de la source :

```
com.adobe.viewsource.ViewSource.addItem(obj:InteractiveObject, url:String,  
hideBuiltins:Boolean = true)
```

L'exemple suivant illustre l'utilisation de cette fonction dans les applications ActionScript :

```
package {  
    import flash.display.MovieClip;  
    import com.adobe.viewsource.ViewSource;  
  
    public class MyASApp extends MovieClip  
    {  
        public function MyASApp()  
        {  
            ViewSource.addItem(this, "srcview/index.html");  
  
            // ... additional application code here  
        }  
    }  
}
```

Cet exemple se base sur l'emplacement par défaut du dossier source (srcview). En cas de modification de cet emplacement, veillez à ce que le code utilise l'emplacement correct.

Génération par ligne de commande de Flash Builder

Flash Builder fournit la tâche Ant <fb.exportReleaseBuild>. Utilisez-la pour implémenter des générations par ligne de commande qui synchronisent les paramètres de génération d'un développeur avec une génération nocturne. Vous pouvez également utiliser la tâche <mxmcl> dans les scripts personnalisés pour la génération nocturne.

Tâche <fb.exportReleaseBuild>

La tâche <fb.exportReleaseBuild> garantit que les paramètres de génération nocturne correspondent exactement à ceux utilisés par les développeurs pendant leur travail de la journée.

Par exemple, si un développeur modifie le chemin de la bibliothèque d'un projet Flex, le nouveau chemin est écrit dans le projet Flash Builder. A son exécution par l'ordinateur de génération nocturne, la tâche <fb.exportReleaseBuild> charge le projet Flash Builder et tous ses paramètres.

La tâche <fb.exportReleaseBuild> présente un autre avantage : elle prend automatiquement en charge les tâches de maintenance supplémentaires qui font normalement partie d'une génération Flash Builder. Par exemple :

- La compilation automatique des projets de bibliothèque associés
- La copie de ressources, telles que les fichiers JPEG, dans le répertoire de sortie
- La copie du modèle HTML, y compris le remplacement des macros en fonction des résultats de compilation (comme la largeur et la hauteur)

Remarque : la tâche <fb.exportReleaseBuild> nécessite l'installation de Flash Builder sur l'ordinateur de génération nocturne.

Tâche <mxmcl>

Si vous écrivez un script personnalisé qui fait appel à la tâche <mxmcl> (un script Ant, par exemple), vous n'avez pas besoin d'installer Flash Builder sur l'ordinateur de génération. Le SDK Flex doit toutefois s'y trouver. L'ordinateur de génération peut par conséquent se trouver sur une plateforme Linux.

En revanche, l'inconvénient de cette approche est que vous disposez de deux jeux de paramètres de génération à synchroniser : le premier, utilisé par les développeurs au cours de la journée, se trouve dans Flash Builder et le second, sur l'ordinateur de génération nocturne.

Utilisation de la tâche <fb.exportReleaseBuild>

- 1 Installez Flash Builder sur un ordinateur de génération.
- 2 Ecrivez `build.xml` avec `fb.exportReleaseBuild` comme cible. Par exemple :

```
<?xml version="1.0"?>
<project default="main">
  <target name="main">
    <fb.exportReleaseBuild project="MyProject" />
  </target>
</project>
```

`build.xml` indique qu'une génération par ligne de commande du projet Flex doit être exécutée avec les paramètres enregistrés dans les fichiers du projet. Pour plus d'informations sur les paramètres disponibles, voir « [Paramètres de la tâche fb.exportReleaseBuild](#) » à la page 88.

- 3 Créez un script de génération nocturne qui demande à Eclipse de rechercher un fichier de génération et d'exécuter sa cible.

Les exemples suivants indiquent `build.xml` comme fichier de génération, qui exécute `MyTarget`.

Si le script de la génération nocturne se trouve sur une plateforme Macintosh, vous pouvez exécuter le script suivant :

```
WORKSPACE="$HOME/Documents/Adobe Flash Builder"

# works with either FlashBuilder.app or Eclipse.app
"/Applications/Adobe Flash Builder/FlashBuilder.app/Contents/MacOS/FlashBuilder" \
  --launcher.suppressErrors \
  --noSplash \
  -application org.eclipse.ant.core.antRunner \
  -data "$WORKSPACE" \
  -file "$(pwd)/build.xml" MyTarget
```

Si la génération nocturne est effectuée sur une plateforme Windows, vous pouvez exécuter le fichier de commandes suivant :

```
set WORKSPACE=%HOMEPATH%\Adobe Flash Builder

REM works with either FlashBuilderC.exe or eclipsec.exe
"C:\Program Files\Adobe\Adobe Flash Builder\FlashBuilderC.exe" ^
  --launcher.suppressErrors ^
  --noSplash ^
  -application org.eclipse.ant.core.antRunner ^
  -data "%WORKSPACE%" ^
  -file "%cd%\build.xml" MyTarget
```

Propriété Ant fb.running

La propriété Ant fb.running a la valeur true lorsque Flash Builder est en cours d'exécution. Vous pouvez utiliser cette propriété lorsque vous exécutez des scripts dans Flash Builder. Par exemple :

```
<target name="myFlashBuilderTasks" if="fb.running">
  <fb.exportReleaseBuild ... />
</target>
```

Tâches Ant d'Eclipse

Eclipse propose plusieurs tâches Ant que vous pouvez incorporer comme cibles dans le script de génération. Par exemple :

- eclipse.incrementalBuild
- eclipse.refreshLocal
- eclipse.convertpath

Consultez la documentation d'Eclipse pour en savoir plus sur ces scripts.

Paramètres de la tâche fb.exportReleaseBuild

Attribut	Description	Obligatoire ?	Valeur par défaut
project	Projet à générer. Indiquez le nom d'un projet dans l'espace de travail Flash Builder, sans chemin d'accès. Par exemple : « MonProjetFlex ».	Oui	Sans objet
application	Nom de l'application à compiler. Vous pouvez spécifier simplement le nom de l'application, sans chemin d'accès ou extension (app1, par exemple). Pour éviter toute ambiguïté de nom, vous pouvez spécifier un chemin d'accès complet par rapport à la racine du projet (src/app1.mxml, par exemple). Pour compiler toutes les applications, spécifiez « * » ou omettez cet attribut. Lorsque vous exécutez un projet AIR, vous ne pouvez spécifier qu'une seule application. La valeur « * » n'est pas autorisée.	Non	Application par défaut du projet.
publishsource	Indique si la source de l'application doit être publiée, ce qui permet à l'utilisateur d'afficher les fichiers source à l'aide du menu contextuel Afficher la source.	Non	false
locale	Indique la langue, par exemple fr-FR. Cette valeur est transmise au compilateur à l'aide de l'identifiant - locale du compilateur. Si spécifiée, cette valeur remplace la langue indiquée dans le champ Arguments de compilateur supplémentaires de Flash Builder.	Non	Sans objet
destdir	Dossier de sortie. Le dossier peut être un chemin relatif ou absolu. Si vous spécifiez un chemin relatif, il est relatif par rapport à la racine du projet. Si vous compilez un projet AIR, le dossier est un répertoire temporaire qui est supprimé après la création du fichier .air.	Non	bin-release
failonerror	Indique si des erreurs de compilation provoquent l'échec de la génération.	Non	true

Attribut	Description	Obligatoire ?	Valeur par défaut
verbose	La tâche <fb.exportReleaseBuild> génère des informations supplémentaires. Par exemple, elle répertorie les fichiers qui ont été assemblés dans le fichier AIR et la durée de chaque étape du processus.	Non	false
package	Pour les projets AIR uniquement : indique si le résultat doit être assemblé dans un fichier .air ou .airi. Si la valeur est true, un fichier .air ou .airi est créé et le répertoire de sortie temporaire (par défaut, le répertoire bin-release défini par l'attribut destdir) est supprimé. Si la valeur est false, aucun fichier .air ou .airi n'est créé et le répertoire intermédiaire reste intact après la compilation.	Non	true
destfile	Pour les projets AIR uniquement : le nom d'un fichier .air ou .airi à créer. Vous pouvez spécifier un chemin relatif ou absolu. Si vous spécifiez un chemin relatif, il est relatif par rapport à la racine du projet.	Non	appname.air ou appname.airi (dans la racine du projet)
certificate	Pour les projets AIR uniquement : le chemin d'accès au certificat utilisé pour signer le fichier AIR.	Non	Si cet argument est omis, un fichier .airi non signé est généré. Il pourra être signé ultérieurement.
password	Pour les projets AIR uniquement : le mot de passe du certificat qui est utilisé pour signer le fichier AIR. Si cet argument est omis, un message d'erreur s'affiche. Avertissement : l'utilisation d'une valeur littérale pour un mot de passe risque de compromettre la sécurité.	Non	Sans objet
timestamp	Pour les projets AIR uniquement : indique si le fichier AIR généré inclut un horodatage.	Non	false

Assistant Exporter vers une version validée

Lorsque vous exécutez l'assistant Exporter vers une version validée (Projet > Exporter vers une version validée), les paramètres que vous définissez dans l'assistant sont enregistrés dans le fichier .actionScriptProperties. Une génération par ligne de commande qui utilise la tâche fb.exportReleaseBuild récupère les paramètres de l'assistant. L'assistant Exporter vers une version validée enregistre les paramètres suivants :

- Afficher la source

Les fichiers source que vous spécifiez pour l'option Afficher la source sont enregistrés. Si vous définissez le paramètre publishsource sur fb.exportReleaseBuild, l'assistant inclut ces fichiers comme fichiers source pouvant être affichés.

Important : pour les projets pour serveurs, vous pouvez sélectionner le dossier des services à l'exportation des fichiers source. L'exportation de fichiers implémentant des services a des implications en termes de sécurité. Ces fichiers peuvent donner accès à votre base de données, noms d'utilisateurs et mots de passe inclus. Voir Exportation des fichiers source vers la version validée d'une application.

- Pour les projets AIR, tous les fichiers de sortie supplémentaires dont vous déterminez la prise en compte avec le fichier AIR ou AIRI dans l'assistant.

Exécution des générations par ligne de commande sur Linux et d'autres plateformes

La tâche `<fb.exportReleaseBuild>` est uniquement prise en charge sur des plateformes Windows et Mac.

Si vous rédigez toutefois un script de génération pour une autre plateforme, utilisez l'option `-dump-config` dans le compilateur `mxmxmlc` ou `compc` pour écrire les paramètres de configuration du compilateur dans un fichier. L'option `-load-config` vous permettra ensuite de lire les options de configuration.

Modifiez les paramètres de configuration du fichier en fonction de vos besoins. Par exemple, changez `<debug>true</debug>` en `<debug>>false</debug>` pour que la génération nocturne génère une version validée.

Exécution d'une génération par ligne de commande à l'aide des paramètres du compilateur Flash Builder

- 1 Dans Flash Builder, sélectionnez **Projet > Propriétés > Compilateur Flex**.
- 2 Dans le champ **Arguments de compilateur supplémentaires**, indiquez l'argument suivant :
`-dump-config nom_du_chemin`, où `nom_du_chemin` spécifie le chemin d'accès absolu à un fichier du système.
- 3 Appliquez les modifications dans la fenêtre du projet.
Les paramètres du compilateur sont écrits dans le fichier spécifié. Supprimez l'argument `-dump-config` après avoir vérifié que le fichier a été écrit.
- 4 Modifiez les paramètres de configuration comme il convient.
- 5 Dans le script de génération, exécutez le compilateur afin qu'il intègre les paramètres enregistrés du compilateur :
`mxmxmlc -load-config nom_du_chemin`

Restrictions pour l'exécution des générations par ligne de commande

Certaines restrictions s'appliquent à l'exécution des générations par ligne de commande à l'aide de la tâche `<fb.exportReleaseBuild>`.

Exécution des générations par ligne de commande sur des plateformes Mac avec Eclipse 3.4

Dans Eclipse 3.4 sur des plateformes Mac, la génération sans affichage échoue si le chemin d'installation d'Eclipse comporte des espaces.

Exécution des générations par ligne de commande sur des plateformes 64 bits

Flash Builder s'exécute sur des plateformes qui implémentent Java 32 bits. Pour exécuter une génération par ligne de commande sur des plateformes prenant en charge Java 64 bits (Mac OS X Snow Leopard, par exemple), ajoutez `-d32` aux options de ligne de commande qui sont transmises à Java. Par exemple :

```
java -d32 ...
```

Exécution des applications

Le test d'une application dans Adobe® Flash® Builder™ consiste à exécuter les fichiers SWF de l'application dans un navigateur Web ou dans la version autonome de Flash Player. En présence d'erreurs dans l'application, les outils de débogage vous permettent : de placer des points d'arrêt dans le code et de les gérer ; de contrôler l'exécution de l'application en l'interrompant, la reprenant et la terminant ; d'avancer d'un pas avec entrée ou d'avancer d'un pas sans entrée dans les instructions de code ; de sélectionner les variables critiques à contrôler ; et enfin d'évaluer les expressions de contrôle en cours d'exécution de l'application.

A propos de l'exécution et du débogage des applications

Une fois les projets générés en applications (voir « [Génération de projets](#) » à la page 73), vous pouvez les exécuter et les déboguer dans Flash Builder.

Utilisation de la version de débogage de Flash Player

Pour utiliser la fonction de débogage de Flash Builder, vous devez exécuter la version de débogage de Flash Player. Cette version est disponible sous forme de plug-in ou de contrôle ActiveX pour navigateur ou en tant que version autonome. Elle est installée avec Flash Builder mais peut également être téléchargée à partir du site Web d'Adobe.

Les programmes d'installation de la version de débogage de Flash Player se trouvent dans le répertoire `flex_builder_install/Player`.

Vous pouvez déterminer par programmation la version de Flash Player exécutée en utilisant la méthode `Capabilities.isDebugger()`. Pour plus d'informations, voir [Determining Flash Player version in Flex](#).

Utilisation des configurations de lancement pour exécuter et déboguer des applications

Une configuration de lancement définit le nom du projet, le fichier principal de l'application et le chemin des versions d'exécution et de débogage de l'application. Flash Builder contient une configuration de lancement d'application Flex par défaut. Cette configuration est utilisée pour créer automatiquement des configurations de lancement pour chacun de vos projets. Pour plus d'informations, voir « [Gestion des configurations de lancement](#) » à la page 94.

Remarque : dans la configuration plug-in de Flash Builder, la création des configurations de lancement n'est pas automatique. Vous devez en créer une à la première exécution de l'application. Voir « [Exécution des applications](#) » à la page 92.

Personnalisation des configurations de lancement

Vous pouvez personnaliser les configurations de lancement que Flash Builder crée automatiquement pour vous. Vous pouvez par exemple commuter le fichier d'application principal ou modifier le chemin afin qu'il dirige vers un fichier SWF plutôt que vers le fichier d'enveloppe HTML. Cela permet d'exécuter et de déboguer les applications directement dans la version autonome de Flash Player plutôt que dans un navigateur Web. Vous pouvez en outre créer des configurations de lancement distinctes pour chaque fichier d'application du projet. Pour plus d'informations, voir « [Création ou modification d'une configuration de lancement](#) » à la page 94.

Exécution et débogage des applications dans un navigateur ou dans la version autonome de Flash Player

Par défaut, les chemins d'exécution et de débogage de la configuration de lancement dirigent vers les fichiers d'enveloppe HTML situés dans le dossier de sortie du projet. Les applications sont donc exécutées et déboguées dans la version Flash Player intégrée dans le navigateur Web utilisé. Vous avez toutefois la possibilité d'exécuter et de déboguer les applications dans la version autonome de Flash Player (voir « [Exécution du fichier d'application SWF dans la version autonome de Flash Player](#) » à la page 95). Vous pouvez en outre remplacer le navigateur Web standard par tout autre navigateur installé sur votre ordinateur (voir « [Modification du navigateur Web par défaut](#) » à la page 96).

Remarque : les paramètres de sécurité par défaut d'Internet Explorer 6 (ou des versions ultérieures) bloquent les applications lancées à partir de Flash Builder. Pour les modifier, cliquez sur Outils > Options Internet > Avancé > Sécurité > Autoriser le contenu actif à s'exécuter dans les fichiers de la zone Ordinateur local. Une autre possibilité consiste à autoriser l'exécution de chaque page Internet Explorer après son blocage.

Exécution des outils de débogage à partir de l'éditeur de code

La perspective Débogage Flash contient toutes les fonctions attendues d'un outil de développement puissant et complet. Les outils de débogage vous permettent :

- de définir et de gérer des points d'arrêt ;
- de déterminer comment interrompre, reprendre ou quitter l'application ;
- d'avancer d'un pas avec et sans entrée dans le code ;
- de contrôler des variables ;
- d'évaluer des expressions.

Pour plus d'informations sur les outils de débogage Flash Builder, voir « [Débogage des applications](#) » à la page 141.

Pour plus d'informations sur la modification du code, voir « [A propos de la modification du code dans Flash Builder](#) » à la page 103.

Activation de la perspective de débogage au niveau d'un point d'arrêt

Des points d'arrêt peuvent être ajoutés à des lignes de code exécutables dans l'éditeur de code. Au lancement d'une session de débogage, l'application est exécutée jusqu'au premier point d'arrêt rencontré. A ce stade, la perspective Débogage Flash est activée. Elle vous permet d'analyser l'état de l'application et de la gérer en faisant appel aux outils de débogage. Pour plus d'informations, voir « [Lancement d'une session de débogage](#) » à la page 141.

Comparaison des versions déboguées et non déboguées d'une application Flex

Par défaut, Flash Builder génère des versions déboguées du fichier SWF de l'application et les enregistre dans le répertoire bin-debug du projet. Le volume de cette application est supérieur à celui d'une version non déboguée car il contient du code et des métadonnées supplémentaires utilisées par le débogueur.

Pour générer une version non déboguée de l'application Flex, effectuez l'une des opérations suivantes :

- Sélectionnez **Projet > Exporter vers une version validée**. Cette opération crée un fichier SWF ou un fichier AIR non débogué dans le répertoire bin-release.
- Ajoutez `-debug=false` dans le champ Arguments de compilateur supplémentaires. Ceci faisant, un fichier SWF non débogué sera généré où que vous exportiez le projet.

Exécution des applications

Les applications sont exécutées (et déboguées) en fonction d'une configuration de lancement. La configuration de lancement d'une nouvelle application Flex ou ActionScript spécifie l'emplacement des fichiers d'application générés et du fichier principal de l'application. Les configurations de lancement peuvent être modifiées ou personnalisées. Pour plus d'informations, voir « [Gestion des configurations de lancement](#) » à la page 94.

L'exécution des projets conduit à l'ouverture du fichier principal SWF de l'application dans le navigateur Web par défaut ou directement dans la version autonome de Flash Player. Pour plus d'informations sur la modification du navigateur par défaut ou sur l'exécution et le débogage dans la version autonome de Flash Player, voir « [Modification du navigateur Web par défaut](#) » à la page 96 et « [Exécution du fichier d'application SWF dans la version autonome de Flash Player](#) » à la page 95.

L'exécution des projets dans Flash Builder peut être effectuée de différentes manières. Vous pouvez par exemple utiliser la commande d'exécution à partir de la barre d'outils et du menu principaux du workbench, de la vue Explorateur de packages et des menus contextuels de l'éditeur de code.

Remarque : le bouton Exécuter comporte deux éléments : le bouton d'action principal et la liste déroulante affichant les fichiers d'application du projet pouvant être exécutés ou débogués. Cliquez sur le bouton d'action principal pour exécuter le fichier d'application par défaut. Vous pouvez également activer la liste déroulante et sélectionner l'un des fichiers d'application du projet ainsi que créer ou modifier une configuration de lancement dans la boîte de dialogue Création, gestion et exécution d'une configuration.

Exécution du projet avec la configuration de lancement par défaut de l'application Flex

1 Dans l'Explorateur de packages, sélectionnez le projet que vous souhaitez exécuter.

2 Dans la barre d'outils principale du workbench, cliquez sur le bouton Exécuter.

Si le projet n'a pas encore été généré, Flash Builder le génère et l'exécute.

L'application s'affiche dans le navigateur Web par défaut ou dans la version autonome de Flash Player.

Vous pouvez exécuter et déboguer tout fichier de projet défini comme fichier d'application. Pour plus d'informations, voir « [Gestion des fichiers d'application du projet](#) » à la page 52.

Création d'une configuration de lancement personnalisée

1 Dans l'Explorateur de packages, sélectionnez le projet que vous souhaitez exécuter et ouvrez le fichier principal de l'application dans l'éditeur.

2 Dans le menu principal, cliquez sur Exécuter > Exécuter > Autre.

3 Dans la boîte de dialogue Création, gestion et exécution des configurations, sélectionnez l'option Application Web ou Application de bureau.

4 Dans la barre d'outils de la boîte de dialogue, cliquez sur le bouton Crée une nouvelle configuration de lancement.

5 Saisissez le nom de la configuration.

6 (Facultatif) Modifiez si nécessaire les propriétés de la configuration.

7 (Facultatif) Cliquez sur Exécuter pour lancer l'exécution de l'application.

Exécution d'autres fichiers d'application du projet

1 Dans l'Explorateur de packages, sélectionnez le projet que vous souhaitez exécuter.

2 Dans le menu principal, cliquez sur Exécuter > Exécuter > Autre.

3 Sélectionnez l'application que vous souhaitez exécuter.

Exécution d'une configuration de lancement personnalisée

❖ Dans le menu principal, cliquez sur Exécuter > Exécuter > Autre.

Dans la boîte de dialogue Création, gestion et exécution des configurations, sélectionnez une configuration personnalisée existante ou créez-en une. Voir « [Gestion des configurations de lancement](#) » à la page 94.

L'action de la commande Exécuter de la version plug-in de Flash Builder est légèrement différente. Elle n'exécute pas le projet sélectionné, mais la dernière configuration à avoir été lancée. Vous pouvez également sélectionner une configuration parmi la liste de celles à avoir été lancées en dernier.

Exécution de la dernière configuration lancée

❖ Cliquez sur le bouton Exécuter situé dans la barre d'outils principale.

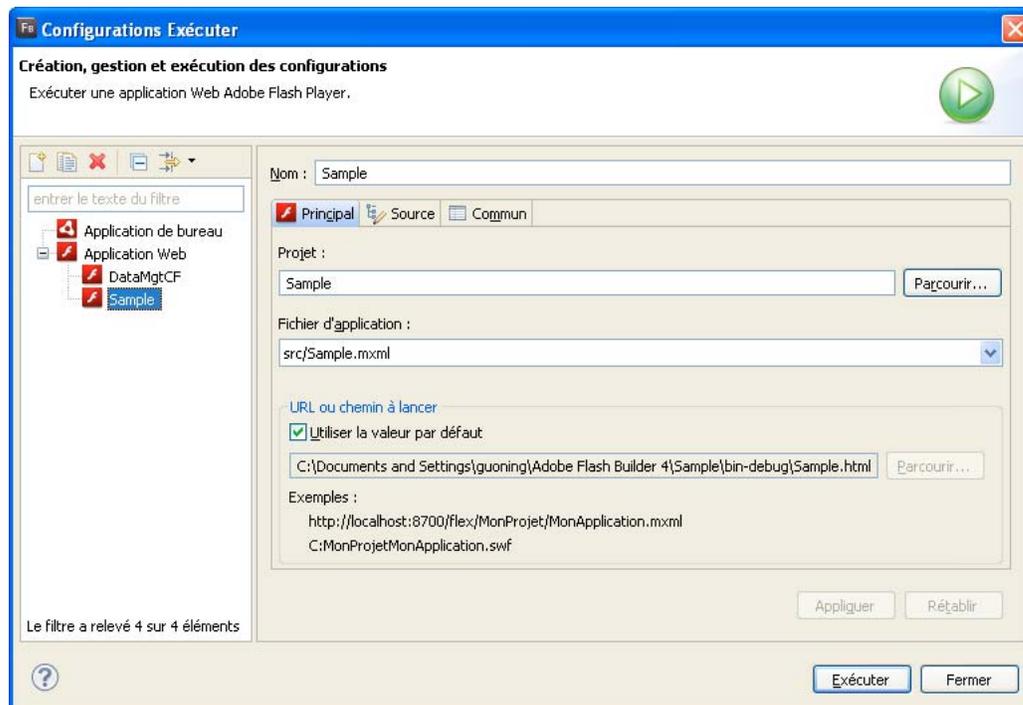
Gestion des configurations de lancement

Les configurations de lancement permettent d'exécuter et de déboguer des applications. Flash Builder comporte une configuration de lancement par défaut pour les applications Flex et ActionScript. Lorsque vous exécutez ou déboguez un projet pour la première fois, le programme crée une configuration de lancement spécifique à ce projet. Vous pouvez ensuite modifier le fichier d'application par défaut spécifié dans la configuration de lancement de même que le chemin de lancement par défaut, afin que l'application soit exécutée ou déboguée dans la version autonome de Flash Player plutôt que dans un navigateur Web.

Création ou modification d'une configuration de lancement

Un projet créé et généré est prêt à être exécuté ou débogué. L'exécution et le débogage des applications du projet sont contrôlés par une configuration de lancement. Par défaut, Flash Builder crée une configuration de lancement pour chaque fichier d'application du projet exécuté ou débogué pour la première fois. Les configurations se basent sur la configuration d'application Flex par défaut et peuvent être modifiées.

Les configurations de lancement sont gérées dans la boîte de dialogue Création, gestion et exécution des configurations.



Ouverture et modification d'une configuration de lancement

- 1 Dans l'Explorateur de packages Flex, sélectionnez un projet.
- 2 Ouvrez un fichier du projet dans l'éditeur de code, puis la boîte de dialogue Création, gestion et exécution des configurations. Cette opération dépend de la version d'installation de Flash Builder.
 - Pour la version autonome de Flash Builder : sélectionnez le projet, puis Exécuter > Exécuter > Autre ou Exécuter > Déboguer > Autre.
 - Pour la version plug-in de Flash Builder avec Eclipse 3.2 : cliquez avec le bouton droit de la souris (ou en enfonçant la touche Ctrl sur Mac OS) pour afficher le menu contextuel et sélectionnez Exécuter en tant que > Exécuter ou Déboguer en tant que > Déboguer.

- Pour la version plug-in de Flash Builder avec Eclipse 3.3 : cliquez avec le bouton droit de la souris (ou en enfonçant la touche Ctrl sur Mac OS) pour afficher le menu contextuel et sélectionnez Exécuter en tant que > Ouvrir la boîte de dialogue d'exécution ou Déboguer en tant que > Ouvrir la boîte de dialogue de débogage.

3 Sélectionnez la configuration de lancement que vous souhaitez modifier.

Il est possible que plusieurs configurations soient répertoriées. C'est le cas notamment si l'espace de travail contient d'autres projets, si vous avez défini d'autres fichiers de projet comme fichiers d'application ou si d'autres plug-ins Eclipse sont installés.

A la première exécution d'un projet, Flash Builder crée par défaut une configuration de lancement spécifique au projet, reposant sur la configuration d'application Flex par défaut.

Les configurations spécifiques aux applications peuvent être modifiées. Vous pouvez également créer une nouvelle configuration de lancement ou développer une nouvelle configuration à partir d'une configuration existante.

4 Adaptez les paramètres de la configuration, puis cliquez sur Exécuter ou Déboguer.

Création de configurations de lancement pour applications de bureau

Pour les applications de bureau, vous pouvez créer des configurations de lancement qui remplaceront les configurations par défaut fournies par Flash Builder.

Lors de la création ou de la modification d'une configuration de lancement pour une application de bureau, vous devez spécifier des arguments de ligne de commande, un ID d'éditeur, un profil et des paramètres de format d'écran.

Pour les applications de bureau, la valeur de Profil peut être soit desktop, soit extendedDesktop.

Pour créer une configuration de lancement :

1 Sélectionnez Exécuter > Exécuter > Autre.

Pour les configurations de lancement de débogage, cliquez sur Exécuter > Déboguer > Autre.

Pour les configurations de lancement de profilage, cliquez sur Exécuter > Profil > Autre.

2 Sélectionnez le projet de bureau. Cliquez sur le bouton de création d'une configuration de lancement.

Accédez aux onglets Principal, Source et Commun pour spécifier les paramètres de la configuration.

3 Sélectionnez Exécuter, Déboguer ou Profiler pour exécuter l'application avec les paramètres définis.

Cliquez sur Fermer pour enregistrer les paramètres.

Voir aussi

« [Gestion des configurations de lancement](#) » à la page 94

Exécution du fichier d'application SWF dans la version autonome de Flash Player

Les applications sont exécutées ou déboguées dans le navigateur Web par défaut. Vous pouvez toutefois en sélectionner un autre (voir « [Modification du navigateur Web par défaut](#) » à la page 96). Les applications peuvent également être exécutées ou déboguées dans la version autonome de Flash Player. Il vous suffit pour ce faire de modifier légèrement la configuration de lancement.

Exécution et débogage des applications dans la version autonome de Flash Player

1 Dans la boîte de dialogue Création, gestion et exécution des configurations, sélectionnez la configuration de lancement que vous souhaitez modifier.

2 Dans l'onglet Principal, désélectionnez l'option Utiliser la valeur par défaut.

- 3 Cliquez sur le bouton Parcourir du champ Exécuter et/ou du champ Débuguer.

La boîte de dialogue de sélection d'un fichier s'ouvre, répertoriant le contenu du dossier de sortie de la génération.

- 4 Sélectionnez le fichier d'application SWF du répertoire bin-debug. Ne sélectionnez pas le fichier SWF de l'éventuel répertoire bin-release, étant donné qu'il ne contient aucune information de débogage.
- 5 Cliquez sur Ouvrir pour sélectionner le fichier et retourner à la boîte de dialogue de configuration.
- 6 Validez les modifications apportées et exécutez ou déboguez l'application en utilisant la configuration modifiée.

Modification du navigateur Web par défaut

L'exécution et le débogage des applications sont effectués dans le navigateur Web défini par défaut. Vous ne pouvez pas définir un navigateur Web différent pour chaque configuration de lancement. Vous pouvez toutefois modifier le navigateur Web défini pour le workbench, ce qui aura une incidence sur l'exécution et le débogage de toutes les applications.

Modification du navigateur Web par défaut

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Général > Navigateur Web.
- 2 Sélectionnez un navigateur dans la liste de ceux installés sur le système.

Remarque : l'option Utiliser le navigateur Web interne ne concerne pas l'exécution et le débogage des applications, ces opérations étant toujours effectuées dans un navigateur Web externe.

Vous pouvez également ajouter des navigateurs à la liste, en supprimer ou éditer ceux qui y sont répertoriés.

- 3 Cliquez sur OK pour appliquer les modifications.

Création de modules

Adobe® Flash® Builder™ permet de créer, d'ajouter, d'optimiser et de déboguer des modules. Pour plus d'informations sur la rédaction de code de module, voir *Creating modular applications*.

Création de modules dans Flash Builder

Les étapes suivantes décrivent la procédure de création d'un nouveau module dans Flash Builder. Après avoir créé un nouveau module, vous pouvez le compiler.

Création de modules dans Flash Builder

- 1 Dans Flash Builder, sélectionnez Fichier > Nouveau > Module MXML. La boîte de dialogue Création d'un(e) module MXML s'affiche.
- 2 Sélectionnez un répertoire parent pour le module. Les modules sont habituellement stockés dans le même répertoire que l'application principale, pour que les chemins relatifs des ressources partagées soient identiques. Les modules étant exécutables, ils doivent se trouver dans le dossier source du projet.
- 3 Entrez un nom de fichier pour le module, par exemple MyModule.
- 4 Entrez les propriétés Largeur, Hauteur et Présentation.
- 5 Indiquez si vous souhaitez ou non optimiser le module.

L'optimisation d'un module pour une application signifie que les classes utilisées par l'application sont exclues du module, ce qui peut contribuer à réduire le volume de téléchargement des fichiers SWF. Pour plus d'informations, voir « [Optimisation de modules dans Flash Builder](#) » à la page 101.

- Optimiser pour l'application

Si vous activez cette option, spécifiez l'application pour laquelle le module doit être optimisé.

- Ne pas optimiser

Si vous activez cette option, toutes les classes sont incluses dans le module, qu'elles soient définies ou non dans l'application principale, améliorant ainsi les performances de la compilation d'incrémentation. Vous pouvez en outre charger le module dans n'importe quelle application, toutes ses dépendances y étant compilées.

6 Cliquez sur Terminer.

Flash Builder ajoute un nouveau fichier de module MXML dans le projet.

Compilation de modules dans Flash Builder

Dans Flash Builder, vous pouvez exécuter le module comme une application ou générer le projet du module. Si les modules figurent dans le même projet que l'application, alors Flash Builder compile automatiquement les fichiers SWF pour tous les modules du projet lorsque vous exécutez l'application. Les fichiers SWF sont ensuite chargés dans l'application lors de l'exécution.

Il est impossible d'exécuter le fichier SWF basé sur un module comme une application Flash autonome ou de le charger dans une fenêtre de navigateur. Il doit être chargé par une application en tant que module. Si vous exécutez le module dans Flash Builder pour le compiler, fermez Adobe Flash Player ou la fenêtre du navigateur et ignorez les erreurs éventuelles. Les modules ne doivent pas être sollicités directement par Flash Player ou par l'intermédiaire d'un navigateur.

Les fichiers SWF du module et le fichier SWF de l'application principale se trouvent généralement dans le même répertoire. Flash Builder compile toutefois les modules en même temps que l'application, quel que soit leur emplacement. Les modules peuvent se situer dans le même répertoire que l'application ou dans des sous-répertoires.

Vous pouvez également créer un projet Flex ou ActionScript distinct pour chaque module ou groupe de modules. Il en résulte une meilleure maîtrise de la compilation des modules, les options de compilateur de chaque projet pouvant différer de celles de l'application ou des autres modules. Cela permet également de compiler le ou les projets d'un module sans compiler l'application. Cette approche requiert toutefois la compilation manuelle de chaque module avant de compiler l'application. Une manière de procéder consiste à compiler simultanément tous les projets ouverts dans Flash Builder.

Si vous compilez les modules indépendamment de l'application principale, veillez à inclure ou exclure les informations de débogage, selon que vous souhaitez déboguer l'application et les modules. Pour plus d'informations, voir « [Débogage de modules dans Flash Builder](#) » à la page 101.

Le déroulement du travail dans Flash Builder est conçu pour l'association de modules avec une même application. Pour utiliser des modules dans plusieurs applications, encapsulez éventuellement le code dans une classe ou un composant de bibliothèque à inclure dans un module simple pour chaque application. Contrairement aux bibliothèques, les modules ne sont pas destinés à une utilisation dans le code commun à plusieurs applications.

Utilisation de plusieurs projets pour des modules

Lors de la définition de l'architecture de votre projet, vous pouvez décider d'inclure des modules dans votre projet d'application, de créer un projet distinct pour chaque module ou de créer un projet distinct pour l'ensemble des modules.

Utilisation d'un projet pour chaque module

L'utilisation d'un projet pour chaque module présente les avantages suivants :

- Les projets de module peuvent figurer à un emplacement quelconque de l'espace de travail.
- Les projets de module peuvent disposer de paramètres de compilateur distincts, tel qu'un chemin d'accès à la bibliothèque personnalisé.

Ils peuvent utiliser l'option de compilateur `load-externs` pour supprimer les dépendances qui se recoupent.

L'utilisation d'un projet pour chaque module présente les inconvénients suivants :

- La mémoire nécessaire augmente avec le nombre de projets.
- L'existence de nombreux projets dans un même espace de travail peut augmenter son encombrement.
- Par défaut, lorsque vous compilez l'application, tous les projets de module ne sont pas compilés, même s'ils ont subi des modifications.
- Pour optimiser la taille du fichier du module, appliquez manuellement les options de compilateur `load-externs` et `link-report`.

Utilisation d'un projet pour tous les modules

Une approche semblable consiste à utiliser un même projet pour l'ensemble des modules, en conservant l'application dans son propre projet distinct. Certains inconvénients, liés à l'utilisation d'un même projet à la fois pour l'application et les modules, subsistent. Cette approche apporte toutefois de nombreux avantages découlant de l'utilisation d'un projet distinct pour chaque module.

L'utilisation d'un projet pour l'ensemble des modules présente les avantages suivants :

- Le projet de module peut figurer à un emplacement quelconque de l'espace de travail.
- Vous pouvez compiler séparément les modules ou l'application, sans nécessité de compiler les deux simultanément.
- Le projet de module peut utiliser l'option de compilateur `load-externs` pour supprimer les dépendances qui se recoupent.

L'utilisation d'un projet de module pour l'ensemble des modules présente les inconvénients suivants :

- Tous les modules du projet de module doivent utiliser les mêmes paramètres de compilateur, tel que le chemin d'accès à la bibliothèque.
- Par défaut, lorsque vous compilez l'application, le projet de module n'est pas compilé, même s'il a subi des modifications.
- Pour optimiser la taille du fichier du module, vous devez appliquer manuellement les options de compilateur `load-externs` et `link-report`.

Création de projets pour des modules

Lors de la création d'un projet séparé pour les modules, vous modifiez le dossier de sortie du projet de module sur un répertoire utilisé par l'application. Vous supprimez également la création de fichiers d'enveloppe.

Création d'un projet distinct pour les modules dans Flash Builder

- 1 Créez un projet principal.
- 2 Créez un nouveau projet pour le ou les modules.
- 3 Dans le menu contextuel du projet du module, sélectionnez Propriétés. La boîte de dialogue Propriétés s'ouvre.
- 4 Sélectionnez l'option Chemin de génération Flex.

- 5 Modifiez le dossier de sortie de manière telle à ce qu'il pointe vers le répertoire des modules MainProject, comme illustré dans l'exemple ci-dessous :

```
 ${DOCUMENTS}\MainProject\assets
```

Cette modification redirige la sortie de compilation du module vers le répertoire des ressources du projet d'application (MainProject). Dans l'application principale, vous pouvez définir la propriété `url` de `ModuleLoader` sur les fichiers SWF du répertoire des ressources. La valeur de cette propriété est relative au dossier de sortie.

- 6 Cliquez sur OK pour enregistrer vos modifications.
- 7 Ouvrez à nouveau les propriétés du projet et sélectionnez l'option Compilateur Flex.
- 8 Désélectionnez l'option Générer le fichier d'enveloppe HTML afin d'éviter que le projet du module ne génère les fichiers d'enveloppe HTML. Ces fichiers ne sont généralement utilisés que pour l'application et sont inutiles pour les modules.
- 9 Cliquez sur OK pour appliquer les modifications.

Compilation de projets pour les modules

La compilation de plusieurs projets dans Flash Builder est une opération courante. Choisissez d'abord l'ordre de compilation souhaité, puis compilez l'ensemble des projets simultanément.

Compilation simultanée de tous les projets dans Flash Builder

- ❖ Dans le menu principal, sélectionnez Projets > Générer tout.

Flex génère alors l'ensemble des projets de l'espace de travail. Les fichiers d'application sont ajoutés au dossier de sortie de chaque projet. Si vous n'avez pas choisi l'enregistrement automatique des fichiers avant le début d'une génération, vous êtes invité à enregistrer les fichiers.

Pour modifier l'ordre de génération, utilisez la boîte de dialogue Ordre de génération. La modification de l'ordre de génération n'est pas toujours nécessaire. Les projets utilisant des modules ne doivent être compilés qu'avant l'exécution de l'application de projet principale, et non lors de sa compilation. Dans la plupart des cas, l'ordre de génération par défaut ne doit pas être modifié.

Si toutefois vous souhaitez éliminer les dépendances qui se recourent, vous pouvez modifier l'ordre de génération afin de compiler d'abord l'application principale. L'option de compilateur `link-report` permet alors de générer le rapport de liaison. Lorsque vous compilez les modules, l'option de compilateur `load-externs` permet d'utiliser le rapport de liaison généré récemment par l'application principale. Pour plus d'informations sur la réduction de la taille des modules, voir « [Optimisation de modules dans Flash Builder](#) » à la page 101.

Modification de l'ordre de génération des projets

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Général > Espace de travail > Ordre de génération.

La boîte de dialogue Ordre de génération apparaît.

- 2 Désélectionnez la case à cocher Utiliser l'ordre de génération par défaut.
- 3 Utilisez les boutons Haut et Bas pour reclasser les projets de la liste Ordre de génération. Vous pouvez également utiliser le bouton Supprimer un projet pour supprimer les projets qui ne font pas partie de l'application principale ou qui ne sont pas des modules utilisés par l'application. Dans ce cas, le projet supprimé sera généré mais uniquement après tous les autres projets de la liste.
- 4 Cliquez sur OK.
- 5 Modifiez si nécessaire l'ordre de génération, puis cliquez sur OK.

En présence de dépendances entre des projets distincts de l'espace de travail, le compilateur détermine automatiquement l'ordre dans lequel les projets sont générés afin de résoudre correctement les dépendances.

Si vous utilisez un projet distinct pour chaque module, vous pouvez compiler un seul module à la fois. Cela permet de gagner du temps par rapport à la compilation simultanée de tous les projets, ou par rapport à la compilation d'un seul projet contenant tous les fichiers de modules et d'application.

Compilation du projet d'un seul module

- 1 Cliquez avec le bouton droit de la souris sur le fichier MXML du module dans le projet du module.
- 2 Sélectionnez Exécuter l'application. La fenêtre du lecteur ou du navigateur tente d'exécuter le module après sa compilation. Vous pouvez fermer la fenêtre du lecteur ou du navigateur et ignorer les messages d'erreur éventuels qui s'affichent lors de l'exécution. Les modules ne sont pas destinés à une exécution directe dans le lecteur ou dans un navigateur.

Ajout de modules au projet

Vous serez amené, dans certaines situations, à utiliser des modules ne figurant pas dans le projet de l'application principale. Le module peut provenir d'un projet distinct, vous pouvez alors utiliser des options de configuration personnalisées. Vous pouvez également vouloir partager le module entre plusieurs applications. Ajoutez le code source du module au chemin source de l'application, puis ajoutez le module à la liste de modules de l'application afin de pouvoir l'utiliser dans le projet.

Ajout d'un module préalablement compilé au projet

- 1 Sélectionnez l'application principale dans l'Explorateur de packages Flex.
- 2 Sélectionnez Projet > Propriétés > Chemin de génération Flex pour ajouter la source du module au chemin source du projet d'application.
- 3 Cliquez sur le bouton Ajouter un dossier et accédez au chemin source du module. Cliquez sur OK pour sélectionner le module. Répétez cette procédure pour chaque module externe à ajouter au projet de l'application.
- 4 Cliquez à nouveau sur OK pour enregistrer vos modifications.
- 5 Sélectionnez Projet > Propriétés > Modules Flex pour ajouter le module à la liste de modules de l'application. La boîte de dialogue Modules Flex liste l'ensemble des modules ajoutés au projet actif ou figurant dans ce projet. Lors de la création initiale d'un projet, cette boîte de dialogue est vide.
- 6 Cliquez sur le bouton Ajouter. La boîte de dialogue Ajouter un module s'affiche.
- 7 Utilisez le bouton Parcourir ou entrez l'emplacement du fichier MXML du module dans la zone de texte Source. Tous les modules figurant dans le chemin source du projet peuvent être ajoutés à l'aide de cette boîte de dialogue.
- 8 Sélectionnez l'un des boutons radio Taille du fichier SWF du module pour activer ou désactiver l'optimisation de module. Si vous sélectionnez Optimiser pour l'application, Flash Builder compile le module pour l'application sélectionnée en excluant toutes les classes définies dans l'application principale. Il peut s'agir de classes de structure ou de classes personnalisées. Lorsque vous sélectionnez cette option, le module ne peut pas être utilisé dans une autre application car la liste de classes exclues peut être différente. Pour plus d'informations, voir « [Optimisation de modules dans Flash Builder](#) » à la page 101.
- 9 Cliquez sur OK pour enregistrer vos modifications. Flash Builder ajoute le module à la liste des modules disponibles dans le projet de l'application.

Optimisation de modules dans Flash Builder

Dans Flash Builder, vous ne sélectionnez généralement qu'une seule application par rapport à laquelle le module doit être optimisé lors de sa création initiale ou de son ajout au projet. Si vous décidez par la suite de changer l'application visée par l'optimisation du module, ou de ne pas optimiser le module, vous pouvez modifier les propriétés du module au sein du projet. Pour plus d'informations, voir [Reducing module size](#).

Optimisation de modules avec Flash Builder

Cette procédure part du principe que le module et l'application se trouvent dans le même projet Flash Builder. Si les modules se situent dans un projet distinct, ajoutez manuellement les options de compilateur `load-externs` et `link-report`.

- 1 Cliquez avec le bouton droit de la souris sur le projet de l'application dans l'Explorateur de package Flex et sélectionnez Propriétés. La boîte de dialogue Propriétés du projet s'ouvre.
- 2 Dans le volet de gauche, sélectionnez Modules Flex.
- 3 Sélectionnez le module dans la liste et cliquez sur le bouton Modifier. La boîte de dialogue Modifier le module s'affiche.
- 4 Pour supprimer l'optimisation, sélectionnez le bouton radio Ne pas optimiser sous Taille du fichier SWF du module.
- 5 Afin d'optimiser le module pour une autre application, sélectionnez la nouvelle application dans la liste déroulante Optimiser pour l'application.
- 6 Cliquez sur OK.

Pour optimiser davantage la taille du fichier d'un module, vous pouvez supprimer les informations de débogage. Lors de la génération d'un module dans Flash Builder, les informations de débogage y sont incluses par défaut. En les supprimant, vous pouvez réduire la taille du module. Pour obtenir des instructions sur la procédure de suppression des informations de débogage des modules, voir « [Débogage de modules dans Flash Builder](#) » à la page 101.

Débogage de modules dans Flash Builder

Pour déboguer des modules et des applications, il est nécessaire d'inclure des informations de débogage dans les fichiers SWF lors de leur compilation. Dans Flash Builder, il suffit d'exécuter l'application, puisque les informations de débogage sont incluses par défaut. A la ligne de commande, définissez l'option de compilateur `debug` sur `true`. La valeur par défaut est `true`, mais si vous l'avez désactivée dans un fichier de configuration, veillez à le remplacer.

Par défaut, Flash Builder génère un seul fichier SWF comprenant des symboles de débogage. Les instructions Déboguer et Exécuter fonctionnent donc toutes les deux lors de l'exécution d'une application utilisant des modules dans Flash Builder. Toutefois, l'inclusion de symboles de débogage dans le fichier SWF d'un module augmente sa taille. Pour exclure les symboles de débogage avant le déploiement, désactivez le débogage pour les modules de l'application. Pour ce faire, exportez la version validée des modules en sélectionnant Projet > Exporter vers une version validée.

Pour exclure les informations de débogage des fichiers SWF dans Flash Builder, vous pouvez définir l'option `debug` sur `false` dans la zone de texte des arguments de compilateur supplémentaires, ou générer des fichiers SWF sans débogage en les exportant à l'aide de la fonctionnalité Exporter vers une version validée. Cela inclut les modules, s'ils figurent dans le projet actif.

Si vous créez un projet distinct pour les modules, vous pouvez activer ou désactiver le débogage pour l'ensemble du projet sans modifier les paramètres de l'application principale.

Pour déboguer un module, vous devez également déboguer son application de chargement. Le débogueur Flex ne se connecte pas aux applications dépourvues d'informations de débogage, même si les modules chargés par l'application les contiennent. En d'autres termes, vous ne pouvez pas exclure les informations de débogage de l'application si vous souhaitez déboguer le module qu'elle charge.

Lors de l'utilisation de modules dans une application AIR, le fichier SWF du module doit figurer dans le même répertoire que le fichier SWF de l'application, ou dans l'un de ses sous répertoires.

Chapitre 4 : Modification du code dans Flash Builder

Dans Adobe® Flash® Builder™, la modification des codes MXML, ActionScript et CSS advient dans des éditeurs différents. Le workbench de Flash Builder intègre à la fois les projets et les documents, permettant ainsi l'ouverture automatique de l'éditeur correspondant au type de ressource. Les éditeurs de Flash Builder partagent les mêmes fonctionnalités, parmi lesquelles les indicateurs de code, la navigation, le formatage, la restructuration ainsi que d'autres fonctions d'amélioration de la productivité.

A propos de la modification du code dans Flash Builder

La création et le développement d'applications Flex et ActionScript a lieu dans la perspective Développement Flash qui se compose des éditeurs Flash Builder ainsi que de toutes les vues prenant en charge les opérations de modification et de création de code. La configuration de la perspective de développement dépend de l'éditeur et du mode utilisés. Lors de la création d'un projet Flex par exemple, l'éditeur MXML présente deux modes (Source et Création), contenant chacun sa propre gamme de vues. Pour une présentation générale de la perspective de développement Flash, voir « [Perspective Développement Flash](#) » à la page 8.

Éditeurs MXML, ActionScript et CSS

Flash Builder contient trois éditeurs pour la rédaction de code MXML, ActionScript et CSS.

L'éditeur MXML permet de modifier les fichiers MXML. Il se compose de deux modes : Source et Création. Le mode Source vous permet de rédiger du code MXML et d'intégrer les codes ActionScript et CSS contenus dans les balises `<fx:Script>` et `<fx:Style>`. Le mode Création vous permet de créer des applications (voir « [Création d'une interface utilisateur avec Flash Builder](#) » à la page 188).

L'éditeur ActionScript permet de modifier les fichiers AS, dont les principaux fichiers des projets ActionScript ainsi que les fichiers de classe et d'interface. Pour plus d'informations, voir « [Projets ActionScript](#) » à la page 66.

L'éditeur CSS permet d'écrire des fichiers CSS (Cascading Style Sheets). Pour plus d'informations, voir « [Application de styles](#) » à la page 213 et [Styles and themes](#).

Assistant de contenu MXML, ActionScript et CSS

En cours de rédaction de code MXML, ActionScript et CSS, des indicateurs et des documents de référence ASDoc s'affichent afin de faciliter la saisie. Cette fonction porte le nom d'*assistant de contenu*. Flash Builder accélère par ailleurs le développement de code en complétant les balises MXML, en proposant une gestion automatique de l'importation, en intégrant le *Guide de référence du langage Adobe Flex* et en permettant de définir des couleurs et des polices différentes pour l'affichage du code dans l'espace de travail. Pour plus d'informations, voir « [Aide au codage dans Flash Builder](#) » à la page 105.

Indicateurs de code pour les composants personnalisés ainsi que pour les classes et interfaces ActionScript

Outre la prise en charge intégrée pour la structure Flex, des indicateurs de code sont fournis pour tous les composants personnalisés ainsi que pour les classes et les interfaces ActionScript comprises dans le projet. Pour plus d'informations, voir « [Utilisation de l'assistant de contenu](#) » à la page 108.

Sélection de types recommandés pour les espaces de noms

Flash Builder comporte des filtres et un assistant de contenu pour vous permettre d'identifier les types recommandés pour un espace de noms. Vous pouvez également faire défiler les différents filtres et paramètres afin de localiser tous les types, propriétés et événements mis à disposition par la structure Flex. Pour plus d'informations, voir « [Utilisation de l'assistant de contenu](#) » à la page 108.

Génération des fonctions des accesseurs (Get et Set)

Les fonctions des accesseurs Get et Set vous permettent de faire en sorte que les propriétés d'une classe restent privées pour une classe ActionScript. Les utilisateurs de la classe peuvent accéder à ces propriétés de la même façon que s'ils accédaient à une variable de classe (plutôt que d'appeler une méthode de classe). Flash Builder peut générer des fonctions d'accesseur Get et Set ActionScript pour les variables de classe. Pour plus d'informations, voir « [Génération des fonctions d'accesseur](#) » à la page 68.

Simplification de la navigation dans le code

Afin de faciliter l'accès aux nombreux fichiers et lignes de code que contiennent les projets, Flash Builder propose des fonctions pratiques telles que, parmi d'autres, le développement et la réduction de blocs de code, l'ouverture des sources de définitions de code externes ainsi que la recherche et l'ouverture de types de classe. De plus, la vue Structure permet d'examiner la structure du code des documents et d'en parcourir les lignes. Pour plus d'informations, voir « [Navigation et organisation du code](#) » à la page 111.

Recherche de références et restructuration de code

Flash Builder peut rechercher toutes les références et les déclarations correspondant aux identifiants d'un fichier, d'un projet ou d'un espace de travail donné. La recherche porte également sur les éléments reliés à partir de fichiers SWC ainsi que les entrées des chemins de bibliothèque (classes, interfaces, fonctions, variables et quelques métadonnées, par exemple). La restructuration renomme les identifiants du code tout en mettant à jour toutes les références qui y sont faites dans l'ensemble de la base de code. Pour plus d'informations, voir « [Recherche de références et restructuration de code](#) » à la page 118.

Vérification automatique de la syntaxe

Flash Builder génère les projets de manière incrémentielle, en vous signalant d'éventuelles erreurs au fur et à mesure que vous progressez dans votre travail. En cas d'erreurs de syntaxe en cours de rédaction de code MXML et ActionScript, des indicateurs d'erreur s'affichent dans l'éditeur à la hauteur de la ligne de code concernée et un message apparaît dans la vue Erreurs. Pour plus d'informations, voir « [Application des préférences de couleurs pour la syntaxe](#) » à la page 125.

Coloration de la syntaxe

La page Couleurs pour la syntaxe des Préférences définit les couleurs à appliquer au code dans les éditeurs MXML, ActionScript et CSS. Vous pouvez également y définir des options de style pour la police et en afficher un aperçu. Pour plus d'informations, voir « [Comment obtenir de l'aide pour la rédaction de code](#) » à la page 110 et « [Utilisation de l'assistant de contenu](#) » à la page 108.

Editeur et intégration d'un débogueur

Les éditeurs MXML et ActionScript associés à la perspective Débogage Flash facilitent le débogage du code. Par la définition de points d'arrêt, vous pouvez interrompre l'application au niveau de lignes de code susceptibles de provoquer des difficultés ou présentant une importance particulière. Au lancement d'une session de débogage, la perspective Débogage Flash s'affiche lorsque le premier point d'arrêt est atteint. Vous pouvez alors examiner l'état de l'application et isoler, puis résoudre tout problème survenant dans le code. Pour plus d'informations sur le débogage du code, voir « [Débogage des applications](#) » à la page 141.

Aide au codage dans Flash Builder

Les éditeurs Flash Builder proposent plusieurs fonctions facilitant le développement de code.

Assistant de contenu

L'assistant de contenu est disponible dans les éditeurs MXML, ActionScript et CSS. Il fournit des indicateurs de code qui vous aident à compléter les expressions de code. Il facilite également la sélection des classes, propriétés et événements recommandés, disponibles dans un espace de noms. Pour plus d'informations, voir « [A propos de l'assistant de contenu](#) » à la page 106.

Gestion automatique de l'importation

Lorsque vous saisissez du code avec l'assistant de contenu, les noms complets des types sont automatiquement importés dans le document. Dans un document ActionScript, le nom complet du type est ajouté en tête de document avec les autres instructions d'importation. Dans un document MXML, l'instruction d'importation est ajoutée au bloc de script existant. En l'absence de bloc de script, Flash Builder en crée un. Les documents ActionScript offrent également la possibilité de trier les instructions d'importation. Pour plus d'informations, voir « [Organisation des instructions d'importation](#) » à la page 116.

Remplissage des balises MXML

Lors de la saisie de code MXML, de nombreux éléments de la syntaxe sont ajoutés automatiquement, dont les balises de fermeture, les mises en retrait et les nouvelles lignes. En outre, les balises CDATA sont automatiquement placées à l'intérieur de balises `<fx:Script>`.

Utilisation des états d'affichage

En mode Source, l'éditeur MXML facilite l'édition des états d'affichage. Lorsque des états d'affichage ont été définis à la racine de l'application, l'éditeur comporte la liste déroulante Afficher état. Servez-vous de cette liste pour mettre en évidence l'état sélectionné dans l'éditeur. L'assistant de contenu facilite également la sélection des états d'affichage pour les composants et les propriétés. Pour plus d'informations sur ce point ainsi que sur d'autres aspects de l'édition des états d'affichage, voir « [Création et édition d'états d'affichage dans le code source](#) » à la page 231.

Documentation de référence ASDoc

Flash Builder affiche les documents de référence ASDoc dans les éditeurs MXML et ActionScript lorsque vous utilisez un assistant de contenu ou survolez un type avec le curseur. La documentation ASDoc apparaît également dans la vue ASDoc pour les classes sélectionnées. Cette fonctionnalité prend en charge la documentation ASDoc générée par les utilisateurs.

Pour plus d'informations, voir « [Définition de commentaires ASDoc pour les composants personnalisés](#) » à la page 136.

Création de gestionnaires d'événement

Les éditeurs Flash Builder aident l'utilisateur à créer des gestionnaires d'événement pour les composants. Pour chaque événement d'un composant, vous pouvez générer un gestionnaire d'événement entouré de balises `<Script>`. Ce gestionnaire est référencé par la propriété event dans la balise du composant.

Pour plus d'informations, voir « [Création de gestionnaires d'événement](#) » à la page 208.

Aide contextuelle de référence du langage

Le *Guide de référence du langage Adobe Flex* est intégré à l'éditeur et est facilement accessible en sélectionnant un élément de langage ActionScript, une balise de composant MXML ou un attribut, puis en actionnant les touches Maj+F2. Pour plus d'informations, voir « [Comment obtenir de l'aide pour la rédaction de code](#) » à la page 110.

Aide pour le formatage

Afin de faciliter le codage des applications, les éditeurs peuvent reformater des blocs de code et procéder à des modifications en bloc. Pour plus d'informations, voir « [Formatage et modification du code](#) » à la page 116.

A propos de l'assistant de contenu

Au cours de la saisie du code dans les éditeurs Flash Builder, l'assistant de contenu affiche une liste d'options de remplissage de l'expression de code (généralement désignées par *indicateurs de code*). Un document MXML propose par exemple une liste de balises pouvant être ajoutées à l'emplacement auquel se trouve le curseur de la souris.

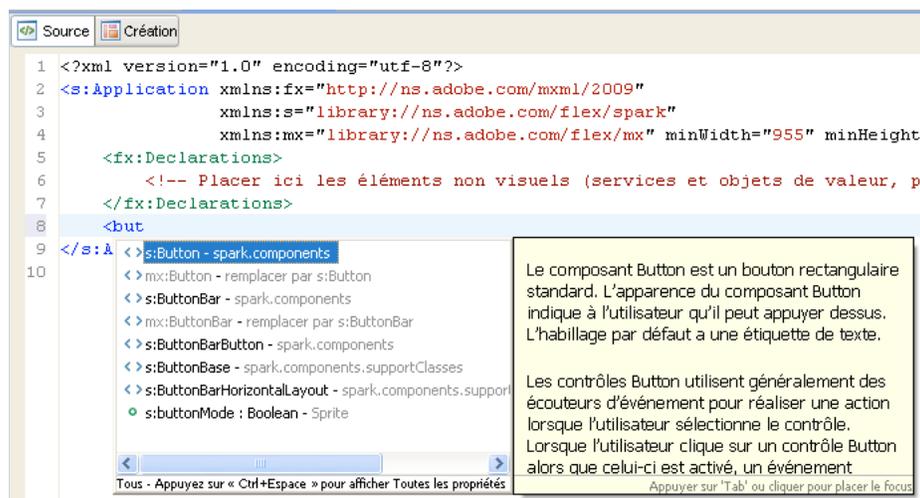
L'assistant de contenu affiche également la documentation de référence ASDoc disponible pour le composant MXML ou le code ActionScript que vous rédigez. La documentation ASDoc s'affiche également lors du passage du curseur de la souris sur un composant MXML ou une classe ActionScript. Si vous sélectionnez une classe dans l'éditeur, le contenu ASDoc apparaît également dans la vue ASDoc.

L'assistant de contenu est disponible dans les éditeurs MXML, ActionScript et CSS. Son utilisation varie en fonction de l'éditeur. Les indicateurs s'affichent à chaque fois qu'une structure ou un langage (MXML, ActionScript et CSS) fournit des options de remplissage de l'expression en cours de rédaction.

L'assistant de contenu affiche des indicateurs pour les composants MXML personnalisés et les classes ActionScript créés faisant partie du projet. Supposons que vous définissiez un composant MXML personnalisé et l'ajoutiez au projet. Les indicateurs de code s'affichent à chaque création d'une référence au composant dans l'application MXML.

Affichage des indicateurs de code avec l'assistant de contenu

Dans l'éditeur MXML, la saisie dans le cadre d'un composant MXML conduit par exemple à l'affichage d'une liste contenant toutes les propriétés de ce composant. L'exemple suivant illustre les indicateurs de code pour les propriétés d'un composant MXML.



La sélection et la saisie d'une propriété entraînent l'affichage des valeurs possibles (à condition que des valeurs prédéfinies existent). L'exemple suivant illustre les indicateurs de code pour les valeurs de propriété.

```

6 <fx:Declarations>
7 <!-- Placer ici les éléments non visuels (services et objets de valeurs) -->
8 </fx:Declarations>
9 <s:Button fontStyle="italic" x="34" y="231" label="Login" id="btu" click="btu_click" title="Sample">
10 <s:Panel x="10" y="10" dataField="col1"/>
11 <mx:DataGrid x="1" dataField="col2"/>
12 <mx:columns>
13 <mx:DataG dataField="col1"/>
14 <mx:DataG dataField="col2"/>
15 <mx:DataG dataField="col3"/>
16 </mx:columns>
17 </mx:DataGrid>
18 </s:Panel>
19 <fx:Script>
20 <![CDATA[

```

Le fonctionnement de l'assistant de contenu dans les éditeurs ActionScript et CSS est analogue.

Assistant de contenu dans l'éditeur MXML

Dans l'éditeur MXML, les indicateurs de code s'affichent automatiquement à la saisie du code. L'exemple suivant illustre les indicateurs de code qui s'affichent lorsque vous ajoutez une balise à une balise Panel ainsi que la documentation de référence ASDoc.

```

<s:Panel>
</s:Panel>
<s:Button>
<s:Panel>
<mx:Accordion - mx.containers
<mx:AccordionHeader - mx.containers.accordionClasses
<mx:AdvancedDataGrid - mx.controls
<mx:AdvancedDataGridGroupItemRenderer - mx.controls
<mx:AdvancedDataGridHeaderRenderer - mx.controls
<mx:AdvancedDataGridSortItemRenderer - mx.controls
<mx:AdvancedListBase - mx.controls.listClasses
<mx:Alert - mx.controls
<s:Application - spark.components

```

Il crée et gère les boutons de navigateur (entêtes d'accordéon) que vous pouvez utiliser pour accéder aux différents enfants. Un bouton de navigateur est associé à chaque conteneur enfant. Chaque bouton de navigateur appartient au conteneur Accordion et non à l'enfant. Lorsque l'utilisateur clique sur un bouton de navigateur, le conteneur enfant associé s'affiche. La transition vers le nouvel enfant fait appel à une animation pour bien faire comprendre à

L'assistant de contenu classe les indicateurs de code par type, répertoriant les composants MXML visuels et non visuels, par propriétés, par événements et par styles.

Par défaut, l'assistant de contenu affiche uniquement les indicateurs correspondant aux *types recommandés*. Les types recommandés sont les composants disponibles dans les espaces de noms ou autres, suivant les balises englobantes. Les composants dont vous disposez dépendent de la déclaration d'espace de noms de l'application concernée ainsi que des balises englobant le point d'insertion dans l'éditeur.

Dans certains contextes par exemple, seuls les composants Spark sont autorisés. D'autres contextes autorisent les composants Spark et les composants Halo. L'assistant de contenu filtre les indicateurs de code en fonction du contenu.

Appuyez sur les touches Ctrl+Espace pour faire défiler les filtres pour les indicateurs de code affichés. Les filtres sont les suivants :

- Affichage initial : Types recommandés
- Tous les composants
- Propriétés
- Événements

- Effets
- Styles
- Retour aux types recommandés

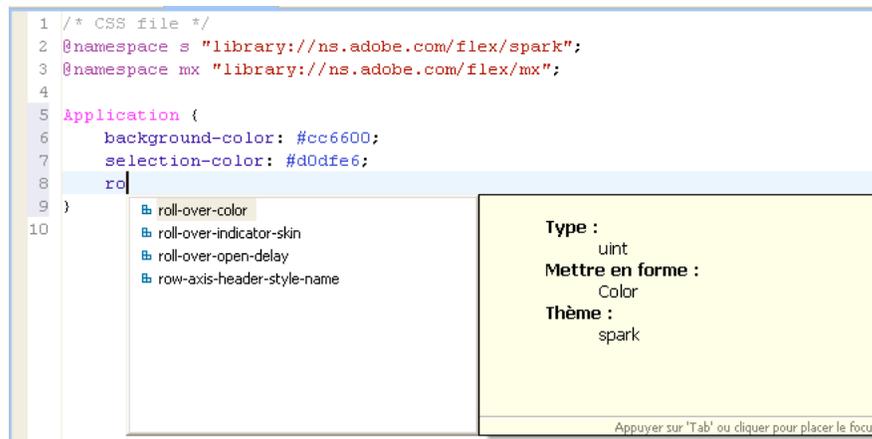
Les indicateurs de code affichés et l'ordre dans lequel ils défilent dépendent des préférences définies par l'utilisateur. Pour modifier les paramètres par défaut dans la boîte de dialogue Préférences :

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Flash Builder > Editeurs > Code MXML > Avancées.
- 2 Spécifiez les types d'indicateurs que vous souhaitez afficher.
- 3 Sélectionnez un type d'indicateur et modifiez-en la position dans le cycle de défilement en cliquant sur les boutons Haut et Bas.

Le premier type d'indicateur de la liste sera affiché en premier.

Affichage des indicateurs de code dans l'éditeur CSS

L'assistant de contenu affiche des indicateurs pour les styles CSS contenus dans des balises intégrées `<fx:Style>` ou dans des documents CSS autonomes, ainsi que l'illustre l'exemple suivant.



Remarque : dans les documents CSS, les indicateurs de code s'affichent uniquement si vous appuyez sur **Ctrl+Espace**.

Utilisation de l'assistant de contenu

Les indicateurs de code accélèrent et facilitent la rédaction de code MXML, ActionScript et CSS. Ils s'affichent en cours de saisie du code dans l'éditeur.

Affichage de l'assistant de contenu et insertion d'indicateurs de code

- 1 Commencez à saisir une ligne de code.

- Dans un document MXML, commencez à saisir une balise :

<

Les indicateurs de code correspondants s'affichent, ainsi que l'illustre l'exemple suivant :

Appuyez sur **Ctrl+Espace** pour parcourir la liste des indicateurs de code disponibles.

- Dans un document ActionScript ou une balise Script d'un document MXML, saisissez une structure de langage typique :

```
public var myVar:
```

- Dans un document CSS ou dans une balise Style d'un document MXML, saisissez une structure de nom de style, puis appuyez sur les touches Ctrl+Espace pour afficher la liste des attributs pouvant être ajoutés.

Vous pouvez également afficher les indicateurs en cours de saisie d'une ligne de code en actionnant les touches Ctrl+Espace.

- 2 Parcourez la liste d'indicateurs de code en utilisant les flèches vers le haut et vers le bas.
- 3 Sélectionnez un indicateur de code et appuyez sur Entrée pour insérer le code dans l'éditeur.
Au fur et à mesure que vous saisissez le code, d'autres indicateurs sont affichés.

Affichage de l'assistant de contenu et des commentaires ASDoc

- 1 Commencez à entrer une ligne de code qui contient une classe MXML ou ActionScript. Vous pouvez également survoler la classe avec le curseur de la souris.

Au fur et à mesure que vous tapez, le contenu ASDoc correspondant à la classe apparaît à côté des indicateurs de code. Si vous passez le curseur de la souris sur une classe, seul le commentaire ASDoc apparaît.

- 2 Cliquez à l'intérieur du contenu ASDoc ou appuyez sur la touche F2 pour afficher le contenu dans une fenêtre distincte que vous pouvez faire défiler.
- 3 Lorsque vous avez terminé la lecture de la documentation, cliquez hors de la fenêtre. La fenêtre ASDoc se ferme.

Détermination de la hiérarchie d'appels

Flash Builder fournit la vue Hiérarchie d'appels qui affiche tous les appelants d'une fonction, d'une variable, d'une classe ou d'un identifiant d'interface ActionScript ou MXML sélectionné. La liste des appelants est hiérarchique. Vous pouvez afficher les appelants pour chaque référence trouvée.

Vous pouvez également afficher le numéro de ligne et le code de ligne de chaque appelant. Double-cliquez sur l'appelant pour ouvrir le fichier source. Ce dernier met en évidence l'appel de fonction ou la référence.

Vous pouvez modifier l'orientation des appelants affichés et spécifier la portée de la recherche.

Affichage de la hiérarchie d'appels d'un élément de langage et ouverture du fichier source d'un appelant

- 1 En mode Source de l'éditeur MXML ou de l'éditeur ActionScript, placez le point d'insertion à l'intérieur d'une fonction, d'une variable, d'une classe ou d'un identifiant d'interface.
- 2 Dans le menu Naviguer, sélectionnez Ouvrir la hiérarchie d'appels.
Vous pouvez également utiliser le menu contextuel ou le raccourci clavier Ctrl+Alt+H.
- 3 Dans la vue Hiérarchie d'appels, développez chaque appelant pour afficher la hiérarchie des appels.
- 4 Double-cliquez sur un appelant pour ouvrir le fichier source à l'origine de l'appel dans l'éditeur. L'appel est mis en évidence.

Modification de l'orientation et de la portée de la hiérarchie d'appels

- 1 Affichez la hiérarchie d'appels d'un élément de langage.
- 2 Dans le menu Hiérarchie d'appels, sélectionnez Présentation.
Choisissez une option pour modifier la présentation.
- 3 Dans le menu Hiérarchie d'appels, sélectionnez Portée de la recherche.

Sélectionnez Espace de travail, Projet ou Fichier. La valeur par défaut est Espace de travail.

Création de gestionnaires d'événement pour les composants

Vous pouvez créer des gestionnaires d'événement pour les composants Flex en cliquant sur le bouton Générer le gestionnaire d'événement pour les événements répertoriés dans la vue Propriétés. Vous pouvez attribuer un nom au gestionnaire d'événement avant de le créer. Si vous ne le faites pas, Flash Builder crée un nom basé sur la propriété ID du composant. En l'absence de propriété ID, Flash Builder génère un nom unique dérivé du nom du composant.

Génération d'un gestionnaire d'événement pour un composant

- 1 Dans la vue Création de l'éditeur de code, sélectionnez un composant.
- 2 Dans la vue Propriétés, sélectionnez la Vue par catégories, développez la liste des événements pour l'élément sélectionné et localisez l'événement pour lequel vous souhaitez générer un gestionnaire d'événement.
- 3 (Facultatif) Saisissez un nom pour le gestionnaire d'événement dans le champ Valeur.
- 4 Cliquez sur le bouton Générer le gestionnaire d'événement.

L'éditeur ouvre la vue Source. Flash Builder insère le gestionnaire d'événement à l'intérieur d'un bloc <Script> et ajoute la propriété d'événement au composant. La propriété d'événement établit une référence à l'événement créé.

Remarque : vous pouvez également utiliser le menu contextuel pour créer des gestionnaires pour les principaux événements d'un composant. Par exemple, le menu contextuel d'un composant Button comporte l'option Générer un gestionnaire Click.

- 5 Ajoutez le code pour l'implémentation du gestionnaire d'événement à la fonction de gestionnaire d'événement générée.

Comment obtenir de l'aide pour la rédaction de code

Le *Guide de référence du langage Adobe Flex* est intégré aux éditeurs MXML et ActionScript, vous permettant ainsi de consulter rapidement l'Aide de référence pour une balise MXML ou pour une propriété, une classe ou tout autre élément de la structure Flex.

Vous pouvez également utiliser l'Aide dynamique, ancrée à proximité de la perspective en cours d'utilisation. Elle affiche les rubriques de référence et d'utilisation portant sur la balise MXML ou la classe ActionScript sélectionnée.

Affichage de l'Aide de la référence du langage

- 1 Dans l'éditeur MXML ou ActionScript, sélectionnez un élément (un mot du code) de la structure Flex en le mettant en surbrillance ou en plaçant le curseur de la souris à l'intérieur de l'élément (ou du mot).
- 2 Pour ouvrir la rubrique d'aide directement dans l'afficheur de l'Aide, appuyez sur les touches Maj+F2 ou sélectionnez Aide > Rechercher dans la référence du langage.

Activation de l'aide dynamique

- ❖ Sélectionnez Aide > Aide dynamique.

Navigation et organisation du code

Les éditeurs Flash Builder offrent de nombreuses fonctions pour parcourir le code. Vous pouvez par exemple réduire ou développer des blocs de code, ouvrir les sources de définitions de code ainsi que rechercher et ouvrir des types. Des blocs de plusieurs lignes peuvent être réduits et développés afin de faciliter la navigation, l'affichage et la gestion de documents présentant un code complexe. Dans Flash Builder, ces deux opérations sont désignées par les termes de *développement* et de *réduction* d'instructions de code de plusieurs lignes.

Configuration, réduction et développement de blocs de code

- 1 Dans l'éditeur, cliquez sur le symbole de réduction (-) ou sur celui de développement (+) dans la marge de gauche.

```
10 public function set cartItems(items:ShoppingCart):Void {  
11     _cartItems = items;  
12     dg.dataProvider = _cartItems.items;  
13 }
```

La réduction d'un bloc de code en masque toutes les lignes à l'exception de la première.

```
10 public function set cartItems(items:ShoppingCart):Void {  
14 }
```

Développez le bloc de code pour en afficher à nouveau le contenu. Placez la souris sur le symbole de développement (+) pour visionner l'ensemble du bloc dans une info-bulle.

```
10 public function set cartItems(items:ShoppingCart):Void {  
14     _cartItems = items;  
15     dg.dataProvider = _cartItems.items;  
16 }  
17  
18     if (_cartItems.items.length == 0)
```

- 2 La réduction de code est activée par défaut dans Flash Builder. Pour la désactiver, ouvrez la boîte de dialogue Préférences et sélectionnez Flash Builder > Editeurs. Désélectionnez ensuite l'option Réduction des codes.

Utilisation de la vue Structure pour parcourir et examiner le code

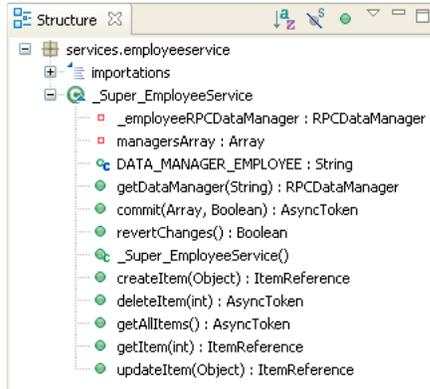
La vue Structure appartient à la perspective Développement Flash (voir « [Perspective Développement Flash](#) » à la page 8). Elle est donc disponible au cours de la modification du code et de la création de l'application. Cette vue permet d'examiner et de parcourir plus facilement la structure des documents MXML, ActionScript et CSS.

Elle présente trois modes : le mode Classe, le mode MXML et le mode CSS. En mode Classe, cette vue affiche la structure du code (classes, variables des membres, fonctions, etc.). En mode MXML, elle affiche la structure MXML (balises, composants, contrôles, etc.). En mode CSS, elle affiche les sélecteurs CSS et les propriétés imbriquées qu'ils contiennent.

La sélection d'un élément dans la vue Structure permet de le localiser et de le mettre en évidence dans l'éditeur, facilitant ainsi considérablement la navigation à l'intérieur du code.

Vue Structure en mode Classe

Lorsque vous éditez un document ActionScript (ou un code ActionScript contenu dans un document MXML), la vue Structure affiche la structure du code. Cette structure comporte les instructions d'importation, les packages, les classes, les interfaces, les variables non contenues dans les fonctions et les fonctions. En revanche, elle ne contient aucune métadonnée, aucun commentaire, aucune déclaration d'espace de noms et aucun contenu de fonction.



Les nœuds et les éléments de l'arborescence de la vue Structure représentent à la fois les différents types d'éléments de langage et leur visibilité. Les icônes rouges signalent par exemple des éléments privés ; les icônes vertes signalent des éléments publics ; les icônes jaunes indiquent que l'élément n'a pas été défini comme étant privé ou public.

Barre d'outils de la vue Structure en mode Classe

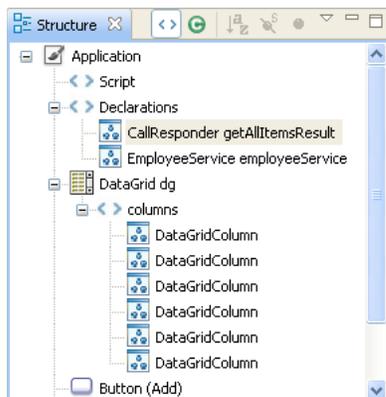
En mode Classe, la barre d'outils de la vue Structure contient les commandes de tri et de filtrage, ainsi que l'illustre l'exemple suivant :



Vue Structure en mode MXML

La vue Structure d'un document MXML en cours d'édition contenant du code MXML et du code ActionScript présente les modes Classe et MXML.

En mode MXML, chaque élément de la vue représente une balise MXML. Les types de balise affichés sont les suivants : les composants, les contrôles, les balises non visuelles (`WebService` ou `State`), les propriétés des composants exprimées sous forme de balises enfant (contraintes de présentation, par exemple) et les balises de compilateur (`Model`, `Array` et `Script`).



Le mode MXML de la vue Structure n'affiche aucun commentaire, aucune règle ou propriété CSS et aucune propriété de composant exprimée sous forme d'attribut (par opposition aux balises enfant qui sont affichées).

Barre d'outils de la vue Structure en mode MXML

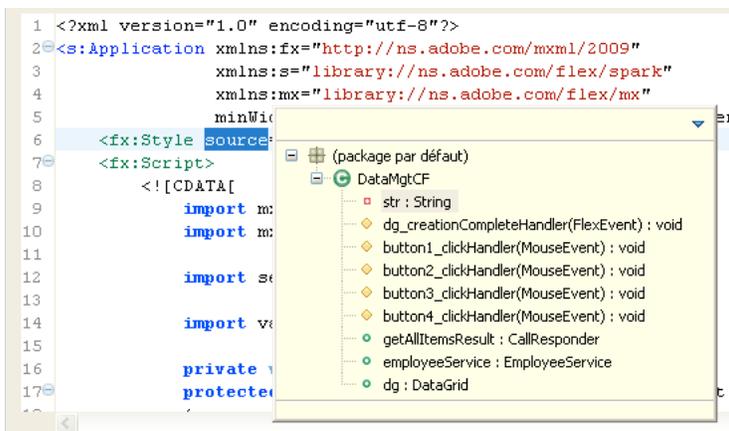
En mode MXML, la barre d'outils de la vue Structure contient des commandes supplémentaires permettant de basculer de la vue MXML à la vue des classes.



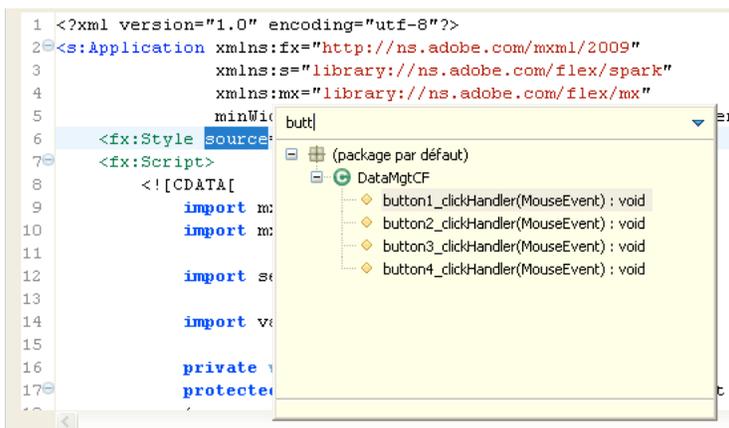
Pour basculer entre les deux vues, sélectionnez Afficher la vue MXML ou Afficher la vue des classes dans le menu de la barre d'outils.

Utilisation de la vue Structure rapide de l'éditeur

Vous pouvez ouvrir la vue Structure rapide à partir des éditeurs ActionScript et MXML pour afficher la vue Structure en mode Classe. La vue Structure rapide apparaît dans une fenêtre contextuelle à l'intérieur de l'éditeur et ne se présente donc pas sous forme de vue distincte. Elle permet de parcourir et d'examiner rapidement le code.



Son contenu est identique à celui du mode Classe, à ceci près que la Structure rapide présente une zone de saisie de texte permettant de filtrer les éléments affichés. Vous pouvez par exemple entrer le nom d'un élément dans la vue Structure rapide afin de limiter l'affichage aux éléments contenant les caractères saisis.



La vue Structure rapide ne contient pas les commandes permettant d'effectuer un tri alphabétique des éléments ou de les masquer.

De même que dans la vue Structure, vous pouvez sélectionner un élément afin de le localiser et de le mettre en évidence dans l'éditeur.

Ouverture de la vue Structure rapide

- ❖ Ouvrez un document ActionScript ou MXML dans l'éditeur. Dans le menu Naviguer, sélectionnez Structure rapide.

Vous pouvez également utiliser le raccourci clavier Ctrl+O.

Fermeture de la vue Structure rapide

- ❖ L'accès à tout emplacement situé en dehors de la vue Structure rapide conduit à la fermeture de cette dernière. Elle peut également être fermée en appuyant sur la touche Echap.

Ouverture de définitions de code

Quelle que soit la complexité des applications, les projets contiennent généralement de nombreuses ressources et lignes de code. Afin de parcourir et d'examiner plus facilement les différents éléments du code, vous pouvez ouvrir la source d'une définition de code externe à partir de l'emplacement, dans le code, auquel il y est fait référence. Si vous créez par exemple un composant MXML personnalisé et l'importez dans l'application MXML, vous pouvez sélectionner la référence au composant MXML et ouvrir le fichier source dans l'éditeur.

Ouverture de la source d'une définition de code

- 1 Dans l'éditeur, sélectionnez la référence du code.
- 2 Dans le menu Naviguer, sélectionnez Accéder à la définition.

Vous pouvez utiliser le raccourci clavier F3.

Le fichier source contenant la définition du code s'ouvre dans l'éditeur.

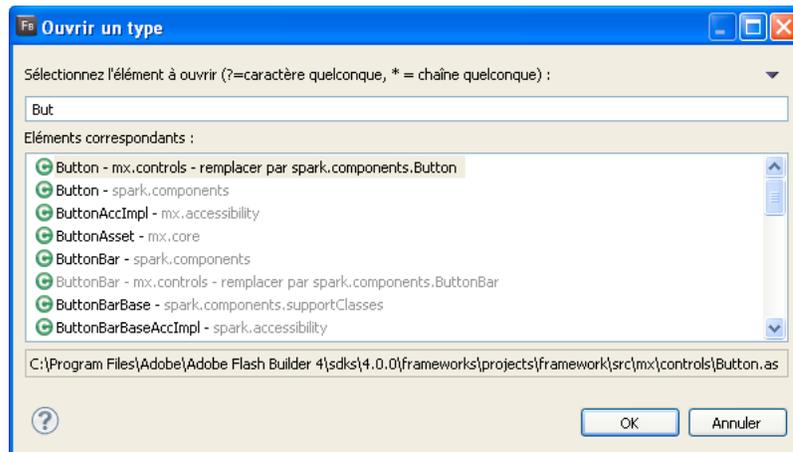
Flash Builder prend également en charge la navigation dans le code par hyperliens.

Ouverture de la source d'une définition de code en utilisant la navigation par hyperliens

- 1 Localisez la référence du code dans l'éditeur.
- 2 Appuyez sur la touche Ctrl (Windows) ou la touche Cmd (Mac OS) et maintenez-la enfoncée, puis positionnez le curseur de la souris sur la référence du code pour afficher l'hyperlien.
- 3 Pour atteindre la référence du code, cliquez sur l'hyperlien.

Recherche et affichage des classes

La boîte de dialogue Ouvrir un type permet de parcourir toutes les classes disponibles (y compris les classes de la structure Flex) du projet. Dans la boîte de dialogue Ouvrir un type, sélectionnez une classe pour afficher l'implémentation.



Boîte de dialogue Ouvrir un type

La boîte de dialogue Ouvrir un type permet également de sélectionner des classes comme classe de base pour une nouvelle classe ActionScript ou un nouveau composant MXML.

Elle comporte une zone de texte permettant de filtrer les classes affichées en fonction du texte ou du joker qui y est saisi. La boîte de dialogue fait appel à différentes couleurs pour signaler les types recommandés et les types exclus. Les types recommandés sont affichés en gris. Les types exclus sont affichés en marron.

Les *types recommandés* sont les classes disponibles dans l'espace de noms par défaut d'un projet. Dans certains contextes par exemple, seuls les composants Spark sont autorisés. D'autres contextes autorisent les composants Spark et les composants Halo.

Les *types exclus* sont les classes qui ne sont pas disponibles dans l'espace de noms par défaut pour un projet.

Ouverture de la boîte de dialogue Ouvrir un type

- (Parcourir les classes) Pour parcourir les classes et visionner leur implémentation :
 - 1 Dans le menu Flash Builder, sélectionnez Naviguer > Ouvrir un type.
 - 2 (Facultatif) Saisissez du texte ou sélectionnez des filtres pour modifier les classes visibles dans la liste.
 - 3 Sélectionnez une classe pour en afficher le code source.
 - Vous ne pouvez pas modifier le code source des classes dans la structure Flex.
- (Nouvelles classes ActionScript) Pour sélectionner une classe de base pour une nouvelle classe ActionScript :
 - 1 Sélectionnez Fichier > Nouveau > Classe ActionScript.
 - 2 En regard du champ Superclasse, cliquez sur Parcourir.
 - 3 (Facultatif) Saisissez du texte ou sélectionnez des filtres pour modifier les classes visibles dans la liste.
 - 4 Sélectionnez une classe de base dans la liste.
- (Nouveaux composants MXML) Pour sélectionner un composant de base pour un nouveau composant MXML :
 - 1 Cliquez sur Fichier > Nouveau > Composant MXML.
 - 2 Dans la liste des projets de l'espace de travail, sélectionnez un projet pour un nouveau composant MXML et spécifiez un nom pour le fichier.
 - Les composants de base disponibles varient en fonction des espaces de noms configurés pour un projet.
 - 3 En regard du champ Basé sur, cliquez sur Parcourir.

Remarque : effacez ou modifiez la classe de base par défaut répertoriée dans le champ Basé sur pour élargir vos choix.

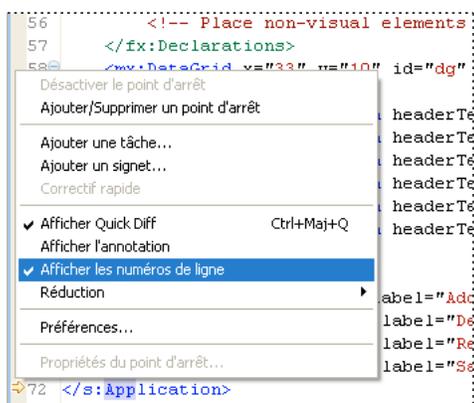
- 4 (Facultatif) Saisissez du texte ou sélectionnez des filtres pour modifier les classes visibles dans la liste.
- 5 Sélectionnez un composant de base dans la liste.

Affichage des numéros de ligne

Vous pouvez ajouter des numéros de ligne dans l'éditeur afin de faciliter la lecture et la consultation du code.

- ❖ Dans le menu contextuel de la marge de l'éditeur, sélectionnez Afficher les numéros de ligne.

La marge de l'éditeur se situe entre la barre de repère et l'éditeur.



Formatage et modification du code

Les éditeurs de Flash Builder offrent des fonctions accélérant la rédaction et le formatage du code, telles que l'ajout rapide de blocs de commentaires, la mise en retrait de blocs de code ainsi que la recherche et le remplacement de texte.

Organisation des instructions d'importation

L'utilisation de l'assistant de contenu dans les éditeurs MXML et ActionScript conduit à l'importation automatique dans le document des packages dans lesquels les classes se situent. Les packages sont ajoutés dans l'ordre dans lequel ils ont été saisis dans le code. Les importations qui ne sont pas utilisées ou ne sont pas nécessaires sont automatiquement supprimées.

Afin de faciliter l'organisation du code dans les documents ActionScript, vous pouvez trier les instructions d'importation par ordre alphabétique. Pour ce faire, ouvrez la boîte de dialogue Préférences, sélectionnez Flash Builder > Éditeurs > Code ActionScript, puis Garder les importations organisées.

Tri des instructions d'importation

- ❖ Dans l'éditeur, ouvrez un document ActionScript contenant les instructions d'importation. Dans le menu Source, sélectionnez Organiser les importations.

Vous pouvez également utiliser le raccourci clavier suivant : Ctrl+Maj+ O (Windows) ou Commande+Maj+O (Mac).

Ajout de commentaires et de blocs de commentaire

Vous pouvez ajouter ou supprimer des commentaires en utilisant les options dans le menu Source ou en faisant appel aux raccourcis clavier. Vous pouvez ajouter les types de commentaires suivants :

- Commentaires de source pour ActionScript (//)
- Commentaires de bloc pour ActionScript (/* */)
- Commentaires ASDoc pour ActionScript (/** */)
- Commentaires de bloc pour MXML (<!---->)
- Blocs CDATA pour MXML (<![CDATA[]]>)

Les commentaires ajoutés au code ActionScript peuvent être activés ou désactivés.

Activer/désactiver les commentaires du code ActionScript

- 1 Dans l'éditeur, sélectionnez au moins une ligne de code ActionScript.
- 2 Appuyez sur Ctrl+Maj+C (Windows) ou Cmd+Maj+C (Mac OS) pour ajouter ou supprimer des commentaires de style C.
- 3 Appuyez sur Ctrl+/ (Windows) ou Cmd+/ (Mac OS) pour ajouter ou supprimer des commentaires de style C++.

Ajout de commentaires XML à du code MXML

- 1 Dans l'éditeur, sélectionnez au moins une ligne de code MXML.
- 2 Appuyez sur Ctrl+Maj+C (Windows) ou Cmd+Maj+C (Mac OS) pour ajouter un commentaire.

Ajout de blocs CDATA à du code MXML

- 1 Dans l'éditeur, sélectionnez au moins une ligne de code MXML.
- 2 Appuyez sur Ctrl+Maj+D (Windows) ou Cmd+Maj+D (Mac OS) pour ajouter un commentaire.

Mise en retrait de blocs de code

L'éditeur met automatiquement en forme les lignes de code en cours de saisie, améliorant ainsi la lisibilité et facilitant la rédaction. Vous pouvez également utiliser la touche de tabulation pour mettre manuellement en retrait certaines lignes de code.

Lorsque vous copiez et collez des blocs de code dans Flash Builder, ce dernier met automatiquement en retrait le code en fonction de vos préférences.

Pour mettre en retrait un bloc de code en une seule opération, vous pouvez utiliser les commandes Décaler vers la droite et Décaler vers la gauche de l'éditeur.

Déplacement d'un bloc de code vers la gauche ou vers la droite

- 1 Dans l'éditeur, sélectionnez un bloc de code.
- 2 Cliquez sur Source > Décaler vers la droite ou Source > Décaler vers la gauche.
- 3 Appuyez sur la touche de tabulation ou sur Maj et la touche de tabulation pour mettre en retrait ou annuler la mise en retrait de blocs de code.

Définition des préférences de la mise en retrait

- 1 Ouvrez la boîte de dialogue Préférences et sélectionnez Flash Builder > Mise en retrait.

- 2 Sélectionnez le type de mise en retrait (tabulations ou espaces) et spécifiez la taille de la mise en retrait ou des tabulations.

Recherche et remplacement de texte dans l'éditeur

Vous disposez de deux options pour rechercher et éventuellement remplacer des chaînes de texte dans le code. Vous pouvez limiter la recherche au document ouvert dans l'éditeur ou l'étendre à toutes les ressources des projets de l'espace de travail. Pour plus d'informations sur la recherche portant sur tout l'espace de travail, voir « [Recherche de références et restructuration de code](#) » à la page 118.

- 1 Ouvrez le document sur lequel la recherche doit porter.
- 2 Utilisez l'une des méthodes suivantes :
 - Appuyez sur Ctrl+F (Windows) ou Cmd+F (Mac OS).
 - Sélectionnez Editer > Rechercher/Remplacer.
- 3 Saisissez la chaîne de texte à localiser.
- 4 (Facultatif) Entrez la chaîne de texte de remplacement.
- 5 (Facultatif) Définissez les critères de recherche avancés.
- 6 Cliquez sur Rechercher, Remplacer, Remplacer tout ou Remplacer/Rechercher.

Une fois localisée, la chaîne de texte est mise en surbrillance et éventuellement remplacée.

Remarque : pour procéder à une recherche incrémentielle, appuyez sur Ctrl+J (Windows) ou Cmd+J (Mac OS).

Recherche de références et restructuration de code

Flash Builder comporte des fonctions de recherche avancées plus puissantes que les fonctions de recherche/remplacement. Afin de faciliter la compréhension de la manière dont les fonctions, variables et autres identifiants sont utilisés, Flash Builder permet de rechercher et de repérer les références ou les déclarations vers les identifiants dans les fichiers ActionScript ou MXML, les projets ou les espaces de travail. La restructuration est utilisée pour apporter des modifications au code en renommant les identifiants suivants, puis en mettant à jour toute référence y renvoyant.

- Variables
- Fonctions
- Types (interface, classe)
- Accesseurs (getter/setter)
- Attributs
- Métadonnées en MXML (effets, événements, styles)

Marquage des références

- 1 En mode Source, cliquez sur le bouton Marquage des occurrences de la barre d'outils.
- 2 Dans l'éditeur, cliquez sur un identifiant. Toutes les occurrences sont marquées en fonction des paramètres des Préférences.

Pour modifier l'apparence des références marquées, sélectionnez Général > Editeurs > Editeurs de texte > Annotations. Pour plus d'informations sur les marqueurs, voir « [A propos des marqueurs](#) » à la page 122.

Recherche de toutes les références et de toutes les déclarations

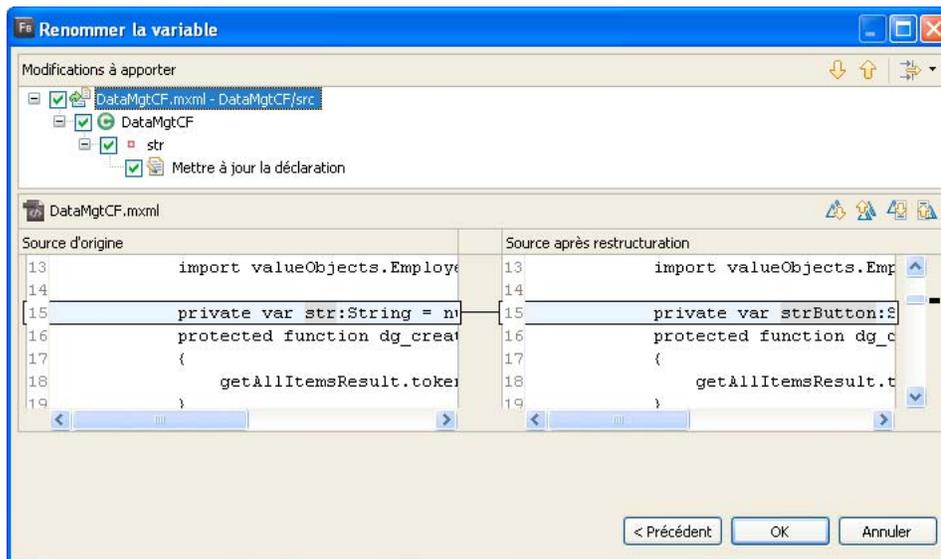
- 1 En mode Source, cliquez sur un identifiant dans l'éditeur.
- 2 Dans le menu principal, sélectionnez Rechercher > Références ou Rechercher > Déclarations. Sélectionnez ensuite Fichier, Projet ou Espace de travail. Les résultats apparaissent dans la vue Recherche.

Restructuration du code

- 1 En mode Source, cliquez sur un identifiant dans l'éditeur.
- 2 Dans le menu principal, sélectionnez Source > Restructurer > Renommer.
- 3 Entrez un nouveau nom.

Flash Builder vérifie que les conditions préalables à la modification du nom sont réunies et vous invite à confirmer les problèmes avant de procéder à la modification du nom. Les conditions préalables à la modification du nom sont les suivantes :

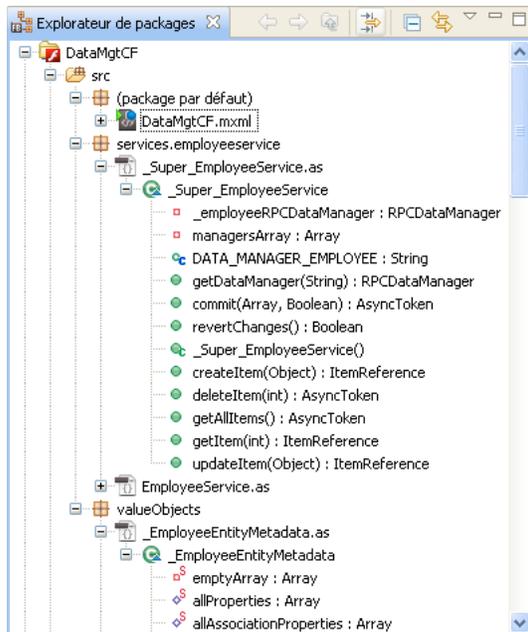
- Les références situées dans des fichiers en lecture seule ne peuvent pas être renommées.
 - Tous les fichiers doivent avoir été sauvegardés.
 - Les projets présentant des erreurs de génération provoquent l'affichage d'un avertissement.
 - Le nouveau nom doit se trouver dans l'étendue, déterminée par le type de l'élément et son emplacement. Les erreurs d'occultation de noms sont également signalées.
 - Le nouveau nom doit être un identifiant valide.
 - La référence définie dans un fichier SWC doit comprendre une connexion de la source.
- 4 Pour vérifier la modification apportée, cliquez sur Aperçu pour visionner la source d'origine et la source restructurée, ou cliquez sur OK pour poursuivre la modification du code.



Icônes représentant les éléments de langage

Flash Builder fait appel à des icônes et à des incrustations afin de fournir des repères visuels pour les éléments de langage. Les icônes sont visibles :

- dans l'Explorateur de packages ;
- dans la vue de débogage Variables ;
- dans les indicateurs des blocs de code ActionScript ;
- dans la vue ASDoc ;
- dans la boîte de dialogue Ouvrir un type lors de la création ou de la sélection de classes.



Explorateur de packages affichant les icônes des éléments de langage

Les icônes représentent des espaces de noms, des classes, des interfaces, des fonctions et des variables.

Icône	Forme	Couleur	Description
	Cercle large	Vert	Classe
	Cercle large	Violet	Interface
	N	Violet	Espace de noms
	Cercle (contour)	Vert	Variable, étendue publique
	Cercle (plein)	Vert	Fonction, étendue publique
	Triangle (contour)	Bleu	Variable, étendue interne
	Triangle (plein)	Bleu	Fonction, étendue interne

Icône	Forme	Couleur	Description
	Diamant (contour)	Jaune	Variable, étendue protégée
	Diamant (plein)	Jaune	Fonction, étendue protégée
	Diamant (contour)	Violet	Variable, espace de noms personnalisé
	Diamant (plein)	Violet	Fonction, espace de noms personnalisé
	Carré (contour)	Rouge	Variable, étendue privée
	Carré (plein)	Rouge	Fonction, étendue privée

Une incrustation est une lettre ou un symbole affiché avec une icône fournissant des informations supplémentaires sur l'élément de langage. Par exemple, l'incrustation « D » signale une variable ou une fonction dynamique.

Incrustation	Symbole	Couleur	Description
	S	Rouge	Statique
	D	Rouge	Dynamique
	C	Bleu	Constante
	C	Vert	Constructeur pour une fonction
	F	Bleu	Final
	L	Gris	Local
	=	Sans objet	Accesneur, méthode Get ou méthode Set
	=	Sans objet	Exemple de fonction d'accesneur public

Flash Builder fait également appel aux icônes pour signaler des événements, des effets et des styles.

Icône	Description
	Événements
	Effets
	Styles

Affichage d'indicateurs de code dans l'éditeur ActionScript

L'assistant de contenu pour ActionScript fait appel à des icônes avec indicateurs de code pour signaler les types d'éléments de langage disponibles.

L'assistant de contenu affiche des indicateurs de code pour ActionScript 3.0. Ces indicateurs de code sont disponibles dans l'éditeur ActionScript ainsi qu'à l'intérieur des balises `<fx:script>` et dans les attributs d'événement dans les documents MXML. L'assistant de contenu fournit des indicateurs pour tous les éléments du langage ActionScript 3.0 (interfaces, classes, variables, fonctions et types de retour), ainsi que l'illustre l'exemple suivant.



A propos des marqueurs

Les marqueurs sont des raccourcis vers les lignes de code d'un document, vers un document ou vers un dossier. Ils représentent des tâches, des signets et des problèmes. Ils sont affichés et peuvent être gérés. La sélection de marqueurs provoque l'ouverture du document associé dans l'éditeur et met éventuellement en évidence la ligne concernée du code.

Pour mettre à jour les marqueurs de problèmes dans Flash Builder, vous devez enregistrer le fichier. Seuls les fichiers auxquels l'application fait référence sont vérifiés. La syntaxe d'une classe isolée qui n'est pas utilisée dans le code n'est pas contrôlée.

Le workbench génère automatiquement les marqueurs de tâches et de problèmes suivants. Vous pouvez ajouter des tâches et des signets manuellement.

Tâches Les marqueurs de tâches représentent un élément de travail. Les éléments de travail sont générés automatiquement par le workbench. Vous pouvez ajouter manuellement une tâche à une ligne de code du document ou au document même. Par exemple, afin de penser à définir une propriété pour un composant, vous pouvez créer une tâche nommée « Définir les propriétés de l'aspect ». Vous pouvez également ajouter des tâches générales qui ne s'appliquent pas directement aux ressources (« Créer un composant personnalisé pour l'invite de connexion de l'employé », par exemple). La vue Tâches permet de gérer tous les marqueurs de tâches. Pour plus d'informations, voir « [Ajout de tâches](#) » à la page 123.

Erreurs Les marqueurs d'erreurs sont générés par le compilateur et signalent différents types d'états non valides. Par exemple, les erreurs de syntaxe et les avertissements générés par le compilateur sont affichés comme marqueurs dans la vue Erreurs. Pour plus d'informations, voir « [Utilisation de la vue Erreurs](#) » à la page 125.

Signets Vous pouvez ajouter manuellement des signets à une ligne de code ou à une ressource (dossier ou document). Les signets sont pratiques pour repérer et rechercher facilement des éléments au sein des projets. Ils sont gérés dans la vue Signets. Pour plus d'informations, voir « [Ajout et suppression de signets](#) » à la page 124.

Remarque : les vues Tâches et Signets ne sont pas affichées par défaut dans la perspective Développement Flash. Pour plus d'informations sur l'affichage de ces vues, voir « [Ouverture des vues](#) » à la page 26.

Recherche de marqueurs

Les *marqueurs* sont à la fois des descriptions et des liens vers des éléments des ressources d'un projet. Qu'ils soient générés automatiquement par le compilateur pour signaler des erreurs dans le code ou ajoutés manuellement pour repérer des tâches ou des fragments de code, les marqueurs sont affichés et gérés dans les vues associées. Les marqueurs d'un projet sont facilement repérables dans les vues Signets, Erreurs et Tâches, à partir desquelles ils sont accessibles.

Accès à l'emplacement d'un marqueur

- ❖ Sélectionnez un marqueur dans les vues Signets, Erreurs ou Tâches.

Le fichier contenant le marqueur est localisé et affiché dans l'éditeur. Si l'emplacement du marqueur est une ligne de code, cette dernière est mise en surbrillance.

Ajout de tâches

Les tâches sont des éléments de l'espace de travail créés automatiquement ou manuellement. Toutes les tâches sont affichées et gérées dans la vue Tâches (Fenêtre > Autres vues > Généralités > Tâches), ainsi que l'illustre l'exemple suivant.

	Description	Ressource	Chemin	Emplacement	Type
<input type="checkbox"/>	par exemple	Sample.mxml	/Sample/src	ligne 7	Tâche
<input type="checkbox"/>	bouton Ajouter	Sample.mxml	/Sample/src	ligne 9	Tâche

Ajout d'une tâche à une ligne de code ou à une ressource

- 1 Ouvrez un fichier dans l'éditeur, puis repérez et sélectionnez la ligne de code à laquelle vous souhaitez ajouter une tâche. Vous pouvez également sélectionner une ressource dans l'Explorateur de packages.
- 2 Dans la vue Tâches, cliquez sur le bouton Ajouter une tâche dans la barre d'outils.
- 3 Entrez le nom de la tâche et sélectionnez une priorité (Elevée, Normale, Faible), puis cliquez sur OK.

Remarque : ainsi que vous pouvez le constater dans l'Explorateur de packages, la ressource ne signale pas qu'elle contient un marqueur. Vous pouvez visionner et gérer tous les marqueurs de tâches dans la vue Tâches.

Achèvement et suppression de tâches

Une fois achevée, une tâche peut être marquée comme ayant été terminée et peut éventuellement être supprimée de la vue Tâches.

Définition d'une tâche comme étant terminée

- ❖ Dans la vue Tâches, sélectionnez une tâche de la colonne, ainsi que l'illustre l'exemple suivant.

	Description	Ressource	Chemin	Emplacement	Type
<input type="checkbox"/>	par exemple	Sample.mxml	/Sample/src	ligne 7	Tâche
<input checked="" type="checkbox"/>	bouton Ajouter	Sample.mxml	/Sample/src	ligne 9	Tâche

Suppression d'une tâche

- ❖ Dans la vue Tâches, accédez au menu contextuel d'une tâche et sélectionnez Supprimer.

Suppression de toutes les tâches terminées

- ❖ Dans la vue Tâches, accédez au menu contextuel et sélectionnez Supprimer les tâches terminées.

Ajout et suppression de signets

Les signets permettent de repérer et rechercher facilement des éléments au sein des projets. Tous les signets sont affichés et gérés dans la vue Signets (Fenêtre > Autres vues > Généralités > Signets), ainsi que l'illustre l'exemple suivant.

Description	Ressource	Chemin	Emplacement
Button	Sample.mxml	/Sample/src	ligne 9
DataGrid	Sample.mxml	/Sample/src	ligne 11

Ajout d'un signet à une ligne de code ou à une ressource

- 1 Ouvrez un fichier dans l'éditeur, puis localisez et sélectionnez la ligne de code à laquelle vous souhaitez ajouter un signet.
- 2 Dans le menu principal, sélectionnez Editer > Ajouter un signet.
- 3 Saisissez le nom du signet et cliquez sur OK.

Une icône de signet (📌) apparaît en regard de la ligne de code.

Remarque : ainsi que vous pouvez le constater dans l'Explorateur de packages, la ressource ne signale pas qu'elle contient un marqueur. Vous pouvez visionner et gérer tous les signets dans la vue Signets.

Suppression d'un signet

- 1 Dans la vue Signets, sélectionnez le signet à supprimer.
- 2 Cliquez sur le signet avec le bouton droit de la souris (Windows) ou en appuyant sur la touche Ctrl (Mac OS), puis sélectionnez Supprimer.

A propos de la vérification de la syntaxe

Le compilateur Flash Builder détecte les erreurs de syntaxe et les signale afin d'en permettre la correction en cours de rédaction, avant d'exécuter l'application. Les couleurs de la syntaxe peuvent être personnalisées.

Lorsqu'une erreur de syntaxe est détectée dans le code, elle vous est signalée de l'une des manières suivantes.

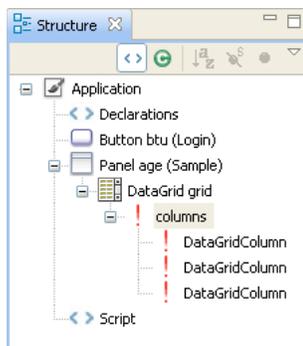
- Un indicateur d'erreur apparaît en regard de la ligne de code, ainsi que l'illustre l'exemple suivant.

```

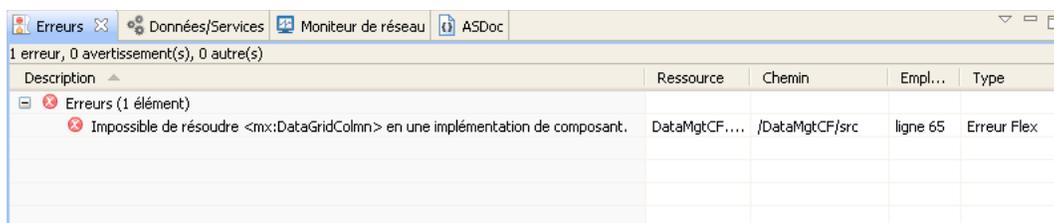
11 <mx:DataGrid x="10" id="grid" y="10" width="203">
12   <mx:columns>
13     <mx:DataGridColumn headerText="Colonne 1" dataField="col1"/|
14     <mx:DataGridColumn headerText="Colonne 2" dataField="col2"/>
15     <mx:DataGridColumn headerText="Colonne 3" dataField="col3"/>
16   </mx:columns>
17 </mx:DataGrid>

```

- La vue Structure signale les erreurs par un point d'exclamation affiché dans les lignes de code concernées, ainsi que l'illustre l'exemple suivant.



- La vue Erreurs répertorie les erreurs en leur attribuant un symbole et en affichant le message correspondant. Double-cliquez sur le message d'erreur pour localiser et mettre en évidence la ligne de code dans l'éditeur, ainsi que l'illustre l'exemple suivant.



Les erreurs de syntaxe du code sont détectées au cours de la génération des projets. Si elles ne sont pas corrigées avant l'exécution de l'application, un message vous avertit de l'existence d'erreurs. En fonction de la nature et de la gravité des erreurs, il est possible que l'application ne soit exécutée correctement qu'après correction de ces erreurs.

Application des préférences de couleurs pour la syntaxe

- ❖ Ouvrez la boîte de dialogue Préférences et sélectionnez Flash Builder > Editeurs > Couleurs pour la syntaxe.

Les couleurs par défaut peuvent également être définies dans les pages Editeurs de texte et Couleurs et polices des Préférences (voir Préférences > Général > Apparence > Couleurs et polices ; voir également Préférences > Général > Editeurs > Editeurs de texte).

Utilisation de la vue Erreurs

Au fur et à mesure que vous saisissez et enregistrez du code, Flash Builder le compile en arrière-plan et affiche les erreurs de syntaxe et les avertissements (erreurs) dans la vue Erreurs. L'Explorateur de packages signale les nœuds contenant des erreurs.

Chaque erreur ou avertissement contient un message, le fichier et le dossier correspondants ainsi que le numéro de la ligne dans le fichier. Les erreurs demeurent affichées dans la vue Erreurs jusqu'à ce que vous les corrigiez ou qu'elles soient résolues.

Accès à la ligne de code correspondant à une erreur ou à un avertissement

- ❖ Dans la vue Erreurs, double-cliquez sur une erreur ou sélectionnez Aller à dans le menu contextuel de l'erreur. Flash Builder ouvre un éditeur dans lequel la ligne de code contenant l'erreur est mise en évidence.

Raccourcis clavier d'édition de code

Le tableau suivant contient une liste de raccourcis clavier utiles pour l'édition de code.

Pour obtenir une liste complète des raccourcis clavier disponibles, voir « [Accès aux raccourcis clavier](#) » à la page 33. Pour plus d'informations sur la modification ou la création de raccourcis clavier, voir « [Modification des raccourcis clavier](#) » à la page 30.

Nom	Raccourci clavier	Description
Commuter entre les modes Source et Création	Ctrl+` (guillemet ouvrant)	Permet de commuter entre les modes Source et Création de l'éditeur MXML.
Accéder à la documentation (Flash Builder version plug-in) Rechercher dans la référence API (Flash Builder version autonome)	Maj+F2	Affiche l'aide de référence pour l'élément de langage sélectionné dans le code MXML ou ActionScript en appuyant sur Maj+F2. Pour plus d'informations, voir « Comment obtenir de l'aide pour la rédaction de code » à la page 110.
Aide contextuelle	F1 (Windows) Cmd+Maj+? (Mac OS)	Affiche l'aide contextuelle de l'élément sélectionné dans le workbench (éditeur, vue, boîte de dialogue et ainsi de suite).
Ajouter un commentaire de bloc	Ctrl+Maj+C (Windows) Cmd+Maj+C (Mac OS)	Ajoute la mise en forme de commentaire de bloc aux lignes de code sélectionnées ou ajoute un commentaire au point d'insertion. Pour plus d'informations, voir « Ajout de commentaires et de blocs de commentaire » à la page 117.
Ajouter des données CDATA	Ctrl+Maj+D (Windows) Cmd+Maj+D (Mac OS)	Ajoute une instruction CDATA au point d'insertion afin de vous permettre d'ajouter du code ActionScript à un document MXML.
Rechercher le crochet correspondant	Ctrl+Maj+P (Windows) Cmd+Maj+P (Mac OS)	Déplace le curseur vers le crochet correspondant de l'instruction de code sélectionnée.
Assistant de contenu	Ctrl+Espace (Windows) Cmd+Maj+Espace (Mac OS)	Affiche les indicateurs de code. Pour plus d'informations, voir « Utilisation de l'assistant de contenu » à la page 108.
Rechercher toutes les déclarations dans l'espace de travail	Ctrl+G (Windows) Cmd+G (Mac OS)	Recherche les déclarations dans la base de code. Voir « Recherche de références et restructuration de code » à la page 118.
Rechercher toutes les références dans l'espace de travail	Ctrl+Maj+G (Windows) Cmd+Maj+G (Mac OS)	Recherche dans la base de code les références correspondant aux identifiants. Voir « Recherche de références et restructuration de code » à la page 118.
Accéder à la définition	F3	Ouvre la source d'une définition de code externe. Pour plus d'informations, voir « Ouverture de définitions de code » à la page 114.
Accéder à la ligne	Ctrl+L (Windows) Cmd+L (Mac OS)	Affiche la boîte de dialogue Accéder à la ligne, dans laquelle vous pouvez saisir un numéro de ligne pour l'afficher dans l'éditeur.
Emplacement de la dernière édition	Ctrl+Q (Windows) Ctrl+Q (Mac OS)	Met en surbrillance la dernière ligne de code modifiée.
Marquer les occurrences	Sans objet	Signale chaque occurrence dans le code de l'élément sélectionné.

Nom	Raccourci clavier	Description
Organiser les importations	Ctrl+Maj+O (Windows) Cmd+Maj+O (Mac OS)	Classe par ordre alphabétique les instructions d'importation du document lorsque vous modifiez du code ActionScript. Pour plus d'informations, voir « Organisation des instructions d'importation » à la page 116.
Ouvrir un type	Ctrl+Maj+T (Windows) Cmd+Maj+T (Mac OS)	Permet de parcourir rapidement tous les types de classe. Pour plus d'informations, voir « Recherche et affichage des classes » à la page 114.
Ouverture d'une ressource	Ctrl+Maj+R (Windows) Cmd+Maj+R (Mac OS)	Affiche la boîte de dialogue Ouverture d'une ressource, dans laquelle vous pouvez rechercher rapidement une ressource, puis l'ouvrir dans l'éditeur.
Structure rapide	Ctrl+O (Windows et Mac OS)	Affiche la vue Structure en mode rapide dans l'éditeur. Pour plus d'informations, voir « Utilisation de la vue Structure rapide de l'éditeur » à la page 113.
Ajouter/Supprimer la mise en commentaire	Ctrl+/ (Windows) Cmd+/ (Mac OS)	Ajoute/supprime les commentaires ActionScript dans le code ActionScript. Voir « Ajout de commentaires et de blocs de commentaire » à la page 117.
Ajouter/Supprimer la mise en commentaire de blocs	Shift+Ctrl+C Shift+Cmd+C	Ajoute/supprime les blocs de commentaire MXML. Voir « Ajout de commentaires et de blocs de commentaire » à la page 117.
Activer/Désactiver la réduction	Ctrl+Diviser_Pavé numérique	Active/désactive la réduction des blocs de code. Voir « Configuration, réduction et développement de blocs de code » à la page 111.

Personnalisation des modèles de fichier

Flash Builder vous permet de personnaliser les informations par défaut contenues dans les nouveaux fichiers MXML, ActionScript et CSS. Il peut s'agir par exemple des variables définissant l'auteur et la date, des variables pour les balises et les attributs d'ouverture et de fermeture ainsi que des variables pour différentes déclarations ActionScript, pour différents préfixes d'espace de noms et pour presque tout contenu que vous souhaitez inclure dans un modèle de fichier. Les modèles de fichier sont utiles notamment pour spécifier des commentaires d'introduction et des informations de copyright.

Le contenu d'un nouveau fichier est spécifié dans un modèle de fichier accessible à partir de Préférences > Flash Builder > Modèles de fichier. Des modèles sont disponibles pour les types de fichier suivants.

ActionScript	Fichier ActionScript Classe ActionScript Interface ActionScript
MXML	Application Web MXML Application de bureau MXML Composant MXML Module MXML Habillage MXML Rendu d'élément pour composants Spark Rendu d'élément pour composants MX Rendu d'élément pour composant MX DataGrid Rendu d'élément pour composant AdvancedDataGrid Rendu d'élément pour composant MX Tree
FlexUnit	Classe TestCase FlexUnit Classe TestSuite FlexUnit Classe TestCase FlexUnit4 Classe TestSuite FlexUnit4
CSS	Fichier CSS

Une fois modifié, un modèle peut être exporté afin de pouvoir être partagé avec les autres membres de votre équipe.

Modification d'un modèle de fichier

- 1 Sélectionnez Préférences > Flash Builder > Modèles de fichier.
- 2 Développez les types de fichier et sélectionnez le modèle que vous souhaitez modifier.
- 3 Cliquez sur Editer et modifiez le modèle.

Vous pouvez saisir les données directement dans l'éditeur Modèle ou sélectionner Variables afin d'insérer des données prédéfinies dans le modèle.

- 4 Cliquez sur OK pour enregistrer les modifications.
Les modifications sont appliquées aux nouveaux fichiers.

Exportation et importation de modèles de fichier

- 1 Sélectionnez Préférences > Flash Builder > Modèles de fichier.
- 2 Développez les types de fichier et sélectionnez un modèle.
- 3 Cliquez sur Exporter pour exporter le modèle vers un système de fichiers ou Importer pour importer un modèle précédemment exporté.

Les modèles sont exportés dans des fichiers XML.

Restauration des valeurs par défaut

Remarque : le bouton *Restaurer les valeurs par défaut* rétablit les valeurs par défaut de tous les modèles de fichier. Vous ne pouvez pas restaurer les valeurs par défaut d'un seul modèle.

- ❖ Pour restaurer les modèles par défaut, ouvrez la fenêtre Préférences > Flash Builder > Modèles de fichier et sélectionnez Restaurer les valeurs par défaut.

Variables des modèles

Variables des modèles pour tous les types de fichiers

Variable	Description	Exemple
\$(date)	Date actuelle	15 février 2009
\$(year)	Année actuelle	2009
\$(time)	Heure actuelle	15h15
\$(file_name)	Nom du fichier créé	HelloWorld.mxml
\$(project_name)	Nom du projet Flex ou ActionScript	Hello_World_Project
\$(utilisateur)	Nom d'utilisateur de l'auteur	jdoe
\$\$	Symbole du dollar	\$
\$(dollar)		

Variables des modèles pour les fichiers MXML

Variable	Description	Exemple
\$(application) \$(component) \$(module)	Spécifient les noms des balises de l'application, du composant ou du module MXML. Pour une application Web, \$(application) s'étend à « Application ». Pour une application de bureau, \$(application) s'étend à « WindowedApplication ». \$(component) s'étend à « Component ». \$(module) s'étend à « Module ». Ces balises permettent généralement de positionner les balises d'ouverture et de fermeture d'un fichier.	Les balises suivantes : <pre><\${application} \${xmlns}\${wizard_attributes}\${min_size }> \${wizard_tags} </\${application}></pre> s'étendent à : <pre><s:Application xmlns:fx="http://ns.adobe.com/mxml/2009" xmlns:s="library://ns.adobe.com/flex/spark" xmlns:mx="library://ns.adobe.com/flex/halo" minWidth="1024" minHeight="768"> <s:layout> <s:BasicLayout/> </s:layout> </s:Application></pre>
\$(xml_tag)	Version XML	<?xml version="1.0" encoding="utf-8"?>
\$(xmlns)	Correspond à la définition d'espace de noms en fonction du type de SDK Flex du projet et du préfixe d'espace de noms défini dans les Préférences.	Pour un projet SDK Flex 4 : xmlns="http://ns.adobe.com/mxml/2009"
\$(min_size)	Taille minimale d'une application Web MXML.	minWidth="1024" minHeight="768"

Variable	Description	Exemple
<code>\${ns_prefix}</code>	Préfixe d'espace de noms pour le SDK Flex du projet. Vous ne pouvez pas modifier les valeurs par défaut de cette variable.	Pour Flex 3 : <code>mx</code> : Pour Flex 4 : <code>fx</code> :
<code>\${wizard_attributes}</code>	Spécifie la position des attributs définis par l'assistant Nouveau <i>Fichier</i> .	Pour une nouvelle application Web : <code>\${application} \${xmlns}\${wizard_attributes}></code> s'étendent à : <code><Application xmlns="http://ns.adobe.com/mxml/2009" layout="vertical"></code>
<code>\${wizard_tags}</code>	Spécifie la propriété de présentation des conteneurs définis par l'assistant Nouveau <i>Fichier</i> .	Pour une nouvelle application basée sur le SDK Flex 4 : <code><s:layout></code> <code><s:BasicLayout/></code> <code></s:layout></code>

Variables des modèles pour les fichiers ActionScript

Variable	Description	Exemple
`\${package_declaration}`	Génère la déclaration du package.	Pour un fichier du package com/samples, génère : package com.samples
`\${import_declaration}`	Génère les déclarations d'importation requises pour une nouvelle classe ou interface ActionScript.	Pour une sous-classe de TextBox, génère : import flex.graphics.TextBox;
`\${interface_declaration}`	Génère la déclaration d'interface pour une nouvelle interface ActionScript.	Pour une nouvelle interface qui étend l'interface IButton, génère : public interface IMyButton extends IButton
`\${class_declaration}`	Génère la déclaration de classe pour une nouvelle classe ActionScript.	Pour une nouvelle sous-classe de CheckBox, génère : public class MyCheckBox extends CheckBox
`\${class_body}`	Génère toutes les instructions requises pour une nouvelle classe.	Pour une nouvelle sous-classe de Button implémentant l'interface IBorder, génère ce qui suit pour le corps de la classe : <pre>public function MyButton() { super(); } public function get borderMetrics():EdgeMetrics { return null; }</pre>
`\${interface_name}` `\${class_name}` `\${package_name}`	Spécifie le nom de l'interface, de la classe ou du package. Cette variable est utilisée généralement lors de la création de commentaires.	Par exemple, la spécification de modèle : <pre>/* * \${class_name} implements. . . */</pre> génère le code : <pre>/* * MyButton implements. . . */</pre>

Variables de modèles pour fichiers CSS

Variable	Description	Exemple
`\${css_namespaces}`	Définit les espaces de noms pour les sélecteurs de style Spark et Halo.	Valeurs par défaut pour Flex 3 : "" (Dans Flex 3, les déclarations d'espace de noms ne sont pas requises dans les fichiers CSS). Valeurs par défaut pour Flex 4 : <pre>@namespace s "library://ns.adobe.com/flex/spark"; @namespace mx "library://ns.adobe.com/flex/halo";</pre>

Exemple de fichier modèle

Le code suivant illustre un exemple de modèle de fichier de composant MXML, suivi par un nouveau fichier de composant MXML généré à partir du modèle.

Exemple de modèle pour un fichier de composant MXML

```

${xml_tag}
<!--
* ADOBE SYSTEMS Confidential
*
* Copyright ${year}. All rights reserved.
*
* ${user}
* ${project_name}
* Created ${date}
*
-->
<${component} ${xmlns}${wizard_attributes}>
    ${wizard_tags}

    <${ns_prefix}Script>
    <![CDATA[

        ]]>
    </${ns_prefix}Script>
</${component}>

```

Nouveau fichier de composant MXML généré à partir du modèle d'exemple

```

<?xml version="1.0" encoding="utf-8"?>
<!--
* ADOBE SYSTEMS Confidential
*
* Copyright 2009. All rights reserved.
*
* jdoe
* FileTemplates
* Created Jul 13, 2009
*
-->

<s:Group xmlns:fx="http://ns.adobe.com/mxml/2009"
xmlns:s="library://ns.adobe.com/flex/spark"
xmlns:mx="library://ns.adobe.com/flex/halo" width="400" height="300">
    <s:layout>
        <s:BasicLayout/>
    </s:layout>

    <fx:Script>
        <![CDATA[

            ]]>
    </fx:Script>
</s:Group>

```

Chapitre 5 : Création de composants MXML personnalisés

Généralement, une application d'Adobe® Flex® se compose d'un fichier d'application MXML (c'est-à-dire un fichier contenant une balise parent `<s:Application>`), d'un ou de plusieurs composants Flex standard ainsi que d'un ou de plusieurs composants personnalisés définis dans des fichiers de composants (SWC) MXML, ActionScript ou Flash. En scindant l'application en blocs de dimensions maniables, vous pouvez rédiger et tester chaque composant individuellement. Dans un souci d'efficacité, les composants peuvent être réutilisés, dans la même application ou dans une application différente.

Vous pouvez utiliser Adobe Flash® Builder™ pour créer des composants MXML et ActionScript personnalisés, puis les intégrer à vos applications. Pour plus d'informations sur la création de composants ActionScript, voir « [Création d'une classe ActionScript](#) » à la page 67.

Vous pouvez également créer un composant MXML directement à partir de code. Pour plus d'informations, voir Simple MXML Components.

A propos des composants personnalisés

Les raisons à l'origine de la personnalisation des composants peuvent être multiples. Vous voulez par exemple simplement ajouter certaines fonctions à un composant existant ou créer un composant réutilisable, tel que par exemple une zone de recherche ou l'affichage d'un élément dans une grille de données. Peut-être souhaitez-vous rédiger un type de composant complètement nouveau, n'existant pas dans la structure Flex.

Si le composant est créé principalement à partir de composants existants, l'utilisation de MXML en facilitera la définition. Toutefois, s'il s'agit d'un type de composant entièrement nouveau, il sera préférable de le définir en tant que composant ActionScript. Pour plus d'informations, voir « [Création d'une classe ActionScript](#) » à la page 67.

Création de composants MXML avec Flash Builder

Flash Builder permet de créer des composants MXML personnalisés.

- 1 Cliquez sur Fichier > Nouveau > Composant MXML.

La boîte de dialogue pour la création d'un nouveau composant s'affiche.

- 2 Spécifiez le dossier parent pour le fichier du composant personnalisé.

Enregistrez le fichier dans un dossier situé dans le répertoire du projet actuel ou dans le chemin source du projet actuel. Le composant s'affichera ainsi dans la vue Composants.

- 3 Spécifiez le nom de fichier du composant.

Il définit le nom du composant. Par exemple, le nom du composant dont le fichier est nommé `LoginBox.mxml` est `LoginBox`.

- 4 Dans le champ Basé sur, sélectionnez le composant de base de votre composant personnalisé.

Les composants personnalisés sont généralement dérivés de composants existants. Les composants de présentation personnalisés se basent généralement sur des composants Container.

Pour Flex 4, Flash Builder suggère **spark.components.Group** comme composant de base.

Sélectionnez Parcourir pour ouvrir la boîte de dialogue Ouvrir un type, puis choisissez un composant.

Modifiez ou effacez le composant suggéré afin d'élargir la sélection dans la boîte de dialogue Ouvrir un type. Par exemple, spécifiez **spark.components** avant de cliquer sur Parcourir.

Vous pouvez filtrer la sélection dans la boîte de dialogue Ouvrir un type en fonction de vos besoins. Pour plus d'informations sur l'utilisation de la boîte de dialogue Ouvrir un type, voir « [Recherche et affichage des classes](#) » à la page 114.

- 5 (Facultatif) Si vous basez le composant sur un conteneur, vous avez la possibilité de définir la largeur et la hauteur du composant.

Vous pouvez saisir des valeurs fixes ou des pourcentages. Vous pouvez également vider les deux champs. Lorsque vous créez une occurrence du composant, vous pouvez remplacer la largeur et la hauteur du composant dans l'occurrence.

Si vous saisissez des pourcentages ou ne spécifiez aucune valeur, vous avez la possibilité de visualiser un aperçu du composant à différentes tailles. Pour ce faire, utilisez la liste déroulante située dans la barre d'outils de la vue Création de l'éditeur MXML. Pour plus d'informations, voir « [Création de composants visuels](#) » à la page 135.

- 6 Cliquez sur Terminer.

Flash Builder enregistre le fichier dans le dossier parent et l'ouvre dans l'éditeur.

Si vous avez enregistré le fichier dans le projet actif ou dans le chemin source correspondant, Flash Builder affiche également le composant dans la vue Composants pour permettre son insertion rapide dans les applications. Pour plus d'informations, voir « [Ajout de composants en mode Création MXML](#) » à la page 192.

Remarque : la vue Composants ne répertorie que les composants personnalisés visibles (c'est-à-dire ceux ayant hérité de la classe `UIComponent`). Pour plus d'informations, voir le Guide de référence du langage Adobe Flex.

- 7 Créez un composant personnalisé.

Pour plus d'informations, voir Simple MXML Components.

Création de composants visuels

Vous pouvez présenter un composant personnalisé visuellement dans l'éditeur MXML à l'instar d'un fichier d'application MXML. Tous les outils du mode Création sont disponibles. Vous pouvez par exemple ajouter des contrôles enfant à partir de la vue Composants et définir les propriétés dans la vue Propriétés. Pour plus d'informations, voir « [Utilisation visuelle des composants](#) » à la page 196.

La taille du composant personnalisé dans la vue Création dépend des règles suivantes :

- Si le composant possède une hauteur et une largeur fixes, Flash Builder adapte automatiquement la zone de création à ces dimensions.
- Si le composant ne possède pas de largeur et de hauteur ou si ces dernières ont une valeur de 100%, vous pouvez sélectionner les dimensions de la zone de création dans la liste déroulante de la barre d'outils de l'éditeur.

Sélectionnez différentes dimensions afin d'obtenir différents aperçus du composant. Les dimensions par défaut de la zone de création sont de 400x300 pixels pour les conteneurs. Pour les contrôles, la zone de création s'ajuste au contenu.

- En présence de valeurs autres que 100% pour la hauteur et la largeur, le composant est représenté sous forme de pourcentage de la dimension sélectionnée dans le menu de la zone de création.

Définition de commentaires ASDoc pour les composants personnalisés

Vous pouvez documenter les composants personnalisés en ajoutant des commentaires ASDoc au code qui les implémente. Les commentaires ASDoc sont ensuite disponibles dans l'assistant de contenu des éditeurs MXML et ActionScript. Pour plus d'informations, voir « [Utilisation de l'assistant de contenu](#) » à la page 108.

Ajoutez des commentaires ASDoc aux fichiers source ActionScript pour fournir la documentation de référence de l'API. Vous pouvez également ajouter des commentaires ASDoc à des éléments MXML du document. Voir ASDoc pour en savoir plus sur la création de commentaires ASDoc pour vos fichiers source.

Modification et distribution des composants MXML personnalisés

Flash Builder représente tous les composants personnalisés visibles (c'est-à-dire tous ceux ayant hérité de UICComponent) dans la vue Création de l'éditeur MXML. Vous pouvez double-cliquer sur un composant personnalisé pour en ouvrir le fichier et le modifier.

- 1 Ouvrez un fichier MXML utilisant un composant personnalisé.
- 2 En mode Création, double-cliquez sur un composant personnalisé.

Le fichier du composant s'ouvre dans l'éditeur.



Vous pouvez également sélectionner Ouvrir le composant personnalisé dans le menu contextuel.

- 3 Modifiez le composant personnalisé.

Distribution des composants personnalisés

Distribuez les composants personnalisés en créant des projets de bibliothèque. Pour plus d'informations, voir « [Projets de bibliothèque](#) » à la page 69.

Chapitre 6 : Développement d'applications AIR avec Flash Builder

Adobe® Flash® Builder™ comporte les outils permettant la création de projets Adobe® AIR®, l'utilisation des composants Flex AIR ainsi que le débogage et l'assemblage d'applications Adobe AIR. Le processus de développement des applications AIR dans Flash Builder est analogue à celui de la plupart des applications Flex.

Création de projets AIR avec Flash Builder

Si vous ne l'avez pas déjà fait, téléchargez et installez l'exécution AIR.

- 1 Ouvrez Flash Builder.
- 2 Sélectionnez Fichier > Nouveau > Projet Flex.
- 3 Entrez le nom du projet.
- 4 Dans Flex, les applications AIR sont considérées comme un type d'application. Vous disposez de deux options : une application exécutée sur le Web dans Adobe® Flash® Player et une application de bureau exécutée dans Adobe AIR. Sélectionnez Application de bureau comme type d'application.
- 5 Spécifiez si nécessaire la technologie de serveur à utiliser avec l'application AIR. En l'absence de technologie de serveur, sélectionnez Aucun/Autre, puis cliquez sur Suivant.
- 6 Sélectionnez le dossier dans lequel vous souhaitez créer l'application. Le dossier affiché par défaut est bin-debug. Cliquez sur Suivant.
- 7 Modifiez si nécessaire les chemins source et d'accès à la bibliothèque, puis cliquez sur Terminer pour créer le projet AIR.

Conversion de projets Flex en projets Adobe AIR

Les projets Flex spécifient soit un type d'application Web (exécutée dans Adobe Flash Player), soit un type d'application de bureau (exécutée dans Adobe AIR). L'application d'un projet Flex peut être convertie du type Web au type de bureau. Voir « [Conversion de projets Flex en projets Adobe AIR](#) » à la page 50.

Remarque : si vous importez un projet Catalyst à partir d'un fichier FXP, Flash Builder importe le projet avec le type d'application Web. Vous pouvez convertir le projet en type d'application de bureau (exécutée dans Adobe AIR).

Débogage des applications AIR avec Flash Builder

Flash Builder prend entièrement en charge le débogage des applications AIR. Pour plus d'informations sur les possibilités de débogage de Flash Builder, voir « [Débogage des applications](#) » à la page 141.

- 1 Dans Flash Builder, ouvrez un fichier source pour l'application (un fichier MXML, par exemple).
- 2 Cliquez sur le bouton Déboguer situé dans la barre d'outils principale .

Vous pouvez également sélectionner Exécuter > Déboguer.

L'application est lancée. Elle est exécutée dans l'application ADL (AIR Debugger Launcher, application de débogage du lanceur AIR). Le débogueur Flash Builder détecte les points d'arrêt et les erreurs d'exécution. L'application peut être déboguée comme toute autre application Flex.

Vous pouvez également déboguer une application à partir de la ligne de commande en utilisant l'outil de ligne de commande de l'application de débogage du lanceur AIR. Pour plus d'informations, voir *Utilisation de l'application de débogage du lanceur AIR (ADL)* dans la documentation AIR.

Création de packages d'application AIR avec Flash Builder

Une fois l'application achevée et prête à être distribuée (ou testée à partir du bureau), elle doit être mise en package dans un fichier AIR. La création de packages se compose des étapes suivantes :

- Sélection de l'application AIR à publier
- Affichage du code source (en présence de l'autorisation requise) et sélection des fichiers d'application à inclure
- Sélection d'un fichier AIR générique ou d'un programme d'installation natif pour l'installation de l'application
- Signature numérique de l'application AIR par un certificat commercial de signature de code ou par la création et l'application d'une signature autosignée
- Création (facultative) d'un fichier AIR intermédiaire pouvant être signé ultérieurement

Mise en package d'une application AIR

- 1 Ouvrez le projet et vérifiez que l'application ne présente aucune erreur de compilation ni d'exécution.
- 2 Sélectionnez **Projet > Exporter vers une version validée**.
- 3 Si plusieurs projets et applications sont ouverts dans Flash Builder, sélectionnez le projet AIR à assembler.
- 4 Vous pouvez également sélectionner **Activer l'affichage de la source** pour permettre aux utilisateurs de visionner le code source à l'exécution de l'application. Cliquez sur **Choisir des fichiers source** pour sélectionner les fichiers à exclure. Par défaut, tous les fichiers source sont sélectionnés. Pour plus d'informations sur la publication des fichiers source dans Flash Builder, voir l'Aide de Flash Builder.

Important : pour les projets pour serveurs, vous pouvez sélectionner le dossier des services à l'exportation des fichiers source. L'exportation de fichiers implémentant des services a des implications en termes de sécurité. Ces fichiers peuvent donner accès à votre base de données, noms d'utilisateurs et mots de passe inclus. Voir *Exportation des fichiers source avec version validée d'une application*.

- 5 Sélectionnez l'option **Exporter vers un fichier AIR** ou l'option **Exporter vers le programme d'installation natif**. Cliquez sur **Suivant**.

L'option **Exporter vers un fichier AIR** crée un fichier de programme d'installation générique permettant l'installation de l'application sur les plates-formes Windows ou Mac OS X.

L'option **Exporter vers le programme d'installation natif** crée un programme d'installation pour la plate-forme cible (Windows ou Mac OS X).

- 6 Vous avez également la possibilité de renommer le fichier AIR généré. Dès que vous êtes prêt à poursuivre, cliquez sur **Suivant** pour apposer une signature numérique sur l'application.

Signature numérique des applications AIR

Avant de poursuivre l'exportation vers une version validée, spécifiez comment apposer la signature numérique à l'application AIR. Vous disposez de plusieurs options. Vous pouvez signer l'application par un certificat commercial de signature du code, créer et utiliser un certificat numérique autosigné ou mettre l'application en package et la signer ultérieurement.

Les certificats numériques délivrés par des autorités de certification telles que VeriSign, Thawte, GlobalSign et ChosenSecurity valident votre identité d'éditeur auprès des utilisateurs et vérifient que le fichier d'installation n'a pas été modifié depuis sa signature. Les certificats numériques autosignés sont utilisés dans le même but, mais se distinguent par le fait qu'ils ne présentent aucune validation par un tiers.

Vous pouvez également assembler l'application AIR sans signature numérique en créant un fichier AIR intermédiaire (.airi). La validité de ce type de fichier est limitée. Il ne peut en effet pas être installé. Il est utilisé dans les procédures de test (effectuées par les développeurs) et peut être lancé avec l'outil de ligne de commande AIR ADT. AIR destine cette fonctionnalité aux environnements de développement au sein desquels la signature est prise en charge par un développeur ou une équipe spécifique. Cette manière de procéder augmente le degré de sécurité de gestion des certificats numériques.

Pour plus d'informations sur la signature des applications, voir *Signature numérique d'un fichier AIR* dans la documentation AIR.

Signature numérique d'une application AIR

Vous pouvez apposer une signature numérique sur une application AIR en sélectionnant le certificat numérique existant ou en créant un nouveau certificat autosigné.

- 1 Sélectionnez **Projet > Exporter vers une version validée**.
Sélectionnez le projet AIR que vous voulez exporter ainsi que le fichier vers lequel vous souhaitez exporter le projet. Cliquez sur **Suivant**.
- 2 Sélectionnez l'option **Exporter et signer un fichier AIR avec un certificat numérique**.
- 3 Si vous disposez déjà d'un certificat numérique, cliquez sur **Parcourir** pour y accéder, puis sélectionnez-le.
- 4 Pour créer un certificat numérique autosigné, cliquez sur **Créer**.
- 5 Spécifiez les informations requises et cliquez sur **OK**.
- 6 (Facultatif) Cliquez sur **Suivant**. Sélectionnez les fichiers de sortie qui devront être inclus dans le fichier AIRI exporté.
Par défaut, tous les fichiers sont inclus.
- 7 Cliquez sur **Terminer** pour créer le fichier AIR.

Création d'un fichier AIR intermédiaire

Vous pouvez créer un fichier AIRI intermédiaire et le signer ultérieurement. Utilisez cette option uniquement lorsque vous procédez à des tests.

- 1 Sélectionnez **Projet > Exporter vers une version validée**.
Sélectionnez le projet AIR que vous voulez exporter ainsi que le fichier vers lequel vous souhaitez exporter le projet. Cliquez sur **Suivant**.
- 2 Sélectionnez l'option **Exporter vers un fichier AIRI intermédiaire à signer ultérieurement**.
- 3 (Facultatif) Cliquez sur **Suivant**. Sélectionnez les fichiers de sortie qui devront être inclus dans le fichier AIRI exporté.

Par défaut, tous les fichiers sont inclus.

4 Cliquez sur Terminer.

Une fois généré, le fichier AIR intermédiaire peut être signé en utilisant l'outil ADT (AIR Developer Tool). Pour plus d'informations sur l'outil de ligne de commande ADT, voir *Signature d'un fichier intermédiaire AIR à l'aide de l'outil ADT* dans la documentation AIR.

Création d'un projet de bibliothèque AIR

Pour créer une bibliothèque de code AIR pour plusieurs projets AIR, générez un projet de bibliothèque AIR en faisant appel à l'assistant standard de projet de bibliothèque Flex.

- 1 Sélectionnez Fichier > Nouveau > Projet de bibliothèque Flex.
- 2 Entrez un nom de projet.
- 3 Activez l'option Inclure les bibliothèques Adobe AIR, puis cliquez sur Suivant.

Remarque : la version du SDK Flex sélectionnée doit prendre en charge AIR. Le SDK Flex 2.0.1 ne le prend pas en charge.

- 4 Modifiez si nécessaire le chemin de génération, puis cliquez sur Terminer. Pour plus d'informations sur la création de projets de bibliothèque, voir « A propos des projets de bibliothèque » dans l'Aide de Flash Builder.

Chapitre 7 : Débogage, test et surveillance des applications

Débogage des applications

Le débogage est similaire à l'exécution des applications. Il diffère toutefois par la possibilité de contrôler le code en cas d'arrêt de l'application, de surveiller des variables importantes ou de tester les corrections apportées au code. L'exécution et le débogage font appel à une configuration pour contrôler la manière dont les applications sont lancées. Le débogage d'une application consiste à exécuter la version déboguée du fichier d'application.

Pour une présentation générale des outils de débogage disponibles dans la perspective de débogage Flash, voir « [Perspective Débogage Flash](#) » à la page 16.

Dans certains cas, le programme vous invite à consulter le fichier journal Eclipse. Pour plus d'informations, voir « [Erreurs d'environnement Eclipse dans le fichier journal](#) » à la page 76.

Lancement d'une session de débogage

Pour lancer une session de débogage, exécutez la configuration de lancement de l'application dans la perspective Débogage Flash.

Débogage d'une application

- 1 Dans l'Explorateur de packages, sélectionnez le projet que vous souhaitez déboguer.
- 2 Cliquez sur le bouton Déboguer de la barre d'outils principale.

Remarque : le bouton Déboguer comporte deux éléments : le bouton d'action principal et la liste déroulante affichant les fichiers d'application du projet pouvant être exécutés ou débogués. Cliquez sur le bouton d'action principal pour exécuter le fichier d'application par défaut du projet. Vous pouvez également cliquer sur la liste déroulante et sélectionner l'un des fichiers d'application du projet que vous voulez déboguer. Vous pouvez en outre ouvrir la boîte de dialogue des configurations de lancement et créer ou éditer une configuration en cliquant sur la commande Déboguer.

Si le projet n'a pas encore été généré, Adobe® Flash® Builder™ le génère et l'exécute en mode de débogage.

- 3 L'application est affichée dans le navigateur Web par défaut ou dans la version autonome de Flash Player. Vous pouvez ensuite faire appel au débogueur de Flash Builder pour interagir avec elle.
- 4 La perspective Débogage Flash est activée dans le workbench à chaque point d'arrêt rencontré.

Lancement d'une session de débogage dans la configuration plug-in

L'action de la commande Déboguer de la version plug-in de Flash Builder est légèrement différente. Elle n'exécute pas le projet sélectionné, mais débogue la dernière configuration à avoir été lancée. Vous pouvez également sélectionner une configuration parmi la liste de celles qui ont été lancées en dernier.

Ajout et suppression de points d'arrêt

Les points d'arrêt permettent d'interrompre l'exécution de l'application pour analyser le code et utiliser les outils de débogage Flash Builder afin de corriger les erreurs. Les points d'arrêt sont ajoutés dans l'éditeur de code et gérés dans la vue Points d'arrêt lorsque vous déboguez les applications.

Ils sont placés au niveau des lignes de code exécutables. Le débogueur ne s'arrête qu'aux points d'arrêt situés sur les lignes présentant les éléments suivants :

- Balises MXML contenant un gestionnaire d'événement ActionScript, par exemple `<mx:Button click="dofunction()" ...>`
- Lignes ActionScript, telles que celles contenues dans une balise `<mx:Script>` ou dans un fichier ActionScript
- Toute ligne de code exécutable dans un fichier ActionScript

Vous pouvez définir des points d'arrêt en cours de rédaction du code ou pendant le débogage.

Positionnement d'un point d'arrêt dans l'éditeur de code

- 1 Ouvrez un fichier de projet contenant du code ActionScript.
- 2 Localisez la ligne de code sur laquelle vous souhaitez placer un point d'arrêt et double-cliquez sur la barre de repère pour en ajouter un.

La barre de repère longe le côté gauche de l'éditeur de code.

Un point d'arrêt est ajouté à la barre de repère ainsi qu'à la liste de la vue Points d'arrêt de la perspective Débogage Flash.

A chaque fois que le débogueur rencontre un point d'arrêt, l'application est interrompue et la perspective Débogage Flash est affichée. La ligne de code est signalée par un point d'arrêt et mise en surbrillance dans l'éditeur de code. Les commandes de débogage vous permettent d'interagir avec le code (voir « [Gestion de la session de débogage dans la vue Débogage](#) » à la page 144).

Suppression d'un point d'arrêt dans l'éditeur de code

- ❖ Dans la barre de repère, double-cliquez sur un point d'arrêt.

Le point d'arrêt est supprimé de la barre de repère ainsi que de la vue Points d'arrêt de la perspective Débogage Flash.

Les points d'arrêt sont gérés dans la vue Points d'arrêt. Vous pouvez supprimer un ou tous les points d'arrêt de la liste ou les désactiver, puis les réactiver ultérieurement (voir « [Gestion des points d'arrêt dans la vue Points d'arrêt](#) » à la page 143).

Définition de points d'arrêt conditionnels

Vous pouvez spécifier les conditions régissant l'interruption de l'exécution du débogueur par les points d'arrêt. La définition d'un point d'arrêt conditionnel consiste à spécifier une expression ActionScript qui sera évaluée au cours de la session de débogage. Les points d'arrêt conditionnels stoppent l'exécution lorsque l'une des conditions suivantes est remplie :

- L'expression prend la valeur « true ».
- La valeur de l'expression change.
- Le nombre d'occurrences spécifié a été atteint.

Définition d'un point d'arrêt conditionnel

1 Dans le menu contextuel d'un point d'arrêt, cliquez sur Propriétés du point d'arrêt.

2 Dans la boîte de dialogue qui s'ouvre, spécifiez les paramètres suivants :

- Activé

Activez ou désactivez le point d'arrêt.

- Nombre d'occurrences

Sélectionnez l'option Nombre d'occurrences afin de définir un compteur pour le point d'arrêt. Saisissez un nombre dans le champ Nombre d'occurrences.

Si vous activez les deux options, Nombre d'occurrences et Activer la condition, le nombre d'occurrences représente le nombre de fois que la condition spécifiée est remplie (elle prend la valeur « true » ou elle change).

Si vous n'activez que l'option Nombre d'occurrences, le nombre d'occurrences représente le nombre de fois que le point d'arrêt est atteint.

- Activer la condition

Sélectionnez l'option Activer la condition et saisissez une expression ActionScript à évaluer. Voir « [Exemples d'expressions](#) » à la page 147.

Remarque : Flash Builder vérifie la syntaxe de l'expression et vous en signale les erreurs. La présence d'un opérateur d'affectation dans l'expression déclenche l'affichage d'un avertissement.

- Suspendre lorsque

Spécifiez à quel moment l'exécution doit être stoppée : lorsque la condition a la valeur « true » ou lorsque la valeur de la condition change.

Gestion des points d'arrêt dans la vue Points d'arrêt

La vue Points d'arrêt vous permet de gérer les points d'arrêt au cours d'une session de débogage. Vous pouvez les supprimer, les désactiver et les activer ou les ignorer.

Les commandes disponibles dans la barre d'outils de la vue Points d'arrêt sont les suivantes (de gauche à droite).



Bouton/commande	Description
Supprimer les points d'arrêt sélectionnés	Cette commande supprime les points d'arrêt sélectionnés.
Supprimer tous les points d'arrêt	Cette commande supprime tous les points d'arrêt.
Afficher les points d'arrêt pris en charge par la cible sélectionnée	Cette commande affiche les points d'arrêt s'appliquant à la cible de débogage sélectionnée.
Aller au fichier relatif au point d'arrêt	Cette commande ouvre (s'il n'est pas déjà ouvert) le fichier contenant le point d'arrêt dans l'éditeur de code et met en surbrillance la ligne de code au niveau de laquelle le point d'arrêt a été placé. Vous pouvez également double-cliquer sur le point d'arrêt pour l'afficher dans l'éditeur de code.
Ignorer tous les points d'arrêt	Cette commande permet d'ignorer tous les points d'arrêt.

Bouton/commande	Description
Développer tout	Cette commande développe tous les points d'arrêt.
Réduire tout	Cette commande réduit tous les points d'arrêt.
Lier à la vue Débogage	Cette commande crée une liaison à la vue Débogage.

Suppression de points d'arrêt de la vue Points d'arrêt

Vous pouvez supprimer un seul, plusieurs ou tous les points d'arrêt de la vue en utilisant les commandes de la barre d'outils.

- ❖ Sélectionnez un ou plusieurs points d'arrêt dans la liste, puis cliquez sur Supprimer les points d'arrêt sélectionnés.

Vous pouvez également supprimer tous les points d'arrêt en une seule opération.

Suppression de tous les points d'arrêt de la vue Points d'arrêt

- ❖ Dans la vue Points d'arrêt, cliquez sur Supprimer tous les points d'arrêt.

Gestion de la session de débogage dans la vue Débogage

La vue Débogage est le centre de contrôle de la perspective Débogage Flash. Elle permet de contrôler l'exécution d'une application, d'interrompre, de reprendre ou de terminer l'application ainsi que d'avancer d'un pas dans le code avec ou sans entrée.

Les commandes disponibles dans la barre d'outils de la vue Débogage sont les suivantes (de gauche à droite).



Bouton/commande	Description
Supprime tous les programmes terminés	Cette commande supprime toutes les sessions de débogage terminées.
Reprendre	Cette commande provoque la reprise de l'application interrompue.
Interrompre	Cette commande interrompt l'application afin que vous puissiez analyser le code, avancer d'un pas dans le code avec entrée et ainsi de suite.
Arrêter	Cette commande termine la session de débogage.
Déconnecter	Cette commande déconnecte le débogueur en cas de débogage à distance.
Avancer d'un pas avec entrée	Cette commande permet d'avancer dans la fonction appelée et de stopper à la première ligne de la fonction.
Avancer d'un pas sans entrée	Cette commande exécute la ligne actuelle de la fonction, puis s'arrête à la ligne suivante.
Exécuter jusqu'à l'instruction de retour	Cette commande provoque la poursuite de l'exécution jusqu'à ce que la fonction actuelle soit retournée à son appelant.
Passer au cadre	Cette commande n'est pas prise en charge dans Flash Builder.
Utiliser les filtres de progression	Cette commande n'est pas prise en charge dans Flash Builder.

Utilisation de la vue Console

La vue Console affiche la sortie des instructions trace du code ActionScript ainsi que les rétroactions du débogueur (états, avertissements, erreurs, etc.).

Les commandes disponibles dans la barre d'outils de la vue Console sont les suivantes (de gauche à droite).

Bouton/commande	Description
Arrêter	Cette commande termine la session de débogage.
Supprimer le lancement	Cette commande supprime toutes les sessions de débogage lancées.
Supprimer tous les programmes terminés	Cette commande supprime toutes les sessions de débogage terminées.
Effacer la console	Cette commande efface le contenu de la vue Console.
Arrêter le défilement	Cette commande permet d'éviter le défilement de la vue.
Afficher la console en cas de modification de la sortie standard	Cette commande affiche la console en cas d'écriture dans la sortie standard.
Afficher la console en cas de modification de l'erreur standard	Cette commande affiche la console en cas d'écriture dans l'erreur standard.
Fixer la console	Cette commande bloque l'actualisation du contenu de la console lorsqu'un autre processus est sélectionné.
Afficher la console sélectionnée	Cette commande affiche les consoles sélectionnées.
Ouvrir la console	Cette commande ouvre une nouvelle console et affiche une liste déroulante permettant de sélectionner d'autres consoles.

Gestion des variables dans la vue Variables

La vue Variables affiche les variables définies par le cadre de pile actuellement sélectionné (dans la vue Débogage). Les variables simples (nom et valeur) sont affichées sur une seule ligne. Les icônes affichées avec les variables fournissent des informations sur le type de la variable.

Les variables complexes peuvent être développées pour en afficher les membres. La vue Variables permet de contrôler les variables en les ajoutant à la vue Expressions. Elle permet également de modifier la valeur des variables au cours de la session de débogage. Vous pouvez également y définir des points de contrôle (voir « [Utilisation des points de contrôle](#) » à la page 149).

Tous les membres de super-classes sont groupés dans un nœud d'arborescence distinct. Par défaut, seuls les membres de la classe actuelle sont affichés. Cela permet de limiter le nombre de variables affichées dans la vue.

Les commandes disponibles dans la barre d'outils de la vue Variables sont les suivantes (de gauche à droite).



Commande	Description
Afficher les noms de type	Cette commande affiche les noms de types des variables.
Afficher la structure logique	Cette commande n'est pas prise en charge dans Flash Builder.
Réduire tout	Cette commande réduit la vue Variables.

Modification de la valeur d'une variable

- 1 Sélectionnez la variable que vous souhaitez modifier.
- 2 Cliquez avec le bouton droit de la souris (Windows) ou en maintenant la touche Ctrl enfoncée (Mac OS) pour afficher le menu contextuel et sélectionnez Modifier la valeur.
- 3 Saisissez la nouvelle valeur et cliquez sur OK.

La variable contient la nouvelle valeur.

Les variables modifiées sont affichées en rouge.

Recherche de variables

- ❖ Pour localiser une variable ou un membre de variable dans la vue Variables, sélectionnez la vue, puis commencez à saisir le nom de la variable recherchée. Vous pouvez utiliser le caractère générique (*) pour rechercher les mots figurant dans le nom d'une variable (« *couleur », par exemple).

Icônes représentant des variables dans la vue Variables

La vue Variables fait appel à des icônes et à des incrustations pour fournir des informations visuelles sur le type des variables affichées. Pour obtenir une liste des icônes avec leurs significations respectives, voir « [Icônes représentant les éléments de langage](#) » à la page 120.

Nom	Valeur
this	newButton (@baf1851)
[inherited]	
\$alpha	1
\$blendMode	"normal"
\$blendShader	<setter>
\$height	100 [0x64]
\$parent	spark.components.Button (@baf10a1)
\$scaleX	1
\$scaleY	1
\$transform	flash.geom.Transform (@ba94921)
\$visible	false
\$width	100 [0x64]
\$x	0
\$y	0
accessibilityDescription	""
accessibilityEnabled	true
accessibilityImplementation	mx.accessibility.ButtonAccImpl (@b9d7)
accessibilityName	""

Vue Variables

Utilisation de la vue Expressions

Accédez à la vue Expressions pour contrôler les variables sélectionnées dans la vue Variables et pour ajouter et évaluer des expressions de contrôle en cours de débogage des applications.

En cours de débogage, vous pouvez analyser et modifier la valeur des variables contrôlées. Vous pouvez également ajouter des expressions de contrôle. Il s'agit d'expressions de code évaluées à chaque interruption du débogage. Les expressions de contrôle sont utiles pour contrôler les variables susceptibles de dépasser l'étendue et de ne pas être visibles dans la vue lorsque vous avancez d'un pas avec entrée dans une fonction différente.

Les commandes disponibles dans la barre d'outils de la vue Expressions sont les suivantes (de gauche à droite) :

Commande	Description
Afficher les noms de type	Cette commande affiche les types d'objet des éléments de la vue Expressions.
Afficher la structure logique	Cette commande n'est pas prise en charge dans Flash Builder.
Réduire tout	Cette commande réduit toutes les expressions de la vue.
Supprimer les expressions sélectionnées	Cette commande supprime la variable ou l'expression de contrôle sélectionnée.
Supprimer toutes les expressions	Cette commande supprime toutes les variables et expressions de contrôle de la vue Expressions.

Faites glisser le curseur de la souris sur une expression ou une variable dans l'éditeur de source pour en afficher la valeur dans une info-bulle. Vous pouvez ajouter l'expression à la vue correspondante en cliquant avec le bouton droit de la souris, puis en sélectionnant la commande de contrôle dans le menu contextuel.

Exemples d'expressions

Le débogueur de Flash Builder prend en charge une large gamme d'expressions simples et complexes. Le tableau suivant répertorie des exemples d'expressions pouvant être évaluées en cours de session de débogage. Cette liste d'expressions prises en charge n'est pas exhaustive et est fournie uniquement à titre indicatif.

Exemples d'expressions prises en charge

Expression	Description
<code>myString.length</code>	Retourne la longueur d'une chaîne.
<code>myString.indexOf('@')</code>	Effectue le suivi de l'index du caractère « @ ».
<code>"constant string".charAt(0)</code>	Effectue le suivi du caractère à un emplacement spécifique de la chaîne. Les constantes de chaîne sont prises en charge.
<code>employees.employee.@nom</code>	« employees » est une variable XML. Ce type d'expression est utile pour déboguer les applications E4X.
<code>x == null</code>	Mots réservés représentant des valeurs dans les expressions.
<code>user1 === user2</code>	La plupart des opérateurs ActionScript sont pris en charge.
<code>MyClass.myStaticFunc()</code>	Fonctions interprétées comme une classe.
<code>this.myMemberFunc()</code>	Fonctions interprétées en utilisant le mot-clé <code>this</code> .
<code>String.fromCharCode(33)</code>	String est en fait une fonction, et non une classe, dont <code>String.fromCharCode</code> est un membre dynamique.
<code>myStaticFunc()</code>	Ne peut être évalué que si <code>myStaticFunc</code> est visible à partir de la chaîne d'étendue actuelle.
<code>myMemberFunc()</code>	Ne peut être évalué que si <code>myMemberFunc</code> est visible à partir de la chaîne d'étendue actuelle.
<code>Math.max(1,2,3)</code>	Les fonctions mathématiques sont prises en charge.
<code>mystring.search(/myregex/i)</code>	Les expressions régulières sont prises en charge.
<code>["my", "literal", "array"]</code>	Création de tableaux.
<code>new MyClass()</code>	Instanciation de classes.
<code>"string" + 3</code>	Traitement correct d'une chaîne plus un nombre entier.
<code>x >>> 2</code>	Les opérations de décalage logique sont prises en charge.
<code>3.5 + 2</code>	Exécution correcte des opérations arithmétiques.

Limitations de l'évaluation des expressions

Il existe quelques limitations à l'évaluation des expressions :

- Les espaces de noms ne sont pas pris en charge.
- Les objets insérés ne sont pas pris en charge.
- Le mot-clé `super` n'est pas pris en charge.
- Les noms de classe complets ne sont pas pris en charge.

Par exemple, vous ne pouvez pas évaluer `mx.controls.Button`.

Vous pouvez faire référence au nom de classe non qualifié. Par exemple, vous pouvez spécifier que `Button` fait référence à `mx.controls.Button`.

En présence d'un nom de classe ambigu (deux classes portant le même nom dans deux packages différents), il vous est impossible de contrôler quelle classe est évaluée. Vous pouvez cependant spécifier :

```
getDefinitionByName("mx.controls.Button")
```

- La plupart des expressions E4X peuvent être évaluées, mais les expressions de filtre E4X ne sont pas prises en charge.

Par exemple, vous ne pouvez pas évaluer `myxml. (@id=='3')`.

- Vous ne pouvez pas appeler des fonctions définies en tant que variables.

Utilisation des points de contrôle

Au cours du débogage d'une application, vous pouvez définir des points de contrôle pour des occurrences spécifiques de variables. Ils auront pour tâche d'interrompre l'exécution à tout changement de valeur de la variable contrôlée. Les points de contrôle étant définis pour l'occurrence spécifique d'une variable, ils ne peuvent pas être placés dans l'éditeur de code. Ils sont donc définis à partir de la vue Variables au cours de la session de débogage.

Pour la définition des points de contrôle, veillez aux points suivants :

- A la fin d'une session de débogage, tous les points de contrôle sont supprimés.
- Vous ne pouvez pas définir des points de contrôle sur les getter, mais vous pouvez en définir sur les champs des getter.

Par exemple, vous ne pouvez pas définir un point de contrôle sur `width`, mais vous pouvez en définir un sur `_width`.

- Vous ne pouvez pas définir des points de contrôle sur des variables locales, mais vous pouvez en définir sur les membres des variables locales, ainsi que l'illustre le fragment de code suivant :

```
public class MyClass
{
    // These are fields of a class, so you can set a watchpoint on
    // 'memberInt', and on 'memberButton', and on 'memberButton._width':
    private var memberInt:int = 0;
    private var memberButton:Button = new Button();

    public function myFunction():void {
        // You CANNOT set a watchpoint on 'i', because it is local:
        var i:int = 0;

        // You CANNOT set a watchpoint on 'someButton', because it is local;
        // but you CAN set a watchpoint on 'someButton._width':
        var someButton:Button = new Button();

        ...
    }
}
```

- L'exécution s'interrompt pour un point de contrôle lorsque la valeur d'origine d'une occurrence d'objet change. Ce comportement diffère de l'utilisation d'une expression dans un point d'arrêt conditionnel pour arrêter l'exécution lorsque la valeur d'une variable change.

Définition de points de contrôle

- ❖ Deux méthodes permettent de définir un point de contrôle au cours d'une session de débogage :
 - Dans la vue Variables, ouvrez le menu contextuel d'une variable et sélectionnez Ajouter/Supprimer un point de contrôle.
 - Dans le menu Exécuter de Flash Builder, sélectionnez Ajouter un point de contrôle.

Dans la boîte de dialogue d'ajout d'un point de contrôle, sélectionnez la variable que vous souhaitez contrôler.

La vue Variables affiche une icône représentant un crayon en regard de la variable pour laquelle un point de contrôle a été défini.

Remarque : si vous tentez de définir un point de contrôle pour un getter, une boîte de dialogue s'ouvre. Elle suggère une variable valide pour le point d'arrêt. Si vous supprimez la variable suggérée, la boîte de dialogue répertorie toutes les variables valides de l'objet.

Utilisation de la commande Exécuter jusqu'à la ligne

La commande Exécuter jusqu'à la ligne permet de quitter une boucle au cours d'une session de débogage.

En cours de débogage, vous constaterez peut-être que le code est exécuté en boucle à plusieurs reprises. Pour quitter cette boucle, faites appel à la commande Exécuter jusqu'à la ligne située dans le menu Exécuter.

Environnement de test FlexUnit

L'environnement de test FlexUnit permet de générer et de modifier des tests reproductibles pouvant être exécutés à partir de scripts ou directement dans Flash Builder. Flash Builder prend en charge les structures « open source » FlexUnit 4 et FlexUnit 1.

Procédez comme suit :

- Créez des scripts et des suites de tests unitaires.

Les assistants de Flash Builder vous guident tout au long de la création de scripts et de suites de tests en générant du code stub pour les tests.

- Exécutez les scripts et les suites de tests.

Vous pouvez exécuter des scripts et des suites de tests de différentes manières à partir de Flash Builder ou en dehors de l'environnement Flash Builder. Les résultats des tests sont affichés dans une application de test. Flash Builder ouvre la vue Résultat des tests unitaires FlexUnit permettant d'analyser l'exécution du test.

- A partir de la vue Résultat des tests unitaires FlexUnit, accédez au code source.

Dans le panneau Résultats des tests, double-cliquez sur un test pour en ouvrir l'implémentation.

Le panneau Détails des tests ayant échoué répertorie la source et le numéro de la ligne correspondant à l'échec. Si la source répertoriée se trouve dans l'espace de travail en cours d'utilisation, double-cliquez sur son nom pour accéder directement à l'échec.

Création de tests FlexUnit

Vous pouvez créer des classes de scripts de tests et des suites de scripts de tests unitaires FlexUnit pour les types de projets suivants :

- Projet Flex
- Projet ActionScript
- Projet de bibliothèque Flex
- Projet AIR

Lorsque vous créez une classe de scripts de tests, vous pouvez spécifier les options suivantes :

- Le dossier `src` d'un projet Flex pour la classe
- Un package pour la classe
- Les classes à tester
- Les méthodes à tester pour chaque classe spécifiée

Une suite de scripts de tests unitaires FlexUnit est une série de tests basés sur des classes de scripts déjà créées, des méthodes spécifiées dans ces classes et d'autres suites de scripts de tests.

Création d'une classe de scripts de tests unitaires FlexUnit

Lors de la création d'une classe de scripts de tests unitaires FlexUnit, Flash Builder génère un fichier ActionScript pour la classe de scripts de tests et le place dans un package de scripts de tests.

La procédure suivante part du principe que vous avez créé un projet Flash Builder, dans lequel vous souhaitez générer et exécuter des tests unitaires FlexUnit.

- 1 Sélectionnez le projet Flex, puis Nouveau > Classe de script de tests dans le menu contextuel.
Si le fichier du projet sélectionné correspond à une classe ActionScript, cette classe sera automatiquement sélectionnée pour le script de test FlexUnit dans l'assistant Nouvelle classe de script de tests.
- 2 Dans l'assistant de création d'une classe de script de tests, indiquez si vous souhaitez créer une classe de style FlexUnit 4 ou FlexUnit 1.
- 3 Attribuez-lui un nom.
- 4 (Facultatif) Spécifiez un dossier source et un package pour la classe de script de tests ou validez les paramètres par défaut.
Le dossier source par défaut est le dossier `src` du projet en cours. Le package par défaut est `flexUnitTests`, qui correspond au niveau supérieur de la structure de packages par défaut du projet.
- 5 (Facultatif) Activez la case Sélectionner la classe à tester et recherchez la classe voulue. Cliquez sur Suivant.
- 6 (Facultatif) Sélectionnez les méthodes que vous souhaitez tester.
- 7 Cliquez sur Terminer.
Codez le script de test créé en vous basant sur les stubs de code générés.

Création d'une suite de scripts de tests unitaires FlexUnit

Cette procédure part du principe que vous avez déjà créé des classes de script de tests.

- 1 Sélectionnez le projet Flex, puis créez une suite de script de tests à partir des menus contextuels en sélectionnant Nouveau > Classe de suite de tests.
- 2 Dans l'assistant de création d'une classe de suite de tests, indiquez si vous souhaitez créer une classe de style FlexUnit 4 ou FlexUnit 1.
- 3 Nommez la suite de tests.
- 4 Recherchez ensuite, dans les suites et les scripts de tests, les classes et les méthodes à inclure dans la suite de tests. Cliquez sur Terminer.

Personnalisation des classes de script et de suite de tests FlexUnit

Vous pouvez personnaliser les classes de suite et de script de tests FlexUnit par défaut créées par Flash Builder. Flash Builder fait appel à des modèles de fichiers pour créer les versions par défaut de ces fichiers.

Les modèles de fichiers pour les tests FlexUnit se trouvent dans la boîte de dialogue Préférences, sous Flash Builder > Modèles de fichier > FlexUnit. Les classes de suite et de script de tests FlexUnit 1 et FlexUnit 4 correspondent à des modèles distincts.

Pour plus d'informations sur la modification des modèles de fichiers par défaut, voir « [Personnalisation des modèles de fichier](#) » à la page 127.

***Remarque :** `FlexUnitCompilerApplication.mxml` et `FlexUnitApplication.mxml` dérivent du modèle pour applications MXML Web ou de bureau. Le modèle utilisé dépend de la configuration du projet Flex : pour une application Web (s'exécute dans Adobe® Flash® Player) ou une application de bureau (s'exécute dans Adobe AIR®).*

Voir aussi

[Référence de langage « open source » pour FlexUnit](#)

[Documentation « open source » pour FlexUnit](#)

Exécution de tests FlexUnit

Les tests unitaires FlexUnit peuvent s'exécuter à l'intérieur comme à l'extérieur de Flash Builder à l'aide des fichiers SWF générés pour le test unitaire FlexUnit. Dans les deux cas, les résultats des tests s'affichent dans la vue Résultat des tests unitaires FlexUnit.

Vous pouvez également configurer et enregistrer un test unitaire FlexUnit avant de l'exécuter.

Par défaut, les tests FlexUnit sont exécutés dans la perspective Débogage Flash. Vous pouvez les lancer à partir des perspectives de développement et de profilage Flash. Dans ce cas, Flash Builder basculera toutefois vers la perspective de débogage à l'exécution du test.

Vous pouvez modifier la perspective par défaut des tests FlexUnit. Ouvrez la fenêtre des préférences et accédez à Flash Builder > FlexUnit.

Application de compilation FlexUnit et application FlexUnit

Lorsque vous créez un script de test FlexUnit, Flash Builder crée l'application de compilation FlexUnit suivante et une application FlexUnit :

- `FlexUnitCompilerApplication.mxml`
- `FlexUnitApplication.mxml`

Flash Builder utilise ces applications lors de la compilation et de l'exécution des tests FlexUnit. Flash Builder place les applications dans le répertoire `src` du projet.

Cette application contient des références à tous les scripts et à toutes les suites de tests FlexUnit générés par Flash Builder. Ce dernier place tous les tests FlexUnit à l'intérieur des balises `<fx:Declarations>` de cette application. Vous ne serez généralement pas amené à éditer ou à modifier ce fichier directement.

Actualisez l'application de compilation FlexUnit dans les circonstances suivantes :

- Vous ajoutez manuellement un script de tests.
Si vous créez une classe de script de tests sans utiliser l'assistant de création de script de tests, actualisez `FlexUnitCompilerApplication.mxml`. Placez le nouveau script de tests dans le package avec les autres scripts.
- Vous renommez un script de tests.
- Vous supprimez un script de tests.

Actualisez `FlexUnitCompilerApplication.mxml` :

- 1 Si la vue Résultat des tests unitaires FlexUnit n'est pas ouverte, vous pouvez y accéder en sélectionnant Fenêtres > Autres vues > Résultat des tests unitaires FlexUnit. Cliquez sur OK.
- 2 Dans la vue Résultat des tests unitaires FlexUnit, cliquez sur le bouton Actualiser.

Exécution d'un test unitaire FlexUnit dans Flash Builder

Vous pouvez exécuter des tests FlexUnit pour un projet dans son ensemble ou pour des scripts de tests individuels.

Vous lancerez généralement l'exécution à partir du menu contextuel d'un projet ou d'un script de test individuel.

Vous pouvez toutefois également lancer les tests en accédant au menu Exécuter, en cliquant sur le bouton Exécuter ou en sélectionnant le bouton Exécuter des tests unitaires FlexUnit dans la vue Résultat des tests unitaires FlexUnit.

L'exécution à partir du menu Exécuter de Flash Builder entraîne l'ouverture d'une boîte de dialogue de configuration du test, dans laquelle vous pouvez sélectionner les classes de test et les méthodes à exécuter. Les scripts de tests pour les projets de bibliothèque ne peuvent pas être exécutés à partir du menu Exécuter de Flash Builder.

Flash Builder fournit les raccourcis clavier suivants pour faciliter le lancement des tests FlexUnit :

- `Alt+Maj+A, F`
Exécute tous les tests FlexUnit du projet.
- `Alt+Maj+E, F`
Exécute le test FlexUnit sélectionné.

Exécutez les tests FlexUnit de la sélection actuelle dans l'éditeur (voir « [Configuration des tests FlexUnit](#) » à la page 154).

- 1 Sélectionnez un projet et exécutez les tests :
Dans le menu contextuel du projet, sélectionnez Exécuter des tests unitaires FlexUnit.
Dans le menu Exécuter ou dans la liste déroulante du bouton Exécuter, sélectionnez Exécuter > Tests FlexUnit.
- 2 (Menu Exécuter de Flash Builder) Dans la boîte de dialogue Exécution d'une configuration de test unitaire FlexUnit, sélectionnez les scripts de tests et les méthodes à exécuter dans le cadre des tests. Cliquez sur OK pour exécuter les tests.
- 3 Visionnez les résultats des tests.
Flash Builder génère un fichier SWF dans le dossier bin-debug du projet.
Une application s'ouvre pour afficher des informations sur les tests et en indiquer la fin.
Le panneau Résultat des tests unitaires FlexUnit s'ouvre et affiche les résultats des tests. Voir « [Affichage des résultats de l'exécution d'un test FlexUnit](#) » à la page 154.

Exécution de tests FlexUnit individuels :

- 1 Dans l'Explorateur de projets, accédez au package `flexUnitTest` :
Dans le menu contextuel du fichier de test FlexUnit, sélectionnez Exécuter des tests unitaires FlexUnit.
- 2 Visionnez les résultats des tests.
Flash Builder génère un fichier SWF dans le dossier bin-debug du projet.
Une application s'ouvre pour afficher des informations sur les tests et en indiquer la fin.

Le panneau Résultat des tests unitaires FlexUnit s'ouvre et affiche les résultats des tests. Voir « [Affichage des résultats de l'exécution d'un test FlexUnit](#) » à la page 154.

Exécution d'un test unitaire FlexUnit en dehors de l'environnement Flash Builder

Cette procédure part du principe que vous avez déjà exécuté un test unitaire FlexUnit dans Flash Builder et que Flash Builder est en cours d'exécution.

- 1 Copiez le fichier SWF généré pour un test du dossier bin-debug du projet vers un dossier situé en dehors de votre environnement de développement.

Vous pouvez copier le fichier SWF généré automatiquement ou un fichier SWF provenant d'un test unitaire FlexUnit que vous avez précédemment enregistré et configuré.

- 2 Exécutez la copie du fichier SWF.

Flash Player apparaît pour afficher des informations sur le test et indiquer la fin du test.

La vue Résultat des tests unitaires FlexUnit s'ouvre dans Flash Builder et affiche les résultats du test.

Configuration des tests FlexUnit

- 1 Ouvrez la boîte de dialogue Exécution d'une configuration de test unitaire FlexUnit.

Vous pouvez y accéder à partir du menu Exécuter ou à partir de la vue Résultat des tests unitaires FlexUnit :

- Sélectionnez un projet. Dans le menu Flash Builder, sélectionnez Exécuter > Exécuter > Tests FlexUnit.
- Dans la vue Résultat des tests unitaires FlexUnit, cliquez sur le bouton Exécuter des tests unitaires FlexUnit.

Si cette vue n'est pas ouverte, vous pouvez y accéder en sélectionnant Fenêtre > Autres vues > Flash Builder > Résultat des tests unitaires FlexUnit.

- 2 Dans la boîte de dialogue Configuration de test, sélectionnez les scripts et les méthodes à enregistrer comme configuration de test.

Remarque : la boîte de dialogue Configuration de test n'est pas disponible si vous exécutez un test à partir du menu contextuel de l'Explorateur de packages.

- 3 (Facultatif) Sélectionnez Charger pour importer une configuration précédemment sauvegardée dans un fichier XML.
- 4 Cliquez sur Enregistrer.

Flash Builder enregistre un fichier XML et un fichier MXML dans le dossier `.FlexUnitSettings` du projet.

Vous pouvez utiliser le fichier XML dans les scripts de génération pour exécuter le test.

Vous pouvez générer un fichier SWF à partir du fichier MXML. Ce fichier SWF peut servir à effectuer des tests en dehors de l'environnement Flash Builder. Pour générer le fichier SWF, le fichier MXML est généralement copié dans le dossier `src` du projet.

Affichage des résultats de l'exécution d'un test FlexUnit

Le panneau de la vue Résultat des tests unitaires FlexUnit affiche les résultats d'un test unitaire FlexUnit, en détaillant les échecs. Vous pouvez parcourir les résultats, filtrer l'affichage, écrire les résultats dans un fichier et charger les résultats à partir d'un fichier.

Vous pouvez également relancer les tests, annuler un test en cours d'exécution et effacer les résultats de la vue.

Si cette vue n'est pas ouverte, vous pouvez y accéder en sélectionnant Fenêtre > Autres vues > Flash Builder > Résultat des tests unitaires FlexUnit.

Panneau Résultats des tests

Ce panneau répertorie tous les tests de la série, en indiquant si le test a échoué ou réussi.

Double-cliquez sur un test de la liste pour y accéder dans l'éditeur ActionScript.

Panneau Détails des tests ayant échoué

Sélectionnez un test dans le panneau Résultats des tests pour afficher les détails de l'échec.

Chaque détail indique le fichier source et la méthode, y compris le numéro de la ligne correspondant à l'échec.

Si le fichier source appartient à l'espace de travail, double-cliquez dans la liste pour accéder à l'échec dans l'éditeur ActionScript.

Menu de la vue Résultat des tests unitaires FlexUnit

Le menu de la vue Résultat des tests unitaires FlexUnit donne accès aux fonctions suivantes.

- Filtrage des résultats affichés
 - Masquage du panneau Détails des tests ayant échoué.
 - Limitation de l'affichage aux tests ayant échoué
- Défilement des tests affichés dans le panneau Résultat des tests
- Annulation d'un test en cours d'exécution
- Enregistrement des résultats ou de la configuration d'un test
- Chargement des résultats précédemment enregistrés dans un fichier
- Suppression des résultats du panneau
- Réexécution du test actuel. Vous pouvez choisir :
 - d'exécuter tous les tests ;
 - d'exécuter uniquement les échecs ;
 - d'exécuter le test sélectionné.
- Actualisation de la configuration FlexUnit

Si vous avez modifié un test ou avez ajouté ou supprimé des tests, cliquez sur le bouton d'actualisation pour charger la nouvelle configuration FlexUnit.
- Configuration et exécution de tests FlexUnit

Utilisez le bouton Exécuter des tests unitaires FlexUnit pour configurer et exécuter des tests FlexUnit.

Surveillance des applications accédant aux services de données

Le Moniteur de réseau est un outil ingénieux permettant de surveiller et de déboguer les applications accédant aux services de données. Le Moniteur de réseau permet d'analyser les données transmises entre une application et un service de données. Il examine également les données XML, AMF et JSON envoyées avec les protocoles SOAP, AMF, HTTP et HTTPS.

Le Moniteur de réseau est disponible dans les vues Flash de développement et de débogage.

Activation de la surveillance du réseau

Le Moniteur de réseau peut être activé individuellement pour chaque projet Flex. L'état du moniteur (activé ou désactivé) s'applique à toutes les applications de ce projet. Il est impossible d'activer ou de désactiver le Moniteur de réseau pour des applications individuelles.

Par défaut, le Moniteur de réseau n'est pas activé. Activez le Moniteur de réseau en sélectionnant l'icône Activer le moniteur dans la barre d'outils de la vue du Moniteur.

Activation du Moniteur de réseau

Pour effectuer cette procédure, vous devez vous trouver dans la perspective de développement ou de débogage Flash.

- 1 Si la vue Moniteur de réseau n'est pas ouverte, sélectionnez Fenêtre dans le menu Flash Builder, puis Autres vues > Flash Builder > Moniteur de réseau.
- 2 Si le Moniteur de réseau n'est pas activé, cliquez sur l'icône Activer le moniteur dans la barre d'outils de la vue Moniteur de réseau.

Cette icône permet de l'activer et de le désactiver.

Surveillance des services distants

Pour surveiller l'application, activez le Moniteur de réseau et exécutez la version de développement ou de débogage de l'application.

En général, le Moniteur de réseau capture et stocke toutes les données d'événements jusqu'à ce que vous quittiez l'application ou effaciez les données de manière explicite. Les événements s'affichent dans l'ordre chronologique.

Démarrage d'une session de surveillance

- 1 Exécutez la version de développement ou de débogage de l'application qui accède aux services distants.
- 2 Pour chaque accès à un service distant, le Moniteur de réseau répertorie les éléments suivants :
 - Heure de la requête
 - Service
 - Demande d'un service
 - Heure de la réponse
 - Temps écoulé
- 3 Sélectionnez un en-tête de colonne pour trier les données renvoyées en fonction des valeurs de ladite colonne. Cliquez de nouveau sur la colonne pour inverser l'ordre des données.
- 4 Sélectionnez les onglets de requête et de paramètres situés au bas du moniteur pour afficher les détails de l'opération.

Les données envoyées dans la requête, ainsi que les autres informations concernant la requête, peuvent être visualisées à partir de ces onglets.
- 5 Sélectionnez les onglets de réponse et de résultats situés au bas du moniteur pour afficher les détails de la réponse.

Les données envoyées dans la requête, ainsi que les autres informations concernant la réponse, peuvent être visualisées à partir de ces onglets.
- 6 Double-cliquez sur une entrée pour accéder au code source de l'opération.

L'éditeur de source de Flash Builder s'ouvre ; la ligne correspondante du code source est mise en évidence.

Remarque : dans la plupart des cas, le Moniteur de réseau peut faire correspondre un événement au code source Flex. Certains événements toutefois sont déclenchés en dehors de la portée du Moniteur de réseau. Dans ce cas, le Moniteur ne trouvera pas le code source Flex.

- 7 Cliquez sur le bouton Enregistrer de la barre d'outils du Moniteur de réseau pour inscrire toutes les informations saisies dans un fichier XML.

Remarque : utilisez le fichier XML généré pour étudier les données hors ligne. Il est impossible d'importer des données de ce fichier vers le Moniteur de réseau.

- 8 Cliquez sur l'icône Effacer dans la barre d'outils du Moniteur de réseau pour en supprimer toutes les informations.

Interruption d'une session de surveillance

Vous pouvez interrompre, puis reprendre, la surveillance du réseau. L'interruption et la reprise d'une session s'appliquent à toutes les applications du projet Flex. Par exemple, il est impossible d'interrompre l'application d'un projet et de continuer la surveillance d'une autre.

- 1 Cliquez sur le bouton Interrompre dans la barre d'outils du Moniteur de réseau pour interrompre la surveillance d'une session.
- 2 Cliquez sur le bouton Reprendre pour continuer la surveillance de la session.

Arrêt d'une session de surveillance

Pour arrêter la surveillance d'une session, désactivez le Moniteur de réseau.

- 1 (Facultatif) Fermez le Moniteur de réseau.

Remarque : la fermeture du Moniteur de réseau ne suffit pas à mettre un terme à la session de surveillance. La surveillance reste active, même si le Moniteur de réseau est fermé.

- 2 Cliquez sur le bouton Activer le moniteur.

Cette icône permet de l'activer et de le désactiver.

Remarque : la désactivation du Moniteur de réseau s'applique à toutes les applications du projet Flex.

Prise en charge du protocole HTTPS

Le Moniteur de réseau prend en charge la surveillance des appels HTTPS dirigés vers un serveur certifié par une autorité de certification ou par un certificat autosigné.

Pour surveiller les appels passés par protocole HTTPS, modifiez la préférence par défaut du Moniteur de réseau en activant l'option Ignorer les vérifications de sécurité SSL. Ouvrez la boîte de dialogue Préférences et accédez à Flash Builder > Moniteur de réseau.

Affichage des données du Moniteur de réseau

Le volet de gauche du Moniteur de réseau fournit des informations sur la source des données. Il affiche les informations suivantes :

- L'URL source du service de données
- Le type de service affiché :
Par exemple RemoteService, HTTPService ou Webservice.
- Le temps de demande, le temps de réponse et le temps écoulé pour la demande de données
- Le nom de l'opération appelée à partir du service de données

Le Moniteur de réseau comporte deux onglets, l'un pour l'affichage des données de demande et l'autre pour l'affichage des données de réponse.

Les données des demandes et réponses peuvent être visionnées en structure arborescente, dans une vue brute ou en affichage hexadécimal. Passez d'un affichage à l'autre en sélectionnant l'icône correspondante.

- Arborescence
Affiche les données XML, JSON et AMF en structure arborescente. L'arborescence est la vue par défaut.
- Vue brute
Affiche les données en cours de transfert.
- Affichage hexadécimal
Affiche les données au format hexadécimal. Cet affichage est indiqué en particulier pour le débogage des données binaires envoyées par l'intermédiaire d'un réseau.

Par défaut, le Moniteur de réseau efface toutes les données enregistrées à chaque lancement d'une application. Vous pouvez cependant modifier ce comportement par défaut et conserver toutes les données de surveillance. Les données conservées sont celles issues de toutes les applications de tous les projets. Ouvrez la boîte de dialogue Préférences et accédez à Flash Builder > Moniteur de réseau. Désactivez l'option Effacer les entrées au démarrage.

Enregistrement des données du Moniteur de réseau

Vous pouvez enregistrer les données du Moniteur de réseau dans un fichier XML. Pour ce faire, cliquez sur le bouton Enregistrer de la vue Moniteur de réseau.

Surveillance de plusieurs applications

Vous pouvez surveiller plusieurs applications en même temps. Il existe deux scénarios possibles pour surveiller plusieurs applications :

- Surveillance de plusieurs applications d'un même projet
Vous pouvez uniquement disposer d'un Moniteur de réseau par projet Flex. Lors de la surveillance de plusieurs applications d'un même projet, les événements de toutes les applications s'affichent dans le moniteur en fonction de l'heure à laquelle ils surviennent.
Le moniteur ne permet pas de filtrer les événements selon une application spécifique.
- Surveillance de plusieurs applications dans différents projets
Vous pouvez ouvrir un Moniteur de réseau pour chaque projet Flex actif. Chaque Moniteur de réseau est indépendant : il affiche uniquement les événements de son projet spécifique.
L'interruption ou la désactivation d'un Moniteur de réseau dans un projet ne s'applique pas aux moniteurs d'autres projets.

Limites du Moniteur de réseau

La surveillance des données du réseau présente les deux restrictions suivantes :

- Le Moniteur de réseau ne prend pas en charge les applications créées dans le cadre de projets purement ActionScript ou de bibliothèque.
- De même, il ne prend pas en charge le protocole Real Time Messaging Protocol (RTMP). Vous ne pouvez pas, par exemple, surveiller de la vidéo en flux continu.

Chapitre 8 : Profilage des applications Flex

Le profileur d'Adobe® Flex® d'Adobe Flash® Builder™ vous permet d'identifier les goulets d'étranglement des performances et les fuites de mémoire des applications avec lesquelles vous interagissez. Il est disponible uniquement avec Flash Builder Premium.

A propos du profilage

Le profileur d'Adobe Flex détecte tout ralentissement des performances ainsi que les fuites de mémoire de vos applications. Le profileur est lancé à partir d'Adobe Flash Builder. Il enregistre les données concernant l'état de l'application avec laquelle vous interagissez, notamment le nombre d'objets, leur taille, le nombre d'appels de méthode et le temps passé dans ces appels.

Le profilage d'une application peut vous aider à comprendre les éléments suivants.

Fréquence des appels Vous pouvez dans certains cas vous rendre compte que certaines méthodes faisant appel à un grand nombre de ressources sont appelées plusieurs fois, alors qu'un seul appel est nécessaire. En identifiant les méthodes les plus fréquemment appelées, vous pouvez concentrer le temps que vous passez à ajuster les performances sur une zone plus réduite de l'application, où l'impact sur les performances est plus important.

Durée de la méthode Le profileur peut vous indiquer le temps passé dans une méthode particulière, ou, si la méthode est appelée plusieurs fois, le laps de temps moyen passé dans cette méthode au cours d'une session de profilage. Si vous trouvez que certaines méthodes entraînent des goulets d'étranglement des performances, vous pouvez tenter de les optimiser.

Piles d'appels En suivant la pile d'appels d'une méthode, vous pouvez voir l'intégralité du chemin que poursuit l'application lorsqu'elle appelle des méthodes successives. Cela peut vous amener à constater que certains appels de méthodes sont superflus.

Nombre d'occurrences (allocation d'objet) Si vous constatez que le même objet est créé plusieurs fois alors que seul un nombre spécifique d'occurrences est requis, vous pouvez envisager de mettre en œuvre un patron Singleton, pour autant que vous n'ayez vraiment besoin que d'un seul de ces objets. Vous pouvez également appliquer d'autres techniques visant à réduire l'allocation excessive d'objets. Si le nombre d'objets est important, mais nécessaire, vous pouvez optimiser l'objet même afin de réduire sa ressource Agrégat et l'utilisation qu'il fait de la mémoire.

Taille de l'objet Si vous remarquez que certains objets sont de taille disproportionnée, vous pouvez tenter de les optimiser pour réduire leur encombrement mémoire. Cette opération s'avère particulièrement utile si vous optimisez des objets créés de nombreuses fois dans l'application.

Récupération de place Lors de la comparaison d'instantanés de profilage, vous pouvez vous rendre compte que certains objets qui ne sont plus requis par l'application restent stockés dans la mémoire. Pour prévenir ces fuites de mémoire, vous pouvez ajouter une logique qui supprime toute référence restante à ces objets.

Ne considérez pas le profilage seulement comme une étape unique, distincte, du processus de développement d'une application. Il doit plutôt faire partie intégrante de chaque étape du développement d'une application. Vous devez, si possible, profiler une application précocement et fréquemment pendant son développement, de manière à identifier rapidement les éventuels problèmes. Le profilage étant un processus itératif, vous tirez un maximum de bénéfices en profilant le plus souvent possible.

A propos des types de profilage

Avant de faire appel au profileur, vous devez décider du type de profilage que vous allez effectuer : profilage des performances ou profilage de la mémoire.

Le *profilage des performances* est le processus de recherche des méthodes de l'application qui s'exécutent lentement et qui peuvent être améliorées. Une fois identifiées, ces zones à problèmes peuvent être optimisées pour accélérer l'exécution de l'application et pour que celle-ci réponde plus rapidement aux interactions de la part de l'utilisateur. Lors du profilage des performances, vous cherchez généralement deux choses : une méthode appelée une seule fois mais qui met plus de temps à s'exécuter que des méthodes similaires, ou une méthode qui ne met pas autant de temps à s'exécuter, mais qui est appelée de nombreuses fois. Les données de profilage des performances permettent d'identifier les méthodes que vous optimisez par la suite. Vous trouverez peut-être que le fait de réduire le nombre d'appels à une méthode est plus efficace que la restructuration du code au sein de la méthode.

Le *profilage de la mémoire* est l'examen de la quantité de mémoire qu'utilise chaque objet ou type d'objet de l'application. Les données de profilage de la mémoire s'utilisent de diverses façons : pour voir si des objets sont plus volumineux que nécessaire, pour voir si trop d'objets d'un même type existent et pour identifier des objets qui ne sont pas nettoyés dans la mémoire (fuites de mémoire). Grâce aux données de profilage de la mémoire, vous pouvez tenter de réduire la taille des objets, le nombre des objets créés ou permettre le nettoyage des objets dans la mémoire en supprimant les références à ces objets.

Le profilage de la mémoire peut ralentir les performances de l'application car il fait appel à une plus grande quantité de mémoire que le profilage des performances. N'effectuez le profilage de la mémoire que lorsque cela s'avère nécessaire.

Il convient souvent d'effectuer à la fois un profilage des performances et un profilage de la mémoire pour trouver la source des problèmes de performances. Le profilage des performances permet d'identifier les méthodes qui entraînent une allocation de mémoire excessive et un allongement des temps d'exécution. Vous pouvez ensuite recourir au profilage de la mémoire afin d'identifier les fuites de mémoire dans ces méthodes.

Lorsque vous savez le type de profilage que vous allez effectuer, vous pouvez démarrer le profileur.

Ressources supplémentaires

Le profileur seul n'améliore ni la taille, ni la vitesse, ni les performances perçues d'une application. Après avoir utilisé le profileur pour identifier les méthodes et classes qui posent problèmes, consultez les ressources suivantes dans la documentation Flex, qui vous aideront à améliorer l'application :

- Optimizing Flex Applications dans *Utilisation d'Adobe Flex 4*
- Improving Startup Performance dans *Utilisation d'Adobe Flex 4*

Fonctionnement du profileur Flex

Le profileur est un agent qui communique avec l'application s'exécutant dans Flash Player. Il se connecte à l'application par l'intermédiaire d'une connexion à un socket local. Vous devrez donc éventuellement désactiver votre logiciel antivirus si ce dernier entrave la communication avec les sockets.

Lorsque le profileur est en cours d'exécution, il prend un instantané des données à intervalles très courts et enregistre les actions d'Adobe Flash Player à ces moments. C'est ce qu'on appelle l'*échantillonnage*. Le profileur enregistre par exemple une méthode exécutée par l'application au moment de la prise de l'instantané. Si au moment de l'instantané suivant, l'application exécute toujours la même méthode, le profileur continue d'enregistrer l'heure. Lorsque le profileur effectue l'instantané suivant et que l'application est passée à l'opération suivante, le profileur peut consigner le temps qu'il a fallu à la méthode pour s'exécuter.

L'échantillonnage vous permet de profiler sans ralentir notablement l'application. L'intervalle écoulé s'appelle la *fréquence d'échantillonnage*, qui survient environ toutes les millisecondes pendant la période de profilage. Cela signifie que les opérations ne sont pas toutes enregistrées et que les instantanés ne sont pas tous précis à une fraction de milliseconde près. Cela vous permet toutefois d'avoir une idée plus précise des opérations qui prennent plus de temps que les autres.

En analysant les données d'échantillonnage, le profileur peut afficher toutes les opérations effectuées dans l'application et enregistrer leur délai d'exécution. Il consigne également l'utilisation de la mémoire et les traces de pile et affiche les données dans une série de vues, ou panneaux. Les appels de méthode sont classés par durée d'exécution et nombre d'appels, ainsi que par nombre d'objets créés dans la méthode.

Le profileur calcule également les valeurs cumulées des données. Par exemple, si vous consultez des statistiques liées aux méthodes, les données cumulées comprennent le temps et la mémoire alloués au cours de cette méthode, ainsi que le temps et la mémoire alloués au cours de toutes les méthodes appelées à partir de cette méthode. Vous pouvez effectuer une analyse plus poussée des appels de méthode suivants, jusqu'à trouver la source des problèmes de performances.

A propos des API de profilage

Le profileur utilise les API ActionScript définies dans le package `flash.sampler.*`. Ce package contient les classes `Sample`, `StackFrame`, `NewObjectSample` et `DeleteObjectSample`. Vous pouvez faire appel aux méthodes et aux classes contenues dans ce package pour écrire votre propre application de profileur ou pour inclure un sous-ensemble de fonctionnalités de profilage dans vos applications.

Outre les classes du package `flash.sampler.*`, le profileur utilise les méthodes contenues dans la classe `System`.

A propos des actions internes de Flash Player

Le profileur enregistre généralement des données sur les méthodes de la classe dont l'exécution était en cours au moment de l'instantané d'échantillonnage. Il arrive toutefois que le profileur enregistre également des actions internes de Flash Player. Ces actions, consignées entre crochets, sont notamment `[keyboardEvent]`, `[mark]` et `[sweep]`.

Par exemple, si `[keyboardEvent]` se trouve dans la liste des méthodes avec une valeur de 100, vous savez que le lecteur a effectué une action interne liée à cet événement au moins 100 fois au cours de la période d'interaction.

Le tableau suivant décrit les actions internes de Flash Player qui apparaissent dans les données de profilage.

Action	Description
<code>[generate]</code>	Le compilateur juste à temps (JIT) génère du code machine AS3.
<code>[mark]</code>	Flash Player marque les objets actifs pour la récupération de place.
<code>[newclass]</code>	Flash Player définit une classe. Cette opération a généralement lieu au démarrage. Le chargement d'une nouvelle classe peut avoir cependant lieu à tout moment.
<code>[pre-render]</code>	Flash Player se prépare à rendre les objets (notamment les calculs de géométrie et le parcours de la liste d'affichage qui se produisent avant l'affichage).

Action	Description
[reap]	Flash Player récupère les objets de décompte différé des références (DRC, deferred reference counting).
[render]	Flash Player rend les objets dans la liste d'affichage (pixel par pixel).
[sweep]	Flash Player récupère la mémoire des objets non marqués.
[verify]	Le compilateur JIT effectue la vérification du code binaire d'ActionScript 3.0.
[event_typeEvent]	Flash Player distribue l'événement spécifié.

Ces informations peuvent vous aider à identifier d'éventuels problèmes de performances. Par exemple, si vous voyez un grand nombre d'entrées pour [mark] et [sweep], vous pouvez partir du principe qu'un grand nombre d'objets sont en cours de création, puis sont marqués pour la récupération de place. En comparant ces nombres dans différents profils de performances, vous pouvez constater par vous-même si les modifications que vous apportez ont un effet.

Pour consulter les données correspondant à ces actions internes, affichez un profil de performances dans la vue Profil de performance ou un profil de mémoire dans la vue Suivi des allocations. Pour plus d'informations, voir « [Utilisation de la vue Profil de performance](#) » à la page 176 et « [Utilisation de la vue Suivi des allocations](#) » à la page 173.

Utilisation du profileur

Le profileur nécessite Flash Player version 9.0.124 ou une version ultérieure. Vous pouvez profiler des applications qui ont été compilées pour Flex 2, Flex 2.0.1 et Flex 3. Vous pouvez utiliser le profileur pour profiler des applications ActionScript 3.0 créées avec Flash Authoring, ainsi que des applications de bureau exécutées sur Adobe® AIR™.

Le profileur a besoin d'informations de débogage dans l'application que vous êtes en train de profiler. Lorsque vous compilez une application et que vous lancez le profileur, Flash Builder inclut par défaut les informations de débogage dans l'application. Vous pouvez inclure explicitement les informations de débogage dans n'importe quelle application en définissant l'option `debug` du compilateur sur `true`. Si vous exportez une application par le biais de l'option Exporter vers une version validée, l'application ne contient aucune information de débogage.

Lancement, arrêt et reprise du profileur

Vous pouvez profiler des applications que vous êtes en train de créer dans Flash Builder. Flash Builder inclut des informations de débogage lors de la compilation et de l'exécution d'une application lors d'une session de débogage. Vous pouvez également profiler dans Flash Builder des applications externes que vous n'êtes pas en train de développer, mais dont le fichier SWF est disponible avec une URL ou dans le système de fichiers. Le fichier SWF d'une application doit contenir des informations de débogage pour que vous puissiez la profiler. Pour plus d'informations, voir « [Profilage d'applications externes](#) » à la page 167.

Lancement du profilage d'une application Flex

- 1 Fermez toutes les occurrences du navigateur.
- 2 Ouvrez l'application dans Flash Builder.
- 3 Cliquez sur le bouton Profiler *nom_application* dans la barre d'outils principale. Flash Builder vous informe que vous devez fermer toutes les occurrences de vos navigateurs si ce n'est pas déjà fait.
- 4 Cliquez sur le bouton OK. Flash Builder compile l'application et la lance dans une fenêtre de navigateur distincte. Flash Builder ouvre également la boîte de dialogue Configurer le profileur.

- 5 Sélectionnez les options de la boîte de dialogue Configurer le profileur puis cliquez sur Reprendre. Pour profiler une application, vous devez sélectionner l'option Activer le profilage de la mémoire ou l'option Activer le profilage des performances. Vous pouvez sélectionner ces deux options lors du profilage d'une application.

Le tableau suivant présente les options disponibles.

Paramètre	Description
Connecté à partir de	Indique le serveur depuis lequel vous lancez l'application. Si l'application s'exécute sur le même ordinateur que le profileur, cette valeur est localhost. Vous ne pouvez pas modifier cette valeur. Vous pouvez toutefois profiler une application s'exécutant sur un ordinateur distinct. Voir « Profilage d'applications externes » à la page 167.
Application	Précise l'application que vous êtes sur le point de profiler. Vous ne pouvez pas modifier cette valeur.
Activer le profilage de la mémoire	<p>Demande au profileur de recueillir des données concernant la mémoire. Utilisez cette option pour détecter des fuites de mémoire ou un nombre excessif de créations d'objets.</p> <p>Si vous êtes en train d'effectuer le profilage des performances, vous pouvez désélectionner cette option.</p> <p>L'option Activer le profilage de la mémoire est sélectionnée par défaut.</p>
Surveiller les données de mémoire active	<p>Demande au profileur d'afficher des données concernant la mémoire dans la vue Objets actifs lors du profilage. Cela n'est pas obligatoire pour le profilage de la mémoire ni le profilage des performances. Vous pouvez sélectionner cette option uniquement si vous avez sélectionné Activer le profilage de la mémoire.</p> <p>L'option Surveiller les données de mémoire active est sélectionnée par défaut.</p>
Générer des traces de pile d'allocation d'objets	<p>Demande au profileur de capturer une trace de pile à chaque création d'objet. L'activation de cette option ralentit le profilage. Cette option n'est généralement sélectionnée qu'en cas de nécessité. Elle n'est disponible que si l'option Activer le profilage de la mémoire a été activée.</p> <p>L'option Générer des traces de pile d'allocation d'objets est désactivée par défaut.</p> <p>Elle doit être activée pour pouvoir afficher les informations liées au suivi des allocations dans la vue Références de l'objet ou Suivi des allocations.</p>
Activer le profilage des performances	<p>Demande au profileur de recueillir les données liées à la trace de la pile aux intervalles d'échantillonnage. Utilisez ces échantillons pour déterminer les processus auxquels la majeure partie du temps d'exécution de l'application est employée.</p> <p>Si vous êtes en train d'effectuer le profilage de la mémoire, vous pouvez désélectionner cette option.</p> <p>L'option Activer le profilage des performances est sélectionnée par défaut.</p>

Vous pouvez changer les valeurs par défaut de ces options en modifiant les préférences de profilage. Pour plus d'informations, voir « [Définition des préférences du profileur](#) » à la page 167.

- 6 Vous pouvez maintenant commencer à interagir avec l'application et à examiner les données du profileur.

Interruption et reprise du profilage d'une application Flex

Une fois que vous avez démarré le profileur, vous pouvez interrompre et redémarrer les applications dans la vue Profiler. Sélectionnez une application puis sélectionnez l'action à effectuer sur cette application. L'exemple suivant illustre la vue Profiler avec plusieurs applications. Une application est en cours d'exécution ; toutes les autres sont fermées.

Arrêt du profilage d'une application Flex

- 1 Sélectionnez l'application dans la vue Profiler.
- 2 Cliquez sur le bouton Arrêter pour mettre fin à la session de profilage. Cette opération ne ferme pas le navigateur ni ne termine le processus Flash Player. Vous devrez y procéder manuellement.
- 3 Pour retourner à la perspective Développement Flex, sélectionnez Développement Flex dans la liste déroulante de perspective. Vous pouvez également changer les perspectives en appuyant sur les touches Ctrl+F8 sous Windows.

A propos des boutons du profileur

Le tableau suivant présente les boutons de la barre d'outils du profileur.

Bouton	Nom	Description
	Reprendre	Reprend la session de profilage. Cette option est active uniquement si le nom d'une application est sélectionné et qu'elle est interrompue.
	Interrompre	Interrompt la session de profilage. Cette option est active uniquement si le nom d'une application est sélectionné et qu'elle est en cours d'exécution.
	Arrêter	Met fin à la session de profilage. Cette option est active uniquement si le nom d'une application est sélectionné et qu'elle n'a pas déjà été fermée.
	Lancer la récupération de place	Demande à Flash Player de récupérer de la place. Cette option est active uniquement si le nom d'une application est sélectionné et qu'elle est en cours d'exécution. Pour plus d'informations sur la récupération de place, voir « A propos de la récupération de place » à la page 182.
	Prendre un instantané de la mémoire	Enregistre l'utilisation que fait une application de la mémoire pour que vous puissiez la consulter ou la comparer à d'autres instantanés. Cette option est active uniquement si le nom d'une application est sélectionné, que celle-ci est en cours d'exécution et si vous sélectionnez Activer le profilage de la mémoire dans la boîte de dialogue de lancement. Le profileur ajoute de nouveaux instantanés de la mémoire en tant qu'enfants de l'application sélectionnée dans la vue Profiler. Pour ouvrir le nouvel instantané de mémoire dans la vue Instantané de la mémoire, double-cliquez sur l'entrée d'instantané de la mémoire. La récupération de place se fait implicitement avant l'enregistrement d'instantanés de la mémoire. En d'autres termes, un clic sur le bouton Prendre un instantané de la mémoire équivaut à un clic sur le bouton Lancer la récupération de place puis à un clic sur le bouton Prendre un instantané de la mémoire. Pour plus d'informations sur l'utilisation des instantanés de mémoire, voir « Utilisation de la vue Instantané de la mémoire » à la page 171.

Bouton	Nom	Description
	Rechercher les objets vagabonds	Compare deux instantanés de la mémoire dans la vue Objets vagabonds. Cette option est active uniquement si deux instantanés de mémoire sont sélectionnés et que l'option Activer le profilage de la mémoire est sélectionnée dans la boîte de dialogue de lancement. Pour plus d'informations sur la vue Objets vagabonds, voir « Utilisation de la vue Objets vagabonds » à la page 180.
	Afficher le suivi des allocations	Compare les méthodes entre deux instantanés de la mémoire dans la vue Suivi des allocations. Cette option est active uniquement si deux instantanés de mémoire sont sélectionnés et que l'option Activer le profilage de la mémoire est sélectionnée dans la boîte de dialogue de lancement. Pour plus d'informations sur la vue Suivi des allocations, voir « Utilisation de la vue Suivi des allocations » à la page 173.
	Réinitialiser les données de performance	Efface les données de profilage des performances. Cette option est active uniquement si le nom d'une application est sélectionné, que celle-ci est en cours d'exécution, et si vous sélectionnez Activer le profilage de la mémoire dans la boîte de dialogue de lancement. Généralement, vous cliquez sur ce bouton, vous interagissez avec l'application, puis vous cliquez sur le bouton Capturer le profil de performance pour obtenir un instantané des performances de l'application à partir du moment où vous réinitialisez les données. Pour plus d'informations sur la vue Profil de performance, voir « Utilisation de la vue Profil de performance » à la page 176.
	Capturer le profil de performance	Enregistre un nouvel instantané des performances en tant qu'enfant de l'application sélectionnée. Cette option est active uniquement si le nom d'une application est sélectionné, que celle-ci est en cours d'exécution, et si vous sélectionnez Activer le profilage de la mémoire dans la boîte de dialogue de lancement. Pour ouvrir la vue Profil de performance, double-cliquez sur l'entrée d'instantané des performances. Pour plus d'informations sur la vue Profil de performance, voir « Utilisation de la vue Profil de performance » à la page 176.
	Supprimer	Supprime de la mémoire les données de l'instantané sélectionné. Un clic sur ce bouton supprime aussi l'application de la vue du profil, si l'application a été fermée. Cette option est activée uniquement si un instantané de performances ou de mémoire est sélectionné.
Sans objet	Enregistrer	Enregistre les données de profilage sur le disque. L'option Enregistrer est également disponible à partir du menu de la vue Profiler. Cette option est active uniquement si le nom d'une application est sélectionné.

Certaines vues du profileur, telles que Statistiques de la méthode et Statistiques de l'objet, possèdent des boutons de navigation qui permettent de parcourir la pile ou de modifier la vue. Le tableau suivant présente les boutons de navigation dans ces vues du profileur.

Bouton	Nom	Description
	Précédent	Affiche toutes les méthodes que vous avez parcourues à partir de la première méthode sélectionnée, jusqu'à la méthode en cours d'affichage.
	Suivant	Affiche la méthode en cours d'affichage et les méthodes qui mènent à la méthode sélectionnée. Cet élément est actif une fois que vous êtes revenu en arrière.
	Valeur par défaut	Affiche la méthode sélectionnée.
	Ouvrir le fichier source	Ouvre un éditeur de source qui affiche le code source de la méthode sélectionnée.
	Filtres	Permet de contrôler les méthodes à inclure dans la table. Pour plus d'informations, voir « A propos des filtres du profileur » à la page 186.
	Afficher/Masquer les méthodes de durée nulle	Affiche ou masque les méthodes comportant un temps égal à 0,00 dans la colonne de temps moyen, car elles n'apparaissent dans aucun échantillon.

Enregistrement et chargement des données de profilage

Une fois le profileur exécuté, vous pouvez enregistrer les données afin de comparer un instantané de la session de profilage en cours avec un instantané pris après avoir apporté des modifications au code. Vous pouvez ainsi déterminer si vous avez identifié les causes responsables des problèmes et si les modifications apportées améliorent les performances et l'utilisation de la mémoire par l'application.

Lors de l'enregistrement des données de profilage, vous enregistrez toutes les données de l'application dans ce profil, à savoir tous les profils de performance, les instantanés de la mémoire et le suivi des allocations. Flash Builder enregistre ces informations dans un groupe de fichiers binaires à l'emplacement que vous spécifiez.

Enregistrement des données de profilage

- 1 Sélectionnez l'application dans la vue Profiler.
- 2 Ouvrez la liste déroulante dans la vue Profiler et sélectionnez Enregistrer. La boîte de dialogue Rechercher un dossier apparaît.
- 3 Choisissez un emplacement pour enregistrer les données de profil et cliquez sur OK. Vous devez créer un dossier pour chaque jeu de données de profilage à enregistrer. Dans le cas contraire, les nouvelles données remplacent les anciennes si vous sélectionnez le même dossier.

Récupération des données de profilage enregistrées

- 1 Sélectionnez la vue Données de profilage enregistrées.
- 2 Cliquez sur le bouton Ouvrir. La boîte de dialogue Rechercher un dossier apparaît.
- 3 Recherchez le dossier contenant les données de profilage de l'application, puis cliquez sur OK. Flash Builder affiche les données de profilage disponibles dans la vue Données de profilage enregistrées. Vous ne pouvez pas reprendre l'application dans cette vue, mais vous pouvez afficher les instantanés de la mémoire ou les autres données que vous avez enregistrées.

Vous ne pouvez pas supprimer les données d'application enregistrées depuis Flash Builder.

Suppression des données de profilage

- 1 Sélectionnez l'instantané de l'application dans la vue Profiler.
- 2 Cliquez sur le bouton Supprimer.

Définition des préférences du profileur

Vous pouvez définir certaines préférences du profileur de manière telle à ce que les paramètres s'appliquent à toutes les sessions de profilage. Ces paramètres vous permettent de définir le Flash Player/navigateur dans lequel vous profilez l'application, ainsi que les filtres par défaut et le numéro de port sur lequel l'application est disponible, si l'application profilée s'exécute sur un serveur.

Définition des préférences de Flash Builder pour plusieurs sessions de profilage

❖ Ouvrez la boîte de dialogue Préférences et sélectionnez Flash Builder > Profileur.

Sélectionnez les options du menu Profileur pour y accéder. Le tableau suivant répertorie les préférences que vous pouvez définir.

Option de menu	Description
Profileur	Sélectionnez la méthode de profilage par défaut. Vous pouvez activer ou désactiver le profilage de la mémoire et le profilage des performances.
Connexions	Définissez le numéro du port sur lequel Flash Builder écoute l'application profilée. Le numéro de port par défaut est 9999. Vous ne pouvez pas changer le port en 7935, car c'est le port utilisé par le débogueur.
Filtres d'exclusion	Définissez les packages par défaut qui sont exclus des vues du profileur. Pour plus d'informations sur l'utilisation des filtres, voir « A propos des filtres du profileur » à la page 186.
Filtres d'inclusion	Définissez les packages par défaut qui sont inclus dans les vues du profileur. Tous les autres packages sont exclus. Pour plus d'informations sur l'utilisation des filtres, voir « A propos des filtres du profileur » à la page 186.
Lecteur/Navigateur	Définissez l'emplacement du fichier exécutable de Flash Player et du navigateur que Flash Builder utilise pour exécuter votre application profilée.

Profilage d'applications externes

Outre les applications que vous développez dans Flash Builder, vous pouvez profiler des applications externes. Il peut s'agir de fichiers SWF situés dans n'importe quel emplacement accessible, notamment des applications situées sur un serveur Web distant ou d'une application se trouvant dans votre système de fichiers local.

Dans le cas d'un fichier SWF, vous pouvez spécifier une URL ou un emplacement de système de fichiers. Si vous indiquez une URL, Flash Builder lance le fichier SWF de l'application dans le navigateur par défaut. Le navigateur doit utiliser la version débogueur de Flash Player pour pouvoir profiler l'application correctement.

Si vous spécifiez un emplacement de système de fichiers pour le fichier SWF, Flash Builder ouvre l'application dans la version débogueur du Flash Player autonome. Vous devez en règle générale demander le fichier par le biais d'une URL. L'exécution des applications dans la version autonome de Flash Player peut produire des résultats inattendus, plus particulièrement si l'application utilise des services distants ou des appels réseau.

Profilage d'une application externe

- 1 Placez-vous dans la perspective Profil Flash.
- 2 Sélectionnez Profiler > Profiler une application externe. La boîte de dialogue Profiler une application externe apparaît.
- 3 Sélectionnez l'option (par défaut) Lancer l'application sélectionnée puis cliquez sur le bouton Nouveau. La boîte de dialogue Ajouter une application apparaît.

Vous pouvez aussi lancer l'application manuellement en sélectionnant l'option Lancer l'application manuellement hors Flash Builder.

- 4 Saisissez l'emplacement du fichier SWF puis cliquez sur OK ou sur le bouton Parcourir et localisez votre application dans votre système de fichiers.
- 5 Cliquez sur le bouton Lancer. Si vous avez indiqué une URL comme emplacement de l'application, Flash Builder lance l'application à partir du navigateur par défaut. Si vous avez spécifié un emplacement de système de fichiers pour l'application, Flash Builder ouvre l'application dans la version autonome de Flash Player.

Si vous avez spécifié un fichier SWF qui a été compilé sans informations de débogage, Flash Builder retourne une erreur. Recompilez l'application avec l'option `debug` du compilateur définie sur `true`, puis relancez-la.

A propos des vues du profileur

Le profileur Flex se compose de plusieurs vues (ou panneaux) présentant de différentes manières les données de profilage. Le tableau suivant présente chacune de ces vues.

Vue	Description
Profiler	Affiche les applications actuellement connectées, leur état et tous les instantanés de mémoire et de performances associés. Les sessions de profilage démarrent sans instantanés de performances ou de mémoire.
Données de profilage enregistrées	Affiche une liste d'instantanés enregistrés, classés par application. Vous pouvez charger des données de profilage enregistrées en double-cliquant sur l'instantané enregistré dans cette liste. Pour plus d'informations, voir « Enregistrement et chargement des données de profilage » à la page 166.
Objets actifs	Affiche des informations concernant les classes utilisées par l'application en cours. Cette vue montre les classes qui sont instanciées, le nombre créé, combien se trouvent dans le segment mémoire et la quantité de mémoire que les objets actifs occupent. Pour plus d'informations, voir « Affichage des informations dans la vue Objets actifs » à la page 169.
Instantané de la mémoire	Affiche l'état de l'application à un moment donné. Cette vue contraste avec la vue Objets actifs, mise à jour en continu. La vue Instantané de la mémoire affiche le nombre d'objets référencés et utilisés dans l'application et la quantité de mémoire que chaque type d'objets utilisait à ce moment-là. Deux instantanés de la mémoire pris à des moments différents sont comparés en vue de déterminer les fuites de mémoire qui existent entre les deux moments. Vous pouvez afficher la vue Instantané de la mémoire en cliquant sur le bouton Prendre un instantané de la mémoire puis en double-cliquant sur l'instantané de mémoire dans la vue Profiler. Pour plus d'informations, voir « Utilisation de la vue Instantané de la mémoire » à la page 171.
Objets vagabonds	Affiche les objets créés entre deux instantanés de mémoire qui existent toujours dans la mémoire ou qui n'ont pas été nettoyés. Vous pouvez double-cliquer sur le nom d'une classe dans la table pour ouvrir la vue Références de l'objet. Celle-ci vous permet d'examiner la relation entre les objets sélectionnés et d'autres objets. Vous pouvez afficher la vue Objets vagabonds en sélectionnant deux instantanés de mémoire et en cliquant sur le bouton Objets vagabonds. Pour plus d'informations, voir « Utilisation de la vue Objets vagabonds » à la page 180.

Vue	Description
Suivi des allocations	Affiche des statistiques liées aux méthodes lors de la comparaison de deux instantanés de mémoire. Vous pouvez afficher la vue Suivi des allocations en sélectionnant deux instantanés de mémoire et en cliquant sur le bouton Afficher le suivi des allocations. Pour plus d'informations, voir « Utilisation de la vue Suivi des allocations » à la page 173.
Références de l'objet	Affiche les objets et les objets qui les référencent. Vous pouvez afficher la vue Références de l'objet en double-cliquant sur le nom d'une classe dans la vue Instantané de la mémoire ou Objets vagabonds. Pour plus d'informations, voir « Utilisation de la vue Références de l'objet » à la page 172.
Statistiques de l'objet	Affiche des informations concernant l'appelant et l'appelé du groupe d'objets sélectionné. Vous pouvez afficher la vue Statistiques de l'objet en double-cliquant sur une entrée de la vue Suivi des allocations. Pour plus d'informations, voir « Utilisation de la vue Statistiques de l'objet » à la page 175.
Profil de performance	Affiche les performances des méthodes de l'application au cours d'une période donnée. Cliquez ensuite sur le nom d'une méthode dans la table pour ouvrir la vue Statistiques de la méthode, qui vous permet de détecter les goulets d'étranglement des performances. Vous pouvez afficher la vue Profil de performance en double-cliquant sur l'un des instantanés des performances dans la vue Profiler. Pour plus d'informations, voir « Utilisation de la vue Profil de performance » à la page 176.
Statistiques de la méthode	Affiche des statistiques concernant les performances du groupe de méthodes sélectionné. Vous pouvez afficher la vue Statistiques de la méthode en double-cliquant sur une ligne de la vue Profil de performance ou en y sélectionnant une méthode et en cliquant sur le bouton Ouvrir les statistiques de la méthode. Pour plus d'informations, voir « Identification des caractéristiques des performances des méthodes » à la page 178.
Utilisation de la mémoire	Affiche sous forme de graphique les pics d'utilisation de la mémoire et l'utilisation actuelle de la mémoire dans le temps. Pour plus d'informations, voir « Utilisation du graphique Utilisation de la mémoire » à la page 181.

Affichage des informations dans la vue Objets actifs

La vue Objets actifs affiche des informations concernant les classes utilisées par l'application en cours. Cette vue montre les classes qui sont instanciées, le nombre créé, combien se trouvent dans la mémoire et la quantité de mémoire que les objets actifs occupent.

Le profileur met à jour en continu les données dans la vue Objets actifs à mesure que vous profilez l'application. Vous n'avez pas besoin d'actualiser la vue ou de garder le focus dessus pour mettre à jour les données.

Pour pouvoir utiliser la vue Objets actifs, vous devez activer le profilage de la mémoire au démarrage du profileur. C'est la valeur par défaut. Si vous fermez la vue Objets actifs et que vous voulez la rouvrir, déployez la liste déroulante dans la vue Profiler et sélectionnez Surveiller les objets actifs.

L'exemple suivant illustre la vue Objets actifs.

Classe	Package (filtré)	Occurrences cumulées	Occurrences	Mémoire cumulée
Sample		1 (0.72%)	1 (4.76%)	1580 (24.69%)
newButton		1 (0.72%)	1 (4.76%)	1536 (24.0%)
_Sample_mx_managers_SystemManager		1 (0.72%)	1 (4.76%)	748 (11.69%)
_e5c5ce0db1832b2257012c2b335ba33b4720ce97e4...		1 (0.72%)	1 (4.76%)	420 (6.56%)
_d51b89cbb5e0c534017126393d44cf2d1f40b09ad58...		1 (0.72%)	1 (4.76%)	420 (6.56%)
_cee9af537b39795a3bd1729ebe7d63f3803fa353596...		1 (0.72%)	1 (4.76%)	420 (6.56%)
_c56f1cc5cebebc67eb0bf3658f7d3ce009ff7c644263...		1 (0.72%)	1 (4.76%)	420 (6.56%)
_01373b48f3fe562fb974ae981bec109407d2618a9a5...		1 (0.72%)	1 (4.76%)	420 (6.56%)
Vector, <*>	__AS3__vec	3 (2.16%)	3 (14.29%)	168 (2.63%)
ModuleManagerImpl	ModuleManager.as\$577	15 (10.79%)	1 (4.76%)	60 (0.94%)
LocaleID	LocaleSorter.as\$831	4 (2.88%)	0 (0.0%)	40 (0.63%)
fr_FR\$styles_properties		1 (0.72%)	1 (4.76%)	20 (0.31%)
fr_FR\$skins_properties		1 (0.72%)	1 (4.76%)	20 (0.31%)
fr_FR\$layout_properties		1 (0.72%)	1 (4.76%)	20 (0.31%)
fr_FR\$effects_properties		1 (0.72%)	1 (4.76%)	20 (0.31%)
fr_FR\$core_properties		1 (0.72%)	1 (4.76%)	20 (0.31%)
fr_FR\$controls_properties		1 (0.72%)	1 (4.76%)	20 (0.31%)
fr_FR\$components_properties		1 (0.72%)	1 (4.76%)	20 (0.31%)
fr_FR\$collections_properties		1 (0.72%)	1 (4.76%)	20 (0.31%)

Le tableau suivant répertorie les colonnes de la vue Objets actifs.

Colonne	Description
Classe	Classes possédant des occurrences dans l'application en cours d'exécution.
Package	Package dans lequel se trouve chaque classe. Si la classe n'est pas dans un package, la valeur de ce champ est le nom du fichier dans lequel se trouve la classe. Le nombre suivant le signe du dollar est un identifiant unique de cette classe. Si le champ Package est vide, la classe se trouve dans le package global ou le package non nommé.
Occurrences cumulées	Nombre total d'occurrences de chaque classe créées depuis le démarrage de l'application.
Occurrences	Nombre d'occurrences de chaque classe actuellement dans la mémoire. Cette valeur est toujours inférieure ou égale à la valeur contenue dans la colonne Occurrences cumulées.
Mémoire cumulée	Quantité totale de mémoire, exprimée en octets, que toutes les occurrences de chaque classe ont utilisée, y compris les classes n'étant plus dans la mémoire.
Mémoire	Quantité totale de mémoire, exprimée en octets, que toutes les occurrences de chaque classe ont utilisée, y compris les classes n'étant plus dans la mémoire. Cette valeur est toujours inférieure ou égale à la valeur contenue dans la colonne Mémoire cumulée.

Vous utilisez généralement les données contenues dans la vue Objets actifs pour voir la quantité de mémoire utilisée par les objets. A mesure que les objets sont nettoyés de la mémoire, l'utilisation du nombre d'occurrences et de la mémoire diminue, mais l'utilisation des occurrences cumulées et de la mémoire cumulée augmente. Cette vue indique également la façon dont la mémoire est utilisée lors de l'exécution de l'application.

Pour plus d'informations sur l'exécution et l'analyse des résultats du nettoyage de la mémoire, voir « [A propos de la récupération de place](#) » à la page 182.

Vous pouvez limiter les données affichées dans la vue Objets actifs grâce aux filtres du profileur. Pour plus d'informations, voir « [A propos des filtres du profileur](#) » à la page 186.

Utilisation de la vue Instantané de la mémoire

La vue Instantané de la mémoire propose des informations concernant l'utilisation des objets et de la mémoire de l'application à un moment donné. A la différence de la vue Objets actifs, les données de la vue Instantané de la mémoire ne sont pas mises à jour en continu.

Pour pouvoir utiliser la vue Instantané de la mémoire, vous devez activer le profilage de la mémoire au démarrage du profileur. C'est la valeur par défaut.

Création et affichage d'un instantané de la mémoire

- 1 Démarrez une session de profilage.
- 2 Interagissez avec votre application jusqu'à atteindre un point dans l'état de l'application où vous voulez prendre un instantané de la mémoire.
- 3 Sélectionnez l'application dans la vue Profiler.
- 4 Cliquez sur le bouton Prendre un instantané de la mémoire.
Le profileur crée un instantané de la mémoire et y appose un cachet de date. Il déclenche également implicitement la récupération de place avant l'enregistrement de l'instantané de mémoire.
- 5 Pour afficher les données dans l'instantané de mémoire, double-cliquez sur l'instantané de mémoire dans la vue Profiler.

L'exemple suivant illustre la vue Instantané de la mémoire.

Classe	Package (filtré)	Occurrences	Mémoire
Sample		1 (4.55%)	1580 (24.69%)
newButton		1 (4.55%)	1536 (24.0%)
_Sample_mx_managers_SystemManager		1 (4.55%)	748 (11.69%)
_e5c5ce0db1832b2257012c2b335ba33b4720ce97e4210218cb2b2b1f6d90dc53...		1 (4.55%)	420 (6.56%)
d51b89cbb5e0c534017126393d44cf2d1f40b09ad5825f2f9d2cd564c228ca6e...		1 (4.55%)	420 (6.56%)
_cee9af537b39795a3bd1729ebe7d63f3803fa353596b9c82a79ce2891eb0f825...		1 (4.55%)	420 (6.56%)
_c56f1cc5cebebc67eb0bf3658f7d3ce009ff7c644263a843832275a80e64c828_f...		1 (4.55%)	420 (6.56%)
_01373b48f3fe562fb974ae981bec109407d2618a9a5a43ae20dcde6d816d8748...		1 (4.55%)	420 (6.56%)
Vector.<*>	__A53__.vec	3 (13.64%)	168 (2.63%)
ModuleManagerImpl	ModuleManager.as\$577	1 (4.55%)	60 (0.94%)
LocaleID	LocaleSorter.as\$831	1 (4.55%)	40 (0.63%)

Le tableau suivant répertorie les colonnes de la vue Instantané de la mémoire.

Colonne	Description
Classe	Classes qui avaient des occurrences dans la mémoire au moment où vous avez enregistré l'instantané de la mémoire.
Package	Package dans lequel se trouve chaque classe. Si la classe n'est pas dans un package, la valeur de ce champ est le nom du fichier dans lequel se trouve la classe. Le nombre suivant le signe du dollar est un identifiant unique de cette classe. Si le champ Package est vide, la classe se trouve dans le package global ou le package non nommé.
Occurrences	Nombre d'occurrences de chaque classe dans la mémoire au moment où vous avez enregistré l'instantané de la mémoire.
Mémoire	Quantité de mémoire, exprimée en octets, que toutes les occurrences de chaque classe utilisaient au moment de la prise de l'instantané de mémoire.

L'instantané de mémoire est généralement utilisé comme point de départ pour déterminer sur quelles classes vous devez vous concentrer pour optimiser la mémoire ou trouver des fuites de mémoire. Pour ce faire, créez plusieurs instantanés de mémoire à différents moments, puis comparez les différences dans les vues Objets vagabonds ou Suivi des allocations.

Vous pouvez enregistrer des instantanés de la mémoire pour comparer l'état d'une application au cours d'une session de profilage différente. Pour plus d'informations, voir « [Enregistrement et chargement des données de profilage](#) » à la page 166.

Lorsque vous double-cliquez sur une ligne de la vue Instantané de la mémoire, le profileur affiche la vue Références de l'objet. Cette vue affiche les traces de pile correspondant aux occurrences de la classe en cours. Vous pouvez consulter les traces de pile pour les occurrences de la classe en cours dans la vue Références de l'objet. Pour plus d'informations sur la vue Références de l'objet, voir « [Utilisation de la vue Références de l'objet](#) » à la page 172.

Vous pouvez également spécifier quelles données afficher dans la vue Instantané de la mémoire grâce aux filtres du profileur. Pour plus d'informations, voir « [A propos des filtres du profileur](#) » à la page 186.

Utilisation de la vue Références de l'objet

La vue Références de l'objet affiche les traces de pile pour les classes qui ont été instanciées dans l'application.

Pour ouvrir la vue Références de l'objet, double-cliquez sur le nom d'une classe dans les vues Instantané de la mémoire ou Objets vagabonds. La vue Références de l'objet propose des informations concernant les occurrences de la classe sélectionnée.

La vue Références de l'objet affiche les données dans deux tables : Occurrences et Suivi des allocations.

La table Occurrences répertorie tous les objets contenant des références à l'objet actuel. Le nombre entre parenthèses après le nom de la classe est le nombre total de références à l'objet en cours. Vous ne pouvez pas voir le nombre de références futures d'un objet. Si aucun objet ne contient de références à l'objet spécifié, aucun objet n'est répertorié dans cette table. Cela ne devrait pas se produire, car l'objet en question doit être nettoyé de la mémoire s'il ne possède aucune référence.

L'exemple suivant illustre la table Occurrences de la vue Références de l'objet.

Occurrence	Propriété	ID	Racine de la RP
newButton(10 Chemins)		236906	
Chemin 1			
newButton		236906	
spark.components:Button	_skin	236732	
spark.components:Group	[child0]	236024	
spark.components:Group	[child0]	235951	
Object	target	236261	OUI
Chemin 2			
Chemin 3			

Le tableau suivant répertorie les colonnes de la table Occurrences.

Colonne	Description
Occurrence	Classe de l'objet contenant une référence à l'objet spécifié. Développez l'occurrence d'une classe pour afficher les chemins menant à l'objet. Le nombre de chemins peut être configuré en accédant à l'option Filtres de la vue Instantané de la mémoire.
Propriété	Propriété de l'objet contenant une référence à l'objet spécifié. Par exemple, si vous avez un objet <code>o</code> avec une propriété <code>i</code> et si vous attribuez cette propriété pour qu'elle pointe sur l'étiquette du bouton : <pre>o.i = myButton.label;</pre> Une référence est créée à l'étiquette <code>monBouton.label</code> à partir de la propriété <code>i</code> .
ID	Identifiant de référence de l'objet contenant la référence à l'objet sélectionné.
Racine de la RP	Indique si un objet possède une référence arrière à la racine de la RP (récupération de place). Développez un chemin menant à une occurrence d'un objet pour voir s'il existe une référence arrière à la racine de la RP.

La table Suivi des allocations affiche la trace de la pile de l'occurrence sélectionnée dans la table Occurrences. Lorsque vous sélectionnez une occurrence dans la table Occurrences, le profileur affiche la pile d'appels de cette occurrence dans la table Suivi des allocations.

L'exemple suivant illustre la table Suivi des allocations de la vue Références de l'objet.

Méthode	Emplacement	Ligne
spark.components.supportClasses:SkinableComponent:attachSkin()	SkinableComponent.as	598
spark.components.supportClasses:SkinableComponent:validateSkinChange()	SkinableComponent.as	404
spark.components.supportClasses:SkinableComponent:createChildren()	SkinableComponent.as	367
mx.core:UIComponent:initialize()	UIComponent.as	7250
mx.core:UIComponent:http://www.adobe.com/2006/flex/mx/internal:childAdded()	UIComponent.as	7142
mx.core:UIComponent:addChildAt()	UIComponent.as	6848
spark.components:Group:addDisplayObjectToDisplayList()	Group.as	1825
spark.components:Group:http://www.adobe.com/2006/flex/mx/internal:elementAdded()	Group.as	1416
spark.components:Group:setMXMLContent()	Group.as	512
spark.components:Group:set mxmlContent()	Group.as	452

Le tableau suivant répertorie les colonnes de la table Suivi des allocations.

Colonne	Description
Méthode	La méthode de niveau supérieur de cette table est celle qui a créé l'occurrence de la classe répertoriée dans la table Occurrences. Vous pouvez développer la méthode pour en afficher la trace de pile. Cela peut vous aider à déterminer l'endroit où la pile d'appels a commencé.
Emplacement	Fichier dans lequel la méthode est définie.
Ligne	Numéro de la ligne dans le fichier.

Vous pouvez afficher des données dans cette table uniquement si vous activez le suivi d'allocations au démarrage du profileur.

Vous pouvez ouvrir le code source de la classe sélectionnée en double-cliquant sur une classe dans cette table.

Utilisation de la vue Suivi des allocations

La vue Suivi des allocations indique les méthodes qui ont été appelées entre deux instantanés de mémoire et la quantité de mémoire consommée lors de ces appels de méthode. Pour ouvrir la vue Suivi des allocations, sélectionnez deux instantanés de mémoire puis cliquez sur le bouton Afficher le suivi des allocations. Pour plus d'informations sur l'enregistrement d'un instantané de la mémoire, voir « [Utilisation de la vue Instantané de la mémoire](#) » à la page 171.

Le résultat de la comparaison des instantanés de la mémoire est une liste de méthodes que Flash Player a exécutées entre les deux instantanés de mémoire. Pour chacune de ces méthodes, le profileur indique le nombre d'objets créés dans cette méthode.

Ces informations peuvent vous aider à optimiser les performances. Une fois que vous avez identifié des méthodes qui créent un nombre excessif d'objets, vous pouvez optimiser ces zones à problème.

Pour pouvoir utiliser la vue Suivi des allocations, vous devez activer le suivi des allocations au démarrage du profileur. Cette option est désactivée par défaut.

L'exemple suivant illustre la vue Suivi des allocations.

Méthode	Packag...	Occurrences cumul...	Occurrences propres	Mémoire cumulée	Mémoire propre
[newclass]		687 (17.36%)	509 (56.24%)	1988419 (29.57%)	1975667 (97.4...)
_Sample_Styles.init		202 (5.1%)	129 (14.25%)	74273 (1.1%)	29808 (1.47%)
LocaleRegistry.\$cinit		95 (2.4%)	95 (10.5%)	5260 (0.08%)	5260 (0.26%)
_Sample_mx_managers_System...		125 (3.16%)	3 (0.33%)	252689 (3.76%)	2888 (0.14%)
Sample._Sample_DataGrid1_i		67 (1.69%)	1 (0.11%)	67797 (1.01%)	2392 (0.12%)
global.flash.utils.getQualifiedClas...		23 (0.58%)	23 (2.54%)	1830 (0.03%)	1830 (0.09%)
global.flash.utils.getQualifiedSup...		16 (0.4%)	16 (1.77%)	1596 (0.02%)	1596 (0.08%)
Sample._Sample_Button1_i		24 (0.61%)	4 (0.44%)	11453 (0.17%)	1476 (0.07%)
Sample._Sample_Panel1_i		14 (0.35%)	3 (0.33%)	1730 (0.03%)	1320 (0.07%)
global.\$init.global\$init		723 (18.27%)	40 (4.42%)	1989443 (29.58%)	1136 (0.06%)
<anonymous>		119 (3.01%)	22 (2.43%)	297339 (4.42%)	752 (0.04%)
global.flash.net.getClassByAlias		7 (0.18%)	7 (0.77%)	670 (0.01%)	670 (0.03%)

Le tableau suivant répertorie les colonnes de la vue Suivi des allocations.

Colonne	Description
Méthode	Méthode appelée pendant l'intervalle de l'instantané. Cette colonne contient aussi la classe dont l'occurrence a appelé cette méthode.
Package	Package dans lequel se trouve chaque classe. Si la classe n'est pas dans un package, la valeur de ce champ est le nom du fichier dans lequel se trouve la classe. Le nombre suivant le signe du dollar est un identifiant unique de cette classe. Si le champ Package est vide, la classe se trouve dans le package global ou le package non nommé.
Occurrences cumulées	Nombre d'objets instanciés dans cette méthode et toutes les méthodes appelées à partir de cette méthode.
Occurrences propres	Nombre d'objets instanciés dans cette méthode, à l'exception des objets instanciés dans des appels de méthode suivants à partir de cette méthode.
Mémoire cumulée	Quantité de mémoire, exprimée en octets, utilisée par les objets instanciés dans cette méthode et toutes les méthodes appelées à partir de cette méthode.
Mémoire propre	Quantité de mémoire, exprimée en octets, utilisée par les objets instanciés dans cette méthode, à l'exception de la mémoire utilisée par les objets instanciés dans des appels de méthode suivants à partir de cette méthode.

Lors de l'enregistrement de méthodes au cours d'intervalles d'échantillonnage, le profileur consigne aussi les actions internes de Flash Player. Ces actions apparaissent dans la liste de méthodes entre crochets sous la forme [mouseEvent] ou [newclass] ou avec des noms similaires. Pour plus d'informations sur les actions internes de Flash Player, voir « [Fonctionnement du profileur Flex](#) » à la page 160.

Pour ouvrir la vue Statistiques de l'objet, double-cliquez sur une ligne de la table Suivi des allocations. Cette vue fournit des informations concernant les objets créés dans la méthode que vous avez sélectionnée. Elle permet aussi de consulter en détail les objets créés dans des méthodes appelées à partir de cette méthode. Pour plus d'informations, voir « [Utilisation de la vue Statistiques de l'objet](#) » à la page 175.

Vous pouvez limiter les données affichées dans la vue Suivi des allocations grâce aux filtres du profileur. Pour plus d'informations, voir « [A propos des filtres du profileur](#) » à la page 186.

Utilisation de la vue Statistiques de l'objet

La vue Statistiques de l'objet propose des statistiques concernant les performances du groupe d'objets sélectionné. Cette vue permet d'identifier quelles méthodes appellent un nombre disproportionné d'autres méthodes. Elle indique également la quantité de mémoire que les objets instanciés dans ces appels de méthode consomment. Vous pouvez faire appel à la vue Statistiques de l'objet pour identifier d'éventuelles fuites de mémoire et d'autres sources de problèmes de performances dans votre application.

Pour accéder à la vue Statistiques de l'objet, sélectionnez deux instantanés de mémoire dans la vue Profiler et affichez la comparaison dans la vue Suivi des allocations. Double-cliquez ensuite sur une ligne pour afficher les informations dans la vue Statistiques de l'objet.

La vue comporte trois sections :

- Un récapitulatif des statistiques de l'objet sélectionné, notamment le nombre d'occurrences et la quantité de mémoire utilisée.
- La table Occurrences propres : liste d'objets instanciés dans la méthode sélectionnée dans la vue Suivi des allocations. Elle ne répertorie pas les objets instanciés dans les appels de méthode suivants à partir de cette méthode. Le nombre d'objets dans cette liste correspond au nombre d'objets spécifié dans la colonne Occurrences propres de la vue Suivi des allocations.
- La table Occurrences de l'appelé : liste d'objets instanciés dans les méthodes appelées par la méthode sélectionnée dans la vue Suivi des allocations. Elle ne comprend pas les objets instanciés dans la méthode même. Le nombre d'objets dans cette liste correspond au nombre d'objets spécifié dans la colonne Occurrences cumulées de la vue Suivi des allocations.

L'exemple suivant illustre le récapitulatif de méthodes et les tables Occurrences propres et Occurrences de l'appelé de la vue Statistiques de l'objet.

Le tableau suivant répertorie les champs de la table Occurrences propres de la vue Statistiques de l'objet.

Colonne	Description
Classe	Classes qui ont été instanciées uniquement dans la méthode sélectionnée, à l'exception des classes instanciées dans des appels suivants à partir de cette méthode.
Package	Package dans lequel se trouve chaque classe. Si la classe n'est pas dans un package, la valeur de ce champ est le nom du fichier dans lequel se trouve la classe. Le nombre suivant le signe du dollar est un identifiant unique de cette classe. Si le champ Package est vide, la classe se trouve dans le package global ou le package non nommé.
Occurrences propres	Nombre d'occurrences de cette classe créées uniquement dans la méthode sélectionnée, à l'exception des occurrences créées dans des appels suivants à partir de cette méthode.
Mémoire propre	Quantité de mémoire, exprimée en octets, utilisée par les occurrences créées uniquement dans la méthode sélectionnée, à l'exception de la mémoire utilisée par les occurrences créées dans des appels suivants à partir de cette méthode.

L'exemple suivant illustre la table Occurrences de l'appelé de la vue Statistiques de l'objet.

Occurrences de l'appelé :			
Classe	Package	Occurrences cumulées	Mémoire cumulée
Class		6 (8.22%)	41361 (93.02%)
Dictionary	flash.util	28 (38.36%)	1152 (2.59%)
Array		20 (27.4%)	912 (2.05%)
String		11 (15.07%)	792 (1.78%)
Object		8 (10.96%)	248 (0.56%)

Le tableau suivant répertorie les champs de la table Occurrences de l'appelé de la vue Statistiques de l'objet.

Colonne	Description
Classe	Classes qui ont été instanciées dans la méthode sélectionnée, y compris les classes instanciées dans des appels suivants à partir de cette méthode.
Package	Package dans lequel se trouve chaque classe. Si la classe n'est pas dans un package, la valeur de ce champ est le nom du fichier dans lequel se trouve la classe. Le nombre suivant le signe du dollar est un identifiant unique de cette classe. Si le champ Package est vide, la classe se trouve dans le package global ou le package non nommé.
Occurrences cumulées	Nombre d'occurrences de cette classe créées dans la méthode sélectionnée et dans des appels suivants à partir de cette méthode.
Mémoire cumulée	Quantité de mémoire, exprimée en octets, utilisée par les occurrences créées dans la méthode sélectionnée et dans des appels suivants à partir de cette méthode.

Utilisation de la vue Profil de performance

La vue Profil de performance est la principale vue à utiliser lors du profilage des performances. Elle propose des statistiques telles que le nombre d'appels, la durée propre et la durée cumulée pour les méthodes appelées au cours d'un intervalle d'échantillonnage particulier. Ces données permettent d'identifier les goulets d'étranglement des performances.

Le processus de profilage des performances conserve une liste des méthodes et des informations concernant ces méthodes qui ont été appelées entre le moment où vous supprimez les données de performances et le moment où vous capturez les nouvelles données. Cet écart de temps est désigné par le terme de *période d'interaction*.

Pour pouvoir utiliser la vue Profil de performance, vous devez activer le profilage des performances au démarrage du profileur. C'est la valeur par défaut.

Génération d'un profil de performance

- 1 Activez le profilage des performances et démarrez une session de profilage.
- 2 Interagissez avec l'application jusqu'à atteindre le point où vous voulez commencer le profilage.
- 3 Cliquez sur le bouton Réinitialiser les données de performance.
- 4 Interagissez avec l'application et effectuez les actions à profiler.
- 5 Cliquez sur le bouton Capturer le profil de performance.

L'écart de temps entre le moment où vous cliquez sur Réinitialiser les données de performance et le moment où vous cliquez sur Capturer le profil de performance est la période d'interaction. Si vous ne cliquez pas sur Réinitialiser les données de performance, le profil de performance contiendra toutes les données capturées depuis le premier démarrage de l'application.

- 6 Double-cliquez sur le profil de performance dans la vue Profiler.

L'exemple suivant illustre la vue Profil de performance.

Méthode	Package...	Appels	Durée cumulée (ms)	Durée propre (...)	Durée cumulé...	Durée propre moye...
[abc-decode]		1 (0.0%)	265 (64.01%)	176 (42.51%)	265.0	176.0
[mark]		1 (0.0%)	64 (15.46%)	61 (14.73%)	64.0	61.0
Sample.initialize		1 (0.0%)	50 (12.08%)	0 (0.0%)	50.0	0.0
[verify]		1 (0.0%)	48 (11.59%)	43 (10.39%)	48.0	43.0
[reap]		1 (0.0%)	29 (7.0%)	29 (7.0%)	29.0	29.0
global\$init.global\$init		1 (0.0%)	27 (6.52%)	2 (0.48%)	27.0	2.0
[generate]		1 (0.0%)	26 (6.28%)	26 (6.28%)	26.0	26.0
[io]		1 (0.0%)	23 (5.56%)	7 (1.69%)	23.0	7.0
[execute-queued]		1 (0.0%)	21 (5.07%)	4 (0.97%)	21.0	4.0
[sweep]		1 (0.0%)	12 (2.9%)	12 (2.9%)	12.0	12.0
[enterFrameEvent]		1 (0.0%)	10 (2.42%)	5 (1.21%)	10.0	5.0
Sample.FlexInit.init		1 (0.0%)	9 (2.17%)	0 (0.0%)	9.0	0.0

Le tableau suivant répertorie les colonnes de la vue Profil de performance.

Colonne	Description
Méthode	Nom de la méthode et classe à laquelle appartient la méthode. Les actions internes exécutées par Flash Player s'affichent sous la forme d'entrées entre crochets : par exemple, [mark] et [sweep]. Vous ne pouvez pas changer le comportement de ces actions internes, mais vous pouvez utiliser les informations les concernant pour aider au profilage et à l'optimisation. Pour plus d'informations sur ces actions, voir « Fonctionnement du profileur Flex » à la page 160.
Package	Package dans lequel se trouve la classe. Si la classe n'est pas dans un package, la valeur de ce champ est le nom du fichier dans lequel se trouve la classe. Le nombre suivant le signe du dollar est un identifiant unique de cette classe. Si le champ Package est vide, la classe se trouve dans le package global ou le package non nommé.
Appels	Nombre de fois où la méthode a été appelée au cours de la période d'interaction. Si une méthode est appelée un nombre disproportionné de fois par rapport aux autres méthodes, vous pouvez tenter d'optimiser cette méthode en vue de réduire le temps d'exécution.
Durée cumulée	Laps de temps total, exprimé en millisecondes, qu'ont mis à s'exécuter tous les appels à cette méthode, et tous les appels à des méthodes suivantes, au cours de la période d'interaction.

Colonne	Description
Durée propre	Laps de temps, exprimé en millisecondes, qu'ont mis à s'exécuter tous les appels à cette méthode au cours de la période d'interaction.
Durée cumulée moyenne	Laps de temps moyen, exprimé en millisecondes, qu'ont mis à s'exécuter tous les appels à cette méthode, et tous les appels à des méthodes suivantes, au cours de la période d'interaction.
Durée propre moyenne	Laps de temps moyen, exprimé en millisecondes, qu'a mis à s'exécuter cette méthode au cours de la période d'interaction.

Si vous double-cliquez sur une méthode dans la vue Profil de performance, Flex affiche des informations concernant cette méthode dans la vue Statistiques de la méthode. Cela vous permet de consulter le détail de la pile d'appels d'une méthode donnée. Pour plus d'informations, voir « [Identification des caractéristiques des performances des méthodes](#) » à la page 178.

Vous pouvez limiter les données affichées dans la vue Profil de performance grâce aux filtres du profileur. Pour plus d'informations, voir « [A propos des filtres du profileur](#) » à la page 186.

Vous pouvez enregistrer les profils de performances pour une utilisation ultérieure. Pour plus d'informations, voir « [Enregistrement et chargement des données de profilage](#) » à la page 166.

Identification des caractéristiques des performances des méthodes

La vue Statistiques de la méthode présente les caractéristiques des performances de la méthode sélectionnée. Vous pouvez faire appel à la vue Statistiques de la méthode pour identifier des goulets d'étranglement dans l'application. En affichant, par exemple, les temps d'exécution d'une méthode, vous pouvez voir les méthodes qui prennent un temps disproportionné pour s'exécuter. Vous pouvez ensuite optimiser de manière sélective ces méthodes.

Pour plus d'informations, voir « [Utilisation de la vue Profil de performance](#) » à la page 176.

Affichage des informations concernant les méthodes dans la vue Statistiques de la méthode

- 1 Double-cliquez sur une ligne dans la vue Profil de performance ou sélectionnez une méthode dans cette même vue.
- 2 Cliquez sur le bouton Ouvrir les statistiques de la méthode.

La vue comporte trois sections :

- Un récapitulatif des performances de la méthode sélectionnée, notamment le nombre d'appels, la durée cumulée et la durée propre.
- La table Appelants : informations concernant les méthodes qui ont appelé la méthode sélectionnée. Dans certains cas, il est important de savoir si la méthode sélectionnée est trop appelée, la façon dont elle est utilisée et si elle est utilisée correctement.
- La table Appelés : informations concernant les méthodes appelées par la méthode sélectionnée.

L'exemple suivant illustre le récapitulatif de méthodes et les tables Appelants et Appelés de la vue Statistiques de la méthode.

Méthode	Package (filtré)	Durée cumulée (ms)	Durée propre (ms)
[abc-decode]		49 (76.56%)	0 (0.0%)
[verify]		5 (7.81%)	0 (0.0%)

Méthode	Package (filtré)	Durée cumulée (ms)	Durée propre (ms)
[sweep]		3 (4.69%)	3 (4.69%)

Le tableau suivant répertorie les champs de la table Appelants de la vue Statistiques de la méthode.

Colonne	Description
Méthode	Méthodes qui ont appelé la méthode apparaissant dans le récapitulatif de la partie supérieure de cette vue. Si cette liste est vide, la méthode cible n'a été appelée par aucune méthode qui n'est pas filtrée.
Package	Package dans lequel se trouve chaque classe. Si la classe n'est pas dans un package, la valeur de ce champ est le nom du fichier dans lequel se trouve la classe. Le nombre suivant le signe du dollar est un identifiant unique de cette classe. Si le champ Package est vide, la classe se trouve dans le package global ou le package non nommé.
Durée cumulée	Laps de temps, exprimé en millisecondes, qu'a mis à s'exécuter chaque méthode appelante, et toutes les méthodes suivantes.
Durée propre	Laps de temps, exprimé en millisecondes, qu'a mis à s'exécuter, chaque méthode appelante, à l'exception des méthodes appelées par les méthodes suivantes.

Le tableau suivant répertorie les champs de la table Appelés de la vue Statistiques de la méthode.

Colonne	Description
Méthode	Méthodes appelées par la méthode apparaissant dans le récapitulatif de la partie supérieure de cette vue. Si cette liste est vide, la méthode cible n'a été appelée par aucune méthode qui n'est pas filtrée.
Package	Package dans lequel se trouve chaque classe. Si la classe n'est pas dans un package, la valeur de ce champ est le nom du fichier dans lequel se trouve la classe. Le nombre suivant le signe du dollar est un identifiant unique de cette classe. Si le champ Package est vide, la classe se trouve dans le package global ou le package non nommé.
Durée cumulée	Laps de temps (exprimé en millisecondes) qu'a mis à s'exécuter chaque méthode appelée et toutes les méthodes suivantes.
Durée propre	Laps de temps (exprimé en millisecondes) qu'a mis à s'exécuter chaque méthode appelée, à l'exception des méthodes appelées par les méthodes suivantes.

Vous pouvez parcourir la pile d'appels pour rechercher des goulets d'étranglement des performances en cliquant sur les méthodes dans la table Appelants ou Appelés. Si vous double-cliquez sur une méthode dans ces tables, le profileur affiche le récapitulatif de cette méthode dans la partie supérieure de la vue Statistiques de la méthode puis affiche les appelants et les appelés pour la méthode nouvellement sélectionnée dans les deux tables.

Remarque : la durée cumulée moins la durée propre dans cette vue n'équivaut pas toujours à la durée cumulée des appelants. La raison en est que si vous remontez à un appelant, la durée cumulée correspond à la durée propre de cet appelant plus toutes les chaînes à partir desquelles la méthode d'origine a été appelée, mais pas les autres appelés.

Vous pouvez aussi utiliser les boutons Précédent, Suivant et Valeur par défaut du profileur pour parcourir la pile d'appels.

Vous pouvez limiter les données affichées dans la vue Statistiques de la méthode grâce aux filtres du profileur. Pour plus d'informations, voir « [A propos des filtres du profileur](#) » à la page 186.

Utilisation de la vue Objets vagabonds

La vue Objets vagabonds présente les différences entre deux instantanés de mémoire de l'application que vous profilez. Les différences que cette vue illustre sont le nombre d'occurrences d'objets dans la mémoire et la quantité de mémoire que ces objets utilisent. Ces données sont utiles pour identifier les fuites de mémoire. Le laps de temps écoulé entre deux instantanés de la mémoire est l'*intervalle d'instantanés*.

Pour ouvrir la vue Objets vagabonds, sélectionnez deux instantanés de mémoire et cliquez sur le bouton Objets vagabonds. Pour plus d'informations sur l'enregistrement d'un instantané de la mémoire, voir « [Utilisation de la vue Instantané de la mémoire](#) » à la page 171.

L'exemple suivant illustre la vue Objets vagabonds.

Classe	Package	Occurrences	Mémoire
Class		256 (12.77%)	1968107 (92.48%)
Function		209 (10.43%)	52851 (2.48%)
Object		795 (39.67%)	33204 (1.56%)
Array		242 (12.08%)	10976 (0.52%)
String		101 (5.04%)	8918 (0.42%)
Group	spark.components	7 (0.35%)	8848 (0.42%)
Rect	spark.primitives	9 (0.45%)	4104 (0.19%)
Dictionary	flash.utils	88 (4.39%)	3760 (0.18%)
CSSStyleDeclaration	mx.styles	36 (1.8%)	3312 (0.16%)
DataGrid	mx.controls	1 (0.05%)	2392 (0.11%)
Sample		1 (0.05%)	1580 (0.07%)

Le tableau suivant répertorie les colonnes de la vue Objets vagabonds.

Colonne	Description
Classe	Classes créées mais pas détruites pendant l'intervalle d'instantanés.

Colonne	Description
Package	Package dans lequel se trouve chaque classe. Si la classe n'est pas dans un package, la valeur de ce champ est le nom du fichier dans lequel se trouve la classe. Le nombre suivant le signe du dollar est un identifiant unique de cette classe. Si le champ Package est vide, l'objet se trouve dans le package global ou le package non nommé.
Occurrences	Nombre d'occurrences créées pendant l'intervalle d'instantanés. Différence entre le nombre d'occurrences de chaque classe dans le premier intervalle et dans le second intervalle. Par exemple, si le premier instantané contient 22 567 chaînes et le second 22 861 chaînes, la valeur de ce champ est 294.
Mémoire	Quantité de mémoire allouée pendant l'intervalle d'instantanés. Différence entre la quantité de mémoire que les occurrences de chaque classe ont utilisé lors de la prise du premier instantané et lors de la prise du second instantané. Par exemple, si la classe Strings occupe 2 031 053 octets dans le premier instantané et 2 029 899 dans le second, la valeur de ce champ est 1 154 octets.

Pour plus d'informations sur l'identification de fuites de mémoire, voir « [Localisation des fuites de mémoire](#) » à la page 182.

Utilisation du graphique Utilisation de la mémoire

Le graphique Utilisation de la mémoire présente la mémoire utilisée par l'application que vous profilez. Cette valeur est différente de l'utilisation de la mémoire de Flash Player et son navigateur. La raison en est que cette valeur ne comprend pas l'utilisation de la mémoire de l'agent du profileur ou du navigateur. Elle correspond uniquement à la somme des objets actifs de l'application profilée. En conséquence, si vous comparez la valeur de l'utilisation de la mémoire dans ce graphique à la quantité de mémoire que le navigateur utilise telle qu'indiquée, par exemple, dans le Gestionnaire des tâches de Windows, vous obtenez des résultats très différents.

L'image suivante illustre un graphique de la vue Utilisation de la mémoire.



La valeur de Mémoire actuelle est égale à la somme des totaux de la colonne Mémoire de la vue Objets actifs, en supposant que tous les filtres soient désactivés.

La valeur de Mémoire maximale correspond à la quantité la plus élevée de mémoire que cette application a utilisée au cours de la session de profilage en cours.

Le graphique Utilisation de la mémoire montre l'état de la mémoire de l'application au cours des 100 dernières secondes. Vous ne pouvez pas configurer ce nombre et vous ne pouvez pas enregistrer de données historiques pour le diagramme.

Si vous fermez le graphique Utilisation de la mémoire et que vous voulez le rouvrir, cliquez sur le bouton du menu déroulant dans la vue Profiler et sélectionnez Utilisation de la mémoire.

A propos de la récupération de place

La *récupération de place* est l'action de supprimer de la mémoire des objets qui ne sont plus nécessaires. La mémoire utilisée par des occurrences n'ayant plus de références à ces objets doit être désallouée pendant ce processus.

Flash Player nettoie la mémoire en fonction des besoins au cours du cycle de vie d'une application. Le déréférencement d'un objet ne déclenche pas le nettoyage de la mémoire. Ainsi, lorsque vous supprimez toutes les références à un objet, le nettoyage de la mémoire ne désalloue pas nécessairement la mémoire pour cet objet. Cet objet devient une cible potentielle du nettoyage de la mémoire.

Le fait de cliquer sur le bouton Lancer la récupération de place ne garantit pas que tous les objets susceptibles d'être nettoyés le soient. La récupération de place est généralement déclenchée par l'allocation de mémoire à de nouvelles ressources. En effet, lorsque de nouvelles ressources nécessitent de la mémoire qui n'est pas disponible dans l'allocation actuelle, la fonction de récupération de place s'exécute et libère de la mémoire signalée pour la désallocation. Par conséquent, la suppression de toutes les références à la mémoire ne signifie pas qu'elle sera nettoyée immédiatement, mais bien qu'elle le sera probablement lors de la création et de l'utilisation d'autres occurrences. Si une application est inactive, vous pouvez consulter l'allocation de sa mémoire. Même si certains objets sont marqués pour le nettoyage, l'utilisation de la mémoire d'une application inactive ne change pas tant que vous ne commencez pas à interagir avec elle.

Flash Player alloue la mémoire en blocs de plusieurs octets, et non octet par octet. Si une partie d'un bloc est marquée pour le nettoyage sans que d'autres parties le soient, le bloc n'est pas libéré. La récupération de place tente d'associer les parties de la mémoire non utilisées dans des blocs plus volumineux pouvant être nettoyés, mais cela ne se produit pas systématiquement à chaque tentative de récupération.

La récupération de place se fait implicitement avant l'enregistrement d'instantanés de la mémoire. En d'autres termes, un clic sur le bouton Prendre un instantané de la mémoire équivaut à un clic sur le bouton Lancer la récupération de place puis à un clic sur le bouton Prendre un instantané de la mémoire.

Exécution de la récupération de place pendant le profilage d'une application

❖ Sélectionnez l'application dans la vue Profiler, puis cliquez sur le bouton Lancer la récupération de place.

Vous pouvez analyser l'efficacité de ce processus en comparant deux instantanés de mémoire, avant et après le nettoyage de la mémoire.

Identification des zones à problèmes

Le profileur propose différentes techniques d'identification des zones à problèmes des applications.

Localisation des fuites de mémoire

L'un des problèmes les plus courants auxquels les développeurs d'applications sont confrontés est celui des fuites de mémoire. Celles-ci prennent souvent la forme d'objets qui ont été créés dans un laps de temps donné mais qui n'ont pas été récupérés.

L'une des façons de les identifier consiste à observer le nombre de références à un objet dans la table Occurrences de la vue Références de l'objet. Vous pouvez généralement ignorer les références provenant de propriétés et de liaisons d'un document et rechercher des références inattendues ou inhabituelles, plus particulièrement des objets qui ne sont pas des enfants de l'objet. Pour plus d'informations, voir « [Utilisation de la vue Références de l'objet](#) » à la page 172.

Vous pouvez également examiner les chemins menant aux occurrences d'un objet pour déterminer si un chemin possède une référence arrière à la récupération de place (racine de la RP). Une occurrence dont la libération était attendue mais qui possède des références à la racine de la RP indique une perte de mémoire. Dans ce cas, modifiez le code de l'application de manière à supprimer toute référence à la racine de la RP. Une occurrence ne possédant plus aucune référence à la racine de la RP est prête à faire l'objet d'une récupération de place. Flash Player libère finalement cette mémoire.

Une autre façon de localiser les fuites de mémoire consiste à comparer deux instantanés de mémoire dans la vue Objets vagabonds, afin de déterminer quels objets se trouvent toujours dans la mémoire après une série particulière d'événements. Ce processus est décrit dans la présente section.

Il est courant, pour nettoyer les liens de la mémoire, de faire appel aux méthodes `disconnect()`, `clearInterval()` et `removeEventListener()`.

Recherche d'une référence arrière à la racine de la RP pour une occurrence

- 1 Créez un instantané de la mémoire.

Voir « [Création et affichage d'un instantané de la mémoire](#) » à la page 171.

- 2 Spécifiez le nombre de chemins de référence arrière à rechercher.

Dans la vue Instantané de la mémoire, indiquez le nombre maximal de chemins à rechercher ou sélectionnez Afficher tous les chemins de références arrière.

- 3 Double-cliquez sur une classe dans la vue Instantané de la mémoire pour ouvrir la vue Références de l'objet.
- 4 Développez les chemins répertoriés et regardez s'il existe une référence arrière à la racine de la RP.

Rechercher les objets vagabonds

- 1 Créez deux instantanés de la mémoire.

Voir « [Création et affichage d'un instantané de la mémoire](#) » à la page 171.

- 2 Sélectionnez les deux instantanés de mémoire à comparer.

Remarque : si vous avez plus de deux instantanés de la mémoire, vous ne pouvez pas en sélectionner un troisième. Vous pouvez seulement comparer deux instantanés de mémoire en même temps.

- 3 Cliquez sur le bouton Rechercher les objets vagabonds.

La vue Objets vagabonds matérialise d'éventuelles fuites de mémoire. Les colonnes Occurrences et Mémoire indiquent les différences entre le nombre d'occurrences d'une classe et la quantité de mémoire consommée par ces occurrences au cours de l'intervalle entre deux instantanés. Si vous voyez une classe que vous ne vous attendiez pas à voir créée pendant ce laps de temps ou une classe que vous pensiez voir détruite pendant ce laps de temps, examinez l'application pour détecter si ces éléments peuvent être à l'origine d'une fuite de mémoire.

- 4 Pour déterminer la façon dont un objet de la vue Rechercher les objets vagabonds a été instancié, double-cliquez sur l'objet dans la vue. La vue Références de l'objet présente la trace de pile de chaque occurrence que vous avez sélectionnée.

Un moyen d'identifier une fuite de mémoire consiste dans un premier temps à déterminer un ensemble d'étapes que vous pouvez répéter de nombreuses fois avec l'application, là où l'utilisation de la mémoire ne cesse d'augmenter. Il est important d'effectuer ces étapes au moins une fois dans l'application avant de prendre le premier instantané de la mémoire, de manière que tous les objets mis en mémoire cache ou que toute autre occurrence soient inclus dans cet instantané.

Répétez ces étapes un certain nombre de fois (3, 7 ou un autre nombre premier) dans l'application. Prenez un second instantané de mémoire pour le comparer à l'instantané de départ. Dans la vue Rechercher les objets vagabonds, vous trouverez peut-être des objets vagabonds dotés d'occurrences au nombre de multiples de 3 ou de 7. Ces objets présentent probablement des fuites. Double-cliquez sur les classes pour consulter les traces de pile de chacune des occurrences.

Un autre moyen consiste à répéter la séquence d'étapes sur une longue période et d'attendre que l'utilisation de la mémoire atteigne son maximum. Si elle n'augmente pas après cela, il n'y a aucune fuite de mémoire pour cet ensemble d'étapes.

Parmi d'autres causes, les écouteurs d'événement en attente sont souvent à l'origine de fuites de mémoire. La méthode `removeEventListener()` permet de supprimer un écouteur d'événement qui n'est plus utilisé. Pour plus d'informations, voir Object creation and destruction dans *Utilisation d'Adobe Flex 4*.

Analyse des temps d'exécution

En analysant les temps d'exécution des méthodes et la quantité de mémoire allouée pendant ces appels de méthode, vous pouvez déterminer à quels niveaux les ralentissements des performances surviennent.

Cette opération est utile notamment pour identifier le temps d'exécution des méthodes appelées fréquemment, plutôt que celui des méthodes qui le sont rarement.

Détermination de la fréquence des appels de méthodes

- 1 Démarrez une session de profilage et veillez à activer le profilage des performances lors de la configuration du profileur sur l'écran de démarrage.
- 2 Sélectionnez l'application dans la vue Profiler.
- 3 Interagissez avec l'application jusqu'à atteindre le point où vous voulez commencer à analyser le nombre d'appels de méthode. Pour voir le nombre de fois qu'une méthode a été appelée lors du démarrage de l'application, n'interagissez pas avec l'application.
- 4 Cliquez sur le bouton Réinitialiser les données de performance. Toutes les données de performance sont alors effacées, de manière telle à ce que le profil de performance suivant n'inclue que les données saisies à compter de la réinitialisation.
- 5 Interagissez avec l'application jusqu'à atteindre le point où vous voulez vérifier le nombre d'appels de méthode depuis la réinitialisation des données.
- 6 Cliquez sur le bouton Capturer le profil de performance.
- 7 Double-cliquez sur le profil de performance dans la vue Profiler.
- 8 Dans la vue Profil de performance, triez les données selon la colonne Méthode et localisez votre méthode dans la liste.

La valeur figurant dans la colonne Appels correspond au nombre de fois que cette méthode a été appelée au cours de cet intervalle d'échantillonnage. Il s'agit de l'écart de temps qui sépare le moment où vous avez cliqué sur Réinitialiser les données de performance et le moment où vous avez cliqué sur Capturer le profil de performance.

Examinez les valeurs contenues dans les colonnes Durée cumulée, Durée propre, Durée cumulée moyenne et Durée propre moyenne de la vue Profil de performance. Elles indiquent le temps d'exécution des méthodes.

Comparez le temps mis par chaque méthode pour s'exécuter au temps que mettent à s'exécuter toutes les méthodes appelées par une méthode particulière. En règle générale, si la durée propre ou la durée propre moyenne d'une méthode est élevée ou élevée par rapport à celle d'autres méthodes, observez de plus près la façon dont la méthode est implémentée et tentez de réduire le temps d'exécution.

De la même façon, si la durée propre ou la durée propre moyenne d'une méthode est faible, mais que la durée cumulée ou la durée cumulée moyenne est élevée, observez les méthodes que cette méthode appelle pour localiser les goulets d'étranglement.

Localisation d'une allocation excessive d'objets

Un moyen d'identifier les zones à problèmes d'une application consiste à déterminer à quels niveaux un nombre excessif d'objets est créé. La création d'une occurrence d'un objet peut consommer un grand nombre de ressources, plus particulièrement si cet objet se trouve dans la liste d'affichage. L'ajout d'un objet à la liste d'affichage peut engendrer un grand nombre d'appels des méthodes de style et de présentation, ce qui risque de ralentir l'application. Dans certains cas, vous pouvez restructurer le code afin de réduire le nombre d'objets créés.

Une fois que vous avez déterminé si la création de certains objets est superflue, vous pouvez décider s'il est raisonnable ou utile de réduire le nombre d'occurrences de cette classe. Vous pouvez par exemple déterminer la taille des objets, car les objets volumineux présentent généralement le plus grand potentiel d'optimisation.

Pour savoir quels objets sont créés en nombre important, vous pouvez comparer les instantanés de mémoire de l'application à deux moments.

Affichage du nombre d'occurrences d'une classe spécifique

- 1 Démarrez une session de profilage et veillez à activer le profilage de la mémoire lors de la configuration du profileur sur l'écran de démarrage.
- 2 Interagissez avec l'application jusqu'à atteindre l'endroit dont vous souhaitez prendre un instantané de la mémoire.
- 3 Cliquez sur le bouton Prendre un instantané de la mémoire. Le profileur ajoute un nouvel instantané de la mémoire à la liste d'applications dans la vue Profiler.
- 4 Dans la vue Profiler, double-cliquez sur l'instantané de la mémoire pour l'ouvrir.
- 5 Pour afficher le nombre d'occurrences d'une classe particulière, ainsi que la quantité de mémoire que ces occurrences utilisent, effectuez un tri selon la colonne Classe et localisez la classe dans cette colonne. Vous pouvez aussi effectuer un tri selon les autres colonnes afin d'identifier rapidement les objets qui occupent le plus de mémoire ou les objets comptant le plus grand nombre d'occurrences. Dans la plupart des cas, la classe Strings est celle qui compte le plus d'occurrences et qui utilise le plus de mémoire.

Pour plus d'informations sur la vue Instantané de la mémoire, voir « [Utilisation de la vue Instantané de la mémoire](#) » à la page 171.

Localisation d'occurrences d'allocation excessive d'objets

- 1 Démarrez une session de profilage et veillez à activer le profilage de la mémoire lors de la configuration du profileur sur l'écran de démarrage.
- 2 Interagissez avec l'application jusqu'à atteindre le premier endroit dont vous souhaitez prendre un instantané de la mémoire.
- 3 Cliquez sur le bouton Prendre un instantané de la mémoire.
Le profileur enregistre l'instantané de la mémoire dans la vue Profiler en indiquant la date et l'heure.

4 Interagissez avec l'application jusqu'à atteindre le deuxième endroit dont vous souhaitez prendre un instantané de la mémoire.

5 Cliquez à nouveau sur le bouton Prendre un instantané de la mémoire.

Le profileur enregistre le deuxième instantané de la mémoire dans la vue Profiler en indiquant la date et l'heure.

6 Sélectionnez les deux instantanés de mémoire à comparer.

Remarque : si vous avez plus de deux instantanés de la mémoire, vous ne pouvez pas en sélectionner un troisième. Vous ne pouvez comparer que deux instantanés de mémoire à la fois.

7 Cliquez sur le bouton Afficher le suivi des allocations.

La vue Suivi des allocations indique les méthodes qui ont été appelées entre les deux instantanés et la quantité de mémoire consommée lors de ces appels de méthode. Voir « [Utilisation de la vue Suivi des allocations](#) » à la page 173.

A propos des filtres du profileur

La quantité de données contenue dans une vue du profileur peut quelquefois être considérable et le niveau de détail trop important. Les actions internes de Flash Player peuvent occulter les données qui vous intéressent véritablement, telles que vos propres méthodes et classes. En outre, Flash Player crée et détruit de nombreux objets sans votre intervention directe. Vous pouvez par conséquent constater que des milliers de chaînes ou de tableaux sont utilisés dans votre application.

Vous pouvez définir des filtres dans les vues suivantes :

- Objets actifs
- Instantané de la mémoire
- Profil de performance
- Statistiques de la méthode
- Suivi des allocations

Vous pouvez définir quels packages doivent figurer dans les vues du profileur grâce aux filtres. Il existe deux types de filtres.

Filtres d'exclusion Les filtres d'exclusion indiquent au profileur qu'il doit exclure de ses vues les packages qui correspondent aux motifs de sa liste de motifs. Si vous faites appel, par exemple, à des commandes de création de graphiques, mais que vous ne voulez pas les profiler, vous pouvez ajouter le motif `mx.charts.*` au filtre d'exclusion. Vous pouvez également exclure des éléments intégrés globaux, ce qui comprend les classes globales `String` et `Array`.

Filtres d'inclusion Les filtres d'inclusion indiquent au profileur qu'il doit inclure dans ses vues uniquement les packages qui correspondent aux motifs de sa liste de motifs. Si vous possédez, par exemple, un package personnalisé nommé `com.mycompany.*`, vous pouvez afficher des informations uniquement sur les classes contenues dans ce package en l'ajoutant au filtre d'inclusion.

Les exclusions par défaut sont les classes `flash.*.*`, `spark.*.*`, `mx.*.*`, ainsi que les classes d'infrastructure Flex dans le package global ou non nommé, ce qui comprend les classes globales `String` et `Array`. Cela signifie que les inclusions par défaut sont des classes personnalisées se trouvant dans le package non nommé et des classes personnalisées dans des packages hors infrastructure (telles que `com.mycompany.MyClass`).

Vous pouvez exclure les classes personnalisées se trouvant dans le package non nommé à partir des données de profilage. Pour ce faire, ajoutez « * » à la liste d'exclusion.

Nombre maximal de lignes visibles L'option Nombre maximal de lignes visibles définit le nombre de rangées de données qui peuvent être affichées dans une vue. Augmentez cette valeur si les données que vous recherchez ne s'affichent pas dans la vue. Réduisez cette valeur pour améliorer les performances du profileur. Utilisez d'autres filtres pour vérifier que vous affichez les données qui vous intéressent.

Nombre maximal de chemins de références arrière à rechercher L'option Nombre maximal de chemins de références arrière à rechercher définit le nombre de chemins vers un objet référencé à afficher lorsque vous examinez les références d'un objet. Les chemins s'affichent selon le chemin le plus court. Par défaut, les dix chemins les plus courts s'affichent. Augmentez cette valeur pour afficher d'autres chemins ou sélectionnez Afficher tous les chemins de références arrière. L'affichage de chemins supplémentaires peut vous aider à trouver les objets qui ne sont plus référencés. Voir « [Localisation des fuites de mémoire](#) » à la page 182.

Définition de préférences de filtres par défaut

❖ Ouvrez la boîte de dialogue Préférences et sélectionnez Flash Builder > Profileur > Filtres d'inclusion ou Filtres d'exclusion.

Lors de l'affichage des données du profileur, celui-ci applique d'abord les filtres d'exclusion, puis les filtres d'inclusion. Supposons que vous définissiez le filtre d'exclusion sur `mx.controls.*`, mais que vous définissiez le filtre d'inclusion sur `mx.*.*`. Le profileur n'affiche aucune information concernant les classes du package `mx.controls`, même si leur motif correspond à la liste de motifs d'inclusion, car ce package est exclu. Supposons également que vous définissiez le filtre d'exclusion sur `mx.*.*` et le filtre d'inclusion sur `mx.controls.*`. Le profileur n'affiche aucune information concernant les classes du package `mx.controls.*` car elles ont été exclues avant l'inclusion du package.

Lorsque vous filtrez certains points de données, les valeurs de pourcentage des colonnes sont ajustées afin de refléter uniquement le pourcentage de données non filtrées.

Le profileur conserve les filtres d'une session de profilage à l'autre pour la même application.

Les paramètres de filtres ne sont pas hérités par les vues secondaires. Par exemple, si vous appliquez un filtre aux données contenues dans la vue Instantané de la mémoire, puis accédez à la vue Références de l'objet en double-cliquant sur une méthode, la vue Références de l'objet n'applique pas le même filtre.

Déterminer si les données sont filtrées

- 1 Cliquez sur le bouton Filtres ou examinez le titre des tables de données. Si des filtres sont appliqués, l'en-tête de la colonne Package est Package (filtré).
- 2 (Facultatif) Réinitialisez les filtres aux valeurs par défaut en cliquant sur le bouton Restaurer les valeurs par défaut.

Chapitre 9 : Création d'une interface utilisateur avec Flash Builder

A propos de la structure des interfaces utilisateur dans Flex

Les éléments constitutifs des interfaces utilisateur dans Flex sont les conteneurs et les contrôles MXML. Un conteneur est une région rectangulaire qui permet d'organiser et de mettre en forme les contrôles, d'autres conteneurs et les composants personnalisés. Un contrôle est un composant d'interface utilisateur tel que Button, TextArea ou ComboBox. Les composants personnalisés sont ceux que vous avez définis dans des fichiers MXML ou ActionScript distincts et que vous incluez dans votre application.

Remarque : vous pouvez également ajouter à votre application des ressources créées avec Adobe Flash® Professional®. Voir « [Création et modification de composants Flash](#) » à la page 194.

Composants Spark

Flex 4 comporte de nouveaux composants Spark, définis dans les packages spark.*. Les composants MX, qui étaient disponibles dans les versions précédentes de Flex, sont définis dans les packages mx.*.

La principale différence entre les composants Spark et les composants MX réside dans l'utilisation des styles CSS et dans l'habillage. Pour plus d'informations sur l'utilisation de composants visuels dans une application, voir Visual components.

Conteneurs Spark et MX

Flex définit deux types de conteneurs : les conteneurs Spark et les conteneurs MX. Les conteneurs Spark sont définis dans le package spark.components, alors que les conteneurs MX sont définis dans les packages mx.core et mx.containers. Les conteneurs Spark définissent la présentation différemment des conteneurs MX.

Bien que les conteneurs MX fournissent la plupart des options de présentation offertes par les conteneurs Spark, Adobe recommande d'utiliser les conteneurs Spark.

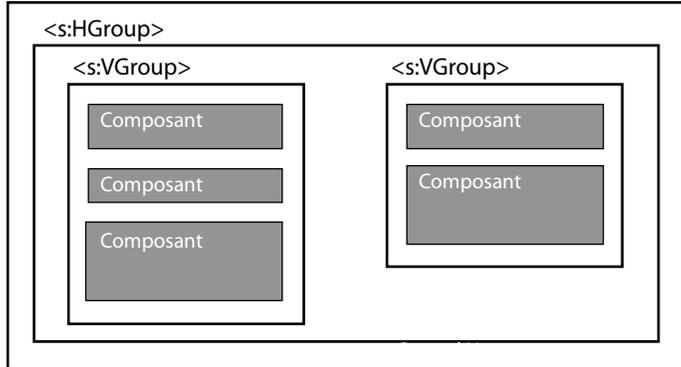
Pour plus d'informations sur l'utilisation de conteneurs pour présenter une application, voir Introduction to containers.

Conteneurs Spark

Les applications dans Flex sont généralement constituées d'un fichier d'application MXML (fichier doté d'un conteneur `<s:Application>` comme balise parent) et d'un ou de plusieurs composants définis dans des fichiers MXML, des fichiers ActionScript ou des fichiers de composants Flash (fichiers SWC) distincts. Vous pouvez insérer des conteneurs et contrôles directement dans le fichier d'application MXML ou dans des fichiers MXML distincts pour créer des composants personnalisés et les insérer ensuite dans le fichier d'application.

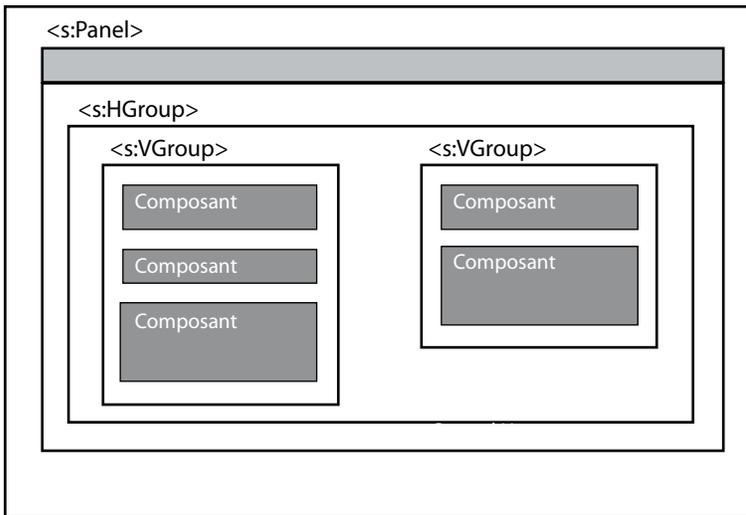
L'exemple suivant illustre une structure d'application simple. Les conteneurs et contrôles sont insérés directement dans le fichier d'application MXML.

```
<s:Application>
```



Voici un exemple analogue qui fait appel au conteneur Panel comme structure de base pour les conteneurs et les composants.

```
<s:Application>
```

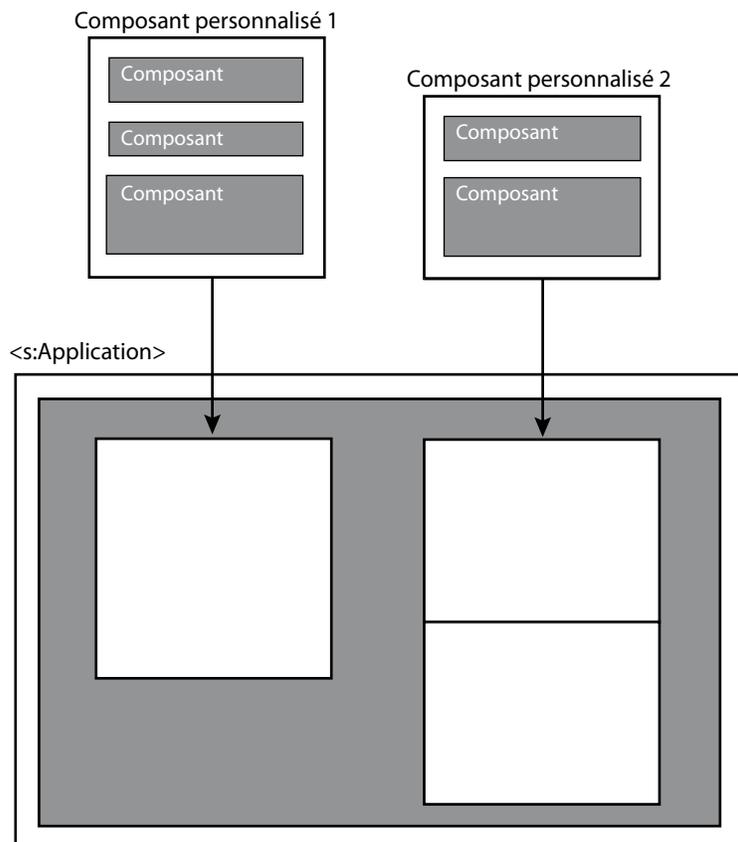


Pour plus d'informations sur les conteneurs, les composants et la présentation d'une interface utilisateur, voir *Building the user interface*, y compris les sections *Introduction to containers* et *Visual components*.

Structure d'application basée sur des composants

Une structure basée sur des composants est utile si votre interface utilisateur est composée d'éléments fonctionnels distincts. Par exemple, votre présentation peut comporter un élément qui extrait et affiche un catalogue de produits, un autre élément qui extrait et affiche des informations concernant tous les produits sur lesquels l'utilisateur peut cliquer dans le catalogue et un élément qui permet à l'utilisateur d'ajouter le produit sélectionné à un panier d'achats. Cette interface utilisateur peut être structurée sous la forme de trois composants personnalisés ou sous la forme d'une association de composants personnalisés et de contrôles insérés directement dans la présentation.

L'exemple suivant illustre une structure d'application basée sur des composants. Vous regroupez les contrôles des éléments d'interface utilisateur dans des fichiers de composants personnalisés séparés, que vous insérez par la suite dans le fichier d'application MXML.



Voir aussi

« [Création de composants MXML personnalisés](#) » à la page 134

Présentations pour conteneurs Spark

La plupart des conteneurs Spark, dont le conteneur de niveau supérieur `<s:Application>`, vous permettent de spécifier une présentation définissant le positionnement des composants ajoutés au conteneur. Par défaut, ces conteneurs Spark font appel à `BasicLayout`.

Vous pouvez spécifier les présentations suivantes pour les conteneurs Spark :

- `BasicLayout`

Cette présentation utilise le positionnement absolu. Tous les enfants du conteneur sont positionnés explicitement ou à l'aide de contraintes. Cette présentation est celle appliquée par défaut à tous les conteneurs, à l'exception de `HGroup` et `VGroup`.

- `HorizontalLayout`

Cette présentation positionne les enfants sur une seule ligne horizontale.

- `VerticalLayout`

Cette présentation positionne les enfants sur une seule colonne verticale.

- `TileLayout`

Cette présentation positionne les enfants en colonnes verticales ou en lignes horizontales, entamant des nouvelles lignes ou colonnes si nécessaire.

Les conteneurs `HGroup` et `VGroup` spécifient respectivement `HorizontalLayout` et `VerticalLayout`. Vous ne devrez généralement pas remplacer la présentation par défaut de ces conteneurs.

Voir aussi

« [Conteneurs MX](#) » à la page 191

Enfants des conteneurs Spark

Tous les conteneurs Spark peuvent avoir des composants visuels Spark et MX comme enfants.

Bien que les conteneurs MX fournissent la plupart des options de présentation offertes par les conteneurs Spark, Adobe recommande d'utiliser les conteneurs Spark.

Conteneurs MX

L'une des principales différences entre les conteneurs Spark et MX réside dans le fait que l'algorithme de présentation des conteneurs MX est fixe, alors que celui des conteneurs Spark peut être sélectionné. MX définit donc un conteneur différent pour chaque type de présentation. Spark définit moins de conteneurs, mais vous permet de modifier l'algorithme de présentation, ce qui accroît la flexibilité des conteneurs.

Les conteneurs Spark peuvent avoir des composants visuels Spark et MX comme enfants. Les composants de navigateur MX ne peuvent toutefois pas avoir des composants Spark comme enfants. Les composants de navigateur MX comptent parmi d'autres les conteneurs `ViewStack`, `TabNavigator` et `Accordion`, ainsi que les composants `TabBar` ou `ButtonBar` et des composants similaires.

Certains composants implémentés dans MX n'ont pas d'équivalents dans Spark. Dans ce cas, les composants MX sont répertoriés comme composants recommandés dans la vue Composants et dans l'assistant de contenu.

Voir aussi

« [Vue Composants](#) » à la page 192

« [A propos de l'assistant de contenu](#) » à la page 106

Ajout et modification des composants

Flash Builder vous permet d'ajouter des composants à une application ainsi que d'en définir la taille, de les modifier ou de les supprimer. Vous pouvez également ajouter et modifier des composants personnalisés définis dans des fichiers MXML et ActionScript distincts.

Ajout de composants en mode Création MXML

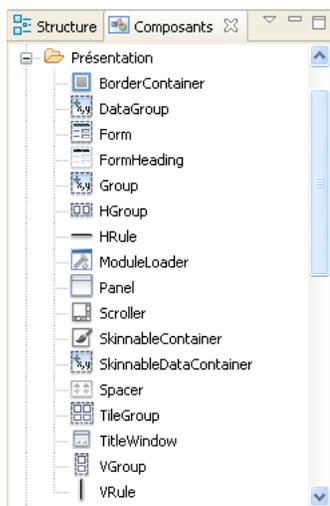
Le mode Création MXML vous permet d'ajouter des conteneurs et des contrôles standard à l'interface utilisateur. Pour ce faire, vous faites glisser les composants de la vue Composants vers la zone de création du fichier MXML et vous les positionnez en fonction de la règle de présentation du conteneur. Vous pouvez aussi ajouter des composants personnalisés que vous définissez dans des fichiers MXML et ActionScript distincts et que vous enregistrez dans le projet en cours ou dans son chemin source.

- 1 En mode Création de l'éditeur MXML, ouvrez le fichier MXML dans lequel vous voulez insérer le composant.

Le fichier MXML doit être ouvert en mode Création pour que vous puissiez utiliser la vue Composants. Le fichier MXML peut être le fichier de l'application principale (fichier doté d'un conteneur Application) ou un fichier de composant MXML personnalisé.

- 2 Dans la vue Composants, localisez le composant à ajouter.

Si la vue n'est pas ouverte, sélectionnez Fenêtre > Composants.



La vue Composants classe les composants par catégories. Elle affiche également les composants recommandés pour un projet. Vous pouvez modifier les paramètres d'affichage de manière à obtenir la liste complète des composants. Pour plus d'informations, voir « [Vue Composants](#) » à la page 192.

- 3 Faites glisser un composant de la vue Composants vers le fichier MXML.

Le composant est placé en fonction de la règle de présentation du conteneur parent.

Vue Composants

La vue Composants s'affiche lorsque vous accédez au mode Création de l'éditeur MXML. Faites-en glisser les composants vers la zone de création pour définir la présentation de votre application.

Cette vue affiche par défaut les composants recommandés pour un projet. La version du SDK Flex d'un projet détermine quels composants sont répertoriés comme composants recommandés.

Lors de la création d'un projet par exemple, Flash Builder utilise par défaut le SDK Flex 4. Les composants recommandés pour le SDK Flex 4 sont les composants Spark. Les composants MX ne possédant aucun équivalent Spark sont également répertoriés dans la liste des composants recommandés.

La vue Composants regroupe les composants en catégories, telles que Contrôles, Présentations, Navigateurs et Graphiques. La catégorie Personnaliser contient les composants définis dans des fichiers MXML et ActionScript distincts. Pour plus d'informations, voir « [Création de composants MXML personnalisés](#) » à la page 134.

Remarque : la vue Composants répertorie les composants visibles.

Modification de l'affichage des composants dans la vue Composants

Accédez au menu de la vue Composants pour en modifier l'affichage.

- 1 Pour afficher tous les composants, désélectionnez Afficher uniquement les composants recommandés.
En mode d'affichage de tous les composants, la vue regroupe les composants en composants Spark et MX.
- 2 Pour afficher les noms qualifiés complets des classes, sélectionnez Afficher les noms qualifiés complets des classes.

Ajout de composants par la rédaction de code

En mode Source de l'éditeur MXML, les indicateurs de code vous assistent lors de l'ajout de conteneurs et de contrôles Flex standard à l'interface utilisateur. Dans Flash Builder, tout comme dans Eclipse, les indicateurs de code sont désignés par le terme Assistant de contenu.

L'assistant de contenu de Flash Builder vous aide à insérer des composants recommandés. L'exemple suivant illustre comment utiliser les indicateurs de code pour insérer un conteneur `<s:VGroup>` dans un conteneur `<s:HGroup>`. `HGroup` étant un composant Spark, l'assistant privilégie l'utilisation d'un conteneur `VGroup` à celle de `<mx:VBox>`.

- 1 Ouvrez un fichier MXML en mode Source dans l'éditeur MXML.

Le fichier MXML peut être le fichier de l'application principale (fichier doté d'un conteneur `Application`) ou un fichier de composants MXML personnalisés.

- 2 Placez le point d'insertion dans la balise de conteneur parent.

Pour insérer par exemple un conteneur `VGroup` dans un conteneur parent `HGroup`, placez le point d'insertion après la balise d'ouverture `<s:HGroup>`.

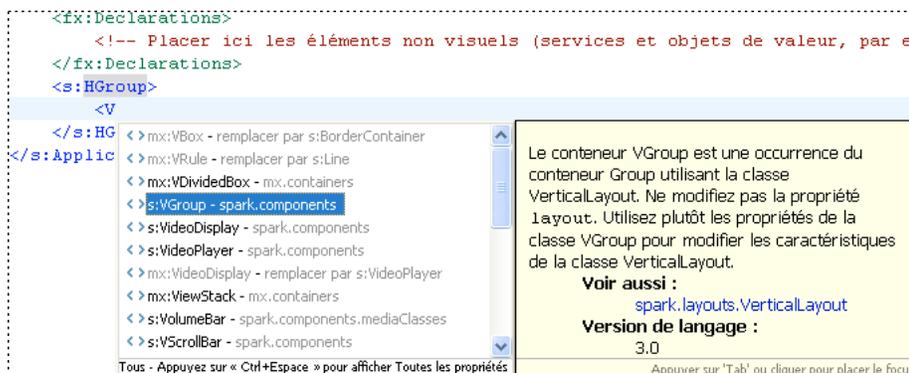
```
<s:HGroup>
    insertion point here
</s:HGroup>
```

- 3 Saisissez la balise du composant.

En cours de saisie, l'assistant de contenu s'affiche pour vous suggérer des entrées possibles. Les composants recommandés s'affichent en noir. Les composants non recommandés s'affichent en gris.

Dans cet exemple, `VGroup` est recommandé, contrairement à `VBox` qui ne l'est pas.

- 4 Appuyez le cas échéant sur les touches représentant des flèches pour sélectionner une balise dans le menu, puis appuyez sur Entrée.



Outre les composants standard, l'assistant de contenu répertorie les composants personnalisés que vous avez définis dans des fichiers MXML et ActionScript distincts et que vous avez enregistrés dans le projet en cours ou dans son chemin source.

L'assistant de contenu peut également suggérer des propriétés, des événements, des effets et des styles. Appuyez sur les touches Ctrl+Espace pour parcourir les recommandations de l'assistant de contenu.

Vous pouvez modifier le type et l'ordre des recommandations de l'assistant de contenu. Dans la boîte de dialogue Préférences, sélectionnez Flash Builder > Editeurs > Code MXML > Avancées.

Voir aussi

« [A propos de l'assistant de contenu](#) » à la page 106

« [Création de composants MXML personnalisés](#) » à la page 134

Création et modification de composants Flash

Adobe Flash® Professional CS5 crée des applications compatibles avec Adobe Flash Player 10. Les applications Adobe prennent également en charge Flash Player 10, ce qui signifie que vous pouvez importer dans vos applications des ressources issues de Flash Professional CS5. Vous pouvez créer des contrôles, des conteneurs, des habillages et d'autres ressources dans Flash Professional CS5, puis les importer dans vos applications sous la forme de fichiers SWC. Pour plus d'informations sur la création de composants à l'aide de Flash Professional CS5, voir [Extensions pour la création d'habillages Flex et kit de composants Flex pour Flash Professional](#).

La vue Création de l'éditeur Flash Builder vous permet d'insérer un nouveau composant Flash en ajoutant un espace réservé pour un composant ou un conteneur Flash. Vous pouvez appeler Flash Professional CS5 à partir de Flash Builder pour créer le composant ou conteneur. Vous pouvez également appeler Flash Professional CS5 pour modifier les composants Flash créés précédemment.

Si l'application contient un composant SWFLoader destiné à lancer des fichiers d'animation Flash, vous pouvez démarrer Flash Professional CS5 pour créer ou modifier les fichiers FLA et SWF associés.

Insertion d'un composant ou d'un conteneur Flash

- 1 Dans l'éditeur Flash Builder, sélectionnez la vue Création et assurez-vous que la vue Composants est visible.
- 2 Dans le dossier Personnaliser de la vue Composants, faites glisser soit un Nouveau composant Flash soit un Nouveau conteneur Flash vers la zone de création.

Vous pouvez ensuite redimensionner ou positionner le composant ou conteneur.

- 3 Dans le menu contextuel ou dans la Vue standard des Propriétés du fichier, sélectionnez Créer dans Adobe Flash.

Remarque : vous pouvez également double-cliquer sur le composant dans la vue Création pour créer l'élément dans Adobe Flash.

- 4 Dans la boîte de dialogue qui s'ouvre, précisez le nom de la classe et du fichier SWC, puis cliquez sur Créer pour ouvrir Adobe Flash Professional CS5.
- 5 Dans Flash Professional CS5, modifiez le composant ou conteneur. Sélectionnez Terminer lorsque vous avez fini pour retourner à Flash Builder.

Modification d'un composant ou d'un conteneur Flash

Cette procédure suppose que vous avez déjà inséré un composant ou un conteneur Flash dans Flash Builder.

- 1 Dans la vue Création de l'éditeur Flash Builder, sélectionnez le composant ou le conteneur Flash à modifier.
- 2 Dans le menu contextuel ou dans la Vue standard des Propriétés Flex, sélectionnez Editer dans Adobe Flash.

Remarque : vous pouvez également double-cliquer sur le composant dans la vue Création pour modifier l'élément dans Adobe Flash.

- 3 Dans Flash Professional CS5, modifiez le composant ou conteneur. Sélectionnez Terminer lorsque vous avez fini pour retourner à Flash Builder.

Création ou modification d'une animation Flash associée à un composant SWFLoader

Cette procédure suppose que l'application contient un composant SWFLoader.

- 1 Dans la vue Création de l'éditeur Flex, sélectionnez le composant SWFLoader.
- 2 A partir du menu contextuel du composant ou de la Vue standard de la fenêtre Propriétés, lancez Flash Professional CS5 en effectuant l'une des opérations suivantes :
 - Sélectionnez Créer dans Adobe Flash pour créer une animation Flash associée au composant SWFLoader.
 - Sélectionnez Editer dans Adobe Flash pour modifier l'animation Flash associée au composant SWFLoader.
- 3 Une fois que vous avez fini de modifier l'animation, sélectionnez Terminer pour retourner à Flash Builder.

Importation de ressources Flash CS3

Vous pouvez ajouter à l'interface utilisateur des composants Flash créés avec Adobe Flash CS3 Professional.

Remarque : pour pouvoir créer des composants Flex dans Flash CS3, vous devez installer le kit de composants Flex pour Flash CS3. Pour plus d'informations, consultez l'article [Importing Flash CS3 Assets into Flex](#).

- 1 Vérifiez que le composant Flash est enregistré dans le chemin d'accès à la bibliothèque du projet en cours.

Le chemin d'accès à la bibliothèque précise l'emplacement d'un ou de plusieurs fichiers SWC que l'application lie au moment de l'exécution. Il est défini dans les paramètres du compilateur Flex pour le projet. Dans les nouveaux projets, le dossier `libs` se trouve par défaut sur le chemin d'accès à la bibliothèque.

Pour définir ou découvrir le chemin d'accès à la bibliothèque, sélectionnez le projet dans l'Explorateur de packages, puis sélectionnez Projet > Propriétés. Dans la boîte de dialogue Propriétés, sélectionnez la catégorie Chemin de génération Flex, puis cliquez sur l'onglet Chemin d'accès à la bibliothèque. Pour plus d'informations, voir « [Génération manuelle de projets](#) » à la page 81.

Il est également possible de définir le chemin d'accès à la bibliothèque dans le fichier de configuration `flex-config.xml` d'Adobe LiveCycle Data Services ES°.

- 2 Ouvrez un fichier MXML et ajoutez un composant Flash de l'une des façons suivantes :
 - En mode Création dans l'éditeur MXML, développez la catégorie Personnaliser de la vue Composants puis faites glisser le composant Flash vers le fichier MXML. Dans le cas de documents déjà ouverts, cliquez sur le bouton Actualiser (l'icône verte contenant des flèches) pour afficher le composant après l'avoir inséré.
 - En mode Source, entrez la balise de composant, puis faites appel à l'Assistant de contenu pour terminer rapidement la balise.

Utilisation visuelle des composants

Flash Builder vous permet de travailler visuellement avec les composants dans l'éditeur MXML pour que vous puissiez voir l'aspect de votre application à mesure que vous la construisez. L'éditeur MXML possède deux modes : le mode Source pour la rédaction de code et le mode Création pour le développement visuel des applications.

Utilisation de l'éditeur MXML en mode Création

Le mode Création vous permet d'afficher, de sélectionner, de déplacer, de redimensionner, de faire défiler, d'agrandir ou d'effectuer un panoramique des éléments dans la zone de création.

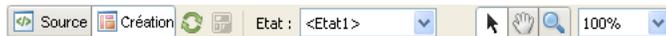
Affichage d'un fichier MXML

- 1 Si le fichier MXML n'est pas déjà ouvert dans l'éditeur MXML, double-cliquez sur le fichier dans l'Explorateur de packages pour l'ouvrir.
- 2 Si l'éditeur MXML affiche le code source, cliquez sur Création dans la partie supérieure de la zone de l'éditeur. Vous pouvez passer rapidement d'un mode à l'autre en appuyant sur les touches Ctrl+' (Guillemet ouvrant).

Le basculement entre les modes Source et Création affiche ou masque automatiquement les vues liées à la création, telles que Composants, Propriétés et Etats. Pour activer et désactiver ce comportement, accédez à la boîte de dialogue Préférences et sélectionnez Flash Builder > Editeurs > Mode Création et enfin l'option Afficher automatiquement les vues liées à la création.

Sélection et déplacement de composants dans la zone de création

- ❖ Cliquez sur le bouton Mode Sélection (en forme de flèche) sur la droite de la barre d'outils de l'éditeur. Ce mode est activé par défaut lorsque vous ouvrez un document. Appuyez sur la touche V du clavier pour passer en mode Sélection. Cliquez sur un composant et faites-le glisser jusqu'à l'emplacement voulu. Vous pouvez aussi le faire glisser pour le redimensionner et cliquer dessus pour le sélectionner.



Panoramique et défilement dans la zone de création

- ❖ Cliquez sur le bouton Mode Panoramique sur le côté droit de la barre d'outils de l'éditeur. Appuyez sur H pour entrer en mode Panoramique à partir du clavier. Pour entrer temporairement en mode Panoramique, appuyez sur la barre d'espace du clavier et maintenez-la enfoncée. Vous pouvez sélectionner ou déplacer des éléments en mode Panoramique.

Zoom dans la zone de création

Il existe plusieurs façons d'utiliser l'outil de zoom. Vous pouvez sélectionner des pourcentages dans le menu principal et le menu contextuel, cliquer sur le bouton Mode Zoom sur la barre d'outils ou utiliser des raccourcis clavier. Le pourcentage d'agrandissement en cours est toujours affiché dans la barre d'outils.

- Dans le menu principal, sélectionnez Création > Zoom avant ou Création > Zoom arrière. Vous pouvez également sélectionner le sous-menu Agrandissement et choisir un pourcentage spécifique.
- Cliquez sur le bouton Mode Zoom dans la barre d'outils ou appuyez sur la touche Z du clavier. Un curseur de symbole + s'affiche dans la zone de création.
- Sélectionnez un pourcentage dans le menu contextuel en regard des boutons Sélectionner, Panoramique et Mode zoom de la barre d'outils de l'éditeur. La zone de création prend le pourcentage sélectionné ou s'ajuste à la taille de la fenêtre.
- Cliquez avec le bouton droit dans la zone de création pour sélectionner le sous-menu Zoom avant, Zoom arrière ou Agrandissement. La zone de création change selon votre choix.

Vous pouvez recourir aux raccourcis clavier suivants dans la zone de création :

- Zoom avant : Ctrl+= (Commande+= sous Mac OS)
- Zoom arrière : Ctrl+- (Commande+- sous Mac OS)

Pour connaître d'autres raccourcis clavier, sélectionnez Aide > Assistant de touches.

Sélection de plusieurs composants dans un fichier MXML

Vous pouvez sélectionner plusieurs composants dans un fichier MXML. Cela peut s'avérer utile si vous voulez définir une valeur commune pour une propriété partagée.

- Appuyez sur la touche Ctrl tout en cliquant sur (Commande-clic sous Macintosh) chaque composant de la présentation.
- Cliquez sur l'arrière-plan de la page et dessinez un rectangle chevauchant les composants.
- Dans la vue Structure (Fenêtre > Structure), appuyez sur la touche Ctrl tout en cliquant sur (Commande-clic sous Macintosh) les composants dans le contrôle d'arborescence.

Désélection de plusieurs composants

- Cliquez sur le conteneur d'arrière-plan.
- Cliquez sur un composant désélectionné.
- Cliquez dans la marge grise encadrant le composant racine.

Positionnement de composants

Vous pouvez changer la position des composants visuellement en fonction de la propriété layout du conteneur parent. Si le conteneur parent spécifie que le positionnement est absolu, vous pouvez faire glisser les composants pour les repositionner. Dans le cas contraire, les composants repositionnés appliquent les règles de présentation du conteneur parent.

Par défaut, les conteneurs Spark font appel à la propriété BasicLayout qui autorise le positionnement absolu. Pour certains conteneurs MX, vous pouvez également définir la propriété layout sur `absolute`. Pour plus d'informations, voir MX layout containers.

Remarque : vous pouvez également utiliser des contraintes de présentation pour positionner un composant dans un conteneur. Ces contraintes définissent de quelle manière les composants doivent être repositionnés en cas de redimensionnement d'un conteneur. Elles permettent toutefois aussi d'en spécifier la position dans des conteneurs dont la taille est fixe. Voir « [Définition de contraintes de présentation pour les composants](#) » à la page 207.

- 1 En mode Création de l'éditeur MXML, sélectionnez le composant dans la présentation et faites-le glisser vers une nouvelle position.

Le composant est placé en fonction des règles de présentation du conteneur parent.

Si le conteneur bénéficie du positionnement absolu, vous pouvez faire glisser les composants et les positionner n'importe où dans le conteneur.

Par exemple, si vous déplacez un conteneur VGroup dans un conteneur HGroup, le conteneur VGroup est placé à l'horizontale avec les autres conteneurs enfant (s'ils existent).

- 2 En mode Création, sélectionnez le conteneur parent du composant et modifiez ses propriétés de présentation dans la vue Propriétés.

Dans certains cas, vous pouvez changer la position des composants enfant en modifiant les propriétés du conteneur parent. Vous pouvez par exemple utiliser les propriétés `verticalGap` et `horizontalGap` du conteneur pour définir l'espacement entre les composants enfant et la propriété `direction` pour préciser une présentation à lignes ou à colonnes.

Voir aussi

« [Conteneurs Spark et MX](#) » à la page 188

« [Définition de contraintes de présentation pour les composants](#) » à la page 207

Dimensionnement des composants

En mode Création, vous pouvez modifier la taille d'un composant en faisant glisser la poignée de redimensionnement, en sélectionnant des options dans les menus ou en modifiant ses propriétés dans la vue Propriétés.

Remarque : utilisez une *présentation basée sur contrainte* pour spécifier la manière dont un composant est redimensionné dynamiquement lors du redimensionnement d'un conteneur. Pour plus d'informations, voir « [Définition de contraintes de présentation pour les composants](#) » à la page 207.

Dimensionnement d'un composant visuellement

- ❖ En mode Création de l'éditeur MXML, cliquez sur le composant et faites glisser la poignée de dimensionnement pour le redimensionner.
 - Pour contraindre les proportions du composant, maintenez enfoncée la touche Maj tout en le faisant glisser.
 - Si l'alignement est activé, lorsque vous effectuez le redimensionnement, des lignes d'accrochage s'affichent pour vous aider à aligner les bords du composant avec les composants voisins. Pour activer ou désactiver l'alignement dans le menu principal, sélectionnez > Activer l'accrochage.

Harmonisation de la largeur ou de la hauteur de deux composants ou plus

- 1 En mode Création, sélectionnez deux composants ou plus.
- 2 Dans le menu Création, sélectionnez l'une des options de dimensionnement suivantes :

Rendre de la même largeur Définit la propriété `width` de tous les composants sélectionnés sur celle du composant que vous avez sélectionné en premier.

Rendre de la même hauteur Définit la propriété `height` de tous les composants sélectionnés sur celle du composant que vous avez sélectionné en premier.

Si tous les composants sélectionnés se trouvent dans le même conteneur, et que le premier composant que vous sélectionnez a un pourcentage de largeur ou de hauteur spécifié, tous les éléments sont définis selon ce pourcentage. Dans le cas contraire, tous les composants sont définis avec la même largeur ou hauteur de pixel.

Dimensionnement d'un composant en modifiant ses propriétés

- 1 En mode Création, sélectionnez le composant.

Pour définir la taille de plusieurs composants simultanément, appuyez sur la touche Ctrl en cliquant (ou Maj-clic sous Mac OS) sur ces composants.

- 2 Dans la vue Propriétés (Fenêtre > Propriétés), définissez la propriété `height` ou `width` du ou des composants sélectionnés.

La vue Propriétés propose trois vues pour l'affichage des propriétés d'un composant : une vue standard sous forme de formulaire, une vue par catégories sous forme de tableau et une vue alphabétique sous forme de tableau. Vous pouvez passer d'une vue à l'autre en cliquant sur les boutons correspondants dans la barre d'outils.

Utilisation de l'alignement pour positionner les composants

Lorsque vous faites glisser un composant visuellement dans un conteneur doté du positionnement absolu, le composant peut s'accrocher dans la bonne position selon l'endroit vers lequel vous le faites glisser par rapport aux composants existants. Les composants peuvent s'aligner à la verticale ou à l'horizontale.

Par défaut, les conteneurs Spark font appel à la propriété `BasicLayout` qui autorise le positionnement absolu. Pour certains conteneurs MX, vous pouvez également définir la propriété `layout` sur `absolute`. Pour plus d'informations, voir [Using Layout Containers](#).

Vous pouvez désactiver l'alignement pour un composant ou pour tous les composants.

Activation ou désactivation de l'accrochage

- ❖ Le fichier MXML étant ouvert en mode Création dans l'éditeur MXML, sélectionnez (ou désélectionnez) Création > Activer l'accrochage.

Activation ou désactivation de l'accrochage dans les Préférences

- 1 Ouvrez la boîte de dialogue Préférences.
- 2 Sélectionnez Flash Builder > Editeurs > Mode Création dans le panneau latéral de la boîte de dialogue Préférences.
- 3 Sélectionnez ou désélectionnez l'option Activer l'accrochage.

Accrochage de composants

Vous pouvez aligner visuellement les composants les uns par rapport aux autres dans un conteneur doté du positionnement absolu. Par défaut, les conteneurs Spark font appel à la propriété `BasicLayout` qui autorise le positionnement absolu. Pour certains conteneurs MX, vous pouvez également définir la propriété `layout` sur `absolute`.

Vous pouvez également centrer les composants dans un conteneur grâce à une présentation basée sur des contraintes. Pour plus d'informations, voir « [Définition de contraintes de présentation pour les composants](#) » à la page 207.

Accrochage de composants dans un conteneur doté du positionnement absolu

- 1 En mode Création de l'éditeur MXML, sélectionnez deux composants ou plus dans le conteneur.

Pour plus d'informations, voir « [Sélection de plusieurs composants dans un fichier MXML](#) » à la page 197.

- 2 Sélectionnez l'une des options d'alignement suivantes dans le menu Création.

Aligner à gauche Positionne tous les composants sélectionnés de manière que leur bord gauche s'aligne avec celui du premier composant que vous avez sélectionné.

Aligner les centres à la verticale Positionne tous les composants sélectionnés de manière que leur centre s'aligne à la verticale avec celui du premier composant que vous avez sélectionné.

Aligner à droite Positionne tous les composants sélectionnés de manière que leur bord droit s'aligne avec celui du premier composant que vous avez sélectionné.

Aligner en haut Positionne tous les objets sélectionnés de manière que leur bord supérieur s'aligne avec celui du premier composant que vous avez sélectionné.

Aligner les centres à l'horizontale Positionne tous les composants sélectionnés de manière que leur centre s'aligne à l'horizontale avec celui du premier composant que vous avez sélectionné.

Aligner en bas Positionne tous les composants sélectionnés de manière que leur bord inférieur s'aligne avec celui du premier composant que vous avez sélectionné.

Aligner les lignes de base Positionne tous les composants sélectionnés de manière que leur ligne de base de texte horizontal s'aligne avec celui du premier composant que vous avez sélectionné. Pour les composants dépourvus de ligne de base de texte (tels que HGroup), le bord inférieur est considéré comme la ligne de base.

Pour les objets dépourvus de contraintes de présentation, Flash Builder ajuste la propriété x pour changer l'alignement vertical et la propriété y pour modifier l'alignement horizontal.

Pour les objets dépourvus de contraintes de présentation, Flash Builder ajuste les contraintes gauche et droite pour changer l'alignement vertical et les contraintes de haut et bas pour modifier l'alignement horizontal. Seules les contraintes existantes sont modifiées ; aucune nouvelle contrainte n'est ajoutée.

Supposons, par exemple, que le composant A possède une contrainte gauche et aucune contrainte droite, et que le composant B soit doté d'une contrainte gauche et d'une droite. Si vous sélectionnez les composants A et B et que vous sélectionnez Création > Aligner les centres à la verticale, Flash Builder ajuste la contrainte gauche de l'objet A et les contraintes gauche et droite de l'objet B pour les aligner. La contrainte de droite non spécifiée de l'objet A reste non spécifiée.

Ajustement de composants

Vous pouvez ajuster la position de composants dans un conteneur doté du positionnement absolu en les déplaçant d'un ou de dix pixels à la fois en utilisant les touches de direction.

Ajustement des composants d'un pixel

- ❖ Sélectionnez un ou plusieurs composants en mode Création de l'éditeur MXML et appuyez sur l'une des touches de direction.

Ajustement des composants de dix pixels

- ❖ Sélectionnez un ou plusieurs composants en mode Création de l'éditeur MXML et appuyez sur l'une des touches de direction tout en maintenant la touche Maj enfoncée.



Maintenez la touche de direction enfoncée pour continuer à déplacer le composant.

Définition des propriétés des composants

Vous pouvez définir visuellement les propriétés de composants dans la zone de création ou dans la vue Propriétés.

Modification du texte affiché par un composant

- ❖ Pour éditer le texte affiché par un composant tel qu'un contrôle Label ou TextInput, double-cliquez sur le composant et saisissez vos modifications.

Changement du texte dans le champ ID

Lorsque vous modifiez du texte dans le champ ID, vous êtes invité à mettre à jour toutes les références de l'espace de travail avec le nouvel identifiant. Vous pouvez définir la suppression de cette boîte de dialogue à la page Mode Création des Préférences :

- 1 Ouvrez la fenêtre Préférences.
- 2 Sélectionnez Flash Builder > Editeurs > Mode Création.
- 3 Sélectionnez ou désélectionnez Toujours mettre à jour les références lors de la modification des ID dans la vue Propriétés.

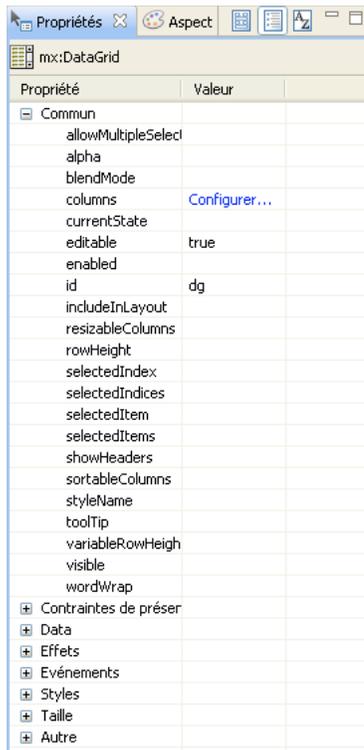
Définition des autres propriétés d'un composant

- ❖ Sélectionnez un composant et définissez-en les propriétés dans la vue du même nom (Fenêtre > Autres vues > Flash Builder > Propriétés).



Vue Propriétés d'un composant DataGrid

Pour définir les propriétés dans la Vue par catégories ou dans la Vue alphabétique, cliquez sur les boutons correspondant aux vues dans la barre d'outils.



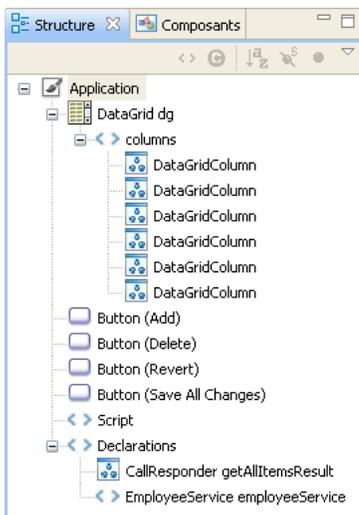
Vue par catégories pour un composant Button

Remarque : pour appliquer votre dernière modification, appuyez sur la touche Entrée ou de tabulation, ou cliquez en dehors de la vue.

Examen de la structure du code MXML

La vue Structure (Fenêtre > Structure) en mode Création permet d'examiner la structure de l'application et de sélectionner un ou plusieurs composants. Lorsque vous avez plusieurs états d'affichage, la vue Structure affiche la structure de l'état d'affichage actif.

- 1 Le fichier MXML étant ouvert en mode Création, sélectionnez la vue Structure.



- 2 Dans la vue Structure, sélectionnez un ou plusieurs composants.

Ces composants apparaissent également sélectionnés en mode Création de l'éditeur.

En mode Source, seul le premier composant sélectionné dans la vue Structure apparaît comme étant sélectionné dans l'éditeur.

Masquage des bordures d'un conteneur

Flash Builder affiche par défaut les bordures des conteneurs en mode Création de l'éditeur MXML. Vous pouvez toutefois les masquer.

- ❖ Sélectionnez Création > Afficher les bordures des conteneurs.

Cette commande est un bouton bascule. Sélectionnez-la à nouveau pour afficher les bordures.

Copie de composants dans d'autres fichiers MXML

Vous pouvez copier et coller visuellement des composants d'un fichier MXML vers un autre.

- 1 Vérifiez que les deux fichiers MXML sont ouverts en mode Création de l'éditeur MXML.
- 2 Sélectionnez des composants dans l'un des fichiers. Cliquez ensuite sur Editer > Copier.
- 3 Accédez à l'autre fichier, cliquez à l'intérieur d'un conteneur et sélectionnez Editer > Coller.

Suppression de composants

Vous pouvez supprimer des composants de l'interface utilisateur. Sélectionnez un composant, puis procédez de l'une des manières suivantes :

- Appuyez sur la touche de suppression de votre clavier.

- Dans le menu contextuel du composant, sélectionnez Supprimer.
- Dans le menu Editer de Flash Builder, sélectionnez Supprimer.

Utilisation de présentations basées sur des contraintes

Vous pouvez appliquer des contraintes à un composant pour en ajuster automatiquement la taille et la position dans le conteneur lorsque l'utilisateur redimensionne la fenêtre de l'application.

Les contraintes de présentation sont généralement définies en mode Création de l'éditeur MXML. Vous pouvez également définir des contraintes de présentation pour un composant en modifiant ses propriétés en mode Source.

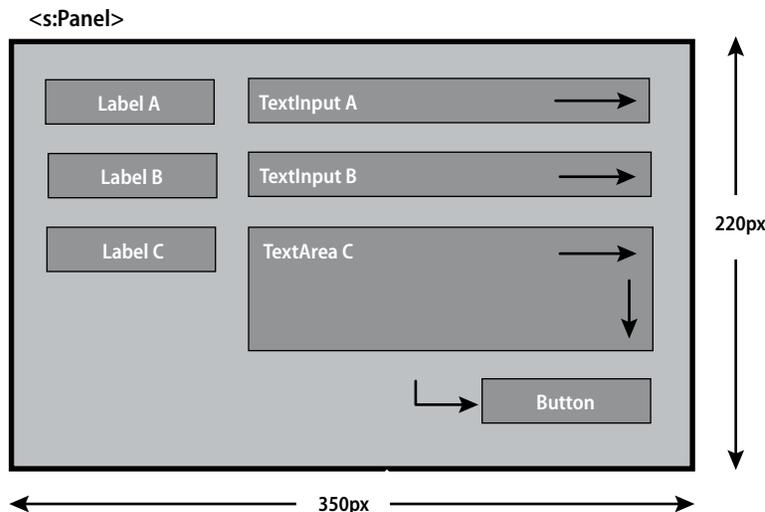
A propos des présentations basées sur des contraintes

Flex prend en charge les présentations basées sur des contraintes. Ce type de présentation est disponible avec les conteneurs qui prennent en charge le positionnement absolu. Pour les conteneurs Spark, le positionnement absolu est pris en charge par la présentation par défaut (BasicLayout).

L'application de présentations basées sur des contraintes se traduit pas l'ancrage d'un ou de plusieurs côtés d'un composant aux bords d'un conteneur ou de sa région de contrainte. Vous pouvez également définir le décalage du composant par rapport à l'ancrage. Lorsqu'un conteneur est redimensionné par l'utilisateur, la taille et la position de ses composants sont déterminées par les points d'ancrage définis.

Exemple de présentation basée sur des contraintes

Dans l'exemple suivant, tous les contrôles bénéficient d'un positionnement absolu dans un conteneur, soit en faisant glisser, soit en définissant les coordonnées x et y dans la vue Propriétés. Les flèches du schéma indiquent de quelle manière les contraintes, définies conformément aux points énumérés ci-dessous, influent sur le comportement des contrôles en cas de redimensionnement de la présentation.



Un certain nombre de contraintes de présentation sont appliquées aux contrôles pour garantir un ajustement correct de la présentation lorsque l'utilisateur redimensionne l'application :

- Label A, Label B et Label C sont ancrés sur les bords gauche et supérieur, de manière que les étiquettes restent en place lorsque l'utilisateur redimensionne la présentation.
- TextInput A et TextInput B sont ancrés sur les bords gauche et droit, pour pouvoir s'étirer ou se comprimer à l'horizontale lorsque l'utilisateur redimensionne la présentation.
- TextArea C est ancré sur les bords gauche et droit et supérieur et inférieur, afin de s'étirer ou de se comprimer à l'horizontale et à la verticale lorsque l'utilisateur redimensionne la présentation.
- Le contrôle Button est ancré aux bords droit et gauche, de façon que le contrôle conserve sa position par rapport au coin inférieur droit du conteneur lorsque l'utilisateur redimensionne la présentation.

Les contrôles TextInput A et TextInput B s'étirent à l'horizontale lorsque la présentation est agrandie. Le contrôle TextArea C s'étire à l'horizontale et à la verticale. Le contrôle Button se déplace vers le bas et vers la droite.

Présentations basées sur des contraintes et positionnement absolu

Pour créer une présentation basée sur des contraintes, utilisez un conteneur prenant en charge le positionnement absolu. Pour les conteneurs Spark, définissez la propriété `layout` sur `BasicLayout`. La propriété par défaut `BasicLayout` est appliquée à tout conteneur Spark pour lequel aucune propriété `layout` n'a été définie.

Pour les conteneurs MX, la présentation absolue ne peut être appliquée qu'à Application, Canvas et Panel. Pour le conteneur Canvas, absolue est la présentation par défaut. Pour les conteneurs MX Application et Panel, définissez la propriété `layout` sur `absolute` pour appliquer le positionnement absolu.

La propriété `layout="absolute"` des conteneurs MX remplace la règle de présentation du conteneur et vous permet de faire glisser les composants et de les positionner n'importe où dans le conteneur.

Utilisation de présentations basées sur des contraintes avancées

Vous pouvez définir des contraintes provoquant l'ancrage à des régions de contrainte horizontales et verticales. Les contraintes avancées sont utiles dans les situations suivantes :

- Les contrôles sont redimensionnés dynamiquement en fonction de leur contenu, par exemple lorsqu'ils affichent des chaînes localisées.
- Vous voulez plusieurs colonnes de taille identique ou devant respecter un pourcentage de largeur spécifique.

Remarque : les conteneurs `HGroup` et `VGroup` imbriqués permettent d'obtenir des résultats proches de ceux obtenus avec les contraintes avancées.

Pour les présentations basées sur des contraintes avancées, définissez des régions `ConstraintColumn` et `ConstraintRow`. Ancrez les composants aux bords ou au centre de ces régions. Les colonnes de contraintes sont disposées dans le conteneur de gauche à droite et les lignes de contraintes du haut vers le bas.

Les régions de contraintes peuvent être définies par des dimensions de pixels fixes (largeur ou hauteur) ou sous la forme d'un pourcentage d'espace dans le conteneur parent. L'ensemble de colonnes et de lignes de contraintes peut posséder un mélange de dimensions fixes et de pourcentage.

Les composants situés dans un conteneur parent peuvent être ancrés au conteneur, à des régions de contraintes ou à toute association de conteneurs et de régions de contraintes.

Utilisez l'éditeur MXML en mode Source pour spécifier des contraintes avancées.

Définition de contraintes de présentation pour les composants

Vous pouvez définir une présentation basée sur des contraintes pour les composants d'un conteneur doté de positionnement absolu.

Remarque : les contraintes de présentation définies par des colonnes et lignes de contraintes peuvent être spécifiées uniquement par l'édition du code source.

- 1 Assurez-vous que le fichier MXML contient un conteneur doté du positionnement absolu.
- 2 En mode Création de l'éditeur MXML, faites glisser un composant de la vue Composants vers le conteneur.
- 3 Positionnez le composant dans le conteneur en le déplaçant dans la zone de création ou en définissant ses propriétés *x* et *y* dans la vue Propriétés (Fenêtre > Propriétés).
- 4 Sélectionnez un composant dans la zone de création.
- 5 Dans la vue Propriétés, accédez à l'outil de contraintes dans la catégorie Présentation.
- 6 En vous servant du tableau suivant comme guide, cochez les cases de contraintes pour obtenir l'effet voulu lorsque l'utilisateur redimensionne l'application.

Effet	Contraintes
Conserver la position et la taille du composant	Aucun(e)
Déplacer le composant à l'horizontale	Gauche ou Droite
Déplacer le composant à la verticale	Haut ou Bas
Déplacer le composant à l'horizontale et à la verticale	Gauche et Haut ou Droite et Bas
Redimensionner le composant à l'horizontale	Gauche et Droite
Redimensionner le composant à la verticale	Haut et Bas
Redimensionner le composant à l'horizontale et à la verticale	Gauche et Droite et Haut et Bas
Centrer le composant à l'horizontale	Centre horizontal
Centrer le composant à la verticale	Centre vertical
Centrer le composant à l'horizontale et à la verticale	Centre horizontal et Centre vertical

- 7 Spécifiez la distance des contraintes par rapport aux bords du conteneur.

Par exemple, vous pouvez définir le composant de manière à conserver sa position à 90 pixels du bord gauche et à 60 pixels du bord droit. Si l'utilisateur redimensionne l'application, le composant s'étire ou se comprime pour conserver ces distances par rapport aux bords de la fenêtre de l'application.

Flash Builder exprime ces contraintes en code MXML comme suit :

```
<s:TextInput y="160" left="90" right="60"/>
```

Remarque : la valeur de *y* varie en fonction des valeurs définies pour les contraintes Gauche et Droite.

Création de gestionnaires d'événement

Les applications Flex sont orientées événements. Les composants de l'interface utilisateur réagissent à divers événements, par exemple lorsqu'un utilisateur clique sur un bouton ou lorsque l'initialisation d'un objet est terminée. Les gestionnaires d'événement, rédigés en code ActionScript, définissent la manière dont les composants réagissent à l'événement.

Remarque : vous pouvez également générer des gestionnaires d'événement pour des éléments non visibles, tels que *RemoteObject* et *HTTPService*.

Flash Builder assiste l'utilisateur par la génération des fonctions des gestionnaires d'événement pour un composant. Dans la fonction générée, rédigez le code qui définit le comportement du composant en réponse à l'événement.

L'assistance pour la génération des gestionnaires d'événement est accessible :

- à partir de la vue Propriétés ;
- à partir du menu contextuel d'un élément en mode Création de l'éditeur MXML ;
- à partir du menu contextuel d'un élément en mode Source de l'éditeur MXML.

A propos des gestionnaires d'événement générés

Lorsque Flash Builder génère une fonction de gestionnaire d'événement, il place le gestionnaire d'événement dans le premier bloc Script du fichier. La fonction est placée à la fin du bloc Script. Le gestionnaire d'événement généré a un accès protégé et accepte la sous-classe appropriée de la classe *Event* comme seul paramètre.

Flash Builder génère soit un nom unique pour le gestionnaire d'événement en fonction du nom de la classe du composant, soit un nom personnalisé pour le gestionnaire d'événement spécifié. Si vous ne précisez pas de nom personnalisé, le nom est généré selon le processus suivant :

- Si une propriété ID est définie, Flash Builder en dérive le nom.
- Si aucune propriété ID n'est définie pour le composant, Flash Builder génère un nom unique, basé sur le nom de la classe du composant.

Vous devez fournir le corps du gestionnaire d'événement. Le bloc de code suivant contient un gestionnaire d'événement généré pour un contrôle *Button*.

```
. . .
<fx:Script>
    <![CDATA[
        protected function myButton_clickHandler(event:MouseEvent):void
        {
            // TODO Auto-generated method stub
        }
    ]]>

</fx:Script>
<s:Button label="Button" id="myButton" click="myButton_clickHandler(event)"/>
. . .
```

Flash Builder désigne un événement par défaut pour chaque composant de l'interface utilisateur. Par exemple, l'événement par défaut d'un contrôle *Button* est l'événement *click*. Vous pouvez préciser le gestionnaire d'événement pour l'événement par défaut dans la Vue standard de la vue Propriétés. Pour spécifier les gestionnaires d'autres événements, sélectionnez Vue par catégories > Événements.

Vous pouvez également générer des gestionnaires d'événement en faisant appel à l'Assistant de contenu en mode Source.

Génération de gestionnaires d'événement à partir de la vue Propriétés

1 En mode Création, sélectionnez un élément, puis sélectionnez la Vue standard dans la vue Propriétés.

La zone Commun comporte un champ d'édition pour le gestionnaire d'événement par défaut.

2 Pour générer un gestionnaire pour l'événement par défaut :

a (Facultatif) Dans le champ de texte Lors d'un *Evénement*, saisissez un nom pour l'événement.

Dans le champ Lors d'un clic d'un composant Button, indiquez par exemple **MyButtonClick**.

Si vous ne spécifiez aucun nom, Flash Builder génère un nom unique pour l'événement.

 Lorsque vous précisez un nom pour le gestionnaire d'événement, vous avez la possibilité de spécifier un paramètre d'événement. Si vous ne le faites pas, Flash Builder génère le paramètre avec un type d'événement approprié.

b Cliquez sur l'icône représentant un éclair, puis sélectionnez Générer un gestionnaire d'événement.

L'éditeur bascule en mode Source et met en évidence le corps du gestionnaire d'événement généré. Saisissez l'implémentation de l'événement.

3 Pour générer un gestionnaire pour un événement d'un élément sélectionné :

a Dans la Vue par catégories, développez le nœud Evénements afin d'afficher tous les événements correspondant à l'élément.

b (Facultatif) Double-cliquez sur le nom de l'événement pour activer la zone de texte du nom du gestionnaire d'événement et saisissez le nom.

c Cliquez sur l'icône dans le champ Valeur pour créer le gestionnaire d'événement.

L'éditeur bascule en mode Source et met en évidence le corps du gestionnaire d'événement généré. Saisissez l'implémentation de l'événement.

Génération de gestionnaires d'événement à l'aide du menu contextuel d'un élément

1 Dans la vue Création, ouvrez le menu contextuel d'un élément.

2 Effectuez l'une des actions suivantes :

- Sélectionnez l'événement par défaut pour l'élément.

Par exemple, pour un élément Button, sélectionnez Génération du gestionnaire de clics.

- Choisissez Afficher tous les événements pour ouvrir la liste d'événements dans la vue Propriétés.

Précisez un gestionnaire d'événement dans la vue Propriétés.

L'éditeur bascule en mode Source et met en évidence le corps du gestionnaire d'événement généré. Saisissez l'implémentation de l'événement.

Génération de gestionnaires d'événement à l'aide de l'assistant de contenu

1 Dans un bloc MXML en mode Source, créez un composant, mais sans préciser d'événements.

2 Activez l'assistant de contenu pour les propriétés d'un composant en insérant un espace après le nom de la classe.

3 Dans la liste de propriétés sélectionnées, sélectionnez un événement (doubleClick, par exemple).

4 Appuyez sur Ctrl+Espace et sélectionnez Générer un gestionnaire d'événement.

Flash Builder génère un nom unique pour le gestionnaire d'événement et place ce dernier dans le bloc Script.

Remarque : si vous avez spécifié un nom personnalisé pour le gestionnaire d'événement, Flash Builder ne peut pas générer le gestionnaire. Pour utiliser un nom personnalisé, générez d'abord un gestionnaire d'événement, puis modifiez-en le nom aussi bien dans la propriété event que dans le gestionnaire généré.

Application de thèmes

Les thèmes permettent de personnaliser l'aspect de vos applications. Flash Builder offre un choix de plusieurs thèmes. Vous pouvez également importer d'autres thèmes ou créer les vôtres.

Flash Builder comporte des thèmes Spark et Halo. Le thème par défaut des composants Flex 4 est Spark et celui des composants Flex 3 est Halo.

Pour plus d'informations sur la prise en charge des thèmes dans Flex, voir [About themes](#).

Spécification d'un thème

Spécifiez des thèmes par projet. Une fois que vous avez précisé un thème pour un projet, toutes les applications du projet le partagent.

- 1 Ouvrez la boîte de dialogue Sélectionner le thème du projet à partir de la vue Création ou de la vue Source de l'éditeur MXML :
 - (Vue Création) Sélectionnez le panneau Aspect. Sélectionnez ensuite le Thème actuel.
 - (Vue Source) Dans la barre de menus de Flash Builder, sélectionnez **Projet > Propriétés > Thème Flex**.
- 2 Sélectionnez un thème puis cliquez sur OK.

Importation de thèmes

Flash Builder vous permet d'importer des thèmes. Les fichiers correspondant à un thème doivent être inclus dans un dossier. Tous les fichiers requis pour un thème Flex doivent être présents.

Le nom du thème est déterminé par l'élément de nom du fichier `metadata.xml` situé dans le dossier des thèmes. Si l'élément de nom n'est pas précisé ou si `metadata.xml` est absent, le thème prend le nom du dossier des thèmes.

Pour plus d'informations sur le format requis pour les thèmes Flex, voir « [Création de thèmes](#) » à la page 211.

Les thèmes de Flash Builder peuvent prendre les formats suivants :

- Fichier de thème ZIP

Extrayez le contenu du fichier ZIP avant d'importer le thème. Le contenu extrait doit englober tous les fichiers requis.
- Fichier CSS ou SWC pour un thème

Le fichier CSS ou SWC doit se trouver dans un dossier contenant tous les fichiers requis pour un thème Flex. Lorsque vous importez un thème à l'aide de Flash Builder, vous sélectionnez le fichier CSS ou SWC correspondant à ce thème.
- Fichier MXP

Adobe Extension Manager CS4 vous permet d'assembler des fichiers correspondant à des thèmes Flex dans un fichier MXP. Vous pouvez ensuite importer le thème dans Flash Builder par le biais de l'Extension Manager.

Pour plus d'informations sur l'assemblage de thèmes dans un fichier MXP, voir « [Création d'un fichier d'extension \(fichier MXP\) pour un thème Flex](#) » à la page 213.

Importation de thèmes Flex à l'aide de Flash Builder

- 1 Ouvrez la boîte de dialogue Sélectionner le thème du projet à partir de la vue Création ou de la vue Source de l'éditeur MXML :
 - (Vue Création) Sélectionnez le panneau Aspect. Sélectionnez ensuite le Thème actuel.
 - (Vue Source) Dans la barre de menus de Flash Builder, sélectionnez Projet > Propriétés > Thème Flex.
- 2 Sélectionnez Importer un thème, accédez au dossier contenant le thème à importer, sélectionnez le fichier CSS ou SWC, puis cliquez sur OK.

Importation de thèmes Flex assemblés dans un fichier MXP

- 1 Importez Adobe Flash® Builder™ 4 dans Adobe Extension Manager CS4 (pour autant que ce ne soit déjà fait) :
Dans Adobe Extension Manager, sélectionnez Fichier > Import Product.
- 2 Dans Adobe Extension Manager, sélectionnez Flash Builder 4.
- 3 Sélectionnez Fichier > Installer l'extension, accédez au fichier MXP correspondant au thème, puis cliquez sur Ouvrir.

Une fois que vous avez accepté la licence, Adobe Extension Manager installe le thème dans Flash Builder. Il est désormais disponible dans la boîte de dialogue Sélectionner le thème du projet de Flash Builder.

Remarque : vous pouvez également double-cliquer sur le fichier MXP pour appeler Adobe Extension Manager, qui installe alors le thème automatiquement.

Téléchargement de thèmes

Vous pouvez télécharger des thèmes que vous pourrez ensuite importer dans Flash Builder.

Téléchargement de thèmes Flex

- 1 Ouvrez la boîte de dialogue Sélectionner le thème du projet à partir de la vue Création ou de la vue Source de l'éditeur MXML :
 - (Vue Création) Sélectionnez le panneau Aspect. Sélectionnez ensuite le Thème actuel.
 - (Vue Source) Dans la barre de menus de Flash Builder, sélectionnez Projet > Propriétés > Thème Flex.
- 2 Sélectionnez Découvrir d'autres thèmes.

Flash Builder ouvre votre navigateur Web par défaut à une page contenant des thèmes à télécharger. Vous pouvez également accéder à tout autre site contenant des thèmes Flex téléchargeables.

- 3 Sélectionnez un thème Flex à télécharger.

Une fois le thème téléchargé, vous pouvez l'importer, comme décrit dans « [Importation de thèmes](#) » à la page 210.

Création de thèmes

Vous pouvez créer vos propres thèmes et les importer dans Flash Builder. Un thème Flex contient généralement les fichiers suivants :

- Des fichiers SWC, SWF, CSS, PNG et JPEG ainsi que d'autres fichiers composant le thème.

Les fichiers composant le thème peuvent varier, mais ils doivent comprendre un fichier SWC ou CSS.

- Un fichier `preview.jpg`.

Il s'agit du fichier image d'aperçu pour le thème. Si le dossier de thème ne contient pas `preview.jpg`, Flash Builder fait appel à une image d'aperçu par défaut pour le thème.

- Un fichier `metadata.xml`.

Ce fichier contient des informations relatives au thème, notamment les versions du SDK avec lesquelles le thème est compatible. Si le dossier de thème ne contient pas ce fichier, Flash Builder en crée un lors de l'importation du thème.

Un thème est généralement assemblé dans un fichier ZIP, mais celui-ci doit être extrait avant l'importation du thème dans Flash Builder. Vous pouvez aussi assembler les fichiers de thèmes dans un fichier Adobe Extension Manager (MXP) et utiliser Adobe Extension Manager pour importer le thème dans Flash Builder.

Pour plus d'informations, voir [About themes](#).

Fichier `metadata.xml`

Le tableau suivant répertorie les éléments pouvant être inclus dans le fichier `metadata.xml`.

Nom d'élément	Description
Name	Nom du thème tel qu'il apparaît dans Flash Builder. Lors de l'importation d'un thème avec Flash Builder, vous pouvez remplacer le nom spécifié dans le fichier <code>metadata.xml</code> .
Category	Auteur du thème. Catégorie sous laquelle le thème est affiché dans Flash Builder.
sdk	Précise les versions du SDK Flex avec lesquelles le thème est compatible. C'est un élément parent de <code>minVersionInclusive</code> et <code>maxVersionExclusive</code> . Si l'élément <code>sdk</code> est absent, le thème est valide pour tous les SDK.
minVersionInclusive	Version la plus ancienne du SDK Flex avec laquelle ce thème est compatible. Si cette information est absente, ce thème est compatible avec toutes les versions antérieures du SDK Flex.
maxVersionExclusive	Version la plus récente du SDK Flex avec laquelle ce thème est compatible. Si cette information est absente, ce thème est compatible avec toutes les versions ultérieures du SDK Flex.
mainFile	Fichier de niveau supérieur pour l'implémentation du thème. Ce fichier peut référencer d'autres fichiers dans le thème. Par exemple, un fichier CSS pourrait référencer un fichier SWC ou SWF. L'argument <code>-theme</code> du compilateur référence le fichier spécifié.

L'exemple suivant illustre un fichier `metadata.xml` standard correspondant à un thème créé par la société ABC.

```
<theme>
  <name>WindowsLookAlike</name>
  <category>ABC</category>
  <sdk>
    <minVersionInclusive>2.0.1</minVersionInclusive>
    <maxVersionExclusive>4.0.0</maxVersionExclusive>
  </sdk>
  <mainFile>WindowsLookAlike.css</mainFile>
</theme>
```

Selon le fichier `metadata.xml`, le thème est compatible avec le SDK Flex 2.0.1. Il est aussi compatible avec des SDK jusqu'à la version Flex 4.0.0 (non incluse). Lorsque ce thème est sélectionné, `WindowsLookAlike.css` est le fichier ajouté à l'argument de compilateur `-themes`.

Création d'un fichier d'extension (fichier MXP) pour un thème Flex

Adobe Extension Manager CS4 vous permet de créer un fichier d'extension (fichier MXP) pour un thème Flex. Vous pouvez ensuite importer le fichier MXP dans Flash Builder par le biais d'Adobe Extension Manager CS4.

Placez tous vos fichiers de thème dans un dossier hiérarchique, puis créez un fichier d'installation d'extensions (fichier MXI), qu'Extension Manager utilisera pour créer le fichier MXP. Pour plus d'informations sur le format d'un fichier MXI, reportez-vous au document [Extension File Format](#).

Lors de la création du fichier MXI, vous devez préciser un chemin de destination pour chacun des fichiers du thème. Les chemins sont au format suivant :

```
$flexbuilder/<Theme Name>
```

- `$flexbuilder` est défini dans le fichier de configuration Flash Builder, `XManConfig.xml`. Extension Manager développe `$flexbuilder` selon cette définition. `XManConfig.xml` se trouve à l'emplacement suivant dans votre système de fichiers :

```
/<Install Dir>/Flash Builder 4/configuration/XManConfig.xml
```

- `<Theme Name>` est le nom du dossier destiné à contenir le thème Flex.

Création d'un fichier d'extension MXP pour un thème Flex

- 1 Placez tous les fichiers correspondant au thème, notamment le fichier MXI, dans un dossier hiérarchique.
- 2 Dans Extension Manager, sélectionnez Fichier > Empaqueter l'extension.
- 3 Accédez au fichier d'installation d'extension souhaité et sélectionnez-le.
- 4 Recherchez un emplacement pour le fichier de package et nommez-le avec l'extension `.mxp`.

Vous pouvez ensuite tester le fichier d'extension en l'installant par le biais d'Extension Manager.

Ajout de thèmes

Vous pouvez définir plusieurs fichiers de thèmes pour une application. En l'absence de chevauchement de styles, les deux thèmes sont appliqués dans leur intégralité. D'autres aspects doivent également être pris en considération lors de l'ajout de thèmes. L'un d'eux est l'ordre des fichiers de thèmes.

Pour ajouter des thèmes, utilisez le compilateur de ligne de commande `mxmlc` avec l'option de compilateur `theme` afin de spécifier le chemin d'accès aux fichiers de thèmes.

`Using themes` fournit des informations sur la définition des arguments du compilateur et sur l'ordre des fichiers de thèmes.

Application de styles

Les styles modifient l'aspect de l'application par la définition de valeurs pour les paramètres visuels des composants. Vous pouvez définir des styles s'appliquant à tous les composants d'une application, à des composants individuels ou à des groupes de composants référencés par un sélecteur de style.

Vous pouvez par exemple définir les styles suivants :

- Texte
Famille, taille, couleur et épaisseur de la police ainsi que d'autres paramètres d'affichage (gras, italique et souligné, par exemple).
- Bordure
Épaisseur, couleur, couleur au survol, style (pleine, incrusté, sortant, aucune), rayon de l'angle et autres.
- Couleur
Couleur de remplissage et alpha.

Remarque : *les styles disponibles varient en fonction des composants.*

Vous pouvez définir des propriétés de style de manière intraligne sur une balise MXML ou séparément à l'aide de code CSS. Le code CSS peut être placé à l'intérieur de balises `<fx:style>` dans une application ou dans un fichier CSS distinct.

Lorsque vous appliquez des styles intralignes à des composants, vous pouvez convertir les styles des composants en règle CSS dans une feuille de style externe. Vous pouvez utiliser l'éditeur CSS pour modifier les fichiers CSS.

Vous pouvez également convertir en styles les habillages appliqués à un composant.

Accédez au mode Création de l'éditeur MXML pour appliquer des styles à une application ou à des composants spécifiques d'une application. Le mode Création vous permet en outre de convertir les styles ou les habillages appliqués en feuilles de style CSS.

Comparaison des styles et des habillages

L'application d'un habillage consiste à changer l'aspect d'un composant par la modification ou le remplacement de ses éléments visuels. Ces éléments peuvent être constitués d'images bitmap, de fichiers SWF ou de fichiers de classe contenant des méthodes de dessin définissant des images vectorielles. Les habillages peuvent définir l'aspect des différents états d'un composant. Vous pouvez par exemple définir l'aspect des états up, down, over et disabled d'un composant Button.

Flash Builder vous permet de convertir l'habillage d'un composant en styles CSS.

Application de styles

Accédez à la vue Aspect pour définir les styles valables pour l'ensemble de l'application. Flash Builder enregistre le style défini dans cette vue en tant que sélecteur de style CSS global.

1 En mode Création de l'éditeur MXML, ouvrez une application MXML contenant plusieurs composants.

2 Dans la vue Aspect, spécifiez des styles globaux pour l'application.

Si la vue Aspect n'est pas ouverte, vous pouvez y accéder en sélectionnant Fenêtre > Autres vues > Flash Builder > Aspect.

Une fois les styles appliqués, Flash Builder crée un sélecteur de style CSS global pour le style spécifié.

S'il s'agit du premier style référencé dans l'application, Flash Builder crée un nouveau fichier CSS et le référence dans le fichier de l'application MXML.

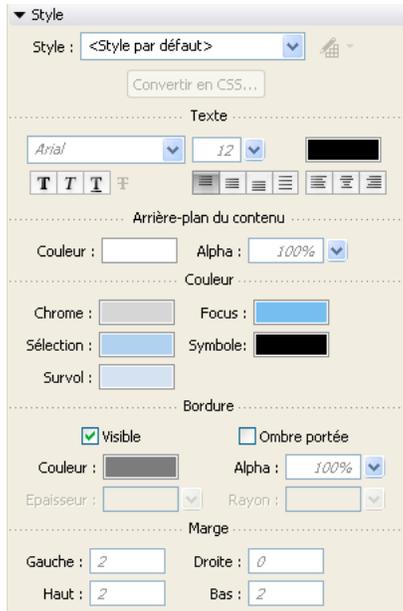
Si l'application référence déjà un fichier CSS ou un bloc `<fx:style>`, Flash Builder met à jour le fichier CSS avec le sélecteur de style global.

Application de styles intralignes à un composant

Vous pouvez définir des styles intralignes pour les composants dans la vue Propriétés.

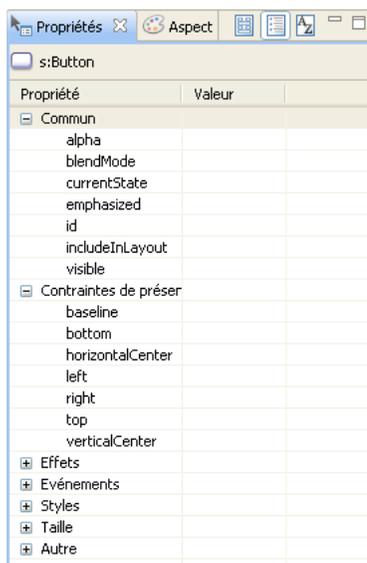
- 1 En mode Création de l'éditeur MXML, ouvrez une application MXML contenant plusieurs composants.
- 2 Sélectionnez un composant et spécifiez les valeurs des propriétés de style dans la zone Style de la vue Propriétés.

La zone Style change, en fonction du composant sélectionné.



Vue Propriétés affichant les styles d'un DataGrid

- 3 Dans la vue Propriétés, sélectionnez la Vue par catégories pour répertorier tous les styles qui peuvent être appliqués au composant sélectionné.



Remarque : dans Flex, les noms de style composés peuvent être orthographiés à la manière des identifiants ActionScript (fontFamily, par exemple) ou des styles HTML (font-family, par exemple).

- 4 Une fois un style spécifié, basculez en mode Source pour afficher le code intraligne généré qui applique le style.

Application d'un style externe ou intégré

Vous pouvez intégrer des styles CSS dans un fichier d'application MXML ou référencer un fichier CSS externe. L'exemple suivant illustre du code CSS appliquant des styles à tous les composants Spark Button d'une application. Ce code génère également le sélecteur `myStyle` qui peut être appliqué à n'importe quel composant :

```
@namespace s "library://ns.adobe.com/flex/spark";
@namespace mx "library://ns.adobe.com/flex/mx";

s|Button { fontSize: 16pt; color: Red } /* type selector */
.myStyle { color: Red } /* class selector */
```

Pour les styles appliqués à des composants (par exemple `s|Button`), le sélecteur pour composants doit spécifier un espace de noms. Dans cet exemple, `s|Button` définit un style qui est automatiquement appliqué à tous les composants Spark Button.

Utilisez la notation à points CSS pour créer un sélecteur pouvant être appliqué à n'importe quel composant. Dans cet exemple, `.myStyle` ne doit pas déclarer d'espace de noms et peut être appliqué à n'importe quel composant.

Flex définit un cadre précis pour la création et l'application de styles. Pour plus d'informations, voir *Using styles in Flex*.

Application de styles

- 1 En mode Création de l'éditeur MXML, créez un fichier d'application MXML contenant plusieurs composants Spark Button et un composant CheckBox.
- 2 (Styles intégrés) En mode Source, ajoutez le code suivant à l'application :

```
<fx:Style>
  @namespace s "library://ns.adobe.com/flex/spark";
  @namespace mx "library://ns.adobe.com/flex/halo";

  s|Button { fontSize: 16pt; color: Red } /* type selector */
  .myStyle { fontSize: 16pt; color: Blue } /* class selector */
</fx:Style>
```

- 3 (Feuilles de style externes) Créez une feuille de style qui implémente les sélecteurs `s|Button` et `.myStyle` à l'étape 1. En mode Source, référez le fichier à partir du fichier d'application MXML :

```
<fx:Style source="styles.css"/>
```

- 4 Basculez en mode Création. Vous remarquerez que le style a été appliqué aux composants Spark Button.
- 5 Sélectionnez le composant CheckBox. Dans la vue Propriétés, sélectionnez ensuite Styles > `myStyle`.

Vous remarquerez que le style `myStyle` a été appliqué au composant CheckBox.

Conversion en CSS

Vous pouvez convertir les styles intralignes et les habillages de composants en styles CSS. La procédure de conversion vous permet de spécifier si les styles doivent être appliqués de manière globale ou uniquement à un composant spécifique. Vous pouvez également définir un sélecteur de style CSS pour les styles générés.

- 1 En mode Création de l'éditeur MXML, sélectionnez un composant dans la zone de création. Choisissez un composant intégrant des styles intralignes ou spécifiant une propriété `skinClass`.
Pour plus d'informations sur la spécification d'une propriété `skinClass` pour un composant, voir « [Génération et modification d'habillages pour des composants Spark](#) » à la page 219.
- 2 Dans la vue Propriétés, cliquez sur Convertir en CSS.
- 3 Si l'espace de travail contient plusieurs projets ouverts, sélectionnez/désélectionnez les fichiers de ressources à enregistrer dans la boîte de dialogue d'enregistrement de ressources. Cliquez ensuite sur OK.
- 4 Dans la boîte de dialogue Nouvelle règle de style, sélectionnez le fichier .css ou cliquez sur Nouveau pour en créer un.
- 5 Spécifiez le Type de sélecteur en sélectionnant l'une des options suivantes :
 - Tous les composants
Le style est global et est appliqué à tous les composants de l'application.
 - Tous les composants ayant le style
Les composants spécifient ce sélecteur de style en utilisant l'attribut `styleName`. Si vous choisissez cette option, spécifiez un nom pour le sélecteur.
 - Composant spécifique
Le style ne s'applique qu'au composant sélectionné.
 - Composant spécifique ayant le style
Le style ne s'applique qu'au composant sélectionné et référence le style d'après le nom du sélecteur de type. Si vous choisissez cette option, spécifiez un nom pour le sélecteur de type.
- 6 Cliquez sur OK.
Flash Builder génère ou met à jour le fichier CSS spécifié. Il modifie également le code source de l'application de manière à référencer le sélecteur de type dans le fichier CSS.
Il supprime les références au style intraligne ou à la propriété `skinClass` pour le composant.

Modification de la règle de style

Lorsque des styles CSS externes sont déjà appliqués à un composant, vous pouvez passer rapidement du composant à l'édition de ces styles.

- 1 Sélectionnez un composant.
- 2 Dans la vue Propriétés, cliquez sur le bouton Modifier la règle de style, situé à droite de la liste déroulante Style, puis sélectionnez le style à modifier.
Le fichier CSS s'ouvre en mode Création dans l'éditeur CSS. La vue Propriétés vous permet d'apporter des modifications supplémentaires. Vous pouvez également modifier le fichier CSS en mode Source.

Création de fichiers CSS

Utilisez l'assistant Nouveau fichier CSS pour créer des fichiers CSS pour un projet. Cet assistant crée un fichier vide que vous pouvez utiliser pour définir vos styles CSS.

Par défaut, Flash Builder ajoute les espaces de noms par défaut pour les styles Spark et MX.

Pour créer un fichier CSS vide :

1 Dans le menu Flash Builder, sélectionnez Fichier > Nouveau fichier CSS.

2 Spécifiez un dossier source.

Le dossier source peut se trouver dans le projet actuel ou dans un autre projet.

3 Spécifiez un package pour le fichier.

Sélectionnez-le dans les packages disponibles dans le projet. Si vous souhaitez placer le fichier dans un nouveau package, créez d'abord ce dernier. Sélectionnez Fichier > Nouveau package.

4 Spécifiez un nom pour le fichier.

5 Cliquez sur Terminer.

Flash Builder utilise des modèles qui définissent le contenu des nouveaux fichiers. Vous avez la possibilité de personnaliser ces modèles. Voir « [Personnalisation des modèles de fichier](#) » à la page 127.

Utilisation de l'éditeur CSS

Flash Builder comporte un éditeur CSS que vous pouvez utiliser pour créer et modifier des feuilles de style pour votre application. L'éditeur CSS est disponible uniquement en mode Source.

A la création d'une nouvelle feuille de style, Flash Builder fournit les déclarations suivantes pour les espaces de noms Spark et MX :

```
@namespace s "library://ns.adobe.com/flex/spark";  
@namespace mx "library://ns.adobe.com/flex/mx";
```

Certains composants Spark et MX partagent le même nom local. Prenons l'exemple d'un composant Spark Button (dans le package `spark.components.*`) et d'un composant MX Button (dans le package `mx.controls.*`). Pour les distinguer l'un de l'autre, vous pouvez spécifier des espaces de noms s'appliquant aux types dans le fichier CSS.

Si vous n'utilisez pas de sélecteurs de type dans vos feuilles de style, vous n'avez pas besoin de déclarer des espaces de noms. Pour plus d'informations (illustrées par des exemples), voir [About namespaces in CSS](#).

Remarque : les styles sont traités différemment pour les applications utilisant le SDK Flex 3. Si votre projet Flex repose sur le SDK Flex 3, l'éditeur CSS applique le comportement implémenté avec Flex Builder 3. Pour plus d'informations sur l'utilisation de l'éditeur CSS avec les applications reposant sur le SDK Flex 3, voir la [documentation de Flex Builder 3](#).

Modification de l'interface utilisateur à l'aide des habillages

Les classes d'habillage modifient l'aspect des contrôles dans une interface utilisateur. La création, la modification et l'importation d'habillages varient en fonction des composants (Spark ou MX).

A propos des habillages Spark

Les habillages Spark contrôlent tous les éléments visuels d'un composant, dont sa présentation. Ils peuvent contenir plusieurs éléments, tels que des éléments graphiques, du texte, des images et des transitions. Les habillages Spark prennent en charge les états. Vous pouvez utiliser un habillage pour définir l'aspect d'un composant pour chacun de ses états. Les habillages spécifient généralement une taille minimale pour le composant. Pour plus d'informations sur l'implémentation d'habillages Spark dans Flex, voir [About Spark skinning](#).

Flash Builder vous permet de générer et de modifier des habillages pour les composants Spark. En générant un habillage, Flash Builder crée une classe d'habillage en MXML. Accédez à l'éditeur MXML pour modifier l'aspect défini par l'habillage. Certaines modifications peuvent être effectuées en mode Création ; d'autres nécessiteront le passage en mode Source. Voir « [Génération et modification d'habillages pour des composants Spark](#) » à la page 219.

A propos des habillages pour les composants MX

Les habillages des composants MX sont soit des graphiques bitmap, soit des graphiques vectoriels. Le graphique bitmap (ou habillage graphique) se compose de pixels individuels formant une image. Le graphique vectoriel (ou habillage par programmation) se compose d'une série de définitions de ligne définissant les points de début et de fin, l'épaisseur, la couleur et d'autres caractéristiques de la ligne requises par Adobe® Flash® Player pour la dessiner. Pour plus d'informations sur l'implémentation des habillages pour les composants MX dans Flex, voir [About MX component skinning](#).

Flash Builder vous permet d'importer des illustrations d'habillage pour des composants MX. Voir « [Importation d'illustrations d'habillage pour les composants MX](#) » à la page 223.

Le package `mx.skins.spark` définit les habillages Spark pour les composants MX.

Génération et modification d'habillages pour des composants Spark

Flash Builder vous permet de générer et de modifier des habillages pour les composants Spark. Lorsque vous générez un habillage, Flash Builder utilise un habillage du thème pour un projet. Le thème par défaut d'un projet est Spark. Vous pouvez modifier le thème d'un projet dans la vue Aspect. Voir « [Application de thèmes](#) » à la page 210.

Lors de la génération d'un habillage pour un composant, Flash Builder crée un fichier MXML qui implémente la classe Skin pour le composant. Vous pouvez spécifier si vous souhaitez générer la classe Skin en tant que copie d'un habillage existant ou générer un fichier de classe Skin vide.

Accédez aux modes Création et Source de l'éditeur MXML pour modifier l'habillage. En mode Création, sélectionnez les éléments d'habillage à modifier dans la vue Structure. Parcourez les états d'un composant dans la vue Etats. Certaines parties d'un habillage ne sont pas modifiables en mode Création. Utilisez le mode Source pour modifier des parties d'habillage qui ne sont pas disponibles en mode Création.

Certains composants contiennent des sous-composants. Par exemple, un composant HSlider contient des composants Button qui définissent le curseur et le rail de HSlider. Les sous-composants sont habillables uniquement en mode Source.

Etats des composants, parties d'habillage et composants hôte

Les habillages définissent l'aspect de chaque état d'un composant. Un bouton Spark possède par exemple quatre états : up, over, down et disabled. La génération d'un habillage pour un bouton Spark permet de spécifier l'aspect de chacun de ces états.

Chaque composant possède des parties auxquelles il est possible d'appliquer un style. Par exemple, pour un composant Button, vous pouvez choisir la couleur de remplissage, les attributs de texte de l'étiquette et les composants Rect formant la bordure du composant.

Lors de la création d'habillages pour un composant, Flash Builder permet de spécifier un composant hôte sur lequel l'habillage généré est basé. Par défaut, le composant hôte est la classe de base du composant auquel l'habillage est appliqué. Cependant, vous pouvez sélectionner un composant hôte différent.

Remarque : la spécification d'un composant hôte pour une classe d'habillage est requise lors de la génération de classes d'habillage à l'aide de Flash Builder. Cependant, la création de classes d'habillage directement dans le code source ne nécessite pas de composant hôte.

Contrat d'habillage entre un habillage et son composant hôte

Le contrat d'habillage entre une classe d'habillage et une classe de composants définit les règles à suivre par chaque membre afin de pouvoir communiquer entre eux.

La classe d'habillage doit déclarer des états d'habillage et définir l'apparence des parties d'habillage. De plus, les classes d'habillage spécifient habituellement le composant hôte et, parfois, effectuent une liaison aux données définies sur le composant hôte.

La classe de composants déclare la classe d'habillage qu'elle utilise. Elle doit également identifier des états d'habillage et des parties d'habillage à l'aide de métadonnées. Si la classe d'habillage établit une liaison aux données du composant hôte, ce dernier doit définir ces données.

Le tableau suivant détaille les règles du contrat d'habillage.

	Classe d'habillage	Composant hôte	Obligatoire ?
Composant hôte	<code><fx:Metadata> [HostComponent("spark.components.Button")] </fx:Metadata></code>	Sans objet	Non
Etats d'habillage	<code><s:states> <s:State name="up"/> </s:states></code>	<code>[SkinStates("up"); public class Button { ... }]</code>	Oui
Parties d'habillage	<code><s:Button id="upButton"/></code>	<code>[SkinPart(required="false") public var upButton Button;</code>	Oui
Données	<code>text="{hostComponent.title}"</code>	<code>[Bindable] public var title:String;</code>	Non

Déclaration Skin

Dans Flash Builder, une déclaration Skin est le fichier qui implémente l'habillage pour un composant. Flex définit une déclaration Skin pour chaque composant visuel. Lors de la génération d'un nouvel habillage pour un composant, Flash Builder génère la déclaration Skin.

Vous pouvez afficher la déclaration Skin de composants que vous avez sélectionnés :

- 1 En mode Création de l'éditeur MXML, sélectionnez un composant Spark dans la zone de création.
- 2 Dans le menu contextuel du composant, sélectionnez Ouvrir une déclaration d'habillage.

La classe implémentant l'habillage s'ouvre dans le mode Source de l'éditeur.

S'il s'agit d'une classe que vous avez créée, vous pouvez modifier le fichier.

Vous pouvez également procéder comme suit pour ouvrir un fichier de déclaration Skin :

- Une fois le composant sélectionné, cliquez sur l'icône en regard du champ Habillage dans la section Style de la vue Propriétés.

- Sélectionnez un composant Spark. En mode Source, sélectionnez ensuite Naviguer > Ouvrir la déclaration d'habillage dans le menu Flash Builder.

Génération et modification d'un habillage pour un composant Spark Button (mode Création)

Cet exemple illustre la génération d'une classe d'habillage pour un composant Spark Button, puis sa modification en faisant appel à différentes vues de Flash Builder. Il se base sur un projet Flex utilisant le thème Spark par défaut.

1 Créez un fichier d'application. En mode Création de l'éditeur, ajoutez un composant Spark Button.

2 Dans le menu contextuel du composant Button, sélectionnez Créer un habillage.

La boîte de dialogue pour la création d'un habillage MXML s'ouvre.

Vous pouvez également procéder comme suit pour accéder à cette boîte de dialogue :

- Dans la section Style de la vue Propriétés, sélectionnez l'icône en regard du champ Habillage.
- Dans le menu Flash Builder, sélectionnez Nouveau > Habillage MXML.
- En mode Création de l'éditeur, sélectionnez Création > Créer un habillage.

3 Dans la boîte de dialogue de création d'un habillage MXML, spécifiez les éléments suivants :

- Dossier source et package de la déclaration Skin générée
- Nom
Nom de la classe Skin créée.
- Composant hôte
Pour modifier le composant par défaut, cliquez sur Parcourir et sélectionnez un composant hôte.
- (Recommandé) Sélectionnez Créer en tant que copie de et ne supprimez pas le code de style ActionScript.
Si vous êtes novice en matière d'habillage, utilisez une copie pour vous lancer dans la création d'une classe Skin. Modifiez le code de style ActionScript.
- (Utilisateurs avancés) Si vous savez comment créer des classes Skin, procédez de l'une des manières suivantes :
Supprimez le code de style ActionScript ou ne créez pas de copie d'une classe existante.
Si vous ne créez pas de copie d'une classe existante, Flash Builder génère un fichier de classe Skin vide comportant des commentaires pour vous guider.
Les étapes restantes de cette procédure supposent que vous avez suivi l'option recommandée pour la génération d'une classe Skin.

4 Cliquez sur Terminer.

Flash Builder génère un fichier de classe Skin et l'ouvre en mode Création dans l'éditeur MXML.

Le composant Button est sélectionné.

L'état Up du composant Button est sélectionné dans la vue Etats.

5 Pour chaque état du composant Button, modifiez les styles Texte, Arrière-plan du contenu et Couleur.

Pour ce faire, utilisez les outils de modification de la section Style de la vue Propriétés.

6 Ouvrez la vue Structure.

Notez que le nœud de niveau supérieur (SparkSkin) est sélectionné.

7 Dans la vue Structure, sélectionnez Rect shadow pour définir les styles de l'ombre du composant Button.

Notez que les outils de la section Style ne sont pas disponibles.

8 Passez en mode Source de l'éditeur.

Flash Builder met en évidence le composant Rect qui définit l'ombre du composant Button. Modifiez l'ombre du composant Button.

9 Enregistrez le fichier de classe Skin et le fichier d'application.

Le mode Création de l'éditeur MXML vous permet d'afficher l'habillage du bouton, en supposant que vous avez suivi l'option recommandée à l'étape 3. Si les styles ne s'affichent pas, sélectionnez l'icône Actualiser de la zone de création.

Notez que le fichier d'application a ajouté une référence à la classe Skin que vous avez créée.

10 Exécutez l'application pour voir comment l'habillage change en fonction des états Up, Over et Down du composant Button.

Création et modification d'habillages pour les composants Spark (mode Source)

Vous pouvez ouvrir la boîte de dialogue de création d'un habillage MXML directement en mode Source de l'éditeur. Par exemple, pour créer une classe skinClass pour un composant Spark Button, procédez comme suit.

1 En mode Source de l'éditeur, placez votre curseur dans une balise `<s:Button>` et saisissez ce qui suit :

```
<s:Button skinClass="
```

Après avoir saisi le premier guillemet pour le nom de la classe skinClass, un menu contextuel s'affiche.

2 Avec Create Skin en surbrillance dans les indicateurs de code, la touche Entrée ouvre la boîte de dialogue de création d'un habillage MXML.

Cette boîte de dialogue est identique à celle qui s'ouvre en mode Création.

Pour créer une classe skinClass, voir les instructions dans « [Génération et modification d'un habillage pour un composant Spark Button \(mode Création\)](#) » à la page 221.

3 Cliquez sur Terminer.

Flash Builder génère une nouvelle classe skinClass en fonction des sélections effectuées dans la boîte de dialogue de création d'un habillage MXML. L'éditeur passe en mode Source pour la nouvelle classe générée.

4 Modifiez la classe skinClass.

Enregistrez le fichier de classe et le fichier d'application.

Remarque : vous pouvez convertir la classe d'habillage générée au format CSS pour afficher les styles appliqués. Voir « [Conversion d'un habillage en style CSS](#) » à la page 222.

Conversion d'un habillage en style CSS

Flash Builder vous permet de convertir l'habillage d'un composant en style CSS. L'avantage d'une telle conversion réside dans la possibilité d'utiliser le style en tant que sélecteur de type pour tous les composants de cette classe. Sinon, définissez la propriété skinClass pour chaque composant.

La procédure suivante illustre la conversion de l'habillage d'un composant Spark Button en style CSS.

1 Générez et éditez l'habillage d'un composant Button.

2 En mode Création, sélectionnez le composant. Dans la zone Style de la vue Propriétés, cliquez sur Convertir en CSS.

3 Dans la boîte de dialogue Nouvelle règle de style, sélectionnez un fichier CSS pour le style ou créez-en un.

En l'absence de fichier CSS dans le projet que vous souhaitez utiliser, cliquez sur Nouveau pour en créer un.

4 Spécifiez le Type de sélecteur en sélectionnant l'une des options suivantes :

- Tous les composants

Le style est appliqué à tous les composants de l'application.

- Tous les composants ayant le style

Les composants spécifient ce sélecteur de style d'après son nom. Si vous choisissez cette option, spécifiez un nom pour le sélecteur de type.

- Composant spécifique

Le style ne s'applique qu'au composant sélectionné.

- Composant spécifique ayant le style

Le style ne s'applique qu'au composant sélectionné et référence le style d'après le nom du sélecteur de type. Si vous choisissez cette option, spécifiez un nom pour le sélecteur de type.

5 Après avoir spécifié un Type de sélecteur, cliquez sur OK.

Flash Builder génère ou met à jour le fichier CSS spécifié. Il modifie également le code source de l'application de manière à référencer le sélecteur de type dans le fichier CSS.

Il supprime les références à la propriété `skinClass` pour le composant.

Importation d'illustrations d'habillage pour les composants MX

L'assistant Importation d'illustrations d'habillage permet l'importation d'illustrations graphiques vectorielles ainsi que d'illustrations bitmap à partir des versions CS4 de Flash Professional, Fireworks, Illustrator et Photoshop (pour les illustrations bitmap, tout fichier .PNG, .JPG ou .GIF peut être utilisé). L'illustration peut alors servir d'habillage pour les composants Flex.

Remarque : Adobe propose un ensemble de modèles d'habillage destinés à faciliter la création d'habillages pour les composants Flex intégrés. Ces modèles, utilisés avec Flash, Fireworks, Illustrator ou Photoshop, permettent de créer l'illustration. Grâce à Flash, vous pouvez aussi créer des composants Flex personnalisés totalement fonctionnels. Pour plus d'informations, voir les articles sous [Importation de ressources Flash Professional dans Flex](#).

1 Sélectionnez Fichier > Importer > Illustrations d'habillage.

Dans la version plug-in, sélectionnez Fichier > Importer > Illustration.

2 Dans la boîte de dialogue Importation d'illustrations d'habillage :

- Choisissez un dossier de bitmaps ou un fichier SWC ou SWF à partir duquel importer des habillages ou cliquez sur Parcourir pour en localiser un. Les types de fichiers pris en charge sont les suivants :
 - Les fichiers SWF AS3 et SWC AS3 créés dans Adobe Flash Professional CS5
 - Les fichiers de graphiques vectoriels créés dans Adobe Illustrator® et exportés en tant que fichiers SWF pour Flash Player 8
 - Les fichiers graphiques bitmap aux formats PNG, GIF et JPG
- Sélectionnez un dossier dans lequel importer les habillages. Il doit s'agir du dossier source d'un projet Flex (vous pouvez également indiquer un sous-dossier du dossier source). Le dossier du projet Flex ouvert est sélectionné par défaut.
- Dans le champ Copier les illustrations dans le sous-dossier, le nom du dossier par défaut est dérivé du dossier ou des ressources en cours d'importation. Cliquez sur Parcourir pour choisir un emplacement différent.

- Dans le champ Créer des règles de style d'habillage dans, précisez un nom pour le fichier CSS qui contiendra les règles de style. Le nom par défaut est dérivé du nom du dossier de l'illustration ou du fichier Flash en cours d'importation.
 - Cochez la case Supprimer toutes les règles existantes dans le fichier si vous voulez que le fichier CSS spécifié soit remplacé lors de l'importation (plutôt que d'importer les habillages et de conserver d'autres définitions existantes dans le fichier CSS). Par défaut, la case est désélectionnée. Si le fichier CSS n'existe pas, elle est désactivée.
 - Dans le champ Appliquer les styles à l'application, la valeur par défaut est le fichier sélectionné dans le Navigateur Flex ou dans la vue d'éditeur active, ou le fichier de l'application principale du projet.
 - Cliquez sur Suivant.
- 3** Dans la boîte de dialogue Importation d'illustrations d'habillage suivante, sélectionnez les habillages à importer et précisez quel type de style CSS et propriété de partie d'habillage seront utilisés. Vous pouvez cocher les éléments un par un ou cliquer sur Sélectionner tout ou Désélectionner tout.
- Si les éléments n'ont pas de style valide ou de nom de propriété de partie d'habillage, ils ne sont pas cochés par défaut. Les exemples suivants illustrent la convention d'attribution de noms dans Flash Builder :
 - Button_upSkin
 - Button_glow_downSkin (correspond à la propriété downSkin de la règle de style Button.glow)
 - TabBar-tab_upSkin (la propriété upSkin correspond à la propriété tabStyleName de la règle de style TabBar)
 - MyCustomComponent_borderSkin
- Pour les composants personnalisés, l'élément est coché si le composant a été défini dans le projet vers lequel vous effectuez l'importation.
- Choisissez au besoin un style et une partie d'habillage dans les listes déroulantes de chaque colonne.
 - Cliquez sur Terminer.
- Un fichier CSS est créé et affiché dans la vue Source. Le fichier CSS sera attaché à l'application spécifiée dans l'assistant. Si vous importez un fichier SWC, il est ajouté automatiquement au chemin d'accès à la bibliothèque du projet.

Génération de rendus d'élément personnalisés

Les contrôles basés sur une liste Spark, tels que List et ComboBox, prennent en charge les rendus d'élément personnalisés. Vous pouvez également utiliser les rendus d'élément Spark avec certains contrôles MX, tels que MX DataGrid et MX Tree.

Utilisez des rendus d'élément personnalisés pour contrôler l'affichage d'un élément de données dans un conteneur DataGroup, SkinnableDataContainer ou dans une sous-classe de ces conteneurs. L'aspect défini par un rendu d'élément peut inclure la police, la couleur d'arrière-plan, la bordure et tout autre aspect visuel de l'élément de données. Un rendu d'élément peut également définir l'aspect d'un élément de données au moment où l'utilisateur interagit avec celui-ci. Par exemple, le rendu d'élément peut afficher l'élément de données d'une certaine manière lorsque l'utilisateur déplace la souris sur ce dernier. Il l'affiche différemment lorsque l'utilisateur clique sur l'élément de données.

Flash Builder permet de générer et de modifier des rendus d'élément. Il utilise l'un des modèles suivants pour générer les rendus d'élément :

- Composants Spark

Utilisez ce modèle pour les contrôles basés sur une liste Spark, tels que List et ComboBox.

- MX Advanced DataGrid
- MX DataGrid
- MX Tree

Vous pouvez ouvrir l'assistant de création d'un rendu d'élément MXML en mode Création et en mode Source de l'éditeur MXML. Dans l'assistant de création d'un rendu d'élément MXML, spécifiez un nom et un modèle pour le rendu d'élément. Flash Builder génère un fichier MXML qui implémente le rendu d'élément.

Les composants de l'application référencent le rendu d'élément généré à l'aide de la propriété `itemRenderer`.

Pour plus d'informations sur la création et l'utilisation des rendus d'élément, voir Custom Spark Item Renderers.

Génération et modification d'un rendu d'élément pour un composant MX Tree (mode Création)

Cet exemple génère un rendu d'élément pour un composant MX Tree et illustre comment combiner des vues Flash Builder pour le modifier. Il se base sur un projet Flex utilisant le thème Spark par défaut.

- 1 Créez un fichier d'application. En mode Création de l'éditeur, ajoutez un composant MX Tree à l'application.

Renseignez le composant Tree de données affichables à l'exécution de l'application.

- 2 Dans le menu contextuel du composant Tree, sélectionnez Créer un rendu d'élément.

La boîte de dialogue de création d'un rendu d'élément MXML s'ouvre.

Pour ouvrir cette boîte de dialogue, vous pouvez également procéder comme suit.

- Dans la section Commun de la vue Propriétés, sélectionnez l'icône en regard du champ Rendu d'élément.
- Dans le menu Flash Builder, sélectionnez Nouveau > Rendu d'élément MXML.

- 3 Dans la boîte de dialogue de création d'un rendu d'élément MXML, spécifiez ce qui suit :

- Dossier source et package pour la déclaration du rendu d'élément généré

- Nom

Nom de la classe de rendu d'élément créée.

- Modèle

Sélectionnez le modèle à utiliser lors de la génération du rendu d'élément.

- 4 Cliquez sur Terminer.

Flash Builder génère un fichier de classe ItemRenderer et l'ouvre en mode Création de l'éditeur MXML.

Le composant ItemRenderer est sélectionné.

L'état normal du composant Tree est sélectionné dans la vue Etats.

- 5 Modifiez l'aspect de chaque état du composant Tree dans la classe ItemRenderer générée.
 - a Ouvrez la vue Structure.

Notez que le nœud de niveau supérieur (MXTreeItemRenderer) est sélectionné.

Dans la section Style de la vue Propriétés, modifiez l'aspect des éléments d'arborescence.
 - b Dans la vue Structure, sélectionnez d'autres composants de MXTreeItemRenderer pour en modifier l'aspect.

Notez que les outils de la section Style ne sont pas toujours disponibles.

S'ils ne sont pas disponibles, utilisez le mode Source de l'éditeur pour définir l'aspect. Lorsque vous passez en mode Source, la source du composant sélectionné dans la vue Structure est mise en évidence.
- 6 Exécutez l'application pour découvrir comment la classe ItemRenderer modifie l'aspect du composant Tree.

Création et modification des rendus d'élément (mode Source)

Vous pouvez ouvrir la boîte de dialogue de création d'un rendu d'élément MXML directement en mode Source de l'éditeur. Par exemple, procédez comme suit pour créer un rendu d'élément pour un composant Spark List.

- 1 En mode Source de l'éditeur, placez votre curseur dans une balise `<s:List>` et saisissez ce qui suit :

```
<s:List itemRender="
```

Après avoir saisi le premier guillemet du nom de la classe du rendu d'élément, un menu contextuel s'affiche.
- 2 Double-cliquez sur Créer un rendu d'élément pour ouvrir la boîte de dialogue de création d'un rendu d'élément MXML. Cette dernière est identique à celle qui s'ouvre en mode Création.

Pour plus d'informations sur la création d'un rendu d'élément, voir les instructions dans « [Génération et modification d'un rendu d'élément pour un composant MX Tree \(mode Création\)](#) » à la page 225.
- 3 Cliquez sur Terminer.

Flash Builder génère un nouveau rendu d'élément en fonction des sélections effectuées dans la boîte de dialogue de création d'un rendu d'élément MXML. L'éditeur passe en mode Source pour la nouvelle classe générée.
- 4 Modifiez la classe du rendu d'élément.

Enregistrez le fichier de classe et le fichier d'application.

Déclaration ItemRenderer

Dans Flash Builder, une déclaration ItemRenderer est le fichier qui implémente le rendu ItemRenderer personnalisé pour un composant.

Vous pouvez afficher la déclaration du rendu ItemRenderer personnalisé pour les composants que vous avez sélectionnés :

- 1 En mode Création de l'éditeur MXML, sélectionnez un composant pour lequel vous avez implémenté un rendu d'élément personnalisé.
- 2 Dans le menu contextuel du composant, sélectionnez Ouvrir la déclaration d'un rendu d'élément.

La classe implémentant le rendu d'élément s'ouvre en mode Source de l'éditeur. Pour ouvrir la déclaration du rendu d'élément, vous pouvez également procéder comme suit :

 - Sélectionnez le composant en mode Création. Dans la section Commun de la vue Propriétés, sélectionnez l'icône en regard du champ Rendu d'élément.

- Sélectionnez le composant. En mode Source, sélectionnez ensuite Naviguer > Ouvrir la déclaration d'habillage dans le menu Flash Builder.

Actualisation du mode Création pour un rendu correct

Vous pouvez le cas échéant actualiser le mode Création des éditeurs MXML et CSS pour rendre correctement la présentation. Le rendu de la présentation peut en effet dans certains cas devenir obsolète, par exemple si vous modifiez un élément visuel dans un composant Flash dépendant (SWC). Les styles et habillages peuvent en outre ne pas être rendus correctement si Flash Builder doit régénérer le fichier.

- ❖ Cliquez sur le bouton Actualiser de la barre d'outils de l'éditeur.



Ajout d'états d'affichage et de transitions

Adobe® Flash® Builder™ permet de créer des applications dont l'apparence varie selon les tâches exécutées par l'utilisateur. Par exemple, l'état de base de l'application peut représenter la page d'accueil et inclure un logo, une barre latérale et un message de bienvenue. Lorsque l'utilisateur clique sur un bouton de la barre latérale, l'apparence (l'état) de l'application change de façon dynamique et la zone de contenu principale est remplacée par un formulaire de bon de commande sans modifier la position du logo et de la barre latérale.

Flex vous permet d'appliquer cette interaction aux états d'affichage et aux transitions. Un *état d'affichage* représente l'un des différents affichages que vous définissez pour une application ou un composant personnalisé. Une *transition* consiste en un ou plusieurs effets regroupés pour être lancés lorsqu'un état d'affichage est modifié. Une transition vise à améliorer le changement visuel d'un état à un autre.

A propos des états d'affichage et des transitions

Un *état d'affichage* représente l'une des différentes mises en forme que vous définissez pour une application ou un composant MXML unique. Vous créez une application ou un composant qui passe d'un affichage à un autre en fonction des actions de l'utilisateur. Vous pouvez utiliser les états d'affichage pour créer une interface que l'utilisateur peut personnaliser ou qui présente progressivement davantage d'informations au fur et à mesure que l'utilisateur exécute des tâches spécifiques.

Chaque application ou composant défini dans un fichier MXML affiche toujours au moins un état, l'*état de base*, représenté par la présentation par défaut du fichier. Vous pouvez utiliser un état de base comme un référentiel de contenus tels que des barres de navigation ou des logos partagés par tous les affichages d'une application ou d'un composant et garantir ainsi une présentation homogène.

Vous pouvez créer un état d'affichage en modifiant la présentation d'un état existant ou en créant une nouvelle présentation. Les modifications pouvant être apportées à un état existant sont l'édition, le déplacement, l'ajout ou la suppression de composants. La nouvelle présentation correspond à ce que l'utilisateur voit lorsqu'il change d'état.

Pour obtenir une présentation conceptuelle complète des états d'affichage, y compris des exemples, voir View states.

En règle générale, l'ajout de pages à une application Flex est différent de l'ajout de pages à une application de type HTML. Vous créez un fichier d'application MXML unique puis ajoutez différentes mises en forme modifiables lorsque l'application est exécutée. Tout comme vous pouvez utiliser des états d'affichage pour ces mises en forme, il est également possible d'utiliser le conteneur de navigateur ViewStack avec d'autres conteneurs de navigateur.

La modification des états d'affichage de l'application change également l'apparence de l'interface utilisateur. Par défaut, les composants semblent sauter d'un état d'affichage à un autre. Vous pouvez adoucir cet effet en ajoutant des transitions.

Une *transition* représente un ou plusieurs effets visuels appliqués de façon séquentielle ou simultanée lors du changement d'un état d'affichage. Par exemple, vous souhaitez redimensionner un composant pour inclure un nouveau composant lorsque l'application passe d'un état à un autre. Vous pouvez définir une transition qui réduit progressivement le premier composant tandis qu'un nouveau composant apparaît petit à petit à l'écran.

Prise en charge des états d'affichage Flex 3

Flash Builder prend en charge les états d'affichage implémentés dans Flex 3. Pour les projets basés sur le SDK Flex 3, l'éditeur MXML rétablit l'implémentation Flex Builder 3 pour ses deux modes (Création et Source). Pour plus d'informations sur la modification des états pour le SDK Flex 3, voir la [documentation de Flex Builder 3](#).

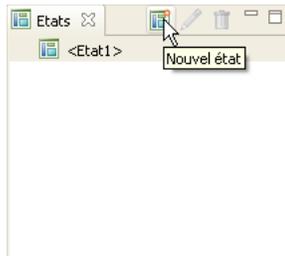
Création d'un état d'affichage

Par défaut, une application comporte un seul état d'affichage, que vous utilisez généralement comme état de base. Utilisez la vue Etats de Flash Builder pour ajouter des états supplémentaires et modifier la présentation et les composants de chaque état.

- 1 A l'aide des outils de présentation de Flash Builder, créez la présentation de l'état de base de l'application ou du composant.

Pour plus d'informations, voir « [Création d'une interface utilisateur avec Flash Builder](#) » à la page 188.

- 2 Dans la vue Etats (Fenêtre > Autres vues > Flash Builder > Etats), cliquez sur le bouton Nouvel état de la barre d'outils.



La boîte de dialogue Nouvel état apparaît.

- 3 Entrez le nom du nouvel état.
- 4 Indiquez si vous voulez créer une copie d'un état existant ou si vous préférez créer un nouvel état vide. Cliquez sur OK.
- 5 Utilisez les outils de présentation de Flash Builder pour modifier l'apparence de l'état.

Vous pouvez éditer, déplacer, ajouter ou supprimer des composants. Au fur et à mesure de vos modifications, les changements définissant le nouvel état sont intégrés au code MXML.

- 6 Définissez un gestionnaire d'événement permettant à l'utilisateur d'afficher le nouvel état.

Pour plus d'informations, voir « [Changement d'état d'affichage lors de l'exécution](#) » à la page 229.

Définition d'un état non de base comme état d'affichage de départ

Par défaut, une application affiche l'état de base au démarrage. Vous pouvez toutefois paramétrer l'affichage d'un autre état au démarrage de l'application.

- 1 Dans la vue Etats (Fenêtre > Etats), double-cliquez sur l'état d'affichage que vous voulez utiliser au démarrage.
- 2 Dans la boîte de dialogue Modifier les propriétés de l'état qui apparaît, sélectionnez l'option Définir comme état de départ, puis cliquez sur OK.

Définition de l'état d'affichage d'un composant

Si l'application comporte plusieurs états, vous pouvez définir l'état d'affichage d'un seul composant.

- 1 Dans la vue Création de l'éditeur MXML, sélectionnez un composant de la présentation.
- 2 Dans la vue Propriétés, utilisez le champ Etats pour sélectionner les états dans lesquels le composant est visible.

Changement d'état d'affichage lors de l'exécution

Lors de l'exécution de l'application, les utilisateurs doivent passer d'un état d'affichage à un autre. Vous pouvez définir des gestionnaires d'événement au niveau des contrôles des utilisateurs afin que ces derniers puissent changer d'état lors de l'exécution.

La méthode la plus simple consiste à affecter la propriété `currentState` à l'événement de clic d'un contrôle tel qu'un bouton ou un lien. La propriété `currentState` sélectionne le nom de l'état d'affichage que vous souhaitez consulter lorsque l'événement de clic se produit. Dans le code, spécifiez la propriété `currentState` de la façon suivante :

```
click="currentState='viewstatename' "
```

Si l'état d'affichage est défini pour un composant spécifique, vous devez également préciser le nom du composant, comme illustré ci-dessous :

```
click="currentState='componentID.viewstatename' "
```

Pour plus d'informations, voir [Using View States](#).

- 1 Vérifiez que l'état initial comporte un contrôle cliquable, par exemple un contrôle de type Button.
En mode Création de l'éditeur MXML, sélectionnez le contrôle et entrez la valeur suivante dans le champ Lors d'un clic de la vue Propriétés :

```
currentState='viewstatename'
```

viewstatename est le nom de l'état.
- 2 Pour passer à l'état de base, saisissez :

```
currentState=''
```

`''` est une chaîne vide représentée par deux guillemets simples.
- 3 Pour vérifier que les états changent correctement dans l'application lorsque vous cliquez sur le bouton, sélectionnez le bouton Exécuter de la barre d'outils de Flash Builder.

Vous pouvez définir une transition afin que le changement d'état soit visuellement plus fluide. Pour plus d'informations, voir « [Création d'une transition](#) » à la page 232.

Création de groupes d'états d'affichage

Flex prend en charge les groupes d'états d'affichage. L'attribut `stateGroups` de la balise `<States>` vous permet de regrouper un ou plusieurs états. Par exemple, si plusieurs composants apparaissent dans le même jeu d'états d'affichage, vous pouvez créer un groupe d'états d'affichage les contenant tous. L'attribution à la propriété `currentState` de l'un des états d'affichage du groupe conduira ensuite à l'affichage des composants. Pour plus d'informations, illustrées d'exemples, voir *Defining view state groups*.

Le mode Création de l'éditeur MXML ne permet pas de modifier les groupes d'états. Basculez en mode Source pour créer et modifier des groupes d'états. Le mode Source facilite la création et l'édition de groupes d'états par des indicateurs de code et par la liste déroulante Afficher état.

Si vous créez un groupe d'états d'affichage, utilisez la vue Création avec prudence. La suppression d'un état dans la vue Création peut conduire par inadvertance au maintien d'une référence à un composant supprimé dans un groupe d'états.

Suppression d'un état d'affichage

Vous pouvez supprimer les états d'affichage d'une application dans la vue Création de l'éditeur MXML. Les groupes d'états créés devront toutefois être supprimés dans la vue Source. Vous évitez ainsi de laisser par inadvertance une référence à un composant dans un état supprimé.

- 1 Dans la vue Création de l'éditeur MXML, sélectionnez l'état d'affichage que vous voulez supprimer de la vue Etats (Fenêtre > Etats).
- 2 Cliquez sur le bouton Supprimer l'état dans la barre d'outils de la vue Etats.

Utilisation de plusieurs états dans une application

Si l'application comporte plusieurs états, le mode Création de l'éditeur MXML vous permet de changer de vue pour chaque état, en affichant uniquement les composants définis pour un état spécifique. Vous pouvez spécifier l'état dans lequel chaque composant est visible.

Modification du composant d'un état spécifique

- 1 Dans la vue Création de l'éditeur de source, utilisez la vue Etats pour ajouter un ou plusieurs états supplémentaires à une application.
- 2 Utilisez le menu déroulant Etats pour basculer la vue sur l'état sélectionné.
- 3 Ajoutez, déplacez, supprimez ou modifiez les composants de l'état.

Les modifications que vous appliquez à un état donné n'apparaissent pas dans les autres états, sauf si vous spécifiez que le composant apparaît dans plusieurs états.

Spécification de l'affichage d'un composant dans plusieurs états

- 1 Dans la vue Création de l'éditeur de source, utilisez la vue Etats pour ajouter un ou plusieurs états supplémentaires à une application.
- 2 Sélectionnez chaque composant d'un état.
- 3 Dans la vue Propriétés, sélectionnez les états dans lesquels le composant apparaît.

Vous pouvez indiquer que le composant doit apparaître dans tous les états ou vous pouvez sélectionner un ou plusieurs états dans lesquels le composant doit apparaître.

Un composant, pour lequel un état particulier a été défini, ne s'affichera pas dans l'éditeur lorsqu'un état autre que celui qui lui a été attribué est sélectionné.

 *Soyez prudent lorsque vous modifiez des applications contenant plusieurs états. Les composants peuvent donner l'impression de disparaître lorsque vous passez à un état qui ne contient pas un composant pourtant visible dans un autre état.*

Création et édition d'états d'affichage dans le code source

Le mode Source de l'éditeur MXML comporte plusieurs fonctions pour la modification du code source des états d'affichage.

Pour toute application déclarant des états d'affichage, l'éditeur MXML comporte la liste déroulante Afficher état. Lorsque vous sélectionnez un état d'affichage dans la liste Afficher état, la mise en évidence dans l'éditeur des composants qui ne figurent pas dans cet état est atténuée.

Les propriétés `includeIn` et `excludeFrom` des composants MXML spécifient l'état d'affichage ou le groupe d'états dans lequel un composant s'affiche. Les indicateurs de code de l'éditeur MXML facilitent la sélection d'un état d'affichage ou d'un groupe d'états pour ces propriétés.

Vous pouvez également utiliser la notation à points pour les attributs de composants. Elle vous permet de spécifier l'état d'affichage dans lequel l'attribut est appliqué. Pour afficher par exemple un composant Button dans deux états d'affichage et obtenir la modification de l'étiquette en fonction de l'état, associez l'opérateur point à la propriété `label`. Les indicateurs de code de l'éditeur MXML facilitent la sélection de l'état d'affichage. Par exemple :

```
<s:Button label.State1="Button in State 1" label.State2="Same Button in State 2">
```

Exemple illustrant l'utilisation des états d'affichage dans le code source

- 1 Créez une application contenant plus d'un état d'affichage.

En mode Source de l'éditeur MXML, ajoutez le code suivant après la balise `<s:Application>`.

```
<s:states>
  <s:State name="State1" />
  <s:State name="State2" />
  <s:State name="State3" />
</s:states>
```

Notez l'affichage de la liste déroulante Afficher état, une fois que des états ont été définis dans l'application.

- 2 En mode Source, ajoutez les composants Button suivants :

```
<s:Button includeIn="State1" label="Show State 2"
  click="currentState='State2'" />
<s:Button includeIn="State2" label="Show State 3"
  click="currentState='State3'" />
<s:Button includeIn="State3" label="Show State 1"
  click="currentState='State1'" />

<s:Button
  label.State1="All States: State 1 Label"
  label.State2="All States: State 2 Label"
  label.State3="All States: State 3 Label"
  x="0" y="30"/>
```

Par défaut, l'éditeur affiche le code de tous les états.

Remarque : les gestionnaires d'événement `click` des trois premiers boutons font défiler les états d'affichage.

- 3 Restez en mode Source et sélectionnez différents états d'affichage dans la liste Afficher état.

L'éditeur affiche en gris clair le code des composants qui ne sont pas visibles dans l'état sélectionné.

L'intégralité du code peut être modifiée. L'atténuation de la mise en évidence des composants n'apparaissant pas dans l'état d'affichage sélectionné facilite toutefois la gestion du code de chaque état d'affichage.

4 Basculez en mode Création de l'éditeur MXML.

Sélectionnez différents états d'affichage dans la liste déroulante Etat ou dans la vue Etats. L'éditeur affiche les composants en fonction des propriétés définies pour l'état d'affichage sélectionné.

5 Exécutez l'application. Cliquez sur le bouton supérieur pour faire défiler les états d'affichage.

Pour plus d'informations sur la création et la modification des états dans le code source, voir *Create and apply view states*.

Création d'une transition

Lorsque vous modifiez les états d'affichage d'une application, les composants semblent sauter d'un état d'affichage à un autre. Vous pouvez adoucir cet effet en ajoutant des transitions. Une transition consiste en un ou plusieurs effets regroupés pour être lancés lorsqu'un état d'affichage est modifié. Par exemple, vous pouvez définir une transition utilisant un effet *Resize* de redimensionnement afin de réduire progressivement un composant dans l'état d'affichage d'origine et un effet *Fade* de fondu pour afficher progressivement un composant dans le nouvel état d'affichage.

1 Assurez-vous de créer au moins un état d'affichage en plus de l'état de base.

2 Dans la vue Source de l'éditeur MXML, définissez un objet Transition en ajoutant une balise `<s:transitions>`, puis une balise enfant `<s:Transition>`, comme l'illustre l'exemple suivant :

```
<s:transitions>
  <mx:Transition id="myTransition">
  </mx:Transition>
</s:transitions>
```

Pour définir plusieurs transitions, insérez d'autres balises enfant `<s:Transition>` dans la balise `<s:transitions>`.

3 Dans la balise `<s:Transition>`, définissez la modification de l'état d'affichage qui déclenche la transition en définissant les propriétés `fromState` et `toState`, comme l'illustre l'exemple suivant (en gras) :

```
<s:transitions>
  <mx:Transition id="myTransition" fromState="*" toState="checkout">
  </mx:Transition>
</s:transitions>
```

Dans l'exemple, vous spécifiez que vous souhaitez appliquer la transition lorsque l'application change d'un état d'affichage (`fromState="*"`) à l'état d'affichage intitulé `checkout` (`toState="checkout"`). La valeur `"*"` est un caractère générique représentant n'importe quel état d'affichage.

4 Dans la balise `<mx:Transition>`, indiquez si vous souhaitez que les effets s'exécutent de façon parallèle ou séquentielle en ajoutant une balise enfant `<mx:Parallel>` ou `<mx:Sequence>`, comme l'illustre l'exemple suivant (en gras) :

```
<mx:Transition id="myTransition" fromState="*" toState="checkout">
  <mx:Parallel>
  </mx:Parallel>
</mx:Transition>
```

Si vous souhaitez appliquer simultanément les effets, utilisez la balise `<mx:Parallel>`. Si vous souhaitez les appliquer les uns après les autres, utilisez la balise `<mx:Sequence>`.

- 5 Dans la balise `<mx:Parallel>` ou `<mx:Sequence>`, spécifiez le ou les composants cible de la transition en définissant la propriété `target` (pour un composant cible) ou `targets` (pour plusieurs composants cible) selon l'ID du ou des composants cible, comme l'illustre l'exemple suivant :

```
<mx:Parallel targets="{ [myVGroup1, myVGroup2, myVGroup3] }" >
</mx:Parallel>
```

Dans cet exemple, trois conteneurs `VGroup` sont ciblés. La propriété `targets` sélectionne un tableau de plusieurs ID.

- 6 Dans la balise `<mx:Parallel>` ou `<mx:Sequence>`, spécifiez les effets à appliquer lors d'un changement d'état d'affichage en ajoutant des balises enfant d'effet, comme l'illustre l'exemple suivant (en gras) :

```
<mx:Parallel targets="{ [myVBox1, myVBox2, myVBox3] }" >
  <mx:Move duration="400"/>
  <mx:Resize duration="400"/>
</mx:Parallel>
```

Pour connaître la liste des effets possibles et savoir comment définir leurs propriétés, voir [Introduction to effects](#).

- 7 Pour tester la transition, cliquez sur le bouton Exécuter de la barre d'outils de Flash Builder, puis changez d'état après le lancement de l'application.

Liaison des contrôles à des données

Pour accéder à un service de données, Flash Builder fournit des outils permettant de lier les données à des contrôles de données, par exemple `DataGrid`. Il crée les colonnes d'un contrôle `DataGrid` en fonction des données retournées par un service. Vous devrez généralement configurer les colonnes ainsi générées. Flash Builder comporte un éditeur pour la configuration des colonnes des composants `DataGrid` et `AdvancedDataGrid`.

Pour plus d'informations sur la liaison des données aux contrôles de données, voir [Liaison d'opérations de service à des contrôles](#).

Configuration des composants `DataGrid` et `Advanced DataGrid`

Passer en mode Création de l'éditeur MXML pour configurer les colonnes `DataGrid`. La procédure suivante illustre la configuration des colonnes d'un composant `DataGrid` accédant à un service de données. La configuration des colonnes d'un composant `AdvancedDataGrid` est analogue.

Configuration des colonnes d'un composant `DataGrid`

- 1 En mode Création de l'éditeur MXML, ajoutez un contrôle `DataGrid` (ou `AdvancedDataGrid`). Liez-le aux données renvoyées par un service de données.

Pour plus d'informations, voir [Liaison d'opérations de service à des contrôles](#).

- 2 Sélectionnez le contrôle `DataGrid`, puis cliquez sur le bouton Configurer les colonnes de la vue Propriétés.

Vous pouvez également sélectionner cette option dans le menu contextuel du contrôle.

- 3 Dans la boîte de dialogue Configurer les colonnes, utilisez les boutons Ajouter, Supprimer, Haut et Bas pour ajouter, supprimer ou réorganiser les colonnes.

4 La Vue standard de cette boîte de dialogue permet de modifier les principales propriétés d'une colonne sélectionnée.

- Liaison de données

Sélectionnez le champ de données qui vous souhaitez afficher dans la colonne. La liste déroulante Liaison au champ répertorie tous les champs disponibles à partir des données renvoyées. Si le contrôle DataGrid est modifiable, vous pouvez déterminer si les données de cette colonne peuvent être modifiées.

Si les données à afficher ne proviennent pas d'un service de données, Liaison au champ se présente sous la forme d'une zone de texte permettant de représenter les colonnes définies dans la source de données. Vous pouvez par exemple saisir le nom des colonnes définies dans une XMLList. Si le nom spécifié ne correspond à aucune source de données définie, la valeur est ignorée et la colonne demeure vide.

Les colonnes ajoutées par programme à un contrôle DataGrid ne peuvent pas être configurées à l'aide des fonctions de liaison de données.

- Propriétés générales

Spécifiez l'en-tête et la largeur de la colonne. Définissez également si la colonne peut ou non être redimensionnée ou triée.

La largeur doit être exprimée en pixels. La valeur par défaut est 100 pixels. Si la valeur de la propriété `horizontalScrollPolicy` du contrôle DataGrid est `false`, toutes les colonnes visibles sont affichées. Pour garantir l'affichage de toutes les colonnes visibles, DataGrid ne respecte pas toujours la valeur spécifiée pour la largeur.

- Propriétés de texte

Définissez le style de mise en forme du texte de la colonne.

5 La Vue avancée permet d'afficher et de modifier les paramètres de toutes les propriétés d'une colonne sélectionnée.

Ajout de composants de représentation graphique

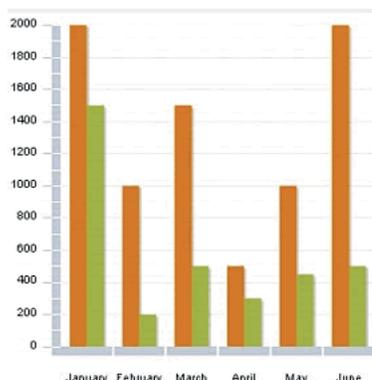
Flash Builder vous permet d'ajouter des composants de représentation graphique pour l'affichage des données dans l'interface utilisateur. Ces composants Flex vous permettent de créer des types de graphiques très courants et d'en déterminer entièrement l'aspect. Pour un récapitulatif des graphiques disponibles, voir *Chart types*.

Cette section décrit l'ajout de composants de représentation graphique à l'interface utilisateur. Pour plus d'informations sur la définition de données de graphique, la mise en forme d'éléments de graphique et la manipulation d'autres aspects des graphiques, voir *Introduction to Charts*.

Ajout d'un composant de représentation graphique

- 1 En mode Création de l'éditeur MXML, faites glisser un composant de représentation graphique de la vue Composants vers la zone de création.
- 2 Saisissez un ID pour le graphique.
- 3 Pour afficher plusieurs séries de données dans le graphique, cliquez sur le bouton Ajouter et saisissez le nom de la nouvelle série dans la boîte de dialogue qui s'affiche.

Par exemple, le contrôle ColumnChart contient deux séries de données. La barre sur la gauche représente le bénéfice brut sur six mois, et la suivante, le bénéfice net pour la même période.



Supprimez une série de données en la sélectionnant dans la liste et en cliquant sur le bouton Supprimer.

4 (Facultatif) Sélectionnez l'option Inclure une légende.

Cette option vous permet d'ajouter un contrôle Legend au graphique. Ce contrôle affiche l'étiquette de chaque série de données dans le graphique et une légende affichant l'élément de représentation graphique de la série.

5 Cliquez sur OK pour insérer le graphique.

Ajout d'interactivité à l'aide d'effets

Un *effet* est la modification visible ou audible du composant cible se produisant sur une période exprimée en millisecondes. Le fondu, le redimensionnement ou le déplacement d'un composant constituent des exemples d'effets.

Les effets sont déclenchés en réponse à un événement, lui-même souvent déclenché par l'action d'un utilisateur, par exemple le fait de cliquer sur un bouton. Les effets peuvent toutefois être programmés ou déclenchés en réponse à des événements qui ne sont pas déclenchés par l'utilisateur.

Vous pouvez par exemple créer un effet faisant rebondir légèrement un composant TextInput auquel l'utilisateur accède par tabulation. Vous pouvez également appliquer un effet de fondu en sortie à un composant Label sur lequel l'utilisateur fait glisser le curseur de la souris.

Dans Flash Builder, les effets peuvent être définis en tant que propriétés des composants MXML. Accédez à la vue Source de l'éditeur MXML pour implémenter l'effet.

L'implémentation des effets varie suivant qu'il s'agit de composants Spark ou MX. Pour plus d'informations sur la création d'effets en code MXML et ActionScript, voir Introduction to effects.

Création d'un effet pour un composant

Vous définissez les effets généralement en mode Source de l'éditeur MXML. Ils sont souvent invoqués à partir du gestionnaire d'événement d'un composant. Vous pouvez par exemple utiliser le gestionnaire d'événement click pour qu'un composant Button invoque un effet. Voir Applying effects.

Dans Flash Builder cependant, vous pouvez définir un effet pour une propriété d'un composant MXML.

1 Accédez au mode Création de l'éditeur MXML, puis cliquez sur un composant dans la zone de création.

- 2 Définissez la propriété d'effet pour les composants Spark :
 - a Dans la vue Propriétés, sélectionnez l'icône Vue par catégories.
 - b Dans la catégorie Effets, sélectionnez une propriété et spécifiez un effet.

Vous pouvez par exemple spécifier l'effet Fade pour la propriété rollOverEffect d'un composant Button. Pour obtenir une liste des effets disponibles, voir Available effects.
- 3 Enregistrez le fichier et exécutez-le pour visionner l'effet.

Chapitre 10 : Utilisation de données dans Flash Builder

Dans Adobe® Flash® Builder™, vous interagissez avec des données et des contrôles gérés par des données directement dans les codes ActionScript et MXML. Vous pouvez travailler avec des données, générer automatiquement des applications de bases de données et utiliser un code proxy pour les services Web, ainsi que générer et utiliser le code fonctionnant avec Flex Ajax Bridge. Vous pouvez également gérer les problèmes de sécurité d'accès aux données d'Adobe Flash Player et utiliser Flash Builder avec un service proxy.

A propos de l'utilisation des données dans Flash Builder

Dans Flash Builder, les données sont manipulées en modifiant directement les codes des applications ActionScript et MXML.

Conteneurs et contrôles gérés par les données

Flex fournit des composants de type conteneur et de type contrôle à partir desquels vous pouvez créer les interfaces utilisateur de vos applications Flex. Certains composants présentent des données que les utilisateurs peuvent sélectionner et avec lesquelles ils peuvent interagir lors de l'utilisation de l'application. Voici quelques exemples d'utilisation des contrôles gérés par les données :

- Dans un formulaire pour la saisie d'une adresse, vous pouvez donner aux utilisateurs la possibilité de sélectionner leur pays de résidence (par exemple) en utilisant les contrôles ComboBox ou List.
- Dans une application de panier d'achats, vous pouvez utiliser un composant Spark List pour présenter les données des produits comportant des images. Pour le composant List, la présentation peut prendre les valeurs suivantes : VerticalLayout, HorizontalLayout ou TileLayout.
- Vous pouvez fournir des options standard de navigation en utilisant des conteneurs tels que les contrôles TabBar et ButtonBar.

Pour tous les contrôles gérés par des données, l'entrée de données est effectuée par un *fournisseur de données*.

Pour plus d'informations sur l'utilisation de contrôles gérés par des données, voir Spark list-based controls.

Fournisseurs de données et collections

Un objet *collection* contient un objet de données, par exemple Array ou XMLList, et fournit un ensemble de méthodes permettant de consulter, trier, filtrer et modifier les éléments de données de cet objet. Plusieurs contrôles Adobe Flex, désignés par l'expression *contrôles de fournisseurs de données*, ont une propriété `dataProvider` renseignée par une collection.

Les exemples suivants illustrent comment un fournisseur de données est défini (en tant qu'ArrayCollection ActionScript) et utilisé par un contrôle :

```

<!-- Simple example to demonstrate the Spark ComboBox control -->
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
               xmlns:s="library://ns.adobe.com/flex/spark"
               xmlns:mx="library://ns.adobe.com/flex/halo">

    <fx:Script>
        <![CDATA[
            import mx.collections.ArrayCollection;

            [Bindable]
            public var complexDP:ArrayCollection = new ArrayCollection(
                [
                    {ingredient:"Salmon", category:"Meat"},
                    {ingredient:"Potato", category:"Starch"},
                    {ingredient:"Cucumber", category:"Vegetable"},
                    {ingredient:"Steak", category:"Meat"},
                    {ingredient:"Rice", category:"Starch"},
                    {ingredient:"Cumin", category:"Spice"}
                ]
            );

            <!-- Function to handel custom input strings -->
            private function myLabelToItemFunction(input:String):*
            {
                <!-- Returns object that matches items in dataProvider -->
                return {ingredient:input, category:"mystery"};
            }
        ]]>
    </fx:Script>

    <s:Panel title="Spark ComboBox Example" width="75%" height="75%">
        <s:layout>
            <s:VerticalLayout paddingTop="10" paddingLeft="10"/>
        </s:layout>

        <!-- Label that displayed current property values -->
        <s:Label text="Index : {cb.selectedIndex}
                Item : {cb.selectedItem.ingredient}
                Type : {cb.selectedItem.category}"/>

        <!-- ComboBox with custom labelToItem function -->
        <s:ComboBox
            id="cb"
            dataProvider="{complexDP}"
            width="150"
            labelToItemFunction="{myLabelToItemFunction}"
            selectedIndex="0"
            labelField="ingredient"/>
    </s:Panel>
</s:Application>

```

Pour plus d'informations sur les fournisseurs de données et les collections, voir [Data providers and collections](#).

Accès aux données distantes

Flex contient des composants d'accès aux données basés sur une architecture orientée services (Service-oriented architecture, SOA). Ces composants utilisent des appels de procédure distants pour interagir avec des environnements de serveur (PHP, Adobe ColdFusion® et Microsoft ASP.NET, par exemple), afin de fournir des données aux applications et d'envoyer des données aux sources de données principales.

Selon le type d'interface dont vous disposez sur une application côté serveur donnée, vous pouvez vous connecter à une application Flex à l'aide de l'une des méthodes suivantes :

- HTTP GET ou POST avec le composant HTTPService
- Services Web conformes au protocole SOAP avec le composant WebService
- Services distants AMF (Adobe Action Message Format) avec le composant RemoteObject

***Remarque :** lorsque vous utilisez Flash Builder pour développer des applications accédant aux données côté serveur, utilisez un fichier `cross-domain.xml` ou un proxy si vous accédez aux données à partir d'un domaine différent du domaine à partir duquel l'application a été chargée. Voir « [Gestion de la sécurité de Flash Player](#) » à la page 245.*

Flash Builder permet également la création d'applications utilisant Adobe LiveCycle® Data Services ES, un produit distinct offrant des fonctionnalités de service de données avancées. LiveCycle Data Services ES propose des applications de service d'appel de procédure distante (RPC), ainsi qu'une configuration de sécurité avancée. Il offre également les services de données suivants :

Service de gestion de données Ce service permet de créer des applications utilisant des données distribuées ainsi que de gérer d'importantes collections de données et de relations de données imbriquées, telles que les relations un-à-un ou un-à-plusieurs.

Service de message Ce service permet de créer des applications pouvant envoyer des messages à d'autres applications et recevoir des messages d'autres applications, dont les applications Flex et JMS (Java Message Service).

Flash Builder fournit des assistants et des outils permettant de se connecter aux services de données et de lier des opérations de services de données à des composants d'application. Voir [Création d'applications centrées sur les données avec Flash Builder](#).

Liaison de données

Comme vous avez pu le constater dans l'exemple de code fourni dans « [Fournisseurs de données et collections](#) » à la page 237, la valeur de la propriété `dataProvider` du contrôle `ComboBox` est "`{complexDP}`". Il s'agit d'un exemple de liaison de données.

La liaison de données copie la valeur d'un objet (la source) dans un autre objet (la destination). Une fois qu'un objet est lié à un autre objet, les modifications apportées à la source sont automatiquement répercutées dans la destination.

L'exemple suivant lie la propriété de texte d'un contrôle `TextInput` (la source) à la propriété de texte d'un contrôle `Label` (la destination), de sorte à ce que le texte saisi dans le contrôle `TextInput` s'affiche dans le contrôle `Label` :

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
               xmlns:s="library://ns.adobe.com/flex/spark"
               xmlns:mx="library://ns.adobe.com/flex/mx" minWidth="955" minHeight="600">
  <fx:Declarations>
    <!-- Place non-visual elements (e.g., services, value objects) here -->
  </fx:Declarations>

  <s:TextInput id="LNameInput" x="10" y="10"/>
  <s:Label text="{LNameInput.text}" x="10" y="50"/>
</s:Application>
```

Pour lier des données d'un objet à un autre, vous pouvez faire appel à la syntaxe d'accolades ({}) (comme l'illustre l'exemple) ou à la balise `<fx:Binding>`. Pour plus d'informations, voir *Using data binding with data models* et *Data binding* dans *Utilisation d'Adobe Flex 4*.

Modèles de données

Un *modèle de données* est un objet que vous pouvez utiliser afin de stocker de manière temporaire des données dans la mémoire pour faciliter leur manipulation. Vous pouvez définir un modèle de données dans ActionScript et dans MXML en utilisant une balise telle que `<fx:Model>` ou en utilisant tout objet contenant des propriétés. Par exemple, le modèle de données suivant affiche des informations telles que le nom d'une personne, son âge et son numéro de téléphone :

```
<fx:Declarations>
  <fx:Model id="reg">
    <registration>
      <name>{nme.text}</name>
      <email>{email.text}</email>
      <phone>{phone.text}</phone>
      <zip>{zip.text}</zip>
      <ssn>{ssn.text}</ssn>
    </registration>
  </fx:Model>
</fx:Declarations>
```

Les champs d'un modèle de données peuvent contenir des données statiques (comme l'illustre l'exemple). Vous pouvez également avoir recours aux liaisons de données pour transmettre des données au modèle de données ou en provenance de ce dernier.

Vous pouvez en outre définir le modèle de données dans un fichier XML. Vous référez alors le fichier XML par le biais du système de fichiers ou d'une URL à l'aide de la propriété `source` de la balise `<fx:Model>`, comme l'illustre l'exemple suivant :

```
<fx:Model source="content.xml" id="Contacts"/>
<fx:Model source="http://www.somesite.com/companyinfo.xml" id="myCompany"/>
```

Pour plus d'informations sur les modèles de données, voir *Storing Data*.

Validation de données

La validation de données vérifie que les données saisies par l'utilisateur dans l'application sont valides. Vous pouvez par exemple utiliser un *validateur de données* de code postal pour vérifier que le code postal indiqué par l'utilisateur est valide.

Flex propose des validateurs prédéfinis pour les types de données suivants : carte de crédit, devise, date, adresse électronique, numéro, numéro de téléphone, expression régulière, sécurité sociale, chaîne et code postal.

Les validateurs de données sont des composants Flex non visuels, ce qui signifie qu'ils ne sont pas accessibles à partir de la vue Composants. Vous devez donc les insérer dans le code, comme l'illustre l'exemple MXML suivant :

```
<!-- Define the ZipCodeValidator. -->
<mx:ZipCodeValidator id="zcV" source="{zipcodeInput}" property="text"/>
<!-- Define the TextInput control for entering the zip code. -->
<s:TextInput id="zipcodeInput"/>
```

Dans cet exemple MXML, le validateur est défini avec la balise MXML appropriée et est lié à la propriété ID d'un contrôle TextInput. Au moment de l'exécution, lorsque l'utilisateur saisit le numéro de téléphone dans le contrôle TextInput, le numéro est immédiatement validé.

Vous pouvez bien sûr utiliser les validateurs de données dans ActionScript en définissant une variable en tant qu'occurrence d'une classe de validateur, puis en créant une fonction pour la lier au contrôle de saisie.

Les validateurs de données sont souvent utilisés avec les modèles de données. Pour plus d'informations, voir Validating Data.

Mise en forme des données

La mise en forme correcte de certains types de données dans l'application est garantie par l'utilisation de *formateurs de données*. Flex fournit des formateurs de données prédéfinis pour les types de données suivants : devise, date, numéro, téléphone et code postal.

Les formateurs de données sont liés aux contrôles de saisie. Ils mettent correctement en forme les données saisies par l'utilisateur. Par exemple, un utilisateur peut saisir une date au format suivant :

120105

Un formateur, lié au contrôle de saisie de texte, stocke et affiche la date au format suivant :

12/01/05

A l'instar des validateurs de données, les formateurs de données sont des composants Flex non visuels que vous pouvez utiliser avec des balises MXML ou des classes ActionScript.

Pour plus d'informations, voir Formatting Data.

Génération automatique du code Flex Ajax Bridge

La fonction Créer un composant Ajax Bridge permet de générer un code JavaScript et un fichier d'enveloppe HTML facilitant l'utilisation d'une application Flex de JavaScript dans une page HTML. Elle peut être utilisée conjointement avec la bibliothèque Flex Ajax Bridge JavaScript qui permet d'exposer une application Flex au script dans le navigateur Web. Le code JavaScript généré est très léger puisqu'il est destiné à exposer la fonctionnalité déjà fournie par Flex Ajax Bridge. Pour plus d'informations sur Flex Ajax Bridge, voir Flex Ajax Bridge.

La fonction Créer un composant Ajax Bridge génère un code proxy JavaScript spécifique aux API d'application Flex que vous voulez appeler à partir de JavaScript. Vous pouvez générer du code pour toute classe ActionScript ou application MXML dans un projet Flash Builder.

Pour les fichiers d'application MXML, vous pouvez générer un code pour tous les éléments suivants ou une partie d'entre eux dans le code MXML :

- Liste des éléments hérités pouvant s'étendre de façon non récursive
- Propriétés publiques, incluant les balises avec les propriétés `id`
- Constantes publiques
- Fonctions publiques incluant les classes définies dans la ligne

Pour les classes `ActionScript`, vous pouvez générer un code pour tous les éléments suivants ou une partie d'entre eux :

- Liste des éléments hérités
- Propriétés publiques (pour chaque propriété, affichage d'une méthode `get` et `set`)
- Constantes publiques
- Méthodes publiques

Dans un répertoire que vous spécifiez, la fonction `Créer un composant Ajax Bridge` génère des fichiers `*.js` et `*.html` correspondant aux applications MXML et aux classes `ActionScript` sélectionnées pour la génération. Elle place une copie de la bibliothèque Flex Ajax Bridge (`fabridge.js`) dans un sous-répertoire du répertoire de génération de code. Cette fonction génère également des fichiers d'aide MXML dans le répertoire `src` du projet ; ces fichiers sont utilisés pour finaliser la génération du code JavaScript.

Génération de code Ajax Bridge

- 1 Cliquez avec le bouton droit sur un projet dans l'Explorateur de packages Flex et sélectionnez `Créer un composant Ajax Bridge`.
- 2 Dans la boîte de dialogue `Création d'un composant Ajax Bridge`, sélectionnez les applications MXML et les classes `ActionScript` pour lesquelles vous souhaitez générer un code JavaScript. Vous pouvez sélectionner la case à cocher de niveau supérieur pour inclure l'intégralité du projet ou vous pouvez sélectionner des membres spécifiques.
- 3 Spécifiez le répertoire dans lequel générer les classes de proxy.
- 4 Cliquez sur `OK` pour générer le code. L'exemple suivant illustre un fichier `.js` généré pour une application affichant des images :

```
*
 * You should keep your JavaScript code inside this file as light as possible,
 * and keep the body of your Ajax application in separate *.js files.
 *
 * Do make a backup of your changes before regenerating this file. (Ajax Bridge
 * display a warning message.)
 *
 * For help in using this file, refer to the built-in documentation in the Ajax Bridge
application.
 *
 */

/**
 * Class "DisplayShelfList"
 * Fully qualified class name: "DisplayShelfList"
 */
function DisplayShelfList(obj) {
    if (arguments.length > 0) {
        this.obj = arguments[0];
    } else {
```

```

        this.obj = FABridge["b_DisplayShelfList"].
            create("DisplayShelfList");
    }
}

// CLASS BODY
// Selected class properties and methods
DisplayShelfList.prototype = {

    // Fields form class "DisplayShelfList" (translated to getters/setters):

    // Methods form class "DisplayShelfList":

    getAngle : function() {
        return this.obj.getAngle();
    },

    setAngle : function(argNumber) {
        this.obj.setAngle(argNumber);
    },

    setCurrentPosition : function(argNumber) {
        this.obj.setCurrentPosition(argNumber);
    },

    setSelectedIndex : function(argNumber) {
        this.obj.setSelectedIndex(argNumber);
    },

    setPercentHeight : function(argNumber) {
        this.obj.setPercentHeight(argNumber);
    },

    setPercentWidth : function(argNumber) {
        this.obj.setPercentWidth(argNumber);
    },

    DisplayShelfList : function() {
        return this.obj.DisplayShelfList();
    },

    setFirst : function(argNumber) {
        this.obj.setFirst(argNumber);
    },

    setFormat : function(argString) {
        this.obj.setFormat(argString);
    },

    setLast : function(argNumber) {
        this.obj.setLast(argNumber);
    }
}

/**
 * Listen for the instantiation of the Flex application over the bridge.

```

```

*/
FABridge.addInitializationCallback("b_DisplayShelfList", DisplayShelfListReady);

/**
 * Hook here all of the code that must run as soon as the DisplayShelfList class
 * finishes its instantiation over the bridge.
 *
 * For basic tasks, such as running a Flex method on the click of a JavaScript
 * button, chances are that both Ajax and Flex have loaded before the
 * user actually clicks the button.
 *
 * However, using DisplayShelfListReady() is the safest way, because it lets
 * Ajax know that involved Flex classes are available for use.
 */
function DisplayShelfListReady() {

    // Initialize the root object. This represents the actual
    // DisplayShelfListHelper.mxml Flex application.
    b_DisplayShelfList_root = FABridge["b_DisplayShelfList"].root();

    // YOUR CODE HERE
    // var DisplayShelfListObj = new DisplayShelfList();
    // Example:
    // var myVar = DisplayShelfListObj.getAngle ();
    // b_DisplayShelfList_root.addChild(DisplayShelfListObj);

}

```

- 5 Modifiez les fichiers .js générés. Dans la fonction `xxxReady()` des fichiers .js générés, ajoutez le code à exécuter dès que la classe correspondante termine son instanciation sur Ajax Bridge. Selon l'application, le code par défaut peut être généré dans cette méthode. Le code en gras de l'exemple suivant illustre le code d'initialisation personnalisé pour l'exemple d'application d'image.

```

...
function DisplayShelfListReady() {

    // Initialize the root object. This represents the actual
    // DisplayShelfListHelper.mxml Flex application.
    b_DisplayShelfList_root = FABridge["b_DisplayShelfList"].root();

    // Create a new object.
    DisplayShelfListObj = new DisplayShelfList();
    // Make it as big as the application.
    DisplayShelfListObj.setPercentWidth(100);
    DisplayShelfListObj.setPercentHeight(100);
    //Set specific attributes.
    DisplayShelfListObj.setFirst(1);
    DisplayShelfListObj.setLast(49);
    DisplayShelfListObj.setFormat("./photos400/photo%02d.jpg");
    //Add the object to the DisplayList hierarchy.
    b_DisplayShelfList_root.addChild(DisplayShelfListObj.obj);

}

```

- 6 Modifiez les fichiers .html générés. Dans la partie des pages HTML contenant le texte d'indication d'emplacement de la description, remplacez le texte par le code HTML à utiliser pour accéder à l'application Flex à partir de la page HTML. Par exemple, ce code ajoute des boutons pour contrôler l'exemple d'application d'image :

```
<h2>Test controls</h2>
<ul>
<li><input type="button" onclick="DisplayShelfListObj.setCurrentPosition(0) "
value="Go to first item"/>
</li>
<li><input type="button" onclick="DisplayShelfListObj.setCurrentPosition(3) "
value="Go to fourth item"/>
</li>
<li><input type="button" onclick="DisplayShelfListObj.setSelectedIndex(0) "
value="Go to first item (with animation)"/>
</li>
<li><input type="button" onclick="DisplayShelfListObj.setSelectedIndex(3) "
value="Go to fourth item (with animation)"/>
</li>
<li><input type="button" onclick="alert(DisplayShelfListObj.getAngle()) "
value="Get photo angle"/>
</li>
<li><input type="button" onclick="DisplayShelfListObj.setAngle(15); "
value="Set photo angle to 15&deg;"/>
</li>
</ul>
```

- 7 Ouvrez la page HTML dans un navigateur Web pour exécuter l'application.

Gestion de la sécurité de Flash Player

Flash Player n'autorise pas une application à recevoir des données d'un domaine autre que celui à partir duquel il a été chargé, à moins d'une autorisation explicite donnée. Si vous chargez le fichier SWF d'application depuis <http://mydomain.com>, il ne peut pas charger de données depuis <http://yourdomain.com>. Ce sandbox de sécurité empêche une utilisation malveillante des fonctions de Flash Player (JavaScript utilise un modèle de sécurité similaire pour empêcher toute utilisation malveillante de JavaScript).

Pour accéder aux données à partir d'une application Flex, vous avez trois possibilités :

- Ajoutez un fichier de régulation interdomaines à la racine du serveur hébergeant le service de données. Voir « [Utilisation des fichiers de régulation interdomaines](#) » à la page 245.
- Placez le fichier SWF d'application sur le même serveur que celui hébergeant le service de données.
- Sur le serveur contenant le fichier SWF d'application, créez un proxy qui appelle un service de données hébergé sur un autre serveur. Voir « [Configuration de Flash Builder pour l'accès par proxy aux données distantes](#) » à la page 246.

Utilisation des fichiers de régulation interdomaines

Un fichier de régulation interdomaines est un simple fichier XML qui accorde à Flash Player l'autorisation d'accéder aux données à partir d'un domaine autre que le domaine sur lequel réside l'application. Sans ce fichier de régulation, l'utilisateur est invité à fournir une autorisation d'accès via une boîte de dialogue. Cette situation est à éviter.

Le fichier de régulation interdomaines (intitulé `crossdomain.xml`) est placé à la racine du serveur (ou des serveurs) contenant les données auxquelles vous souhaitez accéder. L'exemple suivant illustre un fichier de régulation interdomaines :

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM "http://www.adobe.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
  <allow-access-from domain="www.yourdomain.com" />
</cross-domain-policy>
```

Pour plus d'informations sur la configuration des fichiers de régulation interdomaines, voir les notes techniques suivantes : http://www.adobe.com/go/tn_14213.

Configuration d'un proxy pour accéder aux données distantes

Pour gérer la sécurité de Flash Player, au lieu d'utiliser un fichier de régulation interdomaines, vous pouvez également utiliser un proxy. LiveCycle Data Services ES fournit un système de gestion proxy complet pour les applications Flex. Vous pouvez également créer un simple service proxy en utilisant un langage de script Web tel que ColdFusion, JSP, PHP ou ASP.

Le service proxy traite les requêtes de l'application vers le service distant et répond du service distant vers l'application (Flash Player).



Pendant le développement des applications, le proxy est souvent hébergé sur l'ordinateur local. Pour ce faire, vous devez exécuter un serveur Web et un langage de script sur votre ordinateur de développement local.

Pour plus d'informations sur la création de votre propre proxy, voir les notes techniques suivantes : http://www.adobe.com/go/tn_16520.

Configuration de Flash Builder pour l'accès par proxy aux données distantes

Une fois que vous avez configuré un proxy pour accéder aux données d'un service distant, vous placez les fichiers d'application dans le même domaine que le proxy. Dans Flash Builder, vous pouvez modifier les paramètres de version du projet et la configuration de lancement pour gérer l'utilisation d'un proxy.

Si vous utilisez Flash Builder pour compiler vos applications et que le serveur proxy est également configuré sur votre ordinateur de développement local, vous pouvez modifier les paramètres de version de projet de sorte à copier automatiquement les fichiers d'application compilés à l'emplacement approprié sur votre serveur Web.

Modification du chemin de la version du projet

- 1 Dans l'Explorateur de packages Flex, sélectionnez un projet.
- 2 Cliquez avec le bouton droit de la souris et sélectionnez Propriétés. La boîte de dialogue des propriétés du projet apparaît.
- 3 Sélectionnez la page des propriétés Chemin de génération Flex.
- 4 Modifiez le dossier de sortie existant en saisissant un nouveau chemin ou en naviguant jusqu'au dossier approprié du serveur Web (C:\inetpub\wwwroot\myApp\, par exemple).
- 5 Cliquez sur OK.

Pour exécuter et déboguer l'application à partir du serveur Web, modifiez la configuration de lancement du projet.

Remarque : si le serveur proxy n'est pas votre ordinateur local, copiez le contenu du répertoire bin sur le serveur avant l'exécution ou le débogage du programme.

Modification de la configuration de lancement

- 1 Le fichier d'application principale du projet étant ouvert dans Flash Builder, cliquez avec le bouton droit dans l'éditeur et sélectionnez Exécuter en tant que > Exécuter les configurations. La boîte de dialogue Création, gestion et exécution des configurations s'affiche.
- 2 Dans la liste des configurations, sélectionnez la configuration de lancement du projet.
- 3 (Facultatif) Dans l'onglet Principal, désélectionnez l'option Utiliser la valeur par défaut pour modifier l'URL ou le chemin à lancer.
- 4 Dans la zone de texte Déboguer, saisissez l'URL ou le chemin de la version de débogage de l'application.
- 5 Dans la zone de texte Profil, saisissez l'URL ou le chemin de la version du profileur de l'application.
- 6 Dans la zone de texte Exécuter, saisissez l'URL ou le chemin du fichier d'application principale.
- 7 Cliquez sur Appliquer, puis sur Fermer.

Chapitre 11 : Personnalisation de Flash Builder

Adobe® Flash® Builder™ est un plug-in pour la plateforme de développement Eclipse. Outre les préférences générales que vous avez spécifiées pour Eclipse, spécifiez les préférences de Flash Builder. Les paramètres de certaines fonctions devront être définis à la fois dans les nœuds Eclipse et Flash Builder de la boîte de dialogue Préférences. Par exemple, lorsque vous définissez vos préférences pour le débogueur Flash Builder, vous devez spécifier un comportement personnalisé aussi bien sous Exécution/Débogage d'Eclipse que sous Flash Builder > Débogueur.

L'accès à la boîte de dialogue Préférences varie en fonction de la plateforme et de la version de Flash Builder (autonome ou plug-in) utilisées.

Définition des préférences de Flash Builder

- 1 Ouvrez la boîte de dialogue Préférences :
 - (Windows) Sélectionnez Fenêtre > Préférences.
 - (Macintosh, autonome) Sélectionnez Eclipse > Préférences.
 - (Macintosh, plug-in) Sélectionnez Eclipse > Préférences.
- 2 Développez le nœud Flash Builder pour afficher et spécifier les préférences de l'utilisateur.

Préférences Adobe

Définissez les préférences pour les modules plug-in d'Adobe.

Configuration RDS

La configuration Remote Development Server (RDS) s'applique aux utilisateurs d'Adobe LiveCycle® Data Services ou d'Adobe BlazeDS. RDS donne accès aux destinations LiveCycle Data Services et BlazeDS.

La configuration RDS par défaut est le point de départ pour l'établissement d'une connexion aux services de données. Modifiez la configuration par défaut afin de permettre l'accès à votre serveur ou base de données.

Pour plus d'informations sur la configuration RDS, voir la [documentation d'Adobe LiveCycle Data Services](#).

Important : la configuration RDS comporte des implications sécuritaires. Pour plus d'informations sur les implications sécuritaires du serveur d'applications, voir la documentation sur LiveCycle Data Services.

Préférences de Flash Builder

Flash Builder

Avertissement avant la mise à niveau d'anciens projets Flash Builder

Flash Builder met à jour les projets qui ont été créés dans l'une de ses versions antérieures. La mise à jour englobe la mise en conformité des fichiers et de la structure du projet avec la structure actuelle des projets Flash Builder. Un projet converti ne pourra plus être ouvert avec une version antérieure de Flash Builder.

Par défaut, Flash Builder vous avertit avant de procéder à la conversion. Désactivez cette option pour que les conversions s'effectuent sans avertissement.

Données/Services

Les préférences disponibles pour Données/Services sont décrites ci-dessous. Elles s'appliquent à tous les projets se trouvant dans votre environnement de développement Flash Builder.

Vous pouvez les remplacer pour chaque projet individuellement. Sélectionnez Projet > Propriétés > Données/Services pour remplacer les préférences qui y sont spécifiées.

Générateur de code

Flash Builder fournit un utilitaire de génération de code par défaut pour générer l'accès aux services de données. Faites appel aux fonctions d'extensibilité DCD pour créer vos propres utilitaires de génération de code et les ajouter à Flash Builder en tant que plug-in.

Toute extension de génération de code ajoutée à Flash Builder en tant que plug-in est répertoriée dans la liste déroulante Générateur de code.

Activer l'utilisation de l'occurrence de service simple (singleton) lors de la génération de code pour le gestionnaire de services

Par défaut, cette option n'est pas activée. Au cours de la génération de code, chaque application client d'un projet crée sa propre occurrence d'un service de données.

Pour que les applications client du projet partagent une seule occurrence du service, activez cette option.

Cette option est disponible uniquement si vous sélectionnez l'utilitaire de génération de code par défaut de Flash Builder.

Débogage

Les options de débogage décrites ci-dessous sont activées par défaut. Pour les autres options affectant une session de débogage, voir :

- Préférences > Général > Navigateur Web
- Préférences > Exécution/Débogage

M'avertir en cas de tentative d'ouverture de plusieurs sessions de débogage

Certaines combinaisons de navigateurs et plateformes n'autorisent pas l'exécution en parallèle de plusieurs sessions de débogage. Le démarrage d'une seconde session de débogage entraîne la fermeture ou la déconnexion de la première session.

Ne désactivez pas cette option si vous souhaitez que Flash Builder vous avertisse de la tentative de démarrage d'une seconde session de débogage.

Signaler les erreurs de sécurité après le lancement

Dans certains cas, le navigateur Web émet une erreur de sécurité lorsqu'il ne peut pas lire le fichier de confiance de sécurité de Flash Player. Le redémarrage du navigateur Web suffit généralement à résoudre ce problème.

Ne désactivez pas cette option si vous souhaitez que Flash Builder vous avertisse de ce type d'erreur de sécurité.

Voir la [note technique pour l'avertissement de sécurité de Flash Player](#).

Appeler automatiquement les fonctions getter

Au cours des étapes d'une session de débogage, les variables représentant les fonctions d'accesseur (getter) sont automatiquement évaluées. Ce comportement est généralement utile au cours d'une session de débogage.

Désactivez cette option si vous ne souhaitez pas évaluer automatiquement les variables représentant des fonctions d'accesseur lors des étapes d'une session de débogage.

Editeurs Flash Builder

Accolades

Flash Builder comporte des options pour la mise en retrait, l'insertion et la mise en évidence d'accolades représentant des blocs de code.

Assistant de contenu

L'éditeur de source MXML ou ActionScript de Flash Builder fournit des indicateurs de code qui vous aident à compléter vos expressions de code. Il facilite par exemple la sélection des classes recommandées, des propriétés et des événements disponibles.

- **Activer l'activation automatique**

Désactivez l'option Activer l'activation automatique si vous ne souhaitez pas que les indicateurs de code soient automatiquement affichés en cours de frappe. Si cette option est désactivée, vous pouvez accéder aux indicateurs de code en appuyant sur les touches Ctrl+Espace.

- **Activer automatiquement après**

Spécifiez le délai en millisecondes devant s'écouler entre le moment auquel vous commencez à saisir du code et le moment auquel les indicateurs de code doivent s'afficher. Par défaut, ce délai est de 100 millisecondes.

Voir aussi

« [Aide au codage dans Flash Builder](#) » à la page 105

Préférences générales de l'éditeur pour Flash Builder

Par défaut, Flash Builder réduit le code et met automatiquement les lignes en retrait en cours de saisie ou de collage de texte dans l'éditeur de source. Vous pouvez désactiver les préférences par défaut de ces fonctions.

Préférences générales de l'éditeur pour Eclipse

Pour définir d'autres préférences générales de l'éditeur, sélectionnez Préférences > Général > Editeurs.

Code ActionScript

Lorsque vous éditez des fichiers ActionScript dans l'éditeur de source, Flash Builder fournit des fonctions par défaut pour le renvoi à la ligne et l'organisation du code. Par défaut, Flash Builder place toutes les instructions d'importation en tête du fichier ActionScript et supprime toutes celles qui ne sont pas référencées dans le corps du code.

Par défaut, la documentation de référence ActionScript disponible s'affiche simultanément aux indicateurs de code ou en survolant un élément de langage avec le curseur de la souris.

Chacune de ces fonctions par défaut peut être désactivée.

Voir aussi

« [Aide au codage dans Flash Builder](#) » à la page 105

Mode Création

Flash Builder comporte les préférences suivantes pour le mode Création de l'éditeur de source.

- Activer automatiquement les vues liées à la création
Lors du passage en mode Création de l'éditeur de source, Flash Builder ouvre automatiquement les vues associées, telles que Propriétés, Etats, Composants et Aspects.
Désactivez cette préférence pour personnaliser le choix des vues qui s'ouvriront en mode Création.
- Activer l'alignement
En mode Création, Flash Builder comporte une grille invisible qui aligne automatiquement les composants ajoutés ou déplacés.
Désactivez cette préférence pour avoir une meilleure maîtrise du positionnement des composants.
- Générer un rendu des habillages à l'ouverture d'un fichier
Le mode Création représente les composants dotés d'un habillage placés dans la zone de création.
Pour désactiver le rendu des habillages des composants en mode Création, désélectionnez cette préférence.
- Réduire les expressions de liaison de données
En mode Création, les expressions de liaison des données ne reflètent généralement pas précisément la forme ou la taille des données auxquelles elles sont liées.
Activez cette préférence pour autoriser le mode Création de l'éditeur à rendre les composants liés à des expressions de liaison de données avec une plus grande précision.
- Toujours mettre à jour les références lors de la modification des ID dans la vue Propriétés
La modification d'un identifiant de composant dans la vue Propriétés peut avoir une incidence sur le code de l'espace de travail qui fait référence à cet identifiant.
Activez cette préférence pour que Flash Builder mette automatiquement à jour toutes les références aux ID de composant lorsque vous modifiez un ID de composant dans la vue Propriétés.

Code MXML

Lorsque vous éditez des fichiers MXML dans l'éditeur de source, Flash Builder fournit des fonctions par défaut de suggestion de code. Chacune de ces fonctions peut être désactivée.

Flash Builder définit également un comportement par défaut pour l'assistant de contenu lors de l'édition de fichiers MXML. Lorsqu'il est disponible, l'assistant de contenu affiche des indicateurs sur les éléments de langage pouvant être insérés dans le code. Pour parcourir les éléments affichés dans la fenêtre de l'assistant de contenu, appuyez sur les touches Ctrl+Espace.

Utilisez la page Avancées des préférences MXML pour désactiver l'affichage d'un type d'élément de langage dans le cycle de l'assistant de contenu.

Voir aussi

« [Aide au codage dans Flash Builder](#) » à la page 105

« [Utilisation de l'assistant de contenu](#) » à la page 108

Coloration de la syntaxe

La coloration de la syntaxe et la mise en évidence de texte dans les fichiers ActionScript, CSS et MXML sont définies par défaut. Ces paramètres par défaut peuvent être remplacés. Développez le nœud du langage et sélectionnez une fonction de langage pour remplacer la fonction par défaut.

Les préférences générales de l'éditeur pour Eclipse sont également disponibles. Voir :

- Préférences > Général > Editeurs de texte
- Préférences > Général > Couleurs et polices

Modèles de fichiers

Les modèles de fichiers interviennent lors de la création de nouveaux fichiers MXML, ActionScript et CSS et peuvent être personnalisés. Vous pouvez également importer et exporter des fichiers de modèles.

Dans la zone Types de fichier, sélectionnez le modèle d'un type de fichier. Cliquez sur Editer pour le modifier. Désactivez l'option Mise en retrait automatique des nouveaux fichiers si vous ne souhaitez pas faire appel aux préférences de mise en retrait de Flash Builder lors de la création de fichiers.

Vous pouvez modifier les préférences de mise en retrait de Flash Builder. Voir :

Préférences > Flash Builder > Mise en retrait

Voir aussi

« [Personnalisation des modèles de fichier](#) » à la page 127

« [Mise en retrait](#) » à la page 253

FlexUnit

Lorsque vous exécutez un test FlexUnit, Flash Builder est lancé par défaut en mode de débogage.

Cette préférence vous permet de modifier le mode de lancement de Flash Builder pour les tests FlexUnit. Les modes de lancement par défaut de Flash Builder sont les suivants :

- Mode Exécuter
Par défaut, ce mode lance la perspective Flash.
- Mode Déboguer
Par défaut, ce mode lance la perspective Débogage Flash.

- Mode Profiler

Par défaut, ce mode lance la perspective Profil Flash.

Les préférences Eclipse pour le lancement et les perspectives définissent les modes de lancement. Voir :

- Préférences > Exécution/Débogage > Lancement
- Préférences > Exécution/Débogage > Perspectives

Voir aussi

« [Environnement de test FlexUnit](#) » à la page 150

« [A propos des perspectives de Flash Builder](#) » à la page 7

Mise en retrait

Par défaut, la mise en retrait du code s'effectue par tabulations. Ce paramètre peut être modifié en faveur de l'utilisation d'espaces. La taille de la tabulation et de la mise en retrait par défaut équivaut à quatre espaces.

Sélectionnez ActionScript ou MXML sous Mise en retrait dans la boîte de dialogue Préférences afin d'afficher les paramètres correspondants. Vous pouvez également personnaliser les méthodes heuristiques de mise en retrait.

SDK Flex installés

Flash Builder installe les SDK Flex 4.1 et 3.5. Il utilise en outre le SDK Flex 4.1 pour les projets pour lesquels aucun SDK n'a été spécifié.

Vous pouvez ajouter et supprimer des SDK ainsi que modifier le SDK utilisé par défaut pour les projets n'en spécifiant aucun. Flash Builder vous demande de choisir le SDK que vous souhaitez utiliser à l'importation de projets créés avec l'une de ses versions antérieures. Vous pouvez désactiver l'affichage de cette invite à l'importation de projets.

Les types de SDK disponibles sont les suivants :

SDK Adobe Flex gratuit Versions publiées de produits Adobe officiels. Ces SDK contiennent différents composants source ouverts et fermés.

SDK Flex Open Source Ce type est destiné aux utilisateurs souhaitant un package ne contenant que du code « open source ».

Modules complémentaires Adobe pour SDK Flex « open source » Ce package contient tous les éléments qui se trouvent dans le SDK Flex Adobe et non dans le SDK Flex « open source ». Le téléchargement de ce fichier vous permet d'aligner le SDK Flex « open source » sur le SDK Flex Adobe.

Ajout d'un SDK Flex

1 Téléchargez un SDK Flex vers un emplacement temporaire de votre système :

Vérifiez les pages de téléchargement pour obtenir des informations sur les différents types de version, les conditions requises et les licences.

- [Download Flex 4 SDKs](#).
- [Download Flex 3 SDKs](#).

***Remarque :** si vous souhaitez utiliser une version stable et testée, téléchargez une version d'étape. Cette version a été déclarée comme étant une version majeure par l'équipe de développement. Elle comporte une version signée des RSL de la structure Flex (requis pour utiliser la mise en cache RSL).*

- 2 Les SDK Flex téléchargés sont au format ZIP. Dézippez ce fichier dans un nouveau dossier de votre système.
Pour plus de clarté, vous pouvez attribuer au dossier de niveau supérieur le numéro de version du SDK.
Les SDK sont généralement placés dans le répertoire d'installation de Flash Builder à l'emplacement suivant :

```
<Flash Builder Install Directory>/sdks/
```
- 3 Dans les préférences de Flash Builder, sélectionnez Flash Builder > SDK installés.
- 4 Cliquez sur Ajouter, puis recherchez le SDK Flex dézippé. Cliquez sur Appliquer.
- 5 Pour chaque projet utilisant le SDK Flex, accédez à cette boîte de dialogue et sélectionnez le SDK qui sera utilisé par défaut. Cliquez sur OK.

Moniteur de réseau

La page de préférences Moniteur de réseau répertorie les ports sur lesquels le Moniteur de réseau intercepte des événements et écoute les requêtes HTTP.

Par défaut, le Moniteur de réseau supprime au démarrage toutes les entrées. Cette préférence peut être désactivée.

Vous pouvez également activer les préférences suivantes :

- Suspendre la surveillance au démarrage
- Ignorer les vérifications de sécurité SSL

L'activation de cette préférence est utile pour surveiller le trafic réseau à partir d'un serveur autosigné.

Profileur

Par défaut, le profilage de la mémoire et le profilage des performances sont tous les deux activés.

- Profilage de la mémoire

Le profilage de la mémoire permet de déterminer la quantité de mémoire utilisée par un objet ou un type d'objet dans l'application. Faites appel au profilage de la mémoire pour diminuer la taille des objets, réduire le nombre d'objets créés ou permettre le nettoyage des objets dans la mémoire en supprimant les références à ces objets.

La quantité de mémoire utilisée par le profilage de la mémoire est bien supérieure à celle utilisée par le profilage des performances et peut donc ralentir votre application.

- Profilage des performances

Le profilage des performances recherche les méthodes de l'application dont l'exécution est lente et qui sont susceptibles de pouvoir être améliorées ou optimisées.

Vous pouvez définir les préférences suivantes pour le profileur :

- Connexions

Spécifiez le numéro de port que Flash Builder écoute lors du profilage d'une application. Le numéro de port par défaut est 9999. Ce port ne peut pas être défini sur 7935, ce dernier étant utilisé par le débogueur Flash.

- Filtres d'exclusion

Définit les packages par défaut qui sont exclus des vues du profileur.

- Filtres d'inclusion

Définit les packages par défaut qui sont inclus dans les vues du profileur. Tous les autres packages sont exclus.

- Références d'objet

Spécifie le nombre de chemins de références arrière à des occurrences d'un objet à afficher. Les chemins de références arrière permettent de déterminer si un chemin possède une référence arrière à la récupération de place (racine de la RP). Une occurrence dont la libération était attendue mais qui possède des références à la racine de la RP indique une perte de mémoire.

Par défaut, le profileur affiche dix chemins de références arrière. Vous pouvez augmenter cette valeur ou activer l'option d'affichage de tous les chemins de références arrière.

- Lecteur/Navigateur

Vous pouvez spécifier quel lecteur Adobe Flash Player autonome et quel navigateur Web vous voulez utiliser pour le profilage des applications externes. Utilisez une version de débogage de Flash Player pour profiler une application.

Lors du profilage d'applications externes, Flash Builder utilise par défaut les lecteurs Flash Player suivants :

- URL vers un fichier SWF

Flash Builder lance le fichier SWF de l'application en utilisant le navigateur par défaut du système.

- Emplacement du fichier SWF sur le système de fichiers

Flash Builder ouvre l'application avec la version par défaut du débogueur du lecteur Flash Player autonome.

Voir aussi

« [Utilisation du profileur](#) » à la page 162

« [A propos des vues du profileur](#) » à la page 168

Extension de Flash Builder

L'API d'extensibilité de Flash Builder vous permet d'étendre Flash Builder de manière à prendre en charge les composants personnalisés. Elle permet notamment d'étendre les vues Création et Propriétés afin de garantir une prise en charge correcte des composants personnalisés.

Vous pouvez également étendre la prise en charge par Flash Builder des services disponibles à partir de l'assistant des services.

Pour plus d'informations, voir [Flash Builder 4 Extensibility API Reference](#).