

FLASH[®] LITE[™] 2.x アプリケーションの開発

Flash® Lite™ 2.x アプリケーションの開発

本マニュアルがエンドユーザー使用許諾契約を含むソフトウェアと共に提供される場合、本マニュアルおよびその中に記載されているソフトウェアは、エンドユーザー使用許諾契約にもとづいて提供されるものであり、当該エンドユーザー使用許諾契約の契約条件に従ってのみ使用または複製することが可能となるものです。当該エンドユーザー使用許諾契約により許可されている場合を除き、本マニュアルのいかなる部分といえども、**Adobe Systems Incorporated** (アドビ システムズ社) の書面による事前の許可なしに、電子的、機械的、録音、その他いかなる形式・手段であれ、複製、検索システムへの保存、または伝送を行うことはできません。本マニュアルの内容は、エンドユーザー使用許諾契約を含むソフトウェアと共に提供されていない場合であっても、著作権法により保護されていることにご留意ください。

本マニュアルに記載される内容は、あくまでも参照用としてのみ使用されること、また、なんら予告なしに変更されることを条件として、提供されるものであり、従って、当該情報が、アドビ システムズ社による確約として解釈されてはなりません。アドビ システムズ社は、本マニュアルにおけるいかなる誤りまたは不正確な記述に対しても、いかなる義務や責任を負うものではありません。

新しいアートワークを創作するためにテンプレートとして取り込もうとする既存のアートワークまたは画像は、著作権法により保護されている可能性のあるものであることをご留意ください。保護されているアートワークまたは画像を新しいアートワークに許可なく取り込んだ場合、著作権者の権利を侵害することがあります。従って、著作権者から必要なすべての許可を必ず取得してください。

例として使用されている会社名は、実在の会社・組織を示すものではありません。

Adobe、Adobe ロゴ、Flash Lite および Flash は、アドビ システムズ社の米国ならびに他の国における商標または登録商標です。

サードパーティの情報

本マニュアルには、アドビ システムズ社が管理していない、サードパーティの Web サイトへのリンクが掲載されていますが、アドビ システムズ社はいかなるリンク先サイトの内容についても責任を持ちません。本マニュアルに記載されているサードパーティの Web サイトには、自己責任においてアクセスしてください。アドビ システムズ社はこれらのリンクを便宜上の目的においてのみ掲載しています。リンクを掲載することにより、アドビ システムズ社がこれらのサードパーティのサイトの内容について何らかの責任を持つことを示すものではありません。



Sorenson™ Spark™ ビデオ圧縮および圧縮解除テクノロジーは、Sorenson Media, Inc. のライセンス供与によって提供されます。

Fraunhofer-IIS/Thomson Multimedia: MPEG レイヤー 3 音声圧縮テクノロジーは、Fraunhofer IIS および Thomson Multimedia (<http://www.iis.fhg.de/amm/>) によりライセンス供与されています。

Independent JPEG Group: 本ソフトウェアの一部は、Independent JPEG Group による著作物に基づきます。

Nellymoser, Inc.: 音声圧縮および圧縮解除テクノロジーは、Nellymoser, Inc. (<http://www.nellymoser.com>) のライセンス供与によって提供されます。

Opera® browser Copyright © 1995-2002 Opera Software ASA and its suppliers. All rights reserved.

Macromedia Flash 8 ビデオは、On2 TrueMotion ビデオテクノロジーを利用しています。© 1992-2005 On2 Technologies, Inc. All Rights Reserved. <http://www.on2.com>.

Visual SourceSafe は、米国およびその他の国における Microsoft Corporation の登録商標または商標です。

更新情報およびその他のサードパーティのコード情報は、http://www.adobe.com/go/thirdparty_jp/ で入手できます。

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

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

目次

第1章：Flash Lite 2.x の概要	7
Flash Lite 2.x の機能について	7
Flash Lite のコンポーネントについて	12
第2章：インタラクティブ機能とナビゲーション機能の作成	13
Flash Lite でのユーザー操作について	13
Flash Lite でサポートされているキー	14
Flash Lite におけるデフォルトのナビゲーションの使用	15
キーイベントおよびボタンイベントの処理	23
第3章：テキストとフォントの操作	37
Flash Lite のテキストについて	37
テキストの作成とフォーマット	39
テキスト入力フィールドの使用	39
Flash Lite でのフォントのレンダリング方法	47
テキストフィールドのサンプルアプリケーション	51
スクロールテキストの作成	53
第4章：サウンド、ビデオおよびイメージの操作	55
Flash Lite のサウンドについて	55
デバイスサウンドの使用	56
Flash ネイティブサウンドの使用	63
デバイスビデオの使用	66
外部イメージのロード	77
第5章：BREW 用 Flash Lite アプリケーションの開発	79
BREW の基礎	79
BREW 用のシステムのセットアップ	84
BREW 用 Flash Lite ファイルのオーサリング	87
BREW 用 Flash Lite ファイルのパブリッシュ	91
BREW デバイスへのファイルのアップロード	103

第 6 章：コンテンツのパフォーマンスとメモリ使用の最適化	111
第 7 章：Flash Lite コンテンツのプレビュー	113
Flash Lite のプレビュー機能の概要	113
エミュレータでサポートされていないプレビュー機能	114
エミュレータの使用	114
エミュレータでのデバイスビデオの再生について	123
エラー	123
付録 A: 警告メッセージとエラーメッセージ	127
Adobe Device Central エミュレータのエラーメッセージと警告 メッセージ	127
索引	137

Flash Lite 2.x の概要

Adobe® Flash® Lite™ は、デバイス用に設計された Flash Player のバージョンです。本マニュアルでは、アドビ システムズ社の Macromedia® Flash® Lite™ 2.0 と Macromedia® Flash® Lite™ 2.1 ソフトウェアについて説明しますが、これらをまとめて Flash Lite 2.x と呼びます。

Flash Lite 2.x は Macromedia® Flash® Player 7 を基にしており、Flash Player 7 のほとんどの機能をサポートしています。さらに、Flash Player 7 では使用できないモバイル開発専用の機能も含まれています。たとえば、Flash Player 7 でネイティブにサポートされていないデバイス固有のメディアタイプ (イメージ、サウンド、ビデオ) をロードできます。

また、電話をかけたり、テキストメッセージを送信したりするデバイス統合機能も備えています。

本章では、Flash Lite 2.x の新機能の概要を説明し、Flash Lite でサポートされていない Flash Player 7 の機能について説明します。この章では、次のトピックについて説明します。

Flash Lite 2.x の機能について	7
Flash Lite のコンポーネントについて	12

Flash Lite 2.x の機能について

Macromedia® Flash® Lite™ 1.0 および Macromedia® Flash® Lite™ 1.1 は Macromedia® Flash® Player 4 を基にしていたますが、Flash Lite 2.0 および 2.1 (まとめて 2.x と呼びます) は Flash Player 7 を基にしており、XML 処理や ActionScript 2.0 など、Flash Player 7 で使用可能な機能のほとんどをサポートしています。Flash Lite 2.x は、Flash Player 7 では使用できない、モバイルアプリケーション用に設計されたいくつかの機能も備えています。この項の残りの部分では、Flash Lite 2.0 および 2.1 の新機能について説明します。

Flash Lite 2.0 の新機能

Flash Lite 2.0 には次の新機能があります。

- ActionScript 2.0
- デバイスビデオの再生
- ローカルでの永続データストレージ (Flash Lite 共有オブジェクト)
- デバイス固有のサウンドおよびイメージ形式のロードのサポート
- 新しいシステム機能の情報
- QWERTY キーボードサポートおよび最大 11 個のソフトキーのサポートを含む、追加のデバイスキーのサポート
- リッチテキスト形式 (部分的サポート)
- バックライトの長さを制御し、独自のフォーカス矩形の色を設定する機能
- クラス、インターフェイス、および厳密な型指定などの高度なプログラミング手法を使用可能にする、ActionScript 2.0 のサポート
- 同期されたデバイスサウンド
- XML 処理のサポート

Flash Player 7 の以下の機能は、Flash Lite 2.0 では使用できません。

- Flash Player 7 で使用可能ないくつかの ActionScript クラスは、Flash Lite 2.0 ではまったくサポートされていないか、一部しかサポートされていません。使用可能な ActionScript の詳細については、『Flash Lite 2.0 ActionScript 入門』を参照してください。
- XMLSocket クラスを使用したソケット通信 (Flash Lite 2.1 では使用可能)
- Flash Media Server との通信のサポート
- リモート共有オブジェクト (ローカル共有オブジェクトは部分的にサポートされています)
- Flash Video (FLV) 再生のネイティブサポート
- Flash Application Protocol (Flash Remoting で使用されるバイナリデータ通信プロトコル) のサポート
- テキストフィールドでの CSS (Cascading Style Sheet) フォーマット
- デバイスフォントのマスク
- 高品質レンダリング時のビットマップスムージング

Flash Lite 2.1 の新機能

Flash Lite 2.1 では、インラインテキスト入力サポートが追加されました。Flash Lite の以前のバージョンでは、テキスト入力フィールドでデバイスの汎用テキスト入力ダイアログボックスを使用しました。Flash Lite 2.1 では、テキスト入力フィールドを直接編集できます。

×
中

複雑な言語 (ヘブライ語、アラビア語、ペルシャ語、アジアの言語の一部など) をサポートしているデバイスは、インラインテキスト入力をサポートしていません。Flash Lite 2.1 におけるこれらのデバイスのテキスト入力フィールドの機能は、Flash Lite 2.0 の場合と同じです。

Flash Lite 2.1 では、XMLSocket クラスを使用したソケット通信をサポートしているデバイスでこのソケット通信が可能です。詳細については、『Flash Lite 2.x ActionScript リファレンスガイド』を参照してください。テキスト入力フィールドの詳細については、[37 ページの「テキストとフォントの操作」](#)を参照してください。

Flash Lite 2.1 には、QUALCOMM Incorporated によって開発された BREW® (Binary Runtime Environment for Wireless®) を使用するデバイスで実行される、Flash Lite アプリケーションを開発するツールも追加されています。これらのツールの詳細については、[79 ページ、第 5 章の「BREW 用 Flash Lite アプリケーションの開発」](#)を参照してください。

Flash Lite 2.0 ActionScript

Flash Lite 2.0 ActionScript は、Flash Lite 2.0 および Flash Lite 2.1 アプリケーションで使用するスクリプト言語です。この言語は、Flash Player 7 で使用される ActionScript の一部を共有しています。Flash Lite 2.0 には、デバイスの情報を収集したり、電話をかけたり、バックライトの長さを制御したりする、いくつかの ActionScript 追加機能や拡張機能も含まれています。

Flash Lite 2.0 用のアプリケーションを開発するときは、ActionScript 2.0 または ActionScript 1.0 のシンタックスを使用できます。ActionScript 2.0 は、クラス、インターフェイス、および厳密な型指定のオーサリングもサポートしています。ActionScript 2.0 のシンタックスを使用すると、ActionScript コンパイラからより有用なデバッグ情報が提供され、プログラムの設計が向上します。

Flash Lite 2.0 ActionScript の学習については、以下のマニュアルおよびトピックを参照してください。

- Flash Lite 2.x ActionScript 入門
- Flash Lite 2.x ActionScript リファレンスガイド
- 『Flash ユーザーガイド』の「ActionScript 2.0 の学習」

デバイスビデオの再生

Flash Lite 2.0 では、ターゲットデバイスでネイティブにサポートされている任意の形式でビデオを再生できます。たとえば、一部のデバイスでは 3GP ビデオ形式でビデオが録画および再生され、他のデバイスでは AVI または MPEG 形式がサポートされます。再生時には、元のビデオデータが Flash Lite からデバイスに渡され、そのデバイスによりデータがデコードされて画面に直接レンダリングされます。アプリケーションにビデオを組み込む方法は、以下のとおりです。

- 元のビデオデータを SWF ファイルにバンドルします。
- 外部のビデオファイルを、デバイスのメモリカードまたはネットワークからロードします。

ビデオの再生を制御するには、ActionScript Video オブジェクトを使用します。Video オブジェクトは Flash Player 6 で初めて使用可能になったオブジェクトですが、Flash Lite 2.0 ではビデオを制御するためのメソッド、つまり、Video.play() メソッドと Video.pause() メソッドが追加されました。System.capabilities.videoMIMEtypes 配列を使用して、デバイスがサポートしているビデオ形式を判別することもできます。

Video オブジェクト、および Flash Lite でのビデオの使用方法の詳細については、[66 ページの「デバイスビデオの使用」](#)を参照してください。

デバイス固有のサウンドおよびイメージ形式のロード

Flash Lite 2.0 では、デバイスがサポートしている形式の任意のイメージまたはサウンドファイルをロードできます。外部イメージをロードするには、loadMovie() グローバル関数または MovieClip.loadMovie() メソッドを使用します。たとえば、デバイスが PNG ファイル形式をサポートしている場合、次のコードを使用して、Web サーバー上の PNG ファイルをムービークリップインスタンス image_mc にロードできます。

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

外部サウンドをロードするには、Sound.loadSound() メソッドを使用します。Flash Lite 2.0 では、このメソッドを使用して、デバイスがサポートしている任意のサウンド形式 (MIDI、SMAF など) をロードできます。サウンドを再生するには、その外部デバイスサウンドが完全にメモリにロードされている必要があります。

Flash Lite 1.x と同様、Flash Lite 2.0 でも SWF ファイルにバンドルされているデバイスサウンドを再生できます。詳細については、[56 ページの「バンドルされたデバイスサウンドの使用」](#)を参照してください。

外部イメージおよびサウンドのロードの詳細については、以下のトピックを参照してください。

- [77 ページの「外部イメージのロード」](#)
- [61 ページの「外部デバイスサウンドの再生」](#)

Flash Lite 共有オブジェクト

Flash Lite 共有オブジェクトを使用すると、ユーザーのデバイスにデータを永続的に保存できます。たとえば、共有オブジェクトを使用すると、ユーザーの環境設定やゲームスコアなど、アプリケーションセッション間で共有する情報を保存できます。Flash Lite 共有オブジェクトの読み取りおよび書き込みには、SharedObject クラスを使用します。Flash Lite 共有オブジェクトの使用の詳細については、『Flash Lite 2.x ActionScript リファレンスガイド』の SharedObject クラスを参照してください。

×
中

Flash Lite 2.0 で共有オブジェクトを実装しても、それにより複数の SWF ファイルが同じデータを共有できるようになるわけではありません。また、Flash Lite 2.0 では、Flash Media Server とのリモート共有オブジェクトがサポートされていません。

同期されたデバイスサウンド

Flash Lite の以前のバージョンでは、タイムライン上のアニメーションと同期できるのは Flash のネイティブサウンドだけであり、デバイスサウンドは Flash Lite によりネイティブに再生されるのではなく、デバイスにより直接再生されるので同期できませんでした。Flash Lite 2.0 では、新しい `_forceframerate` プロパティを使用して、デバイスサウンドをタイムラインと同期できます。このプロパティを `true` に設定すると、SWF ファイルで指定されたフレームレートを維持するため、必要に応じてアニメーションからフレームが削除されます。詳細については、『Flash Lite 2.x ActionScript リファレンスガイド』の [62 ページの「デバイスサウンドとアニメーションの同期について」](#) および「`_forceframerate` プロパティ」を参照してください。

テキスト機能

Flash Lite 2.0 には、テキスト処理に関連する以下の新しい特長があります。

- Flash Lite 2.0 のすべてのテキストは Unicode に基づいています。
- Flash Lite 2.0 では、HTML 形式および TextFormat ActionScript クラスを部分的にサポートしています。

Flash Lite 2.x でテキストフィールドを操作する方法の詳細については、[37 ページの「テキストとフォントの操作」](#)を参照してください。

追加のキーサポート

Flash Lite 2.0 では、QWERTY キーボードや最大 12 個のソフトキー (標準の左および右のソフトキーを含む) のサポートなど、デバイスキーのサポートが追加されています。

Flash Lite のコンポーネントについて

Flash でインストールされるコンポーネント (DataGrid、Accordion など) は、Flash デスクトップアプリケーション用に設計されています。それらのコンポーネントにはメモリや処理能力の要件があるため、通常、Flash Lite アプリケーションで使用することはできません。Flash Lite アプリケーションでは、標準のユーザーインターフェイスコンポーネントを使用しないことをお勧めします。

インタラクティブ機能とナビゲーション機能の作成

ユーザーが Adobe Flash Lite アプリケーションを操作するには、画面上のどのオブジェクトに現在フォーカスがあるかを判別して、オブジェクト間を移動し、オブジェクトや別のキーを選択してアクションを開始できるようにする必要があります。この基本原則はデスクトップアプリケーションについても同じですが、モバイルデバイスではいくつかの機能が異なります。

この章では、次のトピックについて説明します。

Flash Lite でのユーザー操作について.....	13
Flash Lite でサポートされているキー.....	14
Flash Lite におけるデフォルトのナビゲーションの使用	15
キーイベントおよびボタンイベントの処理	23

Flash Lite でのユーザー操作について

Flash Lite は、デバイスのキーを使用した移動やユーザー操作、または対応するデバイスではペンやタッチスクリーンインターフェイスを使用したユーザー操作をサポートしています。アプリケーションで利用できるオプションは、ターゲットデバイスやコンテンツタイプによって異なります。コンテンツタイプの詳細については、『Flash Lite 2.x ファーストステップガイド』の「Flash Lite コンテンツタイプについて」を参照してください。

Flash Lite アプリケーションにキーによる操作機能を追加する最も簡単な方法は、デスクトップアプリケーションの矢印キーや Tab キーおよび Shift + Tab キーと同じように、デバイスの 4 方向キーを使用する、デフォルトのナビゲーションを使用することです。ユーザーは目的のオブジェクトにフォーカスを移動してから、選択キーを押します。アプリケーションには、これらのボタンイベントに応答するためのイベントハンドラコードがあります。Flash Lite のデフォルトナビゲーションでは、ボタン、テキスト入力フィールド、および必要に応じてムービークリップを使用して操作します。通常、これはメニュー操作などの単純なユーザー操作に最適です。デフォルトのナビゲーションの詳細については、[15 ページの「Flash Lite におけるデフォルトのナビゲーションの使用」](#)を参照してください。

アプリケーションでは、ユーザーが特定のキーを押したときに Flash Lite により生成される任意のキー押下イベントに応答することもできます。イベントに基づいたナビゲーションを使用すると、複雑なユーザー操作モデルを持つゲームなどの Flash Lite アプリケーションを作成できます。イベントの詳細については、[23 ページの「キーイベントおよびボタンイベントの処理」](#)を参照してください。

Flash Lite でサポートされているキー

標準的な電話機の英数字キーに加えて、多くのモバイルデバイスには、デバイスの画面上を移動してアイテムを選択するための方向キーと、2つ以上のソフトキーがあります。デバイスのソフトキーは、画面の表示内容に応じて用途が変わる多機能キーです。

通常、方向キーには、4つの方向キー(上、下、左、右)と1つの選択キー(通常、矢印キーの中央に配置)があります。アプリケーションごとに各キーの用途は異なる場合があります。たとえば、ゲームでは、方向キーを使用して画面上のキャラクタを移動し、選択キーを押してキャラクタをジャンプさせるなどの別の操作を実行することができます。

次の図は、汎用キーおよび実際のデバイスの一般的なキーを示しています。



すべてのデバイスおよび Flash Lite コンテンツタイプで、これらのすべてのキーがサポートされているわけではありません。たとえば、2方向のナビゲーションをサポートするデバイスでは、左右の方向キーがサポートされていません。詳細については、[16 ページの「デフォルトのナビゲーションモード」](#)を参照してください。また、デバイスによってはソフトキーを使用できない場合もあります。

Flash Lite は、モバイルデバイスの次のキーをサポートしています。

説明	キー	利用可能なデバイス
数字、*、#	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #	すべてのデバイスで利用可能です。
5 方向キー	選択キー、上方向キー、下方向キー 左方向キー、右方向キー	すべてのデバイスで利用可能です。 4 方向のナビゲーションをサポートするデバイスでのみ利用可能です (16 ページの「デフォルトのナビゲーションモード」 を参照)。
ソフトキー	左方向キー、右方向キー	SetSoftKeys コマンドをサポートするデバイスで利用可能です。

説明	キー	利用可能なデバイス
キーボードのキー	SOFT3 ～ SOFT12 キー	ソフトキーが3 つ以上あるデバイスで利用可能です。
	!, “, #, \$, %, &, ‘, (,), *, +, ,, -, ., /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, :, ;, <, +, \, ?, @, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, [, \,], ^, _ , ‘, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, {, , }, ~, Backspace	QWERTY キーボードを使用するデバイスで利用可能です。

次の `System.Capabilities` プロパティを使用すると、デバイスで利用可能な方向キーおよび選択キーのオプションを決定できます。

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

`System.Capabilites` クラスの詳細については、『Flash Lite 2.x ActionScript リファレンスガイド』を参照してください。

Flash Lite におけるデフォルトのナビゲーションの使用

デスクトップの Flash アプリケーションでは、Tab キーおよび Shift + Tab キーを使用して画面のオブジェクト間でフォーカスを切り替えることができます。他のアプリケーションでは、矢印キーが同様の動きをする場合もあります。Flash Lite では、デバイスの方向キーは、Flash デスクトップアプリケーションでの矢印キー、Tab キー、および Shift + Tab キーと同じ動きをします。対象オブジェクトにフォーカスを移動した後、ユーザーは選択キーを押して、アプリケーションでアクションをトリガできます。ボタンやムービークリップが選択されたときに応答するイベントハンドラもユーザーが定義します。詳細については、[23 ページの「ボタンイベントの処理」](#)を参照してください。

Flash Lite のデフォルトのナビゲーションは、ボタンとテキスト入力フィールドで使用します。ムービークリップの `tabEnabled` プロパティが `true` に設定されている場合、またはムービークリップにイベントハンドラが関連付けられており、`tabEnabled` プロパティが `false` に設定されていない場合は、ムービークリップも含まれます。

テキスト入力フィールドにフォーカスが置かれているときにユーザーが選択キーを押すと、Flash Lite ではデバイスの汎用テキスト入力ダイアログボックスが開き、ユーザーはそこにテキストを入力できます。

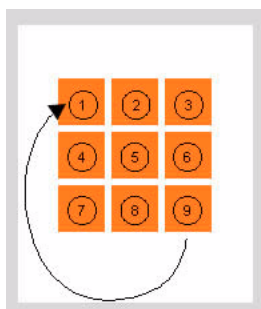
デフォルトのナビゲーションの使用例については、[26 ページの「ボタンとデフォルトのナビゲーションを使用した単純なメニューの作成」](#)を参照してください。

デフォルトのナビゲーションモード

Flash Lite でサポートされているデフォルトのナビゲーションモードは、2 方向、4 方向、および折り返し付き 4 方向の 3 種類です。デバイスおよび Flash Lite コンテンツタイプに応じて、サポートされるナビゲーションモードは異なります。特定のデバイスおよびコンテンツタイプのナビゲーションモードの確認については、[123 ページの「エミュレータでのデバイスビデオの再生について」](#)を参照してください。

Flash Lite の **2 方向ナビゲーション**は、Flash デスクトップアプリケーションのタブナビゲーションと似ており、Tab キーおよび Shift + Tab キーを使用して画面上のオブジェクト間を移動します。デバイスの下方向キーは Tab キーに対応し、上方向キーは Shift + Tab キーに対応します。

2 方向ナビゲーションにおけるデフォルトのタブ順序は、通常、左から右へ、上から下へとなっています。たとえば、次の図は、Flash Lite アプリケーションにおける 3 x 3 のグリッドの Button オブジェクトを表しています。各ボタン上の数字は、デバイスの下方向キーを繰り返し押したときに、各ボタンがフォーカスを得る順番を表しています。右下隅のボタンがフォーカスを取得した後、下方向キーを押すと、フォーカスは左上隅のボタンに折り返されます。



2 方向のナビゲーションにおけるタブ順序の例

Button、MovieClip、および TextField オブジェクトの tabIndex プロパティを使用すると、2 方向のナビゲーションのタブ順序をカスタマイズできます。詳細については、[22 ページの「2 方向のナビゲーションでのタブ順序の制御について」](#)を参照してください。

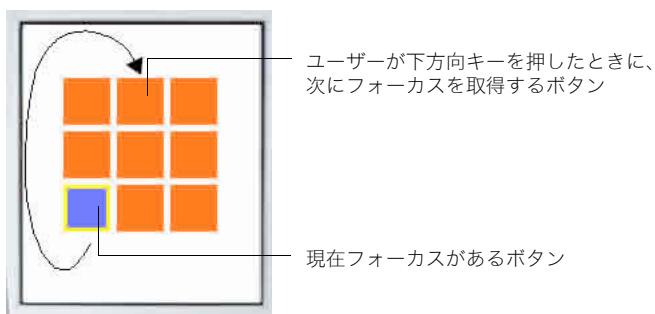
2 方向ナビゲーションの例については、Flash Lite のサンプルとチュートリアルページ (http://www.adobe.com/go/learn_flt_samples_and_tutorials_jp) を参照してください。ActionScript バージョンの .zip ファイルを探してダウンロードおよび解凍し、"Samples" フォルダを選択して "2-way.fla." というサンプルファイルにアクセスします。

Flash Lite の **4 方向ナビゲーション**は、デスクトップコンピュータのキーボードの矢印キーを使用して画面上のオブジェクト間を移動するのに似ています。デバイスの上、下、左、右の方向キーは、コンピュータのキーボードの 4 つの矢印キーに対応しています。方向キーを押すと、その方向にオブジェクトがある場合、そのオブジェクトにフォーカスが移動します。その方向にオブジェクトがない場合は、フォーカスは現在のオブジェクトから移動しません。

× 中	4 方向ナビゲーションに対応しているデバイスでは、tabIndex プロパティはサポートされていません。ただし、tabEnabled および tabChildren プロパティはサポートされています。これらのプロパティは、Flash デスクトップアプリケーションの場合とは動作が異なります。
--------	---

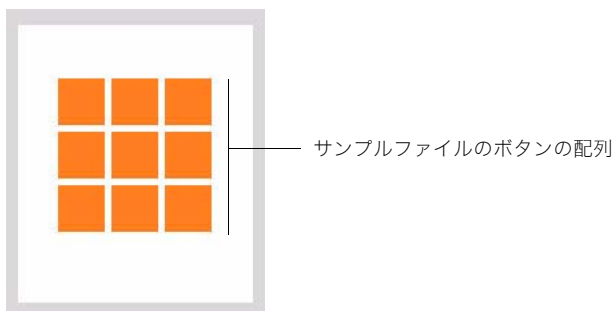
4 方向ナビゲーションの例については、Flash Lite のサンプルとチュートリアルページ (http://www.adobe.com/go/learn_flt_samples_and_tutorials_jp) を参照してください。ActionScript バージョンの .zip ファイルを探してダウンロードおよび解凍し、"Samples" フォルダを選択して "4-way.fla." というサンプルファイルにアクセスします。

折り返し付き 4 方向ナビゲーションは、標準的な 4 方向ナビゲーションと 2 方向ナビゲーションを組み合わせたようなものです。前述の標準的な 4 方向ナビゲーションのように、ユーザーはデバイスの 4 方向ナビゲーションを使用してフォーカスを移動します。ただし、折り返し付き 4 方向ナビゲーションでは、2 方向ナビゲーションと同様に、フォーカスが画面の反対側のオブジェクトに折り返されます。たとえば、次の図では、現在フォーカスのあるボタンが画面の左下隅に表示されています。ユーザーが下方向キーを押すと、最上行の中央のボタンが次にフォーカスを受け取ります。



"2-way fla" と "4-way fla" という名前のサンプルファイルを使用すると、Adobe Device Central エミュレータで 2 方向および 4 方向のナビゲーションモードのビヘイビアをテストできます。サンプルファイルは、http://www.adobe.com/go/learn_flt_samples_and_tutorials_jp にあります。「サンプルとチュートリアル」ページで、Flash Lite バージョンの .zip ファイルを探してダウンロードおよび解凍し、"Samples" フォルダを選択してサンプルファイルにアクセスします。各サンプルファイルは、前に説明したように、同じ 3 × 3 グリッドのボタンで構成されています。それぞれのサンプルファイルの相違点は、各ナビゲーションモード (2 方向または 4 方向) をサポートする Flash Lite コンテンツタイプとデバイスの組み合わせに応じて、各 FLA ファイルが設定されていることです。

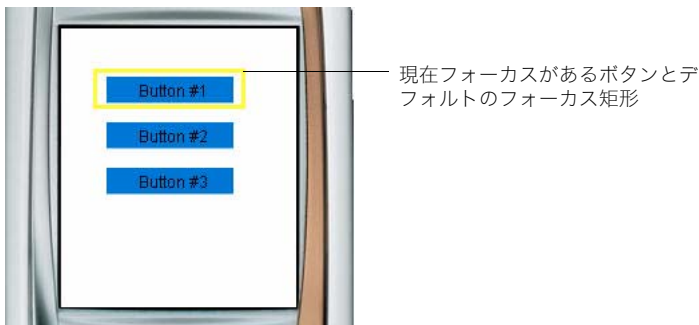
各サンプルファイルを使用するには、ファイルを Flash で開いて Adobe Device Central エミュレータでプレビューします ([制御]-[ムービープレビュー])。エミュレータの矢印キーをクリックするか、キーボードの矢印キーを押して、ナビゲーションモード別にユーザーナビゲーションの動作を確認します。



フォーカス矩形について

デフォルトでは、Flash Lite でフォーカスがあるボタンまたはテキスト入力フィールドの回りに黄色の矩形が表示されます。ムービークリップの `tabEnabled` プロパティが `true` に設定されている場合、またはムービークリップにイベントハンドラが関連付けられており、`tabEnabled` プロパティが `false` に設定されていない場合は、ムービークリップも含まれます。

フォーカス矩形により、デバイスの選択キーを押したときに画面上で応答するオブジェクトを確認できます。たとえば、次の図では、現在フォーカスがあるボタンの周囲にフォーカス矩形が表示されています。



ボタンとムービークリップの場合、フォーカス矩形の境界ボックスはオブジェクトのヒット領域によって決まります。ヒット領域とは、Flash デスクトップアプリケーションで、マウスクリックに応答するボタンやムービークリップの一部を定義する非表示の領域です。テキスト入力フィールドの場合、フォーカス矩形の境界ボックスはテキストフィールドのサイズに対応します。

フォーカス矩形の色をカスタマイズしたり、フォーカス矩形を無効にしたりできます。詳細については、[20 ページの「フォーカス矩形のカスタマイズ」](#)を参照してください。

デフォルトナビゲーションの使用に関するガイドライン

次に、Flash Lite アプリケーションでデフォルトのナビゲーションを使用する際のガイドラインと考慮事項を示します。

- `_focusRect` を `false` に設定して、デフォルトのフォーカス矩形を無効にする場合、ボタン、テキスト入力フィールド、およびタブを有効にしたムービークリップに対して、必ず代替フォーカスインジケータを使用してください。ボタンの場合、ボタンの " オーバー " 状態に対するビジュアルエレメントを追加すると、この操作を行えます。オーバー状態は、ボタンオブジェクトのタイムラインの一部で、ボタンにフォーカスがあるときに表示されます。この手法の例については、[26 ページの「ボタンとデフォルトのナビゲーションを使用した単純なメニューの作成」](#)のサンプルアプリケーションを参照してください。テキスト入力フィールドの場合は、`Selection` オブジェクトを使用して、テキストフィールドがフォーカスを取得したときにそのことを判別し、必要なフォーカスインジケータを表示します。例については、[21 ページの「ActionScript を使用したフォーカス制御」](#)で説明しているサンプルアプリケーションを参照してください。

- 画面に少なくとも2つのオブジェクト (テキスト入力フィールド、ボタン、タブを有効にしたムービークリップ、またはその組み合わせ) を同時に配置してください。画面にテキスト入力フィールド、ボタン、タブを有効にしたムービークリップが1つしかない場合、ユーザーはフォーカスを変更できず、ユーザーインターフェイスが動かなくなってしまうと感じる可能性があります。アプリケーションの画面にユーザーが操作できるオブジェクトとしてボタンが1つしかない場合は、ボタンイベントを使用しないでキー押下イベントを検知するようにします。詳細については、[23 ページの「キーイベントおよびボタンイベントの処理」](#)を参照してください。
- 該当する場合は、`Selection.setFocus()` メソッドを使用して、初期フォーカスを画面上の特定のオブジェクトに設定することを検討してください。これにより、インターフェイスを通してユーザーを手助けし、実行するキーナビゲーションの回数を減らすことができます。たとえば、アプリケーションの画面にテキスト入力フィールドが1つあるとします。通常、テキストフィールドに値を入力する場合、ユーザーはまず方向キーを押してテキストフィールドにフォーカスを設定し、次に選択キーを押してテキスト入力ダイアログボックスを開きます。次の `ActionScript` を使用して、テキスト入力フィールドにフォーカスを自動的に設定できます。

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

`ActionScript` を使用したフォーカス制御の詳細については、[21 ページの「ActionScript を使用したフォーカス制御」](#)を参照してください。

- デフォルトのナビゲーションでは、画面上のオブジェクトの配置が重要です。画面上で各オブジェクトがオフセットに配置されている場合、フォーカスを得る順序がユーザーにはわかりにくいかもしれません。これを避けるには、できる限りオブジェクトのサイズを一定にして垂直方向または水平方向に配置します。2 方向ナビゲーションを使用すると、`tabIndex` プロパティで順序を制御することもできます。詳細については、[22 ページの「2 方向のナビゲーションでのタブ順序の制御について」](#)を参照してください。

フォーカス矩形のカスタマイズ

デフォルトのフォーカス矩形は黄色のハイライトで、現在選択されているボタンまたは入力テキストボックスを示します。ムービークリップの `tabEnabled` プロパティが `true` に設定されている場合、またはムービークリップにイベントハンドラが関連付けられており、`tabEnabled` プロパティが `false` に設定されていない場合は、ムービークリップも含まれます。詳細については、[18 ページの「フォーカス矩形について」](#)を参照してください。

フォーカス矩形のデフォルトのビヘイビアを無効にするには、グローバルプロパティ `_focusRect` を `false` に設定します。特定のボタンやムービークリップのフォーカス矩形を無効にすることもできます。詳細については、『*Flash Lite 2.x ActionScript リファレンスガイド*』の「`_focusrect` (`Button._focusrect` プロパティ)」および「`_focusrect` (`MovieClip._focusrect` プロパティ)」を参照してください。

フォーカス矩形の色は、デフォルトの黄色から任意の色に変更できます。色を変更するには、パラメータとして RGB 値を取得する `SetFocusRectColor` コマンドを使用します。たとえば、次のコードでは、フォーカス矩形の色を赤に設定します。

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

ActionScript を使用したフォーカス制御

ActionScript の `Selection` オブジェクトを使用すると、現在のフォーカスを取得および設定したり、オブジェクトがフォーカスを取得または失ったときに通知を受け取るように設定したりできます。この機能は、アプリケーションが最初にロードされたときに、特定のボタンに自動的にフォーカスを設定する場合などに便利です。また、画面上の特定のオブジェクトがフォーカスを取得または失ったときに通知するように設定すると、必要に応じて画面を更新することができます。

たとえば、次のコードでは、`Selection.setFocus()` メソッドを使用して、`login_btn` という名前のボタンインスタンスにフォーカスを設定します。

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

`Selection.onSetFocus` イベントリスナーを使用すると、フォーカスが変化したときにそのことを判別できます。たとえば、このイベントリスナーを使用すると、デフォルトのフォーカス矩形を使用する代わりに、テキスト入力フィールド用のカスタムのフォーカスマネージャを作成できます。フォーカスのある `TextField` オブジェクトの境界線の色を変更するカスタムフォーカスマネージャの作成手順を次に示します。完成版アプリケーション (`custom_focus_manager fla`) のサンプルについては、Flash Lite のサンプルとチュートリアルページ (http://www.adobe.com/go/learn_flt_samples_and_tutorials_jp) を参照してください。ActionScript バージョンの .zip ファイルを探してダウンロードおよび解凍し、"Samples" フォルダを選択してサンプルファイルにアクセスします。

カスタムテキスト入力フォーカスマネージャを作成するには：

1. 『Flash Lite 2.x ファーストステップガイド』の「Flash Lite ドキュメントテンプレートの作成」で作成した Flash Lite 2.0 テンプレートを使用して新規ドキュメントを作成し、"`custom_focus_manager fla`" という名前で保存します。
2. テキストツールを使用して、テキストフィールドをステージに作成します。
3. テキストフィールドを選択した状態で、プロパティインスペクタの [テキストの種類] ポップアップメニューから [テキスト入力] を選択し、[インスタンス名] テキストボックスに「`inputTxt_1`」と入力し、[テキストの周囲にボーダーを表示] オプションを選択します。
4. 同じ要領で、最初に作成したテキストフィールドの下に `inputTxt_2` という名前の別のテキスト入力フィールドを作成し、2 つ目のテキストフィールドの [テキストの周囲にボーダーを表示] オプションを選択します。
5. タイムラインで、ActionScript というレイヤーのフレーム 1 を選択します。

6. [アクション] パネルを開き ([ウィンドウ]-[アクション])、次のコードを入力 (またはコピー & ペースト) します。

```
// フォーカス矩形をグローバルに無効にします。
_focusrect = false;
// Selection リスナーオブジェクトを作成します。
var focusListener:Object = new Object ();
// onFocus メソッドを定義します。
focusListener.onSetFocus = function (oldFocus, newFocus) {
    // 選択インジケータを有効または無効にします。
    if (newFocus instanceof TextField) {
        // フォーカスを取得したテキストフィールドの境界線の色を赤に設定します。
        newFocus.borderColor = 0xFF0000;
    }
    if (oldFocus != undefined && oldFocus instanceof TextField) {
        // フォーカスを失ったテキストフィールドの境界線の色を黒に設定します。
        oldFocus.borderColor = 0x000000;
    }
};
// Selection オブジェクトにリスナーを追加します。
Selection.addListener (focusListener);
// アプリケーションのロード時の初期フォーカスを設定します。
Selection.setFocus (inputTxt_1);
// フルスクリーンモードを有効にします。
fscommand2 ("FullScreen", true);
```

7. 変更を保存し、エミュレータでアプリケーションをテストします ([制御]-[ムービープレビュー])。
8. エミュレータの上下の矢印キーを押して、2 つのテキストフィールド間でフォーカスを切り替えます。フォーカスがあるテキストフィールドは境界線が赤で、フォーカスがないテキストフィールドは境界線が黒になります。テキストフィールドにフォーカスがあるときに選択キーを押すと、テキスト入力ダイアログボックスが表示されます。

2 方向のナビゲーションでのタブ順序の制御について

Flash Lite の 2 方向のナビゲーションは、Flash のタブナビゲーションと似ており、tabIndex プロパティをサポートしています。このプロパティを使用すると、ボタン、ムービークリップ、およびテキスト入力フィールドのタブ順序を具体的に設定できます。4 方向ナビゲーションをサポートしているデバイスでは、tabIndex プロパティはサポートされません。そのため、4 方向ナビゲーションでは tabIndex プロパティを使用してタブ順序を設定することはできません。

2 方向ナビゲーションでタブ順序を制御するには、各オブジェクトの tabIndex プロパティに、デフォルトのナビゲーションでオブジェクトの順序を指定する数値を割り当てます。たとえば、アプリケーションにボタン (my_button)、ムービークリップ (my_movieclip)、およびテキスト入力フィールド (my_inputTxt) が含まれているとします。次のコードでは、フォーカスが最初にボタン、次にムービークリップ、最後にテキスト入力フィールドに置かれるようにタブ順序を設定します。

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

キーイベントおよびボタンイベントの処理

イベントハンドラおよびイベントリスナーでは、アプリケーションが、ユーザーおよびシステムが生成したイベントに応答する方法を指定します。たとえば、ボタンにフォーカスがある状態でユーザーが選択キーを押すと、onPress イベントが生成されます。Flash Lite アプリケーションでは、デフォルトのナビゲーションを使用したり関連イベントに応答したりするだけでなく、キー押下イベントを検知して応答することができます。

デバイスおよびコンテンツタイプによっては、一部のデバイスキーをサポートしていない場合もあります。たとえば、2 方向のナビゲーション (16 ページの「デフォルトのナビゲーションモード」を参照) をサポートするデバイスでは、Flash Lite は左および右矢印キーについてキー押下イベントを生成しません。利用可能なキーとその詳細の一覧については、14 ページの「Flash Lite でサポートされているキー」を参照してください。

この項では、次のトピックについて説明します。

- 23 ページの「ボタンイベントの処理」
- 26 ページの「ボタンとデフォルトのナビゲーションを使用した単純なメニューの作成」
- 31 ページの「キーリスナーによるキー押下イベントの処理」
- 32 ページの「ソフトキーの使用」

ボタンイベントの処理

ボタンを使用すると、Flash Lite アプリケーションにインタラクティブ機能を簡単に追加できます。Flash Lite は、デスクトップコンピュータの Flash Player と同じボタンイベントをサポートしています。ただし、イベントによっては (onDragOut など)、マウスやペインインターフェイスのあるデバイスでのみ利用可能なものもあります。キーインターフェイスのみを持つデバイスでは、キーフォーカスがボタンにないと、いずれのイベントも生成されません。

Flash Lite は、次の ActionScript ボタンイベントをサポートしています。

ボタンイベント	説明
onDragOut	マウスやペンがあるデバイスでのみサポートされます。ユーザーがマウスボタンをボタン上で押した後、ポインタをボタンの外側にドラッグすると、呼び出されます。
onDragOver	マウスやペンがあるデバイスでのみサポートされます。ユーザーがマウスボタンを押したまま、一度ボタンの外側に移動し、次にボタン上に戻ると、呼び出されます。
onKeyDown	ボタンにフォーカスがあるときにキーを押すと、呼び出されます。
onKeyUp	ボタンにフォーカスがあるときにキーを離すと、呼び出されます。
onKillFocus	ボタンからフォーカスが離れると、呼び出されます。
onPress	ボタンにフォーカスがあるときにデバイスの選択キーを押すと、呼び出されます。
onRelease	ボタンにフォーカスがあるときにデバイスの選択キーを離すと、呼び出されます。
onReleaseOutside	ポインタがボタン内にあるときにボタンを押した後で、ポインタがボタン外にあるときにマウスボタンを離すと、呼び出されます。
onRollOut	ボタンがフォーカスを失うと、呼び出されます。
onRollOver	ボタンがフォーカスを受け取ると、呼び出されます。
onSetFocus	ボタンに入力フォーカスがあるときにキーを離すと、呼び出されます。

次の手順では、ボタンイベントを処理する単純なアプリケーションを作成する方法を説明します。ボタンを使用してメニューを作成する例については、[26 ページの「ボタンとデフォルトのナビゲーションを使用した単純なメニューの作成」](#)を参照してください。

ボタンイベントハンドラを作成するには：

1. 『Flash Lite 2.x ファーストステップガイド』の「Flash Lite ドキュメントテンプレートの作成」で作成した Flash Lite 2.0 テンプレートを使用して新規ドキュメントを作成し、"custom_focus_manager fla" という名前で保存します。
2. [ウィンドウ]-[サンプルライブラリ]-[ボタン]を選択して、作成済みのボタンシンボルの外部ライブラリを開きます。
3. [ライブラリ]パネルで"classic buttons"フォルダをダブルクリックし、"Circle Buttons"フォルダを開きます。
4. メニューボタンシンボルのインスタンスをステージにドラッグします。
5. プロパティインスペクタの[インスタンス名]テキストボックスに、「btn_1」と入力します。
6. 同じボタンの別のインスタンスをステージにドラッグし、最初のボタンのすぐ下に配置します。

7. プロパティインスペクタの[インスタンス名]テキストボックスに、「btn_2」と入力します。
8. タイムラインで、ActionScript というレイヤーのフレーム1を選択します。
9. [アクション] パネルを開き ([ウィンドウ]-[アクション])、次のコードを入力します。

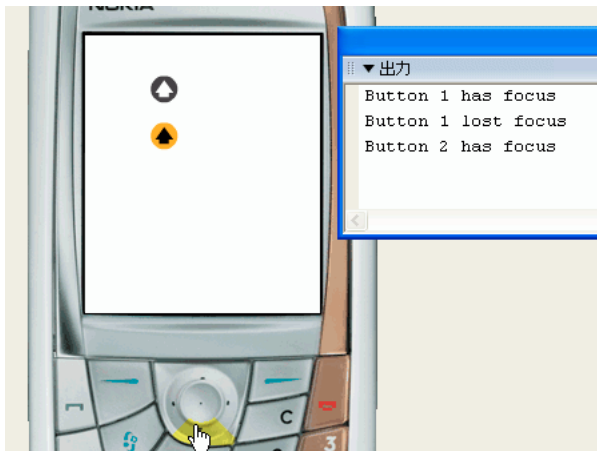
// ボタンがオーバー状態であるためフォーカス矩形を無効にします。
_focusRect = false;

```
// イベントハンドラ btn_1
btn_1.onPress = function() {
    trace("You pressed Button 1");
}
btn_1.onRelease = function() {
    trace("You released Button 1");
}
btn_1.onRollOver = function() {
    trace("Button 1 has focus");
}
btn_1.onRollOut = function() {
    trace("Button 1 lost focus");
}
```

```
// イベントハンドラ btn_2
btn_2.onPress = function() {
    trace("You pressed Button 2");
}
btn_2.onRelease = function() {
    trace("You released Button 2");
}
btn_2.onRollOver = function() {
    trace("Button 2 has focus");
}
btn_2.onRollOut = function() {
    trace("Button 2 lost focus");
}
```

10. エミュレータでアプリケーションをテストします ([制御]-[ムービープレビュー])。

エミュレータの上および下矢印キーを押したときに [出力] パネルに表示されるメッセージを確認します。



オブジェクトタイプによって、サポートされるイベントは異なります。たとえば、TextField オブジェクトは、テキストフィールドの内容が変更されたときに呼び出される onChanged イベントをサポートしています。この手順のボタンイベントハンドラと同じ形式で、これらのイベントのイベントハンドラコードを記述できます。テキストフィールドおよびムービークリップでサポートされているイベントの詳細については、『Flash Lite 2.x ActionScript リファレンスガイド』の「TextField」および「MovieClip」を参照してください。

ボタンとデフォルトのナビゲーションを使用した単純なメニューの作成

この項では、ボタンとデフォルトのナビゲーションを使用して単純なメニューを作成する方法を示します。メニューを作成するには、3 つのボタンシンボルを使用します。各シンボルが各メニューオプションに対応します。次に、ユーザーが各メニュー項目上に移動したとき (対応するボタンにフォーカスを与えたとき)、およびユーザーがデバイスの選択キーを押してメニュー項目を選択したときにメッセージを表示するイベント処理コードを記述します。Flash Lite でのボタンイベントの処理の詳細については、[23 ページの「ボタンイベントの処理」](#)を参照してください。

部分的に完成している Flash ドキュメントから操作を始めます。他のデバイスやコンテンツタイプに合わせてこれらの設定を変更できます ([114 ページの「エミュレータの使用」](#)を参照)。

ボタンを使用する単純なメニューを作成するには：

1. http://www.adobe.com/go/learn_flt_samples_and_tutorials_jp にある "simple_menu_start.fla" という名前のファイルをダウンロードして開きます。「サンプルとチュートリアル」 ページで、Flash Lite バージョンの .zip ファイルを探してダウンロードおよび解凍し、"Samples" フォルダを選択してサンプルファイルにアクセスします。
2. [ライブラリ] パネルを開きます ([ウィンドウ]-[ライブラリ])。
ライブラリ内に、News Button、Weather Button、および Sports Button という 3 つのボタンシンボルがあります。
3. タイムライン内で ([ウィンドウ]-[タイムライン])、Menu Buttons レイヤーを選択します。
4. News Button シンボルのインスタンスを [ライブラリ] パネルからステージにドラッグします。
5. プロパティインスペクタの [インスタンス名] テキストボックスに「btn_news」と入力します。
6. Sports と Weather ボタンについて手順 4 と 5 を繰り返し、それぞれに btn_sports と btn_weather という名前を付けます。
7. 次の図に示すように、3 つのボタンを縦方向に整列します。



8. ツールパネルでテキストツールを選択し、ステージの下部にテキストフィールドを作成します。
このテキストフィールドには、ユーザーが各メニュー項目に移動したときに短いメッセージが表示されます。

9. 新しいテキストフィールドを選択状態にし、プロパティインスペクタで次の操作を行います。
- [テキストの種類] ポップアップメニューから [ダイナミックテキスト] を選択します。
 - [インスタンス名] テキストボックスに「**txt_status**」と入力します。
- ステージは次の図のようになります。



10. タイムラインで、ActionScript というレイヤーのフレーム 1 を選択します。
11. [アクション] パネルを開き ([ウィンドウ]-[アクション])、次のコードを入力します。

```
// ボタンがオーバー状態であるためフォーカス矩形を無効にします。  
_focusRect = false;
```

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

12. [制御]-[ムービープレビュー]を選択して、エミュレータでアプリケーションをプレビューします。

エミュレータで下方向キーをマウスでクリックするか、コンピュータのキーボードで下方向キーを押して、メニューオプション間を移動します。メニュー項目を選択するには、エミュレータの選択キーをマウスで選択するか、コンピュータのキーボードで Enter キーを押します。



キーイベントの処理

Flash Lite では、ユーザーがデバイスのキーを押す操作に対応して、キー押下イベントが生成されます。これらのイベントに応答するキーハンドラコードは記述することができます。Flash Lite でサポートしているキーの一覧については、[14 ページの「Flash Lite でサポートされているキー」](#)を参照してください。

次の表は、一般的に使用されるデバイスキー、それに対応する ActionScript キーコード、およびそれらのキーのキーコード定数を示します。

デバイスキー	ActionScript キーコード / キーコード定数
選択キー	Key.ENTER
上方向キー	Key.UP
下方向キー	Key.DOWN
左方向キー	Key.LEFT
右方向キー	Key.RIGHT
左ソフトキー	ExtendedKey.SOFT1 (または soft1)
右ソフトキー	ExtendedKey.SOFT2 (または soft2)
0	48
1	49
2	50
3	51
4	52
5	53
6	54
7	55
8	56
9	57
*	56
#	51

イベントリスナーの記述

イベントリスナーを使用すると、あるオブジェクトでブロードキャストしたイベントを別のオブジェクトで受け取ることができます。イベントが発生する側のオブジェクトをブロードキャスターオブジェクト、イベントを受け取る側のオブジェクトをリスナーオブジェクトと呼びます。ブロードキャスターオブジェクトは、ブロードキャスターが生成したイベントを受け取るリスナーオブジェクトを登録します。詳細については、『ActionScript 2.x の学習』の「イベントリスナーの使用」を参照してください。

キー押下イベントを処理する簡単な方法は、onKeyDown または onKeyUp 関数を定義するキーリスナーオブジェクトを作成し、次に Key.addListener() メソッドを使用してそのオブジェクトを登録する方法です。次のサンプルコードでは、ユーザーがデバイスの右方向キーを押したときに応答するキーリスナーを定義します。

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

キーリスナーの使用例については、この章の残りのトピックを参照してください。

キーリスナーによるキー押下イベントの処理

次の手順は、キーリスナーを使用して、簡単なアプリケーションのキー押下イベントを処理する方法を示します。このアプリケーションでは、4 方向ナビゲーションキーを使用して、ステージ上でムービークリップを移動します。

イベントリスナーを使用してキー押下イベントを処理するには：

1. 『Flash Lite 2.x ファーストステップガイド』の「Flash Lite ドキュメントテンプレートの作成」で作成した Flash Lite 2.0 テンプレートを使用して新規ドキュメントを作成し、"keylistener fla" という名前で保存します。
2. Content という名前のタイムラインでレイヤーを選択します。
3. 楕円ツールを使用して、ステージ上に楕円または円を作成し、ムービークリップに変換します。
4. 新しいムービークリップが選択されている状態で、プロパティインスペクタの [インスタンス名] テキストボックスに「circle」と入力します。
5. タイムラインで、レイヤー 1 の最初のフレームを選択します。

6. [アクション] パネルを開き ([ウィンドウ]-[アクション]), 次のコードを入力します。

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

7. [制御]-[ムービープレビュー] を選択して、アプリケーションをテストします。

エミュレータの 4 方向キーを押すか、キーボードの対応する矢印キーを押して、ステージ上で円を移動します。

ソフトキーの使用

Flash Lite アプリケーションでソフトキーを使用するには、まず SetSoftKeys コマンドを呼び出します。以後、左ソフトキーを押すと ExtendedKey.S0FT1 イベントが生成され、右ソフトキーを押すと ExtendedKey.S0FT2 イベントが生成されます。これらのイベントに応答して所定の動作を行う ActionScript イベントハンドラコードを記述します。

SetSoftKeys コマンドは 2 つのパラメータを取ります。これらのパラメータは、アプリケーションがフルスクリーンモードで実行されていないときに表示される左ソフトキーと右ソフトキーのラベルを指定します。フルスクリーンモードで実行されているアプリケーションの場合は、指定したラベルは表示されないで、独自のラベルを作成して、ステージ上のソフトキーの場所に配置する必要があります。

たとえば、次の SetSoftKeys コマンド呼び出しについて考えてみます。

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


次の例は、フルスクリーンモードではなく、通常モードの実際のデバイスで実行されるアプリケーションで、このコマンドを実行した場合の結果を示しています。



— 非フルスクリーンアプリケーションで利用可能なスクリーン領域

— デバイスに表示されるソフトキーのラベル

フルスクリーンモードを有効にした場合、つまり `fscommand("fullscreen", true)` コマンドを呼び出した場合は、`SetSoftKeys` コマンドにパラメータとして指定したラベルは表示されません。このため、フルスクリーンモードのアプリケーションでは、次の例のように独自のソフトキーラベルを作成する必要があります。



— カスタムソフトキーのラベル

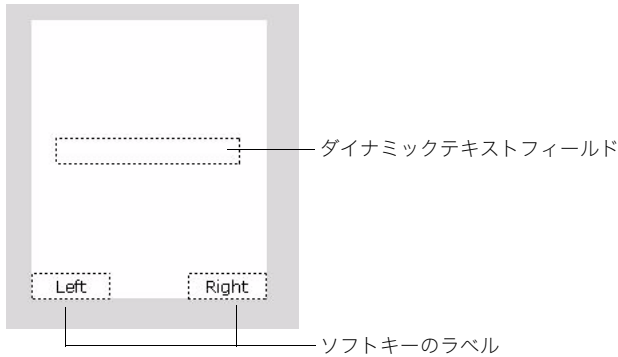
`SetSoftKeys` コマンドの詳細については、『Flash Lite 2.x ActionScript リファレンスガイド』の「`fscommand2` 関数」の項目を参照してください。

アプリケーションでソフトキーを使用するには：

1. 『Flash Lite 2.x ファーストステップガイド』の「Flash Lite ドキュメントテンプレートの作成」で作成した Flash Lite 2.0 テンプレートを使用して新規ドキュメントを作成し、"softkey fla" という名前で保存します。
2. テキストツールを使用して `Left` という名前の静止テキストフィールドを作成し、ステージの左下隅で、デバイスの左ソフトキーの上に配置します。
3. `Right` という別の静止テキストフィールドを作成し、ステージの右下隅でデバイスの右ソフトキーの上に配置します。

4. テキストツールを使用して、ダイナミックテキストフィールドを作成し、ステージの中央に配置します。

ドキュメントのステージは、次の図のようになります。



5. ダイナミックテキストフィールドが選択されている状態で、プロパティインスペクタの[インスタンス名]テキストボックスに「**status**」と入力します。
6. [アクション] パネルを開き ([Window]-[アクション])、タイムラインのレイヤーでフレーム 1 を選択します。
7. [アクション] パネルに次のコードを入力します。

```
fscommand2("SetSoftKeys", "Left", "Right");
fscommand2("FullScreen", true);
```
8. [アクション] パネルに次のコードを入力して、キー押下イベントに応答するオブジェクトを作成および登録します ([31 ページの「キーリスナーによるキー押下イベントの処理」](#)を参照)。

```
var myListener:Object = new Object();
myListener.onKeyDown = function() {
    if (Key.getCode() == ExtendedKey.SOFT1) {
        // 左ソフトキーのキー押下イベントを処理します。
        status.text = "You pressed the Left soft key.";
    } else if (Key.getCode() == ExtendedKey.SOFT2) {
        // 右ソフトキーのキー押下イベントを処理します。
        status.text = "You pressed the Right soft key.";
    }
};
Key.addListener(myListener);
```

9. [制御]-[ムービーレビュー]を選択してエミュレータでアプリケーションをテストします。
アプリケーションをテストするには、マウスでエミュレータの左右のソフトキーをクリックするか、キーボードの PageUp キーおよび PageDown キーを押します。



テキストとフォントの操作

この章では、アドビ システムズ社の Macromedia Flash Lite 2.x アプリケーションに静止テキストフィールドやダイナミックテキストを追加する方法、およびテキスト入力フィールドを追加する方法について説明します。

本章では次のトピックを取り上げます。

Flash Lite のテキストについて	37
テキストの作成とフォーマット	39
テキスト入力フィールドの使用	39
Flash Lite でのフォントのレンダリング方法	47
テキストフィールドのサンプルアプリケーション	51
スクロールテキストの作成	53

Flash Lite のテキストについて

Flash Lite 2.x では、次のテキスト機能をサポートしています。

■ 静止、ダイナミック、テキスト入力フィールド

実行時に、静止テキストフィールドの内容は変化しませんが、ダイナミックテキストフィールドやテキスト入力フィールドの内容は変化します。テキスト入力フィールドにはテキストを入力できます。Flash Lite 2.1 は、ほとんどのデバイスでのインラインテキスト入力をサポートしています。複雑な言語をサポートしているデバイスや Flash Lite 2.0 では、テキスト入力フィールドはデバイスの汎用テキスト入力メカニズムを使用します。テキスト入力フィールドの詳細については、[39 ページの「テキスト入力フィールドの使用」](#)を参照してください。

■ 埋め込みフォントおよびデバイスフォント

Flash Lite では、SWF ファイルに埋め込まれたフォントアウトラインを使用してテキストフィールドをレンダリングするか、またはデバイスで利用可能なフォントを使用してレンダリングできます。フォントのレンダリング方法の詳細については、[47 ページの「Flash Lite でのフォントのレンダリング方法」](#)を参照してください。

- Unicode テキストエンコーディング

Flash Lite では、必要な文字を含むフォントが使用可能であれば、任意の言語でテキストを表示できます。

- HTML 形式および TextFormat クラスのプロパティの部分的サポート

- スクロールテキスト

Flash Lite では、デスクトップバージョンの Flash Player で使用可能なテキスト機能の一部がサポートされていません。Flash Lite には、次の制限があります。

- 高度なアンチエイリアスはサポートされていません。高度なアンチエイリアスは、アドビ システムズ社の Macromedia Flash Player 8 以降で使用可能な拡張されたフォントレンダリングテクノロジーです。
- テキストフォーマットはサポートされますが、デバイステキストには次の制限があります。
 - 色、書体、サイズ、ボールドおよびイタリックのオプションしか使用できません。
 - 選択したオプションがデバイステキストのフォントに含まれない場合、フォーマットは表示されません。たとえば、イタリック形式のフィールドは、デバイスフォントにイタリックが含まれていない場合は、通常のテキストとして表示されます。
- デバイステキストは、マスクの適用、マスクとしての使用、透過表示はできません。
- [テキストを HTML としてレンダリング] オプションは、テキスト入力フィールドおよびダイナミックテキストフィールドで部分的にサポートされています。テキストは HTML タグが非表示の状態が表示されますが、フォーマットが有効になるのは次のタグのみです。p、br、sbr、(face、color、および size 属性を持つ) font、b、および i。
- Flash Lite では、カスケーディングスタイルシート (CSS) をサポートしていません。
- Label、TextArea、TextInput などの Flash コンポーネントはサポートされていません。
- TextField オブジェクトおよび TextFormat オブジェクトは一部サポートされますが、アラビア語、ヘブライ語、タイ語には追加の制限があります。詳細については、『Flash Lite 2.x ActionScript リファレンスガイド』を参照してください。
- XML オブジェクトおよび XMLNode オブジェクトがサポートされています。

Flash Lite 2.1 では、インラインテキスト入力、予測テキスト入力エンジン、および XMLSocket のサポートが追加されました。

- インラインテキスト入力のサポートにより、ユーザーはテキストフィールドに直接テキストを入力できます。
- 予測テキスト入力のサポートにより、単語補完や候補リストなどの機能を使用できます。Flash Lite 2.1 は、予測テキスト入力エンジンのベンダーが提供する標準 API と同じように予測テキスト入力エンジンが実装されていれば、どのプラットフォームでも最も優れた予測テキスト入力エンジン (T9、eZiTap/eZiText、iTap など) をサポートします。
- XMLSocket のサポートにより、Flash デスクトップアプリケーションのサポートが Flash Lite にまで拡張され、開発者はゲームやチャットなどのアプリケーションに連続的で待ち時間の短いデータ接続を作成できます。

テキストの作成とフォーマット

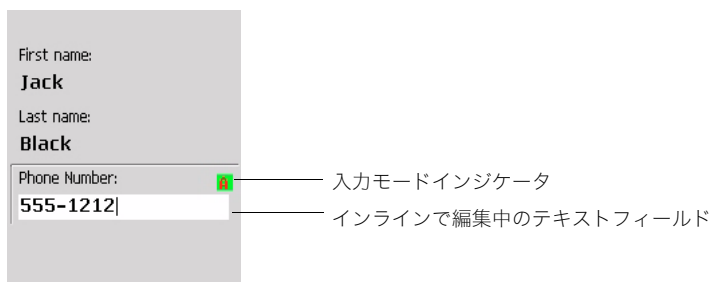
Flash Lite でテキストを作成およびフォーマットする方法は、Flash デスクトップアプリケーションと同じです。

Flash でのテキスト操作の詳細については、『Flash ユーザーガイド』の次のトピックを参照してください。

Flash Lite でサポートされていないテキスト機能の一覧については、[37 ページの「Flash Lite のテキストについて」](#)を参照してください。

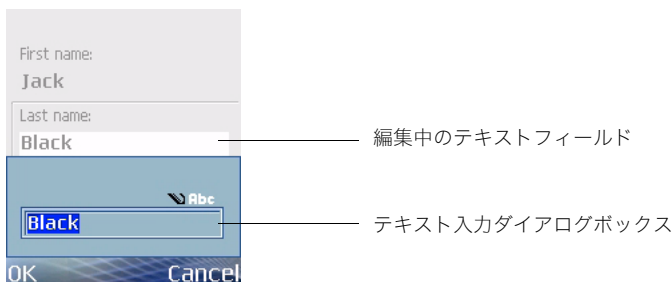
テキスト入力フィールドの使用

Flash Lite 2.1 はインラインテキスト入力をサポートしています。この機能により、ユーザーは以前のバージョンのように別のテキスト入力ボックスでテキストフィールドを編集するのではなく、Flash Lite アプリケーションで直接テキストフィールドを編集できます。たとえば、次の図は、Nokia Series 60 デバイスでインラインテキスト入力フィールドがどのように表示されるかを示しています。



インラインテキスト入力の詳細については、[40 ページの「インラインテキスト入力の使用 \(Flash Lite 2.1\)」](#)を参照してください。

Flash Lite 2.0 および Flash Lite 1.1 を実行しているデバイスと、すべてのバージョンの Adobe Device Central エミュレータでは、ユーザーは Flash Lite コンテンツ上に表示されるモーダルダイアログボックスを使用して入力テキストフィールドの内容を編集します。たとえば、次の図は、Flash Lite 2.0 を実行している Symbian Series 60 デバイスに表示されるテキスト入力ダイアログボックスの例を示しています。



一般的に、既存の Flash Lite 2.0 および Flash Lite 1.1 アプリケーションは変更しなくても Flash Lite 2.1 で動作します。ユーザーは、モーダルダイアログボックスを使用する代わりに、インラインでテキスト入力フィールドを編集できます。しかし、テキスト選択を設定する機能や `activateTextField` コマンドなど、Flash Lite 2.1 だけで使用できる機能を使用するには、古いコンテンツを記述し直す必要があります。

テキスト入力ダイアログボックスの詳細については、[40 ページの「デバイスのテキスト入力ダイアログボックスの使用 \(Flash Lite 2.0\)」](#)を参照してください。

インラインテキスト入力の使用 (Flash Lite 2.1)

Flash Lite 2.1 は、ユーザーがテキスト入力フィールドで直接テキストを入力および編集するのが可能にするインラインテキスト入力をサポートしています。ユーザーは、Flash Lite アプリケーションにテキストを入力するときに、デバイスのネイティブな IME (Input Method Editor: 入力方式エディタ) を操作します。Flash Lite プレーヤーがユーザーの入力自体を処理するではありません。

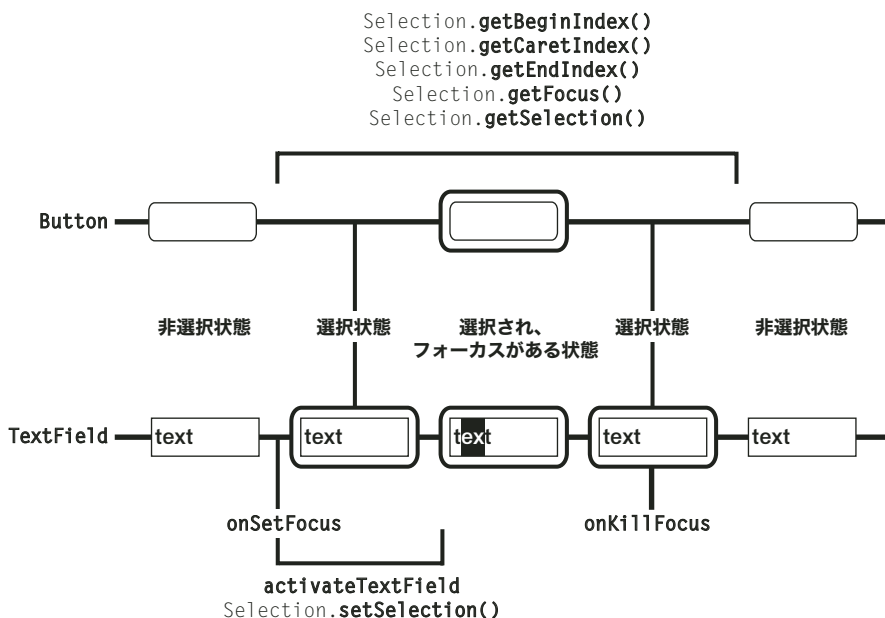


Adobe Device Central の Flash Lite エミュレータはインラインテキストを表示しません。詳細については、[114 ページの「Flash Lite 2.1 でのインラインテキストのプレビュー」](#)を参照してください。

テキスト入力フィールドがアクティブな間、Flash Lite 2.1 プレーヤーは制限モードで実行されます。このモードでは、デバイスのネイティブな IME はすべてのキー押下イベントを処理し、ActionScript は処理されません。また、アニメーション、サウンド、およびビデオの再生を停止するイベントはすべて無視されます。テキストフィールドが非アクティブになると、Flash Lite は通常の動作モードに戻ります。

テキストフィールドは、アクティブになると、ユーザーからの入力を受け入れることができます。テキスト入力フィールドをアクティブにするには、テキスト入力フィールドに選択フォーカスを与え、デバイスの選択キーを押すか、アプリケーションがテキスト入力をフォーカスのある状態に設定するソフトキーを押します。テキストフィールドをアクティブにする詳細については、[42 ページの「ActionScript を使用したテキスト入力フィールドのアクティブ化」](#)を参照してください。

次の図は、テキスト入力フィールドのさまざまな状態 (選択されていない状態、選択されたがアクティブでない状態、およびアクティブな状態) を示しています。



インラインテキスト入力フィールドとナビゲーションについて

アクティブな状態のテキストフィールドには、現在の挿入箇所を示すポインタがあります。デバイスのナビゲーションキーを使用すると、ポインタの位置を変更できます。

デフォルトでは、現在のフォーカスがあるテキストフィールドの周囲にフォーカスの矩形が表示されます。このフォーカスの矩形が原因で、アクティブなテキストフィールドで点滅するポインタ、つまり挿入インジケータが見えにくくなる場合があります。このため、`_focusRect` を `false` に設定してフォーカスの矩形を無効にし、カスタムフォーカスインジケータを使用することをお勧めします。カスタムフォーカスインジケータを使用するアプリケーションの例については、Flash Lite のサンプルとチュートリアルページ (www.adobe.com/go/learn_flt_samples_and_tutorials_jp) を参照してください。ActionScript バージョンの .zip ファイルを探してダウンロードおよび解凍し、"Samples" フォルダを選択してサンプルファイルにアクセスします。

ActionScript を使用したテキスト入力フィールドのアクティブ化

ユーザーは、デバイスの選択キーを押すことにより、キーフォーカスがあるテキスト入力フィールドをアクティブにできます。Flash Lite アプリケーションも、activateTextField コマンドを使用してフォーカスを受け取ると、テキスト入力フィールドを自動的にアクティブにすることができます。このコマンドは、現在選択されているテキストフィールドをアクティブにします。コマンドが実行されたときにテキストフィールドが選択されていない場合は、何も起こりません。

activateTextField を呼び出す最も一般的な場所は、Selection.onSetFocus ハンドラまたは TextField.onSetFocus ハンドラ内からです。たとえば、アプリケーションでステージ上にテキスト入力フィールドが2つまたはそれ以上あるとします。次のコードは、フォーカスを受け取ったテキストフィールドを自動的にアクティブにします。

```
var focusListener:Object = new Object();
focusListener.onSetFocus = function (oldFocus, newFocus) {
    // activateTextField を呼び出します。
    fscommand ("activateTextField", "");
};
TextField1.addListener (focusListener);
```

TextField.prototype.onSetFocus ハンドラを使用して、すべてのテキストフィールドを、フォーカスを受け取るたびにアクティブにすることもできます。

デバイスの選択キー以外のキーを使用して activateTextfield コマンドをトリガすることもできます。次のコードは、すべての数値キーのテキストフィールドをアクティブにします。これにより、"a" という文字などを入力するのが簡単になります。たとえば、アプリケーションで

TextField.onSetFocus ハンドラに activateTextfield が含まれている場合、ユーザーは選択キーを押してから2を押す必要があります。このコードを使用すると、ユーザーは2を2回押すだけで済み、この方が直観的です。

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

この方法を使用する完成済みサンプルアプリケーションについては、インラインテキスト入力のサンプル (www.adobe.com/go/learn_ftl_samples_and_tutorials_jp) にあるを参照してください。

ActionScript バージョンの .zip ファイルを探してダウンロードおよび解凍し、"Samples" フォルダを選択してサンプルにアクセスします。

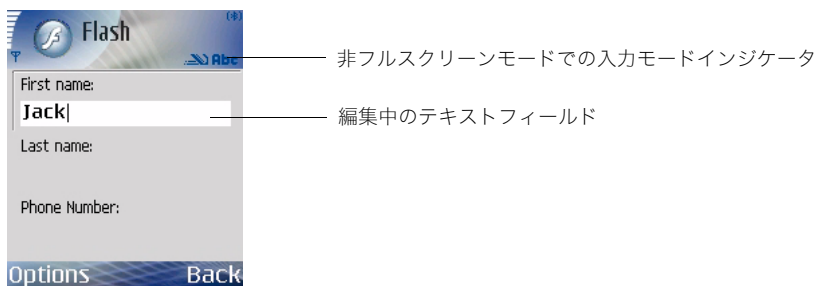
サポートされている言語

Flash Lite 2.1 は、ラテン語およびアジアの言語のインラインテキスト入力をサポートしていますが、複雑な言語のインラインテキスト入力をサポートしていません。これらの複雑な言語に含まれるのは、アラビア語、ヘブライ語、ウルドゥー語、ペルシャ語、イディッシュ語などの右から左に書く言語やアジアの言語の一部です。

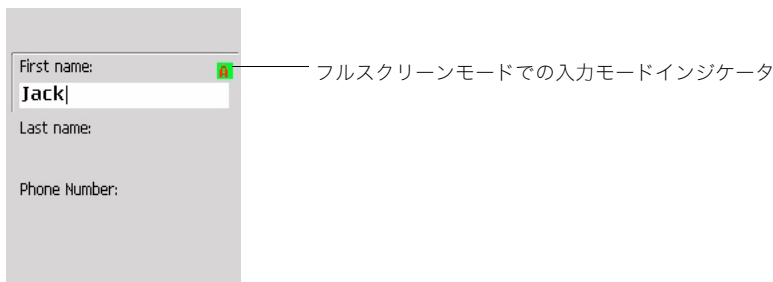
入力モードインジケータについて

ユーザーの一般的なテキスト入力タスクを支援するために、ほとんどのデバイスは、予測または手動 (トリプルタップまたはマルチタップ) テキスト入力モードなどの複数の入力モードや、数字入力モードを提供します。

Flash Lite がフルスクリーンモードで実行されている場合、デバイスには英字入力モードのときは文字 A が、数値入力モードのときは数字記号 (#) が表示されます。Flash Lite がフルスクリーンモードで実行されていない場合、デバイスではディスプレイのステータスバーやその他の場所に入力モードインジケータが表示されることがあります。たとえば、次の図は、Flash Lite 2.1 の Series 60 スタンドアローンバージョンにおける非フルスクリーンモードでの入力モードインジケータを示しています。



プレーヤーがフルスクリーンモードで実行されている場合、デバイスではディスプレイ上の独自に選択した場所に入力モードインジケータが表示されることがあります。たとえば、次の図は、Flash Lite 2.1 の Series 60 スタンドアローンバージョンにおけるフルスクリーンモードでの入力モードインジケータを示しています。



前の図に示されているフルスクリーンアプリケーションの入力モードインジケータは、スタンドアロン Series 60 プレーヤーのリファレンス実装です。表示される特定のインジケータは、デバイスによって決定されます。

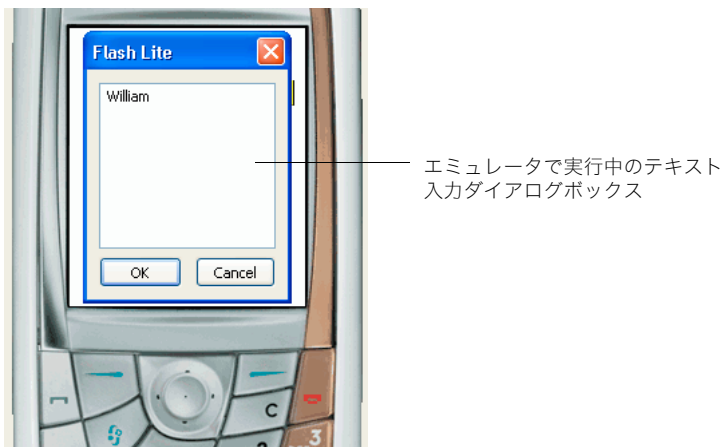
デバイスのテキスト入力ダイアログボックスの使用 (Flash Lite 2.0)

デバイスの入力ダイアログボックスを開くには、最初にテキスト入力フィールドにフォーカスを与え、次にデバイスの選択キーを押します。

テキスト入力ダイアログボックスはモーダルなので、ダイアログボックスにフォーカスがある間は、ユーザーは Flash コンテンツを操作できません。ダイアログボックスにフォーカスがある間は、Flash アプリケーションの再生ヘッドも一時停止します。

ユーザーが [OK] (左ソフトキー) を選択すると、テキスト入力ダイアログボックスが閉じ、Flash Lite によって自動的にテキスト入力フィールドにテキストが入力されます。ユーザーが [キャンセル] (右ソフトキー) を選択すると、テキストはテキスト入力フィールドに入力されません。

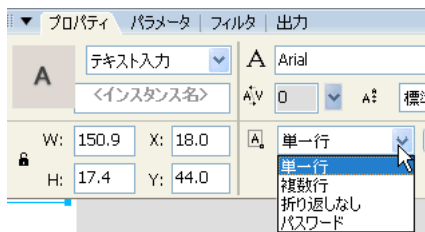
Flash オーサリングツールでアプリケーションをテストする場合、Adobe Device Central エミュレータによってテキスト入力ダイアログボックスの機能が再現されます。次の図は、エミュレータで実行中のテキスト入力ダイアログボックスを示しています。



アプリケーションでテキスト入力フィールドを使用する例については、[51 ページの「テキストフィールドのサンプルアプリケーション」](#)を参照してください。

テキスト入力フィールドの種類の指定

Flash Lite では、単一行、複数行、およびパスワードテキスト入力フィールドをサポートしています。次の図のように、プロパティインスペクタの [行タイプ] ポップアップメニューを使用して、テキスト入力フィールドの種類を指定します。



テキスト入力フィールドについて指定した行のタイプによって、ユーザーがテキストフィールドを編集するときのデバイスのテキスト入力ダイアログボックスの動作が決まります。

たとえば、ユーザーが単一行のテキスト入力フィールドを編集する場合、デバイスのテキスト入力ダイアログボックスには、単一行のテキスト入力ボックスが表示されます。表示できる文字数よりも多くの文字をユーザーが入力すると、テキスト入力ボックスは横にスクロールします。

文字入力の制限

SetInputTextType コマンドを使用して、ユーザーがテキスト入力ダイアログボックスに入力できる文字を制限できます。たとえば、アプリケーションに、ユーザーが年齢などの数値を入力するためのテキスト入力フィールドがあり、このテキスト入力フィールドに ageVar という名前の変数が含まれているとします。このテキスト入力ダイアログボックスには数値しか入力できないようにするには、アプリケーションに次のコードを追加します。

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

テキスト入力ダイアログボックスを開くと、ユーザーはこのテキストフィールドに数値のみを入力できます。

詳細については、『Flash Lite 2.x ActionScript リファレンスガイド』の「fscommand2 関数」の項目にある「SetInputTextType」を参照してください。

テキスト入力ダイアログボックス (Flash Lite 2.0)

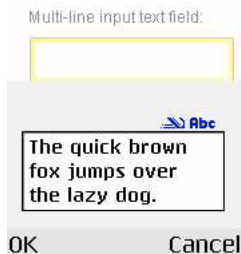
Flash Lite 2.0 では、ユーザーは画面上のテキストフィールドを直接操作するのではなく、別のモーダルダイアログボックス (Flash Lite ではなく、ホストアプリケーションによって制御されます) にテキストを入力します。この操作の間、Flash Lite プレーヤーはユーザーがダイアログボックスを終了するまで実質上、一時停止します。

Flash Lite 2.1 では、ユーザーは画面上のテキストフィールドにテキストを直接入力できます。

次の図は、Flash Lite 2.0 アプリケーションで単一行のテキスト入力フィールドを編集する場合のデバイスのテキスト入力ダイアログボックスを示しています。



ユーザーが複数行のテキスト入力フィールドを編集する場合は、次の図のように、デバイスのテキスト入力ダイアログボックスが必要に応じて拡張され、ユーザーが入力したテキストがすべて表示されます。



ユーザーがパスワードテキスト入力フィールドを編集する場合は、デバイスのテキスト入力ダイアログボックスにユーザーが入力した各文字が表示されます。ユーザーが [OK] をクリックすると、次の図のようにパスワードがアスタリスクでマスクされます。



Flash Lite でのフォントのレンダリング方法

Flash Lite では、次のいずれかの方法でテキストフィールドのフォントをレンダリングできます。

デバイスで利用可能なフォントを使用 テキストフィールドには、デバイスで利用できることがわかっているフォントを適用するか、[フォント] ポップアップメニューで選択可能な 3 つの汎用デバイスフォント (_ゴシック、_明朝、または _等幅) のいずれかを指定できます。汎用デバイスフォントを選択した場合は、実行時にその汎用フォントとデバイスフォントのマッチングが試行されます。たとえば、_sans が sans serif フォント (使用可能な場合) にマップされます。

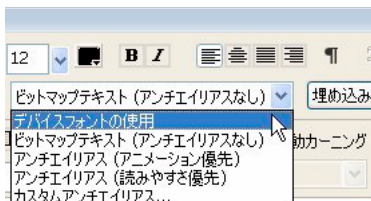
フォントをビットマップとしてレンダリング フォントのアウトラインをピクセルの境界に整列することによってビットマップテキストをレンダリングし、小さいポイントサイズ (10 ポイント以下) でもテキストを読みやすくします。このオプションを使用するには、パブリッシュする SWF ファイルに、選択したフォントのアウトラインを含める必要があります (詳細については、[50 ページの「SWF ファイルへのフォントアウトラインの埋め込み」](#)を参照してください)。

フォントをアンチエイリアス処理されたベクターとしてレンダリング パブリッシュする SWF ファイルに埋め込むフォントアウトラインのベクター形式の表現を使用して、アンチエイリアス処理されたテキストをレンダリングします (詳細については、[50 ページの「SWF ファイルへのフォントアウトラインの埋め込み」](#)を参照してください)。

テキストフィールドのフォントのレンダリング方法を選択するには、プロパティインスペクタの [フォントのレンダリング方法] ポップアップメニューを使用します。[フォントのレンダリング方法] ポップアップメニューには、5 つのレンダリングオプションがありますが、Flash Lite 開発者が使用できるのはそのうちの 3 つだけです。残りの 2 つの方法 ([アンチエイリアス (読みやすさ優先)] と [カスタムアンチエイリアス]) は、デスクトップコンピュータの Flash Player 8 以降に対応するアプリケーションでのみ使用できます。

テキストフィールドのフォントのレンダリング方法を選択するには：

1. ステージ上のテキストフィールドを選択します。
2. プロパティインスペクタで、[フォントのレンダリング方法] ポップアップメニューから次のいずれかのオプションを選択します。



- [デバイスフォントの使用] を選択すると、デバイスで使用可能なフォントが使用されます。パブリッシュする SWF ファイルにはフォントデータが埋め込まれません。
- [ビットマップテキスト (アンチエイリアスなし)] を選択すると、フォントアウトラインがピクセルの境界に揃えられ、小さい文字が判読しやすくなります。このオプションでは、パブリッシュする SWF ファイルにフォントのアウトラインを埋め込む必要があります (詳細については、[50 ページの「SWF ファイルへのフォントアウトラインの埋め込み」](#)を参照してください)。
- [アンチエイリアス (アニメーション優先)] を選択すると、現在のレンダリング品質の設定に従ってテキストフィールドのフォントがアンチエイリアス処理されます。詳細については、[49 ページの「Flash Lite のレンダリング品質とアンチエイリアス処理されたテキスト」](#)を参照してください。このオプションでは、パブリッシュする SWF ファイルにフォントのアウトラインを埋め込む必要があります。

アンチエイリアス、ビットマップ、デバイステキストのいずれかを選択する際のガイドラインを次に示します。

- ダイナミックテキストフィールドまたはテキスト入力フィールドで埋め込みフォントを使用している場合、表示する必要がある文字のアウトラインのみ埋め込みます。これにより、ファイルサイズを縮小できます。たとえば、テキスト入力フィールドを使用してユーザーの年齢 (数字) を取得するには、数字 (0-9) のフォントアウトラインのみを含めます。この場合、文字入力を数字のみに制限することをお勧めします ([47 ページの「Flash Lite でのフォントのレンダリング方法」](#)を参照)。
- Adobe Device Central エミュレータでは、コンテンツの開発に使用しているコンピュータに同じフォントがインストールされていない場合は、デバイスフォントをエミュレートしません。したがって、デバイスのテキストフィールドのレイアウトおよび外観は、エミュレータとデバイスでは異なって表示される場合があります。

- 汎用デバイスフォントフェイス (_ゴシック、_明朝、_等幅、_sans、_serif、または _typewriter) のいずれかを適用すると、Flash Lite はテキストを表示するために類似したデバイスフォントを検索します。ただし、通常モバイルデバイスでは、フォントとフォントスタイルがデスクトップコンピュータよりも少ないため、_sans のようなフォントは sans serif フォントにマップされない可能性があります。適切なフォント設定を決定するには、それぞれのターゲットデバイス上でアプリケーションをテストする必要があります。
- Flash Lite のアンチエイリアス処理されたテキストは、基本的には、複雑なベクターシェイプです。ベクターシェイプと同じように、レンダリングには処理能力が必要です。大抵のデバイスでは処理能力が相対的に遅いため、アンチエイリアス処理された大量のテキストをアニメーション化すると、アプリケーションのパフォーマンスが低下する可能性があります。パフォーマンスを向上させるには、アニメーションの再生中に Flash Lite プレーヤーのレンダリング品質を一時的に低くし、その後アニメーションが完了したら高いレンダリング品質に戻してください。詳細については、次のセクションを参照してください。

Flash Lite のレンダリング品質とアンチエイリアス処理されたテキスト

Flash Lite ではレンダリング品質として低、中、高の 3 種類を設定できます。レンダリング品質を高く設定すると、レンダリングされるベクターのアウトラインがなめらかで正確になります。品質の設定が低いほど、表示されるアウトラインはなめらかでなくなります。デフォルトでは、中の品質を使用してアウトラインがレンダリングされます。SetQuality コマンドを使用して、レンダリング品質を制御することができます。詳細については、『Flash Lite 2.x ActionScript リファレンスガイド』の「fscommand2 関数」の項目にある「SetInputTextType」を参照してください。

Flash Lite では、フォントアウトラインのベクター表現を使用してアンチエイリアス処理されたテキストをレンダリングします。テキストをできるだけなめらかにアンチエイリアス処理するには、プレーヤーのレンダリング品質を高に設定します。レンダリング品質は、アンチエイリアス処理したテキストだけでなく、画面上のすべてのベクターコンテンツに影響します。次の図は、高、中、低のレンダリング品質でアンチエイリアス処理したテキストフィールド (Arial、24 ポイント) を示します。

*High
quality
setting*

*Medium
quality
setting*

*Low
quality
setting*

CPU の負荷が高いアニメーションやトゥイーンシーケンスなどで、アニメーションのパフォーマンスとフレームレートを最大限に向上させる場合は、レンダリング品質を一時的に低く設定して、アニメーションが完了した後で前の設定に戻すことができます。

SWF ファイルへのフォントアウトラインの埋め込み

テキストフィールドのフォントをレンダリングするには、Flash Lite では、デバイスで利用可能なフォントを使用するか、パブリッシュされる SWF ファイルに埋め込まれたフォントアウトラインを使用することができます。詳細については、[47 ページの「Flash Lite でのフォントのレンダリング方法」](#)を参照してください。SWF ファイルにフォントアウトラインを埋め込むと、テキストフィールドのフォントはすべてのターゲットプラットフォームで同じように表示されます。ただし、ファイルサイズは大きくなります。Flash Lite では、フォントアウトラインをビットマップ (非アンチエイリアス) またはアンチエイリアス処理したテキストとしてレンダリングする必要があります。

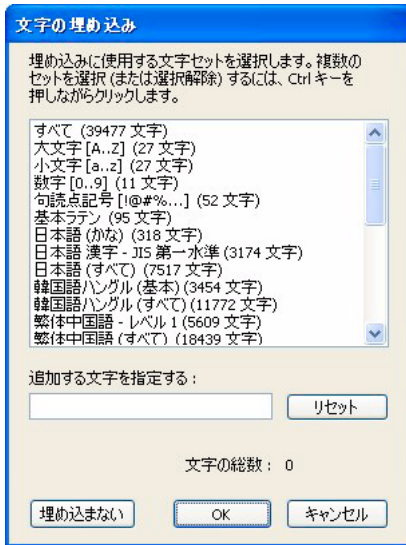
アンチエイリアスまたはビットマップとしてフォントをレンダリングする方法を使用する静止テキストフィールドの場合は、テキストフィールドの内容を表示するのに必要なフォントアウトラインが自動的に埋め込まれます。たとえば、静止テキストフィールドに *Submit* という単語が含まれている場合、この 6 つの文字 (S、u、b、m、i、t) を表示するのに必要なフォントアウトラインが自動的に埋め込まれます。静止テキストフィールドの内容は変化しないので、SWF ファイルに含める必要があるのは、表示される文字のフォントアウトラインのみです。

アンチエイリアスおよびビットマップのフォントレンダリング方法を使用するダイナミックおよびテキスト入力フィールドの場合は、指定する文字のフォントアウトラインを、パブリッシュする SWF ファイルに埋め込む必要があります。この 2 種類のテキストフィールドは、再生中に内容が変動するので、必要なフォントアウトラインを Flash では判断できません。選択したフォントのすべての文字、文字の範囲、特定の文字のフォントアウトラインは含めることができます。パブリッシュする SWF ファイルに埋め込む文字を指定するには、[文字の埋め込み] ダイアログボックスを使用します。

ダイナミックテキストフィールドまたはテキスト入力フィールドにフォントアウトラインを埋め込むには：

1. ステージ上のダイナミックテキストフィールドまたはテキスト入力フィールドを選択します。
2. プロパティインスペクタで、[フォントのレンダリング方法] ポップアップメニューから [ビットマップ (アンチエイリアスなし)] または [アンチエイリアス (アニメーション優先)] を選択します。

3. [フォントのレンダリング方法] ポップアップメニューの横にある [埋め込み] ボタンをクリックし、[文字の埋め込み] ダイアログボックスを開きます。



4. 埋め込む文字を一覧から選択するか、埋め込む文字をテキストボックスに入力するか、または [リセット] をクリックして選択したテキストフィールドにある文字を含めます。
5. [OK] をクリックします。

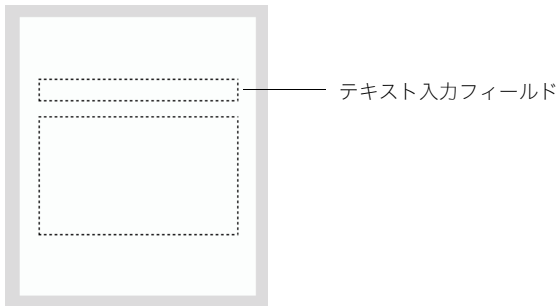
テキストフィールドのサンプルアプリケーション

この項では、ユーザーが入力したテキストを取得およびフォーマットし、そのテキストを HTML 対応のダイナミックテキストフィールドに表示する簡単なアプリケーションを作成する方法について説明します。また、SetFocusRectColor コマンドを使用して、フォーカス矩形の色をデフォルトの黄色から黒に変更します。

この方法を使用する完成済みサンプルアプリケーション (textfield_example.fla) については、www.adobe.com/go/learn_flt_samples_and_tutorials_jp を参照してください。ActionScript バージョンの .zip ファイルを探してダウンロードおよび解凍し、"Samples" フォルダを選択してサンプルにアクセスします。

テキストフィールドのサンプルアプリケーションを作成するには：

1. Flash で、前に『Flash Lite 2.x ファーストステップガイド』の「Flash Lite ドキュメントテンプレートの作成 (Flash Professional のみ)」で作成した Flash Lite 2.0 テンプレートからドキュメントを新規作成します。
2. ツールパネルのテキストツールを使用して、ステージの上部に単一行のテキストフィールドを作成します。
3. テキストフィールドを選択した状態で、プロパティインスペクタの [テキストの種類] ポップアップメニューから [テキスト入力] を選択し、[フォントのレンダリング方法] ポップアップメニューから [デバイスフォントの使用] を選択して、[インスタンス名] テキストボックスに「inputTxt」と入力します。
4. 次に示すように、最初のテキストフィールドの下に、最初のテキストフィールドよりも高さが数倍大きい別のテキストフィールドを作成します。



5. 2 番目のテキストフィールドを選択した状態で、プロパティインスペクタの [テキストの種類] ポップアップメニューから [ダイナミックテキスト]、[行タイプ] ポップアップメニューから [複数行] を選択し、[テキストを HTML としてレンダリング] オプションを選択し、[フォントのレンダリング方法] ポップアップメニューから [デバイスフォントの使用] を選択して、[インスタンス名] テキストボックスに「messageTxt」と入力します。
6. タイムラインで、レイヤー1のフレーム1を選択します。
7. [アクション] パネルを開き ([ウィンドウ]-[アクション])、次のコードを入力します。

```
Selection.setFocus(inputTxt);
fscommand2("SetFocusRectColor", 0, 0, 0);
inputTxt.onChanged = function() {
    messageTxt.htmlText = "入力した文字は <i>" + this.text + "</i> です。";
}
```

Selection.setFocus() メソッドを使用して、初期フォーカスをテキスト入力フィールド (inputTxt) に設定します。次に、fscommand2() 関数の呼び出しがカスタムのフォーカス矩形の色を指定します。最後に、テキスト入力フィールドの onChanged イベントハンドラは、テキスト入力フィールドの内容が変更されるたびに呼び出され、ユーザーが messageTxt テキストフィールドに入力したテキストをフォーマットして表示します。

- 変更を保存し、エミュレータでアプリケーションを起動します ([制御]-[ムービープレビュー])。
- アプリケーションを使用するには、エミュレータの選択キーを押してテキスト入力ダイアログボックスを開き、コンピュータのキーボードを使用して任意のテキストを入力します。その後、[OK] をクリックしてダイアログボックスを閉じます。入力したテキストは、messageTxt テキストフィールドにイタリックで表示されます。

スクロールテキストの作成

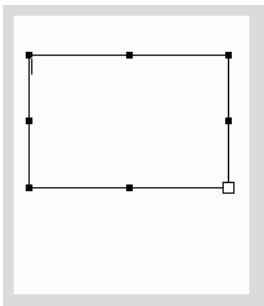
Flash Lite は、TextField.scroll および TextField.maxscroll プロパティをサポートしています。これらのプロパティを使用すると、スクロールテキストフィールドを作成できます。scroll プロパティは、テキストブロック内で表示可能な最初の行を指定します。この値は取得および設定が可能です。たとえば、次のコードは、変数名が story_text であるテキストフィールドを下へ 5 行スクロールします。

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

maxscroll プロパティは、テキストブロック内でテキストの最終行が表示されている場合、テキストブロック内で表示可能な最初の行を指定します。このプロパティは読み取り専用です。テキストフィールドの maxscroll プロパティを scroll プロパティと比較することによって、ユーザーがテキストフィールド内でどれくらいスクロールしたかを特定できます。この機能は、最大スクロール位置に対するユーザーの現在のスクロール位置をフィードバックするようなスクロールバーを作成する場合に便利です。

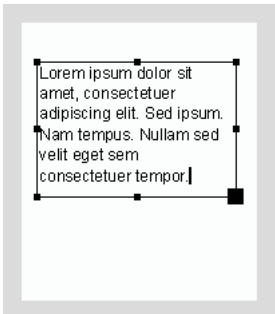
スクロールテキストフィールドを作成して ActionScript で制御するには：

- Flash で、前に『Flash Lite 2.x ファーストステップガイド』の「Flash Lite ドキュメントテンプレートの作成」で作成した Flash Lite 2.0 テンプレートからドキュメントを新規作成します。
- テキストツールを使用して、ステージ上で次の図のようなサイズになるまでテキストフィールドをドラッグします。



- プロパティインスペクタの [行タイプ] ポップアップメニューから [複数行] を選択します。

4. プロパティインスペクタの[テキストの種類]ポップアップメニューから[ダイナミックテキスト]を選択します。
5. プロパティインスペクタの[フォントのレンダリング方法]ポップアップメニューから[デバイスフォントの使用]を選択します。
6. [テキスト]-[スクロール可能]を選択して、テキストフィールドをスクロール可能にします。
7. プロパティインスペクタの[インスタンス名]テキストボックスに、「story」と入力します。
8. テキストフィールド内をダブルクリックし、テキストを入力します。多すぎて一部が下端に表示されないぐらいの量を入力します。



9. タイムラインでレイヤー 1 上の先頭のフレームを選択し、[アクション]パネルを開きます([ウィンドウ]-[アクション])。
10. [アクション]パネルで、次のコードを入力します。

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

11. [制御]-[ムービープレビュー]を選択して Adobe Device Central エミュレータでアプリケーションをテストします。

エミュレータの上下のナビゲーションキーをクリックして、またはコンピュータのキーボードの上矢印キーと下矢印キーを使用して、テキストを上下にスクロールします。

サウンド、ビデオおよびイメージ の操作

この章では、アドビ システムズ社の Macromedia Flash Lite 2.x アプリケーションにサウンド、ビデオ、および外部イメージを組み込む方法について説明します。

この章では、次のトピックについて説明します。

Flash Lite のサウンドについて	55
デバイスサウンドの使用	56
Flash ネイティブサウンドの使用	63
デバイスビデオの使用	66
外部イメージのロード	77

Flash Lite のサウンドについて

Flash Lite 2.0 は、デバイスサウンドと標準 (ネイティブ) の Flash サウンドの 2 種類をサポートしています。ネイティブサウンドは、デスクトップバージョンの Flash Player のように、直接 Flash Lite プレーヤーで再生されます。いくつか例外はありますが、Flash Lite アプリケーションでは、Flash デスクトップアプリケーションで使用すると同じ感覚でネイティブサウンドを使用できます。Flash ネイティブサウンドの使用の詳細については、[63 ページの「Flash ネイティブサウンドの使用」](#)を参照してください。

それに対して、デバイスサウンドは、Flash Lite プレーヤーではなく直接デバイスで再生されます。デバイスのサウンド形式には MIDI や MFi がありますが、サポートされているサウンド形式はデバイスによって異なります。デバイスサウンドを使用するには、パブリッシュされた SWF ファイルにデバイスサウンドをバンドルするか、ネットワークの場所またはデバイスのローカルファイルシステムから外部のサウンドファイルをロードします。デバイスサウンドには、ネイティブサウンドには適用されないいくつかの制限があります。デバイスサウンドの使用の詳細については、[56 ページの「デバイスサウンドの使用」](#)を参照してください。

デバイスサウンドの使用

デバイスサウンドとは、Flash Lite プレーヤーではなく、直接デバイスによって再生されるオーディオのことです。デバイスによって、MIDI、MFi、MP3 など、サポートされているサウンド形式が異なります。デバイスサウンドを Flash Lite コンテンツに組み込むには、パブリッシュされた SWF ファイル内にデバイスサウンドを含めるか、ネットワーク上からまたはデバイスのローカルファイルシステムから外部のサウンドファイルをロードします。いくつかの例外を除いて、デスクトップバージョンの Flash Player でサウンドを制御するように、ActionScript の Sound オブジェクトを使用してデバイスサウンドを制御できます。

このセクションでは、次のトピックについて説明します。

- [56 ページの「バンドルされたデバイスサウンドの使用」](#)
- [59 ページの「サウンドバンドルの作成」](#)
- [61 ページの「外部デバイスサウンドの再生」](#)
- [62 ページの「デバイスサウンドとアニメーションの同期について」](#)
- [62 ページの「サポートされているデバイスサウンド形式の確認」](#)

バンドルされたデバイスサウンドの使用

デバイスサウンドをアプリケーションにバンドルするには、まず、MP3、WAV、AIFF ファイルなどの Flash オーサリングツールが認識する形式でプロキシサウンドを読み込みます。その後、アプリケーションにバンドルする、コンピュータ上のデバイスサウンドファイルにプロキシサウンドをリンクします。SWF ファイルのパブリッシュ時に、Flash オーサリングツールはプロキシサウンドをリンクした外部サウンドに置き換えます。再生中に Flash Lite からデバイスにサウンドデータが渡され、サウンドデータはデコードされて再生されます。

また、形式が異なる複数のデバイスサウンドを、単一の Flash サウンドバンドル (FLS) ファイルにパッケージ化することもできます。この機能は、デバイスサウンド形式が異なる複数のデバイスに対して同じコンテンツを作成する場合に便利です。詳細については、[59 ページの「サウンドバンドルの作成」](#)を参照してください。

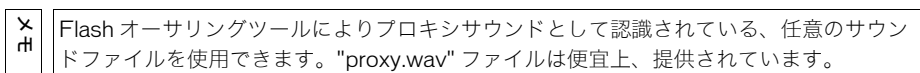
次の例は、バンドルされたデバイスサウンドを読み込んで再生する方法を示しています。デバイスサウンドを再生するには、デバイスサウンドをタイムラインに割り当てるか、Sound オブジェクトを使用してデバイスサウンドを ActionScript で再生します。このセクションでは、両方のテクニックについて説明します。

デバイスサウンドを読み込んで再生するには：

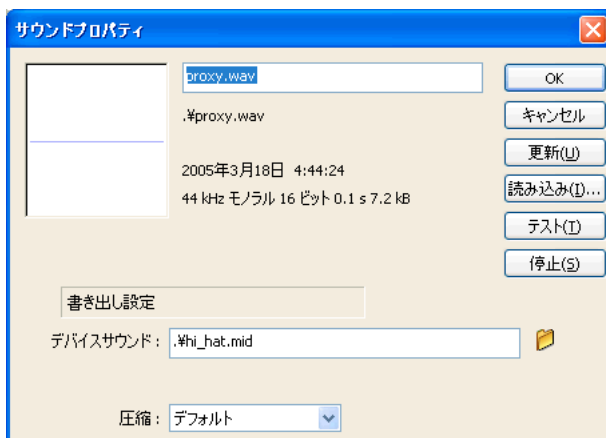
1. Flash Lite 2.0 Symbian Series 60 ドキュメントテンプレートから新しいドキュメントを作成し、"device_sound fla" という名前で保存します。

Flash Lite ドキュメントテンプレートの使用の詳細については、『Flash Lite 2.x ファーストステップガイド』の「Flash Lite ドキュメントテンプレートの作成」を参照してください。

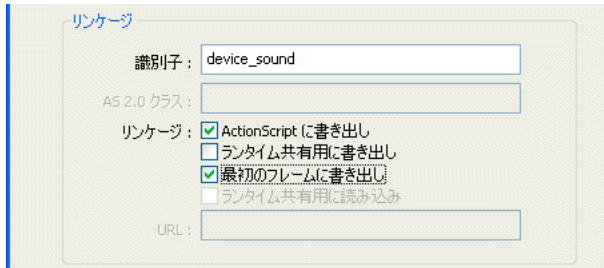
2. [ファイル]-[読み込み]-[ライブラリに読み込み]を選択します。サンプルとチュートリアルページ (http://www.adobe.com/go/learn_flt_samples_and_tutorials_jp) で、Flash Lite バージョンの .zip ファイルを探してダウンロードおよび解凍し、"Samples" フォルダを選択します。
3. "proxy.wav" ファイルを選択して、[OK] をクリックします。



4. プロキシサウンドをデバイスサウンドファイルにリンクするには、次の操作を行います。
 - a. [ライブラリ] パネルで、プロキシサウンドシンボルを右クリック (Windows) または Control キーを押しながらクリック (Macintosh) し、コンテキストメニューから [プロパティ] を選択して、[サウンドプロパティ] ダイアログボックスを開きます。
 - b. [サウンドプロパティ] ダイアログボックスで、[デバイスサウンド] テキストボックスの右側にあるフォルダアイコンをクリックして、[デバイスサウンドの選択] ダイアログボックスを開きます。
 - c. http://www.adobe.com/go/learn_flt_samples_and_tutorials_jp にアクセスします。「サンプルとチュートリアル」ページで、Flash Lite バージョンの .zip ファイルを探してダウンロードおよび解凍し、"Samples" フォルダに移動して hi_hat.mid というファイルを選択します。



- d. (オプション) ActionScript でデバイスサウンドを制御するには、[詳細] をクリックしてサウンドプロパティの詳細なオプションを表示し、[ActionScript に書き出し] を選択して、[識別子] テキストボックスに「device_sound」と入力します。



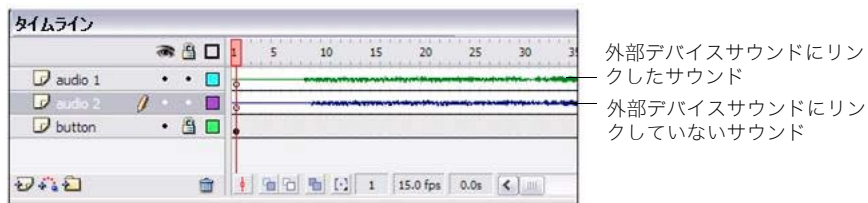
- e. [OK] をクリックして、[サウンドプロパティ] ダイアログボックスを閉じます。

デバイスサウンドを再生するには、プロキシサウンドをタイムラインに割り当てるか、ActionScript サウンドオブジェクトを使用します。ActionScript サウンドオブジェクトを使用するには、手順 6 に進みます。

5. デバイスサウンドをタイムラインに割り当てるには、次の操作を行います。
- Content レイヤーのフレーム 1 のキーフレームを選択します。
 - プロパティインスペクタで、[サウンド] ポップアップメニューから "proxy.wav" を選択します。これによって、プロキシサウンドがキーフレームに割り当てられます。



Flash オーサリングツールでは、タイムライン上にプロキシサウンドの波形が表示されます。次の図に示すように、外部デバイスサウンドにリンクしたサウンドの波形は緑色、外部デバイスサウンドにリンクしていないサウンドの波形は青色です。



6. ActionScript でサウンドを再生するには、次の操作を行います。

- a. タイムラインで Actions レイヤーを選択します。
- b. [アクション] パネルを開き ([ウィンドウ]-[アクション])、次のコードを入力します。

```
var deviceSound:Sound = new Sound();  
deviceSound.attachSound("device_sound");  
deviceSound.start();
```

7. [制御]-[ムービープレビュー] を選択して、Adobe Device Central エミュレータを起動し、SWF ファイルをテストします。

サウンドバンドルの作成

Flash Lite 2.0 では、形式が異なる複数のデバイスサウンドを、単一のサウンドバンドル内にカプセル化できます。たとえば、1つの Flash アプリケーションに、同じサウンドを MIDI と MFi の両方の形式で入れることができます。MIDI オーディオがサポートされているデバイス上でアプリケーションを再生すると、Flash Lite ではサウンドバンドルから MIDI サウンドデータが選択され、再生のためにデバイスに渡されます。同様に、MFi のみがサポートされているデバイスでアプリケーションを再生すると、Flash Lite では MFi サウンドデータがデバイスに渡されます。

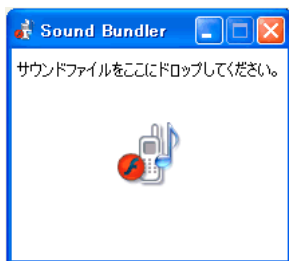
Flash Lite Sound Bundler と呼ばれるユーティリティを使用して、サウンドバンドル (FLS) ファイルを作成できます。ここで、単一のデバイスサウンドと同じように、FLS ファイルを Flash Lite ドキュメントのプロキシサウンドにリンクします。Flash Lite アプリケーションへのデバイスサウンドの追加の詳細については、[56 ページの「バンドルされたデバイスサウンドの使用」](#)を参照してください。



このマニュアルの作成時点では、Sound Bundler ユーティリティは Windows システムでのみサポートされています。

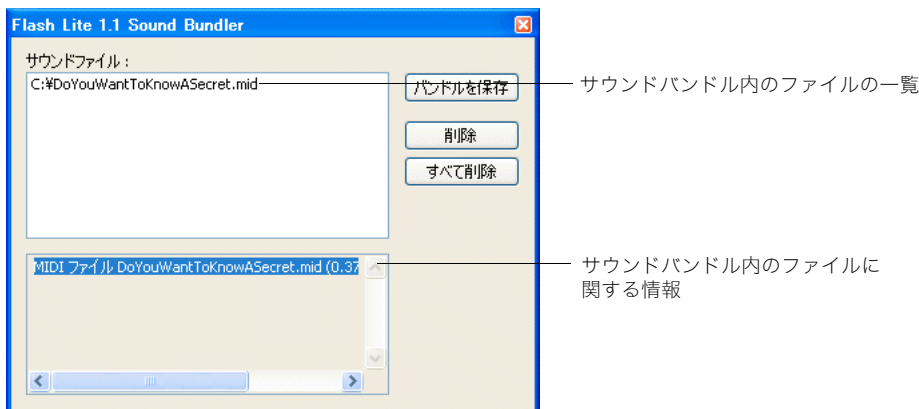
サウンドバンドルファイルを作成するには：

1. Flash Lite Sound Bundler アプリケーション ("FlashLiteBundler.exe") を起動します。このファイルは Flash インストールフォルダにあります (ブートドライブの "¥¥Program Files¥¥Adobe¥¥Flash CS3¥¥FlashLiteBundler.exe" など)。



2. デスクトップから、バンドルする最初のサウンドファイルを Sound Bundler ウィンドウ内にドラッグします。

別のウィンドウに、サウンドバンドルのコンテンツの一覧が表示されます。ウィンドウの下部には、サウンド形式、サウンドデータのサイズ、ファイル名など、サウンドバンドル内のサウンドに関する情報が表示されます。



3. バンドルする残りのサウンドファイルをウィンドウ内にドラッグします。
複数のファイルを同一のオーディオ形式でバンドルすることはできません。たとえば、2つのMIDI ファイルを同じ FLS ファイル内にバンドルすることはできません。
4. サウンドバンドルからファイルを削除するには、一覧からファイルを選択して [削除] をクリックします。
サウンドバンドル内のすべてのファイルを削除するには、[すべて削除] をクリックします。
5. サウンドバンドルを保存するには、[バンドルを保存] をクリックして、FLS ファイルの名前と保存場所を選択します。
6. Sound Bundlerを終了するには、Sound Bundler ウィンドウの閉じるボタン(X)をクリックします。

次の手順では、サウンドバンドル (FLS) ファイルを Flash ドキュメントに追加します。操作手順は Flash ドキュメントに標準のデバイスサウンドを追加する手順と同じですが、プロキシサウンドを置き換える単一のデバイスサウンドファイルを指定する代わりに、作成した FLS ファイルを指定する点が異なります。詳細については、[56 ページの「デバイスサウンドの使用」](#)を参照してください。

外部デバイスサウンドの再生

パブリッシュされた SWF ファイルにバンドルされたデバイスサウンドを再生するだけでなく (56 ページの「バンドルされたデバイスサウンドの使用」を参照)、外部サウンドファイルをロードして再生することもできます。外部デバイスサウンドを再生するには、Sound オブジェクトの loadSound() メソッドを使用します。バンドルされたデバイスサウンドと同様に、Flash Lite プレーヤーでは、外部からロードされたオーディオがデバイスに渡され、デコードおよび再生されます。

次は、Flash Lite 2.0 の外部デバイスサウンドの再生について重要な情報です。

- デスクトップバージョンの Flash Player とは異なり、外部からロードされた Flash Lite 2.0 のサウンドは、常にイベントサウンドとして扱われます。このため、外部デバイスサウンドはストリーミングされません。つまり、ダウンロード中に再生されることはありません。サウンドデータを完全にダウンロードするまで、サウンドを再生できません。同じ理由により、サウンドが完全にロードされた後は、Sound オブジェクトの start() メソッドを呼び出してサウンドを再生する必要があります (次のコード例を参照してください)。
- loadSound() メソッドが実装された Flash Lite 2.0 では、そのメソッドの 2 番目のパラメータ (isStreaming) はサポートされていません。このパラメータが存在する場合、Flash Lite ではそのパラメータは無視されます。
- Flash Lite 2.0 では、外部からロードされた MP3 ファイルはネイティブに再生されません。アプリケーションが外部の MP3 ファイルをロードする場合は、他の外部からロードされたサウンドファイルと同様に、Flash Lite はサウンドデータをデバイスに渡し、デコードして再生します。

次のコードは、外部サウンドファイルをロードし、再生する方法を示しています。

```
// サウンドオブジェクトを作成します。
var mySound:Sound = new Sound();
// サウンドの onLoad ハンドラを定義します。
// このハンドラにより、完全にロードされたサウンドが開始されます。
mySound.onLoad = function(success){
    if(success == true) {
        mySound.start();
    }
}
// サウンドをロードします。
mySound.loadSound("http://www.adobe.com/audio.midi");
```

デバイスサウンドとアニメーションの同期について

Flash Lite のデバイスサウンドは、常にイベントサウンドとして扱われます。つまり、ネイティブ Flash サウンドの場合とは異なり、タイムラインでデバイスサウンドをアニメーションと同期することができません。ただし、デバイスサウンドを使用すると、Flash Lite プレーヤーの `_forceframerate` プロパティを `true` に設定することで、正確な同期サウンドに近づけることができます。このプロパティが `true` に設定されると、Flash Lite はアニメーションのフレームをドロップし、指定された SWF ファイルのフレームレートを維持します。デバイスサウンドデータが正しい持続時間に対して作成されており、サウンドデータが意図した速度で再生される限り、アニメーションとサウンドはほぼ同期されます。

たとえば、5 秒の長さのデバイスサウンドがあるとします。再生中、このサウンドをタイムラインのアニメーションと同期して再生したいとします。また、アプリケーションのフレームレートが 15 FPS に設定されていると想定します。タイムラインのフレームに割り当てるか、`Sound.start()` を呼び出すことでそのサウンドを開始し、同時に `_forceframerate = "true"` と設定します。その後、デバイスオーディオ再生の 1 秒間に、再生ヘッドがタイムラインで 15 フレーム前進していることが確認されます。何らかの理由で、この間にアニメーションの各フレームをプレーヤーでレンダリングできない場合は、指定されたフレームレートを維持するためフレームがドロップされます。

`_forceframerate` プロパティの詳細については、『Flash Lite 2.x ActionScript リファレンスガイド』の「`_forceframerate` プロパティ」を参照してください。

サポートされているデバイスサウンド形式の確認

`System.capabilities.audioMIMETypes` プロパティには、デバイスがサポートする MIME タイプのオーディオの配列が含まれています。この情報から、アプリケーションで再生できるデバイスオーディオのタイプを判別できます。配列のインデックスはサポートされている MIME タイプと同じなので、デバイスが特定のタイプのオーディオをサポートするかどうかは簡単に確認できます。

たとえば、次に示すコードは、外部の MIDI ファイルをロードする前に、デバイスが MIDI オーディオの再生をサポートするかどうかを最初に確認します。

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

Flash ネイティブサウンドの使用

デバイスサウンドのサポートに加え、Flash Lite 2.0 では標準の Flash サウンド、つまり、ネイティブ Flash サウンドもサポートされます。ネイティブサウンドとは、基本的には、Flash オーサリングツールが認識する形式によるサウンドです。Flash Lite のネイティブサウンドは、イベントサウンドまたは同期サウンドのいずれかとして再生できます。[56 ページの「デバイスサウンドの使用」](#)を参照してください。

Flash Lite アプリケーションのネイティブサウンドを使用する一般的なワークフローは、Flash デスクトップアプリケーションの場合と同じですが、次の例外があります。

- Flash Lite は外部の MP3 ファイルの再生をネイティブにサポートしていません。ただし、デバイスが MP3 オーディオを再生できる場合は、外部の MP3 ファイルをアプリケーションにロードできます。詳細については、[61 ページの「外部デバイスサウンドの再生」](#)を参照してください。
- Flash Lite は、[スピーチ] 圧縮オプションをサポートしていません。

Flash でのネイティブサウンドの操作の詳細については、『Flash ユーザーガイド』で次のトピックを参照してください。

- 『Flash ユーザーガイド』の「サウンドの読み込み」
- 『Flash ユーザーガイド』の「ドキュメントへのサウンドの追加」
- 『Flash ユーザーガイド』の「キーフレームでのサウンドの開始および停止」
- 『Flash ユーザーガイド』の「書き出し時のサウンドの圧縮」
- 『Flash ユーザーガイド』の「サウンドの読み込み」
- 『Flash ユーザーガイド』の「ドキュメントへのサウンドの追加」
- 『Flash ユーザーガイド』の「キーフレームでのサウンドの開始および停止」
- 『Flash ユーザーガイド』の「書き出し時のサウンドの圧縮」

8kHz サンプリングレート機能の使用

デフォルトでは、Flash オーサリングツールは、5、11、22、44 キロヘルツ (kHz) のサンプリングレートで、ネイティブサウンドを書き出します。しかし、多くのデバイスでは、これらのサンプリングレートでのオーディオデータの再生はサポートされていません。たとえば、Nokia Series 60 デバイスでは、8 kHz と 16 kHz のみがサポートされています。旧バージョンの Flash Lite では、デバイスのネイティブサウンドのアプリケーションプログラミングインターフェイス (API) を呼び出してサウンドを再生する前に、実行時のオーディオを 8 kHz にリサンプリングします。このリサンプリングでは、処理時間とメモリの消費量がさらに必要です。

Adobe Flash CS3 Professional では、オーサリングツールを使用して、アプリケーションのネイティブオーディオを SWF ファイルのパブリッシュ時に 8 kHz でリサンプリングできます。この設定は、アプリケーションのすべてのストリーミングサウンドとイベントサウンド、またはドキュメントの [ライブラリ] パネルの特定のサウンドのみに適用します。

次に述べる最初の手順では、[サウンドプロパティ] ダイアログボックスで、個々のサウンドに 8kHz のサンプリングレートオプションを有効にする方法について説明しています。次に続く 2 番目の手順では、[パブリッシュ設定] ダイアログボックスを使用して、このオプションをすべてのイベントサウンドとストリーミングサウンドにグローバルに設定する方法について説明しています。

×
H

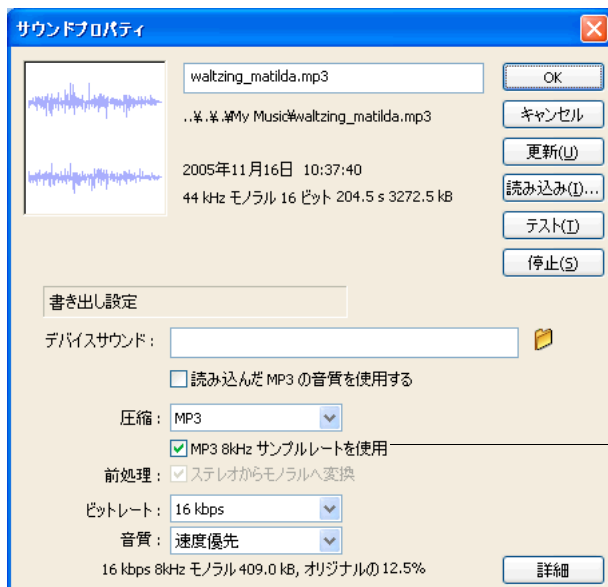
ストリーミングサウンドでは、[パブリッシュ設定] ダイアログボックスで指定した、サウンドのグローバル圧縮オプションが常に適用されます。[サウンドプロパティ] ダイアログボックスで指定したサウンドごとの圧縮オプションは、イベントサウンドにのみ適用されます。

個々のサウンドに 8kHz のサンプリングレート機能を有効にするには：

1. Flash で、[ライブラリ] パネルでサウンドシンボルを右クリック (Windows) または Control キーを押しながらクリック (Macintosh) し、コンテキストメニューから [プロパティ] を選択します。
2. 表示された [サウンドプロパティ] ダイアログボックスで、[圧縮] ポップアップメニューから [MP3] を選択します。

8kHz サンプリング機能は、MP3 圧縮オーディオでのみ使用できます。

3. 選択されていれば、[読み込んだ MP3 の音質を使用する] の選択を解除します。
4. [MP3 8kHz サンプルレートを使用] オプションを選択します。



[MP3 8kHz サンプルレートを
使用] オプション

5. [OK] をクリックします。

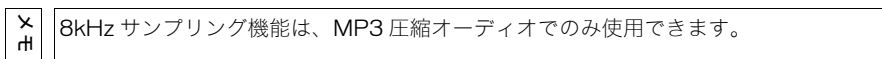
すべてのネイティブサウンドに 8kHz サンプルレート機能をグローバルに有効にするには：

1. Flash で、[ファイル]-[パブリッシュ設定] を選択します。
2. [パブリッシュ設定] ダイアログボックスで、[Flash] タブをクリックします。
3. 8kHz サンプルレートをアプリケーションのストリーミングサウンドすべてに有効にするには、次の操作を行います。
 - a. [ストリーミングの設定] サウンドオプションをクリックします。
 - b. 表示された [サウンドプロパティ] ダイアログボックスで、[圧縮] ポップアップメニューから [MP3] を選択します。

8kHz サンプルレート機能は、MP3 圧縮オーディオでのみ使用できます。
 - c. [MP3 8kHz サンプルレートを使用] オプションを選択します。



- d. [OK] をクリックして [パブリッシュ設定] ダイアログボックスを閉じます。
4. すべてのイベントサウンドに 8kHz サンプルレートを有効にするには、次の操作を行います。
 - a. [イベントの設定] サウンドオプションをクリックします。
 - b. 表示された [サウンドプロパティ] ダイアログボックスで、[圧縮] ポップアップメニューから [MP3] を選択します。



- c. [MP3 8kHz サンプルレートを使用] オプションを選択します。
 - d. [OK] をクリックします。
5. [OK] をクリックして [パブリッシュ設定] ダイアログボックスを閉じます。

デバイスビデオの使用

Flash Lite 2.0 ではデバイスビデオを再生できます。デバイスビデオとは、ターゲットデバイスによってサポートされるビデオ形式やエンコーディングのことです。プレーヤーのサイズを小さく保つため、また、各種のビデオ形式をサポートするため、Flash Lite 2.0 ではデバイスビデオをネイティブにデコードまたはレンダリングしません。その代わりに、Flash Lite 2.0 は、デバイスビデオをデバイスのディスプレイにデコードしてレンダリングすることをデバイスに依存します。このため、デバイスビデオの使用については、いくつかの制限事項があります。詳細については、[76 ページの「デバイスビデオの制限」](#)を参照してください。

一般的なデバイスビデオ形式には、3GP、3G2(または 3GPP2)、MPEG-4 があります。通常、Flash Lite 2.0 アプリケーションでは、ターゲットデバイスがサポートする任意のビデオ形式を再生できます。デバイスによって、サポートしているビデオコーデックおよびビデオ形式が異なります。特定のデバイスがサポートするビデオ形式を判別するには、デバイスの製造元からデバイスの仕様を確認するか、ActionScript を使用して、デバイスがサポートしている形式を確認します。[75 ページの「サポートされているビデオ形式の確認」](#)を参照してください。

デバイスビデオを配置するには、デバイスビデオを SWF ファイルのライブラリにバンドルするか([67 ページの「バンドルされたデバイスビデオの使用」](#)を参照)、ユーザーのデバイス上の外部ファイルまたはネットワークの場所から、デバイスビデオをロードします([73 ページの「外部デバイスビデオの使用」](#)を参照)。

デバイスビデオの再生を制御するには、ActionScript Video オブジェクトを使用します。Flash Lite 2.0 は、Video.play() や Video.pause() などのデスクトップバージョンの Flash Player では利用できないメソッドを Video オブジェクトに追加します。Video オブジェクトを使用してビデオ再生を制御する方法については、[76 ページの「ActionScript によるビデオの制御」](#)を参照してください。

このセクションでは、次のトピックについて説明します。

バンドルされたデバイスビデオの使用	67
外部デバイスビデオの使用	73
デバイスビデオシンボルのプロパティの表示と編集	74
サポートされているビデオ形式の確認	75
デバイスビデオの制限	76
ActionScript によるビデオの制御	76

バンドルされたデバイスビデオの使用

Flash Lite 2.0 アプリケーションにビデオを配置する 1 つの方法に、パブリッシュされた SWF ファイル内にビデオファイルをパッケージ化 (バンドル) する方法があります。バンドルされたビデオを使用するには、デバイスビデオをドキュメントライブラリに読み込みます。アプリケーションがパブリッシュされると、オーサリングツールはデバイスビデオファイルをパブリッシュされた SWF ファイル内にバンドルします。

SWF ファイルにデバイスビデオをバンドルすると、パブリッシュされた SWF ファイルのサイズが大幅に増大しますが、外部ファイルを開いたりストリーミングする必要がないため、非常に高い移植性および再生の信頼性が提供されます。

このセクションでは、次のトピックについて説明します。

- [67 ページの「デバイスビデオの読み込み」](#)
- [69 ページの「バンドルされたデバイスビデオの例」](#)
- [71 ページの「ライブラリからバンドルされたビデオを直接再生」](#)

デバイスビデオの読み込み

Flash Lite アプリケーションでビデオをバンドルするには、Flash ドキュメントにビデオを読み込みます。Flash ドキュメントにビデオを読み込むには、ビデオの読み込みウィザード、または [ビデオプロパティ] ダイアログボックスで使用可能な読み込み機能を使用します。どちらの方法でも、元のデバイスビデオデータが含まれるドキュメントのライブラリに、ビデオシンボルが表示されます。アプリケーションがパブリッシュされると、オーサリングツールは外部ビデオファイルのデータをパブリッシュされた SWF ファイル内にバンドルします。

ビデオの読み込みウィザードを使用してデバイスビデオを読み込むには：

1. Flash で、[ファイル]-[読み込み]-[ビデオの読み込み] を選択します。
ビデオの読み込みウィザードが表示されます。
2. コンピュータ上のファイルを読み込むオプションを選択して、[参照] をクリックします。
3. デバイスビデオファイルのあるフォルダを参照し、そのフォルダを選択します。
[ファイルを開く] ダイアログボックスに目的のビデオファイルが表示されない場合、または表示されても選択できない場合は、[ファイルの種類] ポップアップメニューから [すべてのファイル (*.*)] を選択 (Windows)、または [使用可能] ポップアップメニューから [すべての形式] を選択 (Macintosh) します。Flash オーサリングツールは大部分のデバイスビデオ形式を認識しないため、この操作は必要です。
4. [開く] をクリックします。

5. ビデオの読み込みウィザードで[次へ]をクリックし、ウィザードの[展開]スクリーンに進みます。
Flash Lite 2.0 の SWF ファイルでは、唯一の展開オプションはデバイスビデオを SWF ファイルへバンドルすることです。
6. [終了]をクリックしてビデオを読み込みます。
デバイスビデオファイルにリンクされた新しいビデオシンボルが、ドキュメントの[ライブラリ]パネルに表示されます。ビデオシンボルのプロパティは、[ビデオプロパティ]ダイアログボックスで編集できます。詳細については、[74 ページの「デバイスビデオシンボルのプロパティの表示と編集」](#)を参照してください。

[ライブラリ] パネルを使用してデバイスビデオを読み込むには：

1. Flash で、[ライブラリ]パネルを開きます ([ウィンドウ]-[ライブラリ])。
2. [ライブラリ]オプションメニューを開き、[新規ビデオ]を選択します。
[ビデオプロパティ]ダイアログボックスが表示されます。
3. [ビデオプロパティ]ダイアログボックスで、SWF ファイルのソースビデオをバンドルするオプションを選択して、[読み込み]をクリックします。
4. デバイスビデオファイルのあるフォルダを参照し、そのフォルダを選択します。
[ファイルを開く]ダイアログボックスに目的のビデオファイルが表示されない場合、または表示されても選択できない場合は、[ファイルの種類]ポップアップメニューから[すべてのファイル (*.*)]を選択 (Windows)、または[使用可能]ポップアップメニューから[すべての形式]を選択 (Macintosh) します。Flash オーサリングツールは大部分のデバイスビデオ形式を認識しないため、この操作は必要です。
5. [開く]をクリックします。
[ビデオプロパティ]ダイアログボックスで、[ActionScript に書き出し]オプションを選択して、[識別子]テキストボックスにストリングを入力し、ステージ上の別のビデオオブジェクトを使用してデバイスビデオクリップを再生します。詳細については、[71 ページの「ライブラリからバンドルされたビデオを直接再生」](#)を参照してください。
6. [OK]をクリックして、[ビデオプロパティ]ダイアログボックスを閉じます。
新しいビデオシンボルが、デバイスビデオファイルに関連付けられたドキュメントの[ライブラリ]パネルに表示されます。ビデオシンボルのプロパティは、[ビデオプロパティ]ダイアログボックスで編集できます。詳細については、[74 ページの「デバイスビデオシンボルのプロパティの表示と編集」](#)を参照してください。

バンドルされたデバイスビデオの例

このセクションでは、バンドルされたデバイスビデオファイルを再生するアプリケーションの作成方法について説明します。そのためには、最初にデバイスビデオを Flash ドキュメントに読み込み (67 ページの「[デバイスビデオの読み込み](#)」を参照)、ステージ上にビデオインスタンスを作成し、最後に ActionScript を追加してビデオを再生および停止します。

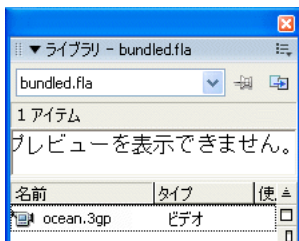
ライブラリから直接ビデオを再生することもできます。詳細については、71 ページの「[ライブラリからバンドルされたビデオを直接再生](#)」を参照してください。

バンドルされたデバイスビデオを読み込んで再生するには：

1. Flash で、Flash Lite 2.0 Symbian Series 60 テンプレートから新しいドキュメントを作成し、"bundled_video.fl" という名前で保存します。
2. ハードディスクの Flash CS3 インストールフォルダ内の "Samples and Tutorials" フォルダにある "ocean.3gp" という名前のデバイスビデオファイルを読み込みます。

デバイスビデオの読み込みの詳細については、67 ページの「[デバイスビデオの読み込み](#)」を参照してください。

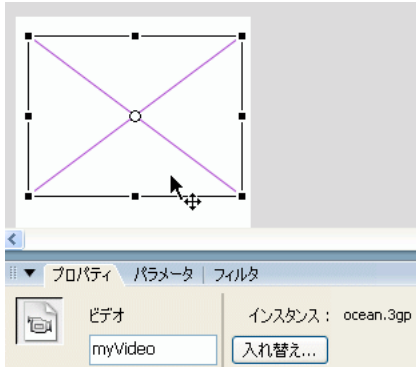
[ライブラリ] パネルに新しいビデオシンボルが表示されます。



3. タイムラインで Content レイヤーを選択して、[ライブラリ] パネルのビデオシンボルのインスタンスをステージにドラッグします。

4. ステージ上でビデオオブジェクトを選択し、プロパティインスペクタで、[インスタンス名] テキストボックスに「myVideo」と入力します。

このインスタンス名は、ActionScript コードでビデオオブジェクトを参照するときに使用します。



5. プロパティインスペクタで、ビデオオブジェクトの幅に 176、高さに 144 を設定します。
このサイズはソースビデオのサイズと同じです。デバイスによっては、Flash Lite のデバイスビデオが、ビデオオブジェクトの境界ボックスのサイズに合わせて拡大および縮小するとは限りません。詳細については、[76 ページの「デバイスビデオの制限」](#)を参照してください。
6. ビデオを制御するボタンを追加するには、作成済みのボタンのライブラリを開きます ([ウィンドウ]-[サンプルライブラリ]-[ボタン])。
7. ボタンライブラリで、"Circle Buttons" フォルダをダブルクリックして開きます。
8. Play ボタンシンボルのインスタンスを、ボタンライブラリからステージにドラッグします。
9. Stop ボタンシンボルのインスタンスを、ボタンライブラリからステージにドラッグします。
10. ステージ上で Play ボタンを選択し、[アクション] パネルを開きます ([ウィンドウ]-[アクション])。
11. [アクション] パネルに次のコードを入力 (またはコピー & ペースト) します。

```
on(press) {  
    myVideo.play();  
}
```
12. ステージ上で Stop ボタンを選択し、[アクション] パネル内で次のコードを入力します。

```
on(press) {  
    myVideo.stop();  
}
```
13. SWF ファイルをパブリッシュ ([ファイル]-[パブリッシュ])、テストのために SWF ファイルをデバイスに転送します。

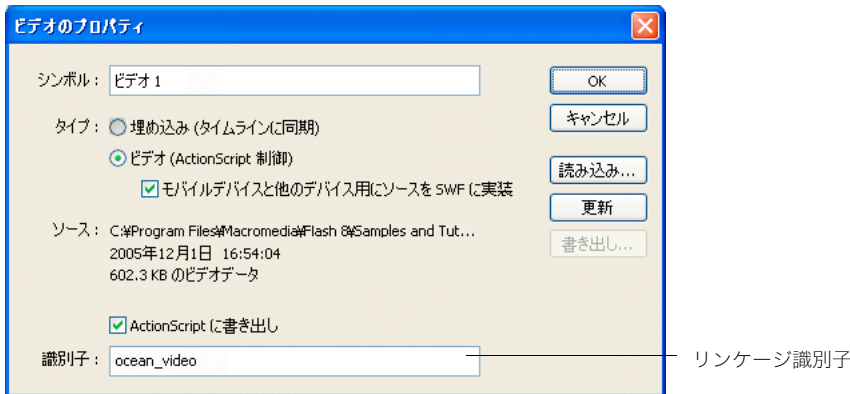


実際のデバイスでデバイスビデオをテストする必要があります。エミュレータでデバイスビデオはプレビューできません。

ライブラリからバンドルされたビデオを直接再生

本マニュアルの前の方で (67 ページの「バンドルされたデバイスビデオの使用」を参照)、バンドルされた単一のデバイスビデオを読み込んで再生する方法について説明しました。これを行うため、デバイスビデオをライブラリに読み込み、ステージ上にビデオシンボルのインスタンスを追加し、ビデオインスタンス上の `Video.play()` メソッドを呼び出しました。

ステージ上の1つの `Video` オブジェクトを使用して、バンドルされた複数のデバイスビデオをライブラリから直接再生することもできます。この場合は、アプリケーションのライブラリでデバイスビデオをバンドルします。また、次の図に示すように、ActionScript でビデオシンボルを参照できるようビデオシンボルに識別子を割り当てます。



その後、別のプレースホルダのビデオシンボルを作成し、ステージにそのインスタンスを追加します。プレースホルダビデオを使用してライブラリでデバイスビデオを再生するには、次の例に示すように、シンボルの ActionScript 識別子を `Video.play()` メソッドに渡します。

```
placeholderVideo.play("symbol://ocean_video");
```

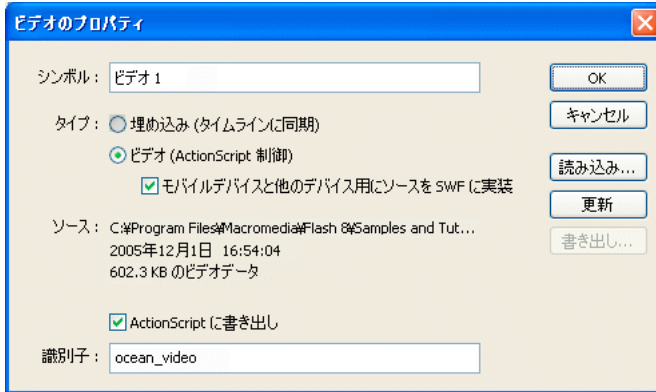
次の手順では、このテクニックを使用して、単一のビデオを直接ライブラリから再生する方法を示します。

ビデオを直接ライブラリから再生するには：

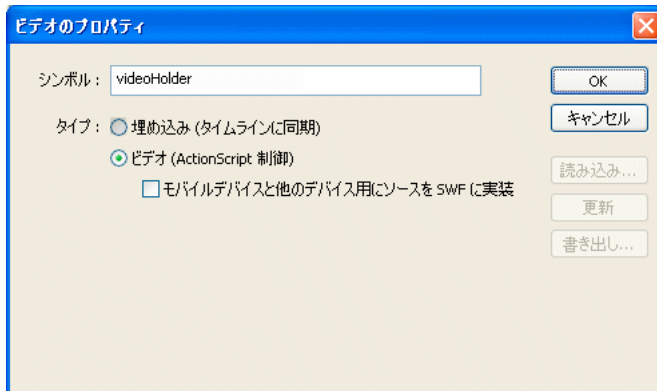
1. Flash で、Flash Lite 2.0 Symbian Series 60 テンプレートから新しいドキュメントを作成し、"library_video.fl" という名前で保存します。
2. www.adobe.com/go/learn_flt_samples_and_tutorials_jp にある "ocean.3gp" という名前のビデオファイルを読み込みます。「サンプルとチュートリアル」ページで、Flash Lite バージョンの .zip ファイルを探してダウンロードおよび解凍し、"Samples" フォルダを選択してファイルにアクセスします。

デバイスビデオの読み込みの詳細については、67 ページの「デバイスビデオの読み込み」を参照してください。

3. ライブラリ内の "ocean.3gp" ビデオシンボルを右クリック (Windows) または Control キーを押しながらクリック (Macintosh) し、コンテキストメニューから [プロパティ] を選択します。[ビデオのプロパティ] ダイアログボックスが表示されます。
4. 次の図に示すように、[ビデオプロパティ] ダイアログボックスで [ActionScript に書き出し] を選択し、[識別子] テキストボックスに「ocean_video」と入力します。



5. [OK] をクリックして、[ビデオプロパティ] ダイアログボックスを閉じます。
6. プレースホルダのビデオクリップを作成するには、次の操作をします。
 - a. [ライブラリ] パネルで、パネルのタイトルバーでオプションメニューボタンをクリックして、[新規ビデオ] を選択します。[ビデオプロパティ] ダイアログボックスが表示されます。
 - b. [シンボル] テキストボックスで、「videoHolder」と入力します。



- c. [OK] をクリックして、[ビデオプロパティ] ダイアログボックスを閉じます。
7. タイムラインで Content レイヤーを選択して、videoHolder シンボルのインスタンスをステージにドラッグします。

8. プロパティインスペクタで、[インスタンス名] テキストボックスに「myVideo」と入力します。
9. ビデオを制御するボタンを追加するには、作成済みのボタンのライブラリを開きます ([ウィンドウ]-[サンプルライブラリ]-[ボタン])。
10. ボタンライブラリで、"Circle Buttons" フォルダをダブルクリックして開きます。
11. Play ボタンシンボルのインスタンスを、ボタンライブラリからステージにドラッグします。
12. Stop ボタンシンボルのインスタンスを、ボタンライブラリからステージにドラッグします。
13. ステージ上でPlay ボタンを選択し、[アクション] パネルを開きます ([ウィンドウ]-[アクション])。
14. [アクション] パネルに次のコードを入力 (またはコピー & ペースト) します。

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

15. ステージ上で Stop ボタンを選択し、[アクション] パネル内で次のコードを入力します。

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

16. SWF ファイルをパブリッシュ ([ファイル]-[パブリッシュ])、テストのために SWF ファイルをデバイスに転送します。



エミュレータがすべての形式をサポートしているわけではないため、ターゲットデバイスでデバイスビデオをテストする必要があります。

外部デバイスビデオの使用

SWF ファイルにバンドルされたビデオを再生するだけでなく ([67 ページの「バンドルされたデバイスビデオの使用」](#)を参照)、デバイスのメモリカードにある外部ビデオファイルや、ネットワークアドレスからの外部ビデオファイルを再生することもできます。バンドルされたデバイスビデオと同様、このデバイスはデバイスビデオのデコードを実行します。

外部ビデオファイルを再生するには、絶対ファイルや相対ファイルの位置、またはビデオファイルの URL を、Video.play() メソッドに渡します。次の例では、SWF ファイルと 3GP ファイルはデバイス上の同じフォルダ内にあります。

```
myVideo.play("ocean.3gp");
```

また、次のように、相対フォルダパスを SWF ファイルに指定することもできます。

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

デバイスによっては、次のように、file:// プロトコルを使用して特定の位置でビデオファイルを再生することもできます。

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



すべてのデバイスが file:// プロトコルをサポートしているわけではありません。このプロトコルを使用する場合は、すべてのターゲットデバイスでアプリケーションのテストを実行してください。

デバイスによっては、ネットワークアドレスからビデオファイルをロードすることもできます。次の例に示すように、現在の実装では RTSP (Real Time Streaming Protocol) を使用したビデオストリーミングのみがサポートされています。HTTP ベースのストリーミングはサポートされていません。

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

デバイスビデオシンボルのプロパティの表示と編集

[ビデオプロパティ] ダイアログボックスでは、[ライブラリ] パネルのビデオシンボルの情報を表示および編集できます。

[ビデオプロパティ] ダイアログボックスを開くには、次のいずれかの操作を行います：

- [ライブラリ] パネルのビデオシンボルを右クリック (Windows) または Control キーを押しながらクリック (Macintosh) し、コンテキストメニューから [プロパティ] を選択します。
- [ライブラリ] パネルでビデオシンボルを選択し、パネルのタイトルバーのオプションメニューから [プロパティ] を選択します。

デバイスビデオをビデオシンボルに読み込むには：

1. [ライブラリ] パネルでビデオシンボルを選択し、[ビデオプロパティ] ダイアログボックスを開きます。
2. [ビデオプロパティ] ダイアログボックスで、SWF ファイルのビデオソースをバンドルするオプションが選択されていない場合は、ここで選択します。
3. ファイルのブラウザで [読み込み] をクリックし、読み込むデバイスビデオファイルを検索して、選択します。

[ファイルを開く] ダイアログボックスに目的のビデオファイルが表示されない場合、または表示されても選択できない場合は、[ファイルの種類] ポップアップメニューから [すべてのファイル (*.*)] を選択 (Windows)、または [使用可能] ポップアップメニューから [すべての形式] を選択 (Macintosh) します。Flash オーサリングツールは大部分のデバイスビデオ形式を認識しないため、この操作は必要です。

4. [開く] をクリックして、ファイルのブラウザを閉じます。
5. [OK] をクリックして、[ビデオプロパティ] ダイアログボックスを閉じます。



[OK] ボタンは、デバイスビデオを読み込むか、オプションの選択を解除してビデオソースを SWF ファイルにバンドルするまで、グレー表示されます。

ビデオシンボルに識別子を割り当てるには：

1. ビデオシンボルを選択し、[ビデオプロパティ] ダイアログボックスを開きます。
2. デバイスビデオをビデオシンボルに読み込んでいない場合は、ここで読み込みます。
3. [ActionScript に書き出し] を選択します。
4. [識別子] テキストボックスで、ビデオシンボルの識別子を入力します。

ActiveScript のすべての識別子のように、識別子の先頭の1文字は、英字、アンダースコア (`_`)、ドル記号 (`$`) のいずれかである必要があります。それ以降の文字には、英字、数字、アンダースコア、ドル記号を使用できます。

5. [OK] をクリックして、[ビデオプロパティ] ダイアログボックスを閉じます。

サポートされているビデオ形式の確認

デバイスによって、サポートしているビデオ形式およびエンコードが異なります。ビデオを使用する Flash Lite アプリケーションを開発する前に、デバイスの製造元の仕様書で、ターゲットデバイスがサポートするビデオ形式を判別します。

`System.capabilities.videoMIMETypes` プロパティを使用して、デバイスがサポートしているビデオ形式を判別することもできます。このプロパティには、デバイスがサポートするビデオの MIME タイプの配列が含まれています。配列の各アイテムには、次の形式があります。

`video/video-type`

たとえば、次のコードは、`mimeTypes_txt` という名前の `TextField` オブジェクトに、デバイスがサポートするすべてのビデオの MIME タイプを表示します。

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

`System.capabilities.videoMIMETypes` 配列に含まれるアイテムのインデックスは、サポートされているデバイスビデオの MIME タイプと同等です。これにより、デバイスが特定のビデオ形式をサポートするかどうかをすぐに検証できます。たとえば、次に示すコードは、3GPP タイプのビデオファイルを再生する前に、デバイスが 3GPP ビデオをサポートするかどうかを確認します。

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

デバイスビデオの制限

次の制限は、Flash Lite 2.0 アプリケーションでデバイスビデオを使用する場合に適用されます。

- デバイスビデオは、回転または傾斜できません。デバイスによっては、拡大 / 縮小をサポートします。
- デバイスビデオは、タイムラインとの同期はできません。
- デバイスビデオは、その他のメディアとの複合または混合はできません。デバイスはビデオを直接ディスプレイにレンダリングし、他の Flash コンテンツの上に表示します。
- ビデオクリップのサウンドボリュームは制御できません。

ActionScript によるビデオの制御

デバイスビデオの再生を制御するには、ActionScript Video オブジェクトを使用します。Flash Lite 2.0 の Video オブジェクトとデスクトップバージョンの Flash Player の Video オブジェクトでは、いくつかの相違点があります。

Video オブジェクトの次のメソッドは、Flash Lite 2.0 でのみ使用できます。デスクトップバージョンの Flash Player では使用できません。

- Video.play()
- Video.stop()
- Video.pause()
- Video.resume()
- Video.close()



Video オブジェクトのこれらのメソッドは、デスクトップバージョン Flash Player の NetStream オブジェクトで利用できる同様のメソッドにほぼ対応しています。

次のデスクトップバージョン Flash Player でのビデオオブジェクトのメソッドおよびプロパティは、Flash Lite 2.0 ではサポートされていません。

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

Flash Lite での Video オブジェクトの使用の詳細については、『Flash Lite 2.x ActionScript リファレンスガイド』の Video オブジェクトの項目を参照してください。

外部イメージのロード

Flash Lite 2.0 では、Flash Lite 1.1 と同じように、loadMovie() 関数 (または、MovieClip オブジェクトの loadMovie() メソッド) を使用して、外部の SWF ファイルをアプリケーションにロードすることができます。また、Flash Lite 2.0 では、loadMovie() 関数を使用して、デバイスがサポートする任意のイメージ形式をロードすることができます。

たとえば、ターゲットデバイスが PNG ファイルをデコードできると想定した場合、次のコードが Web サーバーにある外部 PNG ファイルをロードして表示します。

```
loadMovie("http://www.adobe.com/image.png", "image_target");
```

ターゲットデバイスがサポートしているイメージ形式を判別するには、System.capabilities.imageMIMETypes プロパティを使用します。これはサポートしているイメージの MIME タイプの配列を含んでいます。配列の各エレメントのインデックスは、サポートしている各 MIME タイプと同等です。

たとえば、次の ActionScript は、デバイスが外部 PNG ファイルのロードを試行する前に、PNG イメージをサポートするかどうかを確認します。

```
if (System.capabilities.imageMIMETypes["image/png"]) {  
    loadMovie("images/image.png", "mc_myPngImage");  
}
```

Flash Lite では、アプリケーションが特定のフレーム内で実行できる loadMovie() の操作の数を 5 つに制限し、一度に行う操作は合計 10 に制限しています。たとえば、次に示すように、アプリケーションのフレーム 1 にコードがあり、6 つの外部 JPEG イメージをロードしたとします。

```
image1.loadMovie("image1.jpg");  
image2.loadMovie("image2.jpg");  
image3.loadMovie("image3.jpg");  
image4.loadMovie("image4.jpg");  
image5.loadMovie("image5.jpg");  
image6.loadMovie("image6.jpg"); // ロードしない
```

この場合、最初の 5 つのイメージ (image1.jpg ~ image5.jpg) のみがロードされ、最後のイメージ (image6.jpg) はロードされません。これは、5 つという接続の上限に達したためです。解決策の 1 つは、各フレームが最大 5 つの loadMovie() 呼び出しを含むことができるよう、loadMovie() 呼び出しを複数のフレームに分割することです。

BREW 用 Flash Lite アプリケーションの開発

この章では、QUALCOMM Incorporated によって開発された BREW® (Binary Runtime Environment for Wireless®) を使用するデバイスで実行できる Adobe Flash Lite 2.x アプリケーションを開発する方法について説明します。Flash Lite を使用することには慣れていても、BREW デバイス用のアプリケーションを簡単に作成できるほどには BREW プラットフォームとその要件について十分な知識を持っていない開発者の方は、この章をぜひお読みください。他のアプリケーション開発技術に関する専門知識を持っていて、Flash Lite を使用した BREW アプリケーションの開発について理解したい場合にも、この章は役に立ちます。

この章は、次のトピックを取り上げています。

BREW の基礎	79
Flash Lite と BREW 開発ツール	81
BREW がサポートする Flash Lite の機能	82
BREW がサポートしない Flash Lite の機能	82
サポートされる BREW SDK のバージョン	83
サポートされるデバイス	84
その他の参照情報	84

BREW の基礎

QUALCOMM BREW のモバイルデバイスプラットフォームは、世界中のさまざまなメーカーのさまざまなデバイスでインストールおよびサポートされています。Adobe Flash CS3 Professional には、ユーザーが BREW デバイスにダウンロードして利用できる Flash Lite ベースのアプリケーションやコンテンツを作成するために、開発者が使用するツールとリソースが搭載されています。

Flash 開発者は、Flash オーサリングツールを使用して、BREW 対応デバイス用の Flash Lite コンテンツとアプリケーションを作成します。その後、開発者は、NSTL (National Software Testing Laboratories) にアプリケーションを提出し、アプリケーションの TBT (True BREW Testing) を受けます。TBT に合格したアプリケーションは、BDS (BREW Delivery System) にアップロードされ、キャリアによって配布用に選択されると、キャリアの ADS (アプリケーションダウンロードサーバー) で利用できるようになります。デバイスのユーザーは、このようなアプリケーションをダウンロードして購入し、BDS を介した無線による配布や請求に、BREW プラットフォームの主要な機能を利用できます。

フルスケールの BREW 開発に伴うコストをかけずに BREW を試してみたい開発者は、認証された BREW 開発者として登録せずに、無料の SDK コンポーネントだけをダウンロードできます。このコンポーネントだけでも BREW Simulator を使用してアプリケーションを生成およびテストできますが、アプリケーションをアップロードしてデバイスでテストしたり、カスタマイズされたデバイスパックで特定のデバイスの特性をシミュレートしたりすることはできません。

BREW は、ワイヤレスデバイスコンテンツの開発と配信をサポートするため、"エコシステム"として動作するように作られています。開発者が簡単にワイヤレスアプリケーションを作成して配布し、ワイヤレスアプリケーションから利益を得られるようにすることが目標です。このエコシステムは、以下の主要な関係者グループで構成されます。

開発者は、BREW SDK で利用できるツールを使用し、C++ と API を組み合わせて、または BREW 対応の拡張機能で、コンテンツを開発します。BREW SDK は無料でダウンロードできますが、ツールスイートおよび SDK ツールにアクセスするには QUALCOMM に登録する必要があります。

NSTL (National Software Testing Laboratories) は、開発者が提出したアプリケーションをテストし、合格したアプリケーションは BDS で提供できるようになります。合格しなかったアプリケーションは、修正して再度提出できます。

QUALCOMM は BDS (BREW Delivery System) を管理します。BDS は Web ベースのシステムで、QUALCOMM のカタログにリストされているコンテンツとその価格および請求の情報を提供します。価格には、オペレータのオークションなど、複数のモデルが含まれる場合があります。

オペレータ (キャリア) は、ADS (アプリケーションダウンロードサーバー) を使用して、選択したコンテンツを、顧客が OTA (無線) 配信メカニズムを通して購入できるよう、小売価格で提供します。

デバイスのメーカーは、BREW クライアントを使用します。BREW クライアントは、ワイヤレスアプリケーションの標準化された開発用に、API (アプリケーションプログラミングインターフェイス) の基本セットを公開します。クライアントにはアプリケーションマネージャも含まれ、ユーザーはこれを購入して BREW アプリケーションの管理に使用できます。

Flash Lite と BREW 開発ツール

BREW 環境用の Flash Lite アプリケーションを開発してテストするには、特定のハードウェアツールやソフトウェアツールを入手してインストールする必要があります。

ハードウェア

■ Windows コンピュータ

BREW SDK および BREW ツールスイートのシステム要件については、『BREW 開発スタートガイド』(BREW の Web サイトで入手できます)の「System Requirements」を参照してください。Adobe 製品のシステム要件については、www.adobe.com/go/sysreqs_jp を参照してください。

■ テストビットがセットされた BREW 対応のデバイス

BREW デバイスを購入し、アクティブにして、テスト対応にする方法については、『BREW 入門』(BREW の Web サイトで入手できます)の「Handset Acquisition and Readiness」を参照してください。BREW 2.x をサポートするデバイスは、テスト対応にするには QUALCOMM に送る必要があります。BREW 3x のデバイスはあらかじめ対応になっています。

■ データケーブル

サポートする予定の BREW デバイスに Flash アプリケーションをアップロードするには、データケーブルが必要です。ほとんどのデバイスにはデータケーブルがバンドルされていますが、ケーブルがない場合は、『BREW 入門』(BREW の Web サイトで入手できます)の「Acquire BREW Handsets and Data Cables」を参照してください。

ソフトウェア

■ BREW SDK および BREW ツールスイート

BREW SDK およびツールのインストールについては、『BREW 開発スタートガイド』(BREW の Web サイトで入手できます)の第 2 章「Installing the BREW SDK」を参照してください。BREW ツールスイートをダウンロードしてインストールするには BREW ディベロッパーとして登録済みである必要がありますが、SDK 自体については登録の必要はありません。登録方法については、BREW の Web サイトを参照してください。

■ 対象デバイス用の USB ドライバ

対象とするデバイス用の USB ドライバを入手してインストールします。ドライバの入手方法については、デバイスの製造元に問い合わせてください。

■ BREW 対応デバイス用 Flash Lite 拡張機能 (テスト目的でのみ、BREW の Web サイト から入手できます)。

ユーザーが Flash Lite for BREW アプリケーションを選択してダウンロードすると、拡張機能がユーザーのデバイスに自動的にダウンロードされます。

これらのコンポーネントの詳細については、[84 ページの「BREW 用のシステムのセットアップ」](#)を参照してください。

BREW がサポートする Flash Lite の機能

Flash Lite for BREW は、Flash Lite 2.0 および Flash Lite 2.1 の両方で使用できる機能のサブセットをサポートします。Flash Lite for BREW がサポートする機能は次のとおりです。これらの機能の詳細については、[7 ページの「Flash Lite 2.x の機能について」](#)を参照してください。

- アドビ システムズ社の Macromedia® Flash® Player 7 での構築、ActionScript 2.0 のサポート
- XML データ処理
- 永続データ管理
- デバイス、ベクターフォント
- イベントに対するデバイスのサウンド
- インラインビデオ (デバイスでサポートされるビデオ形式に限られます)
- ストリーミングビデオ
- XML ソケット
- ネットワークアクセスと HTTP ストリーミング
- インラインテキストのサポート (64,000 文字に限られ、個別のデバイスの実装によってさらに限定されます。一部のデバイスでは、1000 文字しかサポートされません)
- 動的マルチメディア
- キーパッドとキーボードのサポート

BREW がサポートしない Flash Lite の機能

BREW デバイスの製造元が、Flash Lite の一部の機能の使用を制限する場合があります。つまり、そのような機能は、そのデバイスでは使用できません。特定のデバイスに関する情報については、BREW の Web サイトの [ディベロッパーリソース] エリアで入手できます。

特定のデバイスでサポートされない機能に加えて、以下の一覧は、現在の Flash Lite BREW の実装でサポートされていない機能です (したがって、どのデバイスでも使用できません)。

Flash Lite サウンドデコード すべてのサウンド形式が、変更されずにデバイスに渡されて再生されます。たとえば、サウンドが MP3 形式で、デバイスが MP3 をサポートする場合は、サウンドが再生されます。Flash Lite は、他のプラットフォームではサウンドをデコードしますが、BREW の内部ではサウンドをデコードしません。

ストリーミングサウンド すべてのサウンドは、完全にロードされてからでないと再生できません。現在の BREW の実装では、ネットワーク経由でロードしながらサウンドを再生することはできません。

ActionScript コマンド ActionScript の以下の `fscommand2()` 関数は、どの BREW デバイスでも使用できません。

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

ActionScript の以下の `fscommand()` 関数は、どの BREW デバイスでも使用できません。

<code>fscommand(Launch)</code>	
--------------------------------	--

getURL() のサポート 実装により、BREW ハンドセットの中には一部の `getURL()` 関数をサポートするものがあります。一般に、以下のプロトコルはサポートされません。

- HTTP (Flash Lite がブラウザを起動できる API がありません)
- MMS (Multimedia Message Service)

<div>× ❌</div>	<code>mailto</code> 要求は直接的にはサポートされません。代わりに、 SMS (Short Message Service) プロトコルを使用できます。 SMS では、メッセージの最大サイズは 160 文字に制限されます。
--------------------	--

壁紙コンテンツ Flash Lite の壁紙コンテンツは、Flash Lite for BREW の実装ではサポートされません。

着信アニメーション 着信アニメーションは、Flash Lite for BREW の実装ではサポートされません。

サポートされる BREW SDK のバージョン

Flash Lite 2.1 for BREW Devices は、現時点では BREW バージョン 2.1.3 以降をサポートしています。BREW Web サイトのデバイス仕様の一覧では、特定のデバイスで稼働している BREW のバージョンを確認できます。現在 Flash Lite for BREW をサポートしている 2 種類のデバイス (Samsung SCH-A950 と LG VX9800) は、サポートしている BREW のバージョンが異なります (それぞれ 2.x と 3.x)。BREW SDK には後方互換性があるので、開発対象のデバイスでサポートされているバージョンではなく、最新のバージョン (現時点では 3.x) に対する SDK とツールをダウンロードできます。たとえば、2.x をサポートする Samsung SCH-A950 が対象の場合でも、バージョン 3.x の BREW SDK とツールスイートをダウンロードして使用できます。

ユーザーインターフェイスに関しては BREW 2.x と 3.x の間に違いはほとんどなく、Flash Lite はどちらのプラットフォームでも同じように動作します。大きな違いは、デバイスへのファイルの格納方法と、BREW ツールスイート内のアプリケーションのロード方法です。ファイルシステムの構造の詳細については、[95 ページの「異なる BREW バージョンのデバイスファイルの構造」](#)を参照してください。

サポートされるデバイス

現在 BREW プラットフォームをサポートしているデバイスは次のとおりです。

デバイス	サポートされるバージョン
Samsung SCH-A950	BREW バージョン 2.1.3
LG VX9800	BREW バージョン 3.1.2

新しく利用できるようになったデバイスがあれば、アドビ システムズ社のカスタマーサポートチームがそれを認定します。現在サポートされているデバイスの一覧については、http://www.adobe.com/jp/mobile/supported_devices/ を参照してください。

その他の参照情報

BREW の詳細については、BREW の Web サイトを参照してください。このサイトのディベロッパーホームページでは、BREW SDK、BREW ツールとユーティリティ、および BREW をサポートする特定のデバイスの特性に関する情報が提供されています。

×
中

BREW ツールスイートをダウンロードして使用したり、デバイスで BREW をテストできるようにしたりするには、BREW ディベロッパーとして登録する必要があります。登録方法の詳細については、サイトを参照してください。

BREW 用のシステムのセットアップ

ここでは、BREW プラットフォームを使用するデバイスで実行できる Flash Lite アプリケーションを開発するためのシステムのセットアップ方法について説明します。BREW 用の Flash Lite ファイルを作成するためのシステムをセットアップするには、複数のソースから適切なソフトウェアを探してインストールする必要があります。また、適切なハードウェア (BREW 対応のデバイスやデータケーブルなど) を入手する必要もあります。これらのコンポーネントをインストールして構成するために必要な具体的な情報の多くは、他の Flash Lite または BREW のマニュアルに記載されています。したがって、ここでは、他のマニュアルに記載されている情報を繰り返すことはせずに、インストールと構成の手順に関する概要と併せて、各セットアップ作業の実行方法に関する詳細情報の適切なソースを示します。

この項では、次のトピックについて説明します。

BREW 用のシステムのセットアップのワークフロー	85
BREW ディベロッパーとしての登録	85
BREW SDK とツールのインストール	85
デバイス用 USB ドライバのインストール	86
その他の参照情報	86

BREW 用のシステムのセットアップのワークフロー

BREW 用の Flash Lite オーサリングをサポートするシステムをセットアップするには、次の作業を行います。

1. BREW ディベロッパーとして登録します。
2. BREW SDK およびツールスイートをインストールします。
3. Flash Lite 2.1 for BREW をインストールします。
4. 対象デバイス用の USB ドライバをインストールします。

BREW ディベロッパーとしての登録

BREW ディベロッパーとして登録するには、QUALCOMM のメイン BREW サイトにアクセスし、ディベロッパーホームページのリンクに従って、認証された BREW ディベロッパーになります。

BREW SDK とツールのインストール

ここでは、コンピュータに BREW SDK をインストールする方法について説明します。BREW 開発者としての登録は済んでいるものとします。BREW SDK またはツールスイートのインストールの詳細については、『BREW 開発スタートガイド』の「Installing the BREW SDK」を参照してください。BREW の Web サイトにアクセスし、[ディベロッパー]-[ディベロッパーサポート]-[BREW ドキュメンテーション]-[アプリケーション開発ドキュメンテーション]を選択します。

BREW SDK とツールスイートをインストールするには：

1. BREW の Web サイトにある BREW ディベロッパーホームで、[ダウンロード BREW ツール] をクリックして [BREW Tools] ページを表示します。
2. [BREW の開発] エリアに移動し、適切なリンクをクリックして、最新バージョンの BREW SDK をダウンロードします (本書作成時点では BREW SDK 3.1)。
画面に表示される指示に従って、SDK をダウンロードしてインストールします。

3. SDK をインストールした後、ツールをインストールします。

[BREW Development and Commercialization Tools] ページの [BREW Commercialization] エリアに移動し、[BREW Testing and Commercialization Utilities] のリンクをクリックします。画面に表示される指示に従って、ツールスイートをコンピュータにインストールします。

[スタート] メニューに、BREW SDK と BREW ツールスイートの項目が追加されます。

Adobe Flash Lite 2.1 for BREW Devices のダウンロード

Flash Lite 2.1 for BREW Devices には、3 つのコンポーネントが含まれます。

- Flash Lite Publisher for BREW (http://www.adobe.com/go/support_flashlite_jp から入手できます)
- Flash Lite 2.1 for BREW Simulator (http://www.adobe.com/go/support_flashlite_jp から入手できます)
- BREW 対応デバイス用 Flash Lite 拡張機能 (テスト目的でのみ、BREW の Web サイトから入手できます。ユーザーが Flash Lite for BREW アプリケーションを選択してダウンロードすると、拡張機能がユーザーのデバイスに自動的にダウンロードされます)

デバイス用 USB ドライバのインストール

BREW 開発の対象となる各デバイスには、独自の USB ドライバのセットが含まれます。これをコンピュータにインストールする必要があります。新しいデバイスには、ドライバが収められた CD-ROM が付属しているはずです。可能な場合には常に、このバージョンを使用します。デバイスの USB ドライバをインストールするには、Windows にデバイスドライバをインストールする際の標準的な手順に従います。

その他の参照情報

Flash Lite アプリケーションの開発とテストについては、このマニュアルの他の項を参照してください。Flash Lite アプリケーションの開発とテストの方法に関する広範な情報が提供されています。Flash CS3 Professional のオンラインヘルプにも、Flash Lite の開発に関するさまざまな情報が含まれます。

BREW の詳細についての最適な情報源は、QUALCOMM BREW の Web サイトです。

BREW の開発プロセスとツールの詳細については、SDK に含まれるヘルプファイルも参照してください。

これを書いている時点では、BREW のマニュアルセットには似た名前の 2 種類のマニュアルが含まれており、どちらも BREW の基本を学習するのに役立ちます。次の 2 つのマニュアルがそれです。

- 『BREW 入門』は、BREW アプリケーションを開発するために必要なハードウェアとソフトウェアを 2 ページにまとめたものです。
- 『BREW 開発スタートガイド』はさらに包括的なマニュアルで、BREW アプリケーションの開発を始めるために必要なすべての情報が含まれます。

BREW 用 Flash Lite ファイルのオーサリング

BREW 向けのオーサリングは一般的な Flash Lite 向けのオーサリングと似ていますが、設計対象の特定の BREW 対応ハンドセットの特徴に合わせて、アプリケーションを調整する点が異なります。ここでは、Flash Lite Publisher for BREW (91 ページの「[BREW 用 Flash Lite ファイルのパブリッシュ](#)」で説明します) を使用して BREW にパブリッシュできる Flash Lite コンテンツを、Flash を使用してオーサリングする方法について説明します。

この項では、次のトピックについて説明します。

BREW 用 Flash Lite ファイルのオーサリングのワークフロー	87
対象デバイスとコンテンツタイプの識別	88
Flash でのアプリケーションの作成	88
Flash でのアプリケーションのテスト	90
その他の参照情報	91

BREW 用 Flash Lite ファイルのオーサリングのワークフロー

BREW プラットフォーム用 Flash Lite コンテンツの作成は繰り返し行われるプロセスであり、次の作業が含まれます。

対象のデバイスと Flash Lite コンテンツのタイプを識別する 本マニュアルの作成時点では、次の 2 つのデバイスが BREW をサポートしています。Samsung SCH-A950 は BREW バージョン 2.1.3 プラットフォームを使用し、LG VX9800 は BREW バージョン 3.1.2 プラットフォームを使用しています。これらのデバイスの特徴は大きく異なるので (たとえば、LG には QWERTY キーボードがあり、Samsung にはありません)、デバイスの固有機能に合わせてアプリケーション開発計画を作成することが重要です。

また、2種類のコンテンツタイプ(アプリケーションとスクリーンセーバー)を作成できますが、必要なアプリケーション設計の考慮事項はコンテンツタイプによって異なります。スクリーンセーバーとアプリケーションの相違点の詳細については、BREW の Web サイトで、[ディベロッパー FAQ]-[BREW ツール]-[MIF の設定]-[スクリーンセーバー]を参照してください。



現時点では、Samsung SCH-A950 のみがスクリーンセーバーのコンテンツタイプをサポートしています。

Flash でアプリケーションを作成してテストする Adobe Flash CS3 Professional には Adobe Device Central CS3 のエミュレータが含まれており、デバイスに転送しなくてもアプリケーションをテストできます。Device Central エミュレータを使用してアプリケーションの全体的な設計を微調整し、モバイルデバイスでテストする前に問題を解決できます。



エミュレータを使用して Flash で行うテストは、BREW SDK に含まれる Simulator を使用して行うテストとは異なります。ここでは Flash エミュレータでのテストについて説明し、BREW Simulator でのテストについては [91 ページの「BREW 用 Flash Lite ファイルのパブリッシュ」](#)で説明します。現時点では、Device Central エミュレータは BREW デバイス用のデバイスパックを提供していないので、Device Central エミュレータを使用して行うことができるのは基本機能のテストのみです。BREW 環境でのアプリケーションの機能をテストするには、BREW Simulator を使用してください。

対象デバイスとコンテンツタイプの識別

QUALCOMM ディベロッパーホームには、BREW のいずれかのバージョンを現在サポートしているすべてのデバイスの具体的な特徴に関する情報が含まれます。まだ市販されていない商用化前のデバイスに関する情報も入手できます。この情報を使用して、BREW アプリケーションの対象にするデバイスを決定できます。BREW SDK には異なるデバイスパックのサンプルアプリケーションもいくつか含まれています。そのサンプルアプリケーションを使用して、特定のデバイスの動作を詳しく知ることができます。

Flash でのアプリケーションの作成

BREW 用の Flash Lite アプリケーションを作成するプロセスは、一般的な Flash Lite アプリケーションの作成と似ています。BREW デバイスがサポートしていない機能 ([82 ページの「BREW がサポートしない Flash Lite の機能」](#)を参照) および BREW 固有のオーサリングに関する注意 (後の説明を参照) を除けば、BREW デバイス用のアプリケーションの開発は、Flash のマニュアルで詳細に説明されているそれ以外の開発手順に従って行うことができます。

以下のトピックでは、BREW 用の Flash Lite オーサリングのみに適用される情報について説明します。

ブラウザの起動

BREW の対象のハンドセットは OpenWave ブラウザを使用しますが、このブラウザでは BREW アプリケーションから起動するための適切なフックが提供されていません。したがって、`getURL()` 関数、`loadVars(send)` 関数、または `XML.send()` 関数で HTTP URL を指定しても、ブラウザは起動しません。サーバーに情報を送信し、新しいウィンドウを開かずに SWF ファイルの再生を続けるには、またはウィンドウやフレームのコンテンツを置き換えるには、`loadVars.sendAndLoad()` 関数を使用します。

デフォルトソフトキーの動作

Flash Lite でのソフトキーの動作をプログラムする方法に関する情報と例については、[32 ページの「ソフトキーの使用」](#)を参照してください。次の表は、Flash Lite 2.1 for BREW でのデフォルトのソフトキーの動作をまとめたものです。

コンテンツのアクション	スクリーンモード	左キーアクション	右キーアクション
デフォルト (コンテンツは <code>SetSoftKeys</code> コマンドを呼び出さない)	フルスクリーンモード	オプションメニューを表示する	プレーヤーを終了する
非フルスクリーンモード (コンテンツは <code>SetSoftKeys</code> を呼び出さない)	非フルスクリーンモード	オプションメニューを表示する	プレーヤーを終了する
コンテンツはソフトキーを無効にする	適用外	なし	なし
コンテンツは一方または両方のソフトキーを上書きする (<code>SetSoftKeys</code> を呼び出す)	適用外	上書きしない場合はオプションメニューを表示する。 上書きする場合は、コンテンツで指定されている動作。	上書きしない場合は、プレーヤーを終了する。 上書きする場合は、コンテンツで指定されている動作。

Flash でのアプリケーションのテスト

アプリケーションの作成と同様に、Adobe Device Central エミュレータを使用した BREW アプリケーションのテストは、一般的な Flash Lite アプリケーションのテストと似ています。前にも説明しましたが、Device Central エミュレータを使用して Flash で行うテストは、BREW SDK に含まれる Simulator を使用して行うテストとは異なります。ここでは Device Central エミュレータでのテストについて説明し、BREW Simulator でのテストについては [91 ページの「BREW 用 Flash Lite ファイルのパブリッシュ」](#) で説明します。

現時点では、Device Central エミュレータは BREW デバイス用のデバイスパックを提供していないので、Device Central エミュレータを使用して行うことができるのは基本機能のテストのみです。BREW 環境でのアプリケーションの機能をテストするには、BREW Simulator を使用してください。

Adobe Device Central エミュレータを使用してアプリケーションをテストするには：

1. Flash でアプリケーションを開きます。
2. [ファイル]-[パブリッシュ設定] を選択します。[Flash] タブで、[バージョン] リストから [Flash Lite 2.1] を選択し、[ActionScript のバージョン] リストから [ActionScript 2.0] を選択します。[OK] をクリックして設定を保存します。
3. Adobe Device Central エミュレータを起動 ([制御]-[ムービープレビュー]) し、[デバイスプロファイル] タブをクリックします。
4. [使用可能なデバイス] ペインで、次のいずれかを行います。
 - a. 対象デバイス用のデバイスパックをダウンロードしていない場合は、[汎用]-「Flash Lite 2.0」-[一般携帯電話] を選択します。
 - b. 対象デバイス用のデバイスパックをダウンロードしてある場合は
(www.adobe.com/jp/products/flash/download/device_profiles から入手できます)、デバイス名をクリックして展開し、適切なデバイスのサイズを選択します。次に、対象デバイスの名前を [使用可能なデバイス] ペインから [デバイスセット] ペインにドラッグします。
5. [ファイル]-[Flash に戻る] を選択し、Flash メニューの [制御]-[ムービープレビュー] を選択します。BREW Publisher ウィザードが表示される場合は、[キャンセル] をクリックしてダイアログボックスを閉じます。Flash からアプリケーションが Device Central に書き出され、エミュレータで開かれます。
6. エミュレータを使用してアプリケーションをテストします。

エミュレータの使用方法については、[114 ページの「エミュレータの使用」](#) を参照してください。エミュレータでのアプリケーションのテストが終了したら、ファイルを保存して、BREW プラットフォームでアプリケーションをパブリッシュする方法に関する [91 ページの「BREW 用 Flash Lite ファイルのパブリッシュ」](#) の説明に進みます。

その他の参照情報

Flash Lite アプリケーションの開発とテストに関する他の情報については、Flash のヘルプに含まれる以下のソースを参照してください。

- Flash Lite 2.x ファーストステップガイド (PDF)
- Flash Lite コンテンツのプレビュー (ヘルプトピック)

BREW 用 Flash Lite ファイルのパブリッシュ

ここでは、Flash Lite Publisher for BREW ウィザードを使用して、BREW プラットフォームを使用するデバイスで実行できる Flash Lite コンテンツを生成する方法について説明します。Flash Lite でのオーサリング方法は既に理解しているものとして、作成済みの Flash Lite ファイルを BREW 対応のデバイスで使用できるようにパブリッシュする方法についてのみ説明します。

この項では、次のトピックについて説明します。

BREW 用の Flash Lite ファイルのパブリッシュについて.....	91
異なる BREW バージョンのデバイスファイルの構造	95
BREW 用 Flash Lite ファイルのパブリッシュとテストのワークフロー	97
BREW Publisher ウィザードを使用したアプリケーションファイルのパブリッシュ..	97
BREW Simulator を使用したアプリケーションのテスト	101
デバイスパック	101
その他の参照情報	103

BREW 用の Flash Lite ファイルのパブリッシュについて

BREW デバイスで実行するには、Flash Lite アプリケーションが BREW 標準に準拠している必要があります。Flash Lite for BREW Publisher ウィザードを使用すると、必要な手順の一部が自動化され、この標準に従うアプリケーションの作成が容易になります。アプリケーション自体のファイルを生成するだけでなく、このウィザードは、BREW SDK の Simulator ツールを使用して、BREW アプリケーションをテストするために使用できるファイルの生成も行います。

BREW Publisher ウィザードを使い始める前に、BREW アプリケーションのファイル形式、ファイル構造、および要件の基本を理解しておくに役に立ちます。

ファイル形式

Flash Lite Publisher for BREW の主な目的は、標準 SWF ファイルをパブリッシュした後、ポストプロセッサを使用して、BREW デバイスで実行できるように SWF ファイルをパッケージすることです。次の表は、BREW デバイスで Flash Lite コンテンツを使用するために必要なファイルの一覧です。

ファイル	ファイルの内容
MIF ファイル	構成情報
MOD ファイル	BREW デバイス用のアプリケーションファイル
SIG ファイル	QUALCOMM から提供されるデバイス固有のファイル

BREW Simulator で Flash Lite コンテンツを使用するために必要なファイルは次のとおりです。

ファイル	ファイルの内容
MIF ファイル	構成情報
DLL ファイル	BREW Simulator 用のアプリケーションファイル

これらのファイルはすべて同じファイル名 (および適切なファイル拡張子である .mif、.mod、.sig、または .dll) でなければならず、ファイル名はアプリケーションフォルダ名とも同じでなければなりません。これらのファイルが機能するためには、デバイス上の正しいフォルダにファイルを格納する必要があります。フォルダの構造は、対象デバイスが使用する BREW のバージョンによって異なります。詳細については、[95 ページの「異なる BREW バージョンのデバイスファイルの構造」](#)を参照してください。

MIF ファイル

MIF ファイルは、BREW アプレットが必要とする情報を提供する構成ファイルです。BREW Publisher ウィザードは、パブリッシュプロセスの一部として、自動的に MIF ファイルを生成します。ただし、個々のコンテンツ開発者は、MIF の形式と要件を理解し、True BREW Testing にアプリケーションを提出する前に、MIF ファイルの設定が正しいことを確認する必要があります。

次のコードは、アプリケーションとスクリーンセーバーに対して BREW Publisher ウィザードから出力されるデフォルトの MIF 値を示しています。これらの値がアプリケーションに対して適切でない場合は、BREW の Web サイトから MIF Editor ツールをダウンロードし、テスト用にアプリケーションを提出する前にファイルを編集してください。

```
<Applet>
- List of Applets defined in this Module: [Class ID in post processor]
- Applet Information:
* Name: [Applet Name in post processor]
* Class ID: [Class ID in post processor]
* Applet Type: [blank]
```

- Graphics
- * Large: [Large Icon in post processor]
- * Medium: [Medium Icon in post processor]
- * Small: [Small Icon in post processor]

<Extension>

- Exported Classes: [blank]
- Exported MIME Types: [blank]

<Dependencies>

- External Classes Used by this Module: [blank]
- External Classes Directory (Checkbox): [checked]
- Used: [Class IDs generated by post processor]

<Privileges>

File [checked]

Network [checked]

Web Access [checked]

TAPI [checked]

<Access Control>

- ACL type [blank]
- Rights [blank]
- Groups [blank]

<License>

No License (Checkbox): [checked]

<General>

- Author Name: [Author in post processor]
- Copyright String: [Copyright in post processor]
- Module Version: [Version in post processor]
- EFS Restrictions
- * Max Files: [blank]
- * Max Space: (bytes) [blank]
- String Encoding Format: [ISOLATIN1]

スクリーンセーバーに対するデフォルトの MIF 出力

<Applet>

- List of Applets defined in this Module: [Class ID in post processor]
- Applet Information:
 - * Name: [Applet Name in post processor]
 - * Class ID: [Class ID in post processor]
 - * Applet Type: [blank]
- Graphics
 - * Large: [Large Icon in post processor]
 - * Medium: [Medium Icon in post processor]
 - * Small: [Small Icon in post processor]
- Notifications, Flags, Settings...
 - * Flags: Screensaver [checked]

<Extension>

- Exported Classes: [blank]
- Exported MIME Types: [brew/ssaver Class ID]

<Dependencies>

- External Classes Used by this Module: [blank]
- External Classes Directory (Checkbox): [checked]
- Used: [Class IDs generated by post processor]

```
<Privileges>
All unchecked
<Access Control>
- ACL type [blank]
- Rights [blank]
- Groups [blank]
<License>
No License (Checkbox): [checked]
<General>
- Author Name: [Author in post processor]
- Copyright String: [Copyright in post processor]
- Module Version: [Version in post processor]
- EFS Restrictions
  * Max Files: [blank]
  * Max Space: (bytes) [blank]
- String Encoding Format: [ISOLATIN1]
```

SIG ファイル

SIG ファイルは、アプリケーション開発に使用するデバイスごとに必要な、デバイス固有の署名ファイルです。これらのファイルは QUALCOMM から取得する必要があり、90 日間は有効です。90 日が経過すると、署名ファイルの有効期間が切れ、QUALCOMM から新しい署名ファイルの発行を受ける必要があります。

必要な SIG ファイルは、デバイスごとに1つだけです。ただし、個別のアプリケーションごとに対応する専用の SIG ファイルが必要なので、元の SIG ファイルをコピーし、作成したアプリケーションの名前と一致するようにファイル名を変更します。SIG ファイルは、BREW デバイス用のアプリケーションファイルである MOD ファイルと同じフォルダに存在する必要があります。

BREW の Web サイトで署名生成ユーティリティにアクセスするには、BREW ディベロッパーとして認定されている必要があります。



SIG ファイルは、デバイスでアプリケーションをテストするためには必要ですが、BREW Simulator を使用するテストには必要ありません。

SIG ファイルを取得するには：

1. BREW の Web サイトにアクセスし、QUALCOMM の Web ベースのテスト署名ジェネレータへのリンクが含まれるページに移動します。
2. [BREW Testing Generator] を選択します。
3. BREW ディベロッパーのユーザー名とパスワードを入力します。
4. [ESN] フィールドに、「Ox<ESN 番号>」と入力します。ESN 番号は、デバイスのバッテリーを持ち上げるとその下に記載されている番号です。
5. [Generate] をクリックし、[Download Signature] のリンクが表示されるのを待ちます。
6. 署名をダウンロードし、ファイルを保存します。このファイルは、デバイスでのすべてのテストに必要です。

BAR ファイル

データテーブルと、AppLoader という名前の Windows アプリケーション (AppLoader は BREW ツールスイートの一部として QUALCOMM から提供されます) を使用して、BREW 対応デバイス上のファイルを追加、削除、および移動できます。ただし、拡張子が .mod、.mif、または .bar のファイルを、BREW デバイスから他のデバイスまたはストレージメディアにコピーすることはできません。この制限は、BREW Delivery System (BDS) を通して購入したファイルを許可されていないユーザーに配布できないようにするために、QUALCOMM によって適用されています。

Flash コンテンツ開発者がこのセキュリティ機能を利用できるようにするため、BREW デバイス上で動作する Flash Lite プレーヤーは、ローカルデバイスのファイルを参照するときに BAR ファイルを検査します。この制限は、ファイルを参照するためのパスを使用するすべての ActionScript 関数または共有ライブラリ関数に適用されます (たとえば、loadMovie()、loadSound()、loadVariables()、XML.load() など)。この機能を使用するには、作成する任意のファイルに .bar 拡張子を付加します。このようにしても、プレーヤーはファイルを探して表示できませんが、ファイルが最初にロードされた BREW デバイスからファイルをコピーすることはできません。Flash Lite プレーヤーは、指定されたローカルパスを使用してファイルが存在するかどうかを検査し (たとえば loadMovie(example.swf))、その場所にファイルが見つからない場合は、.bar 拡張子を付加したファイル名を同じパスで検査します (たとえば loadMovie(example.swf.bar))。その後、ファイルがロードされます。

異なる BREW バージョンのデバイスファイルの構造

BREW 2.x をサポートするデバイスと BREW 3.x をサポートするデバイスでは、ファイルの構造が異なります。BREW Publisher ウィザードを使用して BREW アプリケーションのファイル構造を指定する前に、これらの違いを理解しておくことが重要です。

BREW 2.x デバイスでのファイルシステムは、次のような構造になっています。

- すべてのアプリケーションフォルダは、"brew/" という名前のルートフォルダに格納されます。
- アプリケーションファイル (SIG ファイルを含む) は、アプリケーションフォルダに格納されます (サブフォルダは使用できません)。
- MIF ファイルは、ルートフォルダ "brew/" に格納されます。

- イメージや BAR ファイルなどの共有メディアは、"brew/shared" フォルダに格納されます。共有メディアは、メインアプリケーションファイルが存在するフォルダに格納することもできます。

例：

brew/appname/appname.mod

brew/appname/appname.sig

brew/appname.mif

brew/shared/media.xxx xxx は、グラフィック、ビデオ、サウンド、BAR ファイルなどのリソースファイルの拡張子です。

または

brew/appname/media.xxx xxx は、グラフィック、ビデオ、サウンド、BAR ファイルなどのリソースファイルの拡張子です。

BREW 3.x デバイスでのファイルシステムは、次のような構造になっています。

- すべてのアプリケーションディレクトリは、"brew/mod" フォルダに格納されます。
- アプリケーションファイル (SIG ファイルを含む) は、アプリケーションフォルダに格納されます (サブフォルダは使用できません)。
- MIF ファイルは、"brew/mif" フォルダに格納されます。
- イメージや BAR ファイルなどの共有メディアは、"brew/shared" フォルダに格納できます。共有メディアは、メインアプリケーションファイルが存在するフォルダに格納することもできます。

例：

brew/mod/appname/appname.mod

brew/mod/appname/appname.sig

brew/mif/appname.mif

brew/shared/media.xxx。xxx は、グラフィック、ビデオ、サウンド、BAR ファイルなどのリソースファイルの拡張子です。

または

brew/mod/appname/media.xxx。xxx は、グラフィック、ビデオ、サウンド、BAR ファイルなどのリソースファイルの拡張子です。

BREW 用 Flash Lite ファイルのパブリッシュとテストのワークフロー

BREW 用の Flash Lite ファイルをパブリッシュおよびテストするワークフローは、次のとおりです。

- Flash で BREW Publisher ウィザードを起動し、画面に従って、BREW アプリケーションに対する適切な設定を指定し、ファイルをパブリッシュします。
- BREW SDK で、BREW Simulator を使用してアプリケーションをテストします。

BREW Publisher ウィザードを使用したアプリケーションファイルのパブリッシュ

以下の手順では、アプリケーションファイルを既に Flash で作成および保存してあるものとします。FLA ファイルをまだ保存していない場合は、BREW Publisher ウィザードからファイルの保存を促すメッセージが表示されます。

BREW Publisher ウィザードを使用してアプリケーションファイルをパブリッシュするには：

1. Flash で、[ファイル]-[パブリッシュ設定] を選択します。
[パブリッシュ設定] ダイアログボックスが表示されます。
2. [形式] タブで、[Flash (.swf)] がまだ選択されていない場合は選択します。
BREW アプレットを生成するために必要なファイルは SWF だけなので、他のすべてのファイル形式は選択を解除してもかまいません。
3. [Flash] タブで、[バージョン] リストからサポートされるバージョンのプレーヤーを選択し (Flash Player 7 以前、または任意のバージョンの Flash Lite)、[ActionScript のバージョン] リストから任意のバージョンを選択します (ActionScript はすべてのバージョンがサポートされます)。
4. [Flash] タブの下部近くにある [ポストプロセッサ] リストから、[Flash Lite for BREW] を選択します。

×
h

[Flash] タブに [ポストプロセッサ] リストが表示されない場合は、[84 ページの「BREW 用のシステムのセットアップ」](#)の説明を参考にして、Flash Lite 2.1 for BREW が正しくインストールされていることを確認します。[ポストプロセッサ] リストは、ソフトウェアがインストールされた後でのみ表示されます。

5. [ポストプロセッサ] リストの右側にある [設定] をクリックして、BREW Publisher ウィザードを表示します。

6. (必須) [Identify Applet] 画面の入力フィールドを使用して、BREW アプレットの一意のクラス ID と名前を指定します。

次の表は、これらのフィールドに関する詳細な情報の一覧です。アプレットのクラス ID と名前の指定が終わったら、[次へ] をクリックして続けます。

フィールド名	フィールドの内容
Class ID	このフィールドには、ランダムに生成された値がデフォルトとしてあらかじめ設定されています。認定されたクラス ID がある場合は、それに置き換えます。クラス ID は一意の 32 ビット識別コード (8 桁の 16 進数値) であり、BREW のインターフェイス作成メカニズムが使用します。ローカルシステムでテストする場合は、ランダムに生成されたデフォルトを使用できますが (一意であるなら)、配布を目的とするアプリケーションに対しては QUALCOMM が一意の ID を発行する必要があります。クラス ID ファイルの入手方法の詳細については、BREW の Web サイトを参照してください。
Applet Name	このフィールドには、現在有効になっている FLA ファイルの名前のルート部分があらかじめ設定されています。たとえば、有効な FLA ファイルの名前が "BREW fla" である場合は、アプレット名は BREW になります。このフィールドに入力した名前は、対象デバイスの BREW Application Manager で表示されます。 アプレット名に対しては、次のガイドラインを使用します。 <ul style="list-style-type: none">• アプレット名はすべて小文字でなければならない、アルファベット文字で開始する必要があります。• 数字を使用できますが、1 文字目には使用できません。• 使用できる特殊文字はアンダースコア文字のみで、1 文字目には使用できません。• 名前の最大文字数は、デバイスによって異なります。 詳細については、BREW の Web サイトでデバイスの仕様を確認してください。

7. (省略可能) [Include Applet Information] 画面を使用して、BREW アプレットと共にパブリッシュする情報を入力します。

ここで入力した情報は、アプレットの MIF ファイルに格納されます。次の表は、各フィールドに関する詳細な情報の一覧です。情報の指定が終わったら、[次へ] をクリックして続けます。

フィールド名	フィールドの内容
Author	アプレットの作成者の名前を入力します (23 文字以下)。
Version	アプレットのバージョンを入力します (23 文字以下)。
Copyright	著作権情報を入力します (23 文字以下)。
Domain URL	アプレットの完全修飾されたドメインを入力します。最後はスラッシュで終了します (例: http://www.example.com/)。アプレットのドメインは、アプレットが実行時にアクセスする情報を制御します。以前のバージョンの Flash プレーヤーでは、似たドメイン (たとえば、 www.exampleapps.com と www.examplegames.com) の SWF ファイルは、相互に、そして別のドキュメントと、自由に通信できました。Flash Player 7 以降では、アクセスされるデータのドメインは、データの供給元のドメインと正確に一致していないと通信できません。ドメインとセキュリティの詳細については、Flash のマニュアルで、セキュリティ、ドメイン間のセキュリティ、ドメインが異なる SWF ファイル間でのデータアクセスの許可に関するトピックを参照してください。
Application Type (アプリケーションまたはスクリーンセーバー)	作成している BREW アプレットがアプリケーションかスクリーンセーバーかを指定します。スクリーンセーバーとアプリケーションの相違点の詳細については、BREW の Web サイトで、[ディベロッパー FAQ]-[BREW ツール]-[MIF の設定]-[スクリーンセーバー] を参照してください。

8. (アプリケーションのテスト用としては省略可能、ただし True BREW Testing に合格する必要があるアプリケーションの場合は必須) [Applet Icons] 画面を使用して、アプレット用のアイコンの名前と場所を入力または参照します。

サポートされるファイル形式は、JPEG、BMP、PNG、および BCI (BREW Compressed Image) です。対象デバイスの BREW Application Manager では、これらのアイコンがアプレット名と共に表示されます。アイコンの指定が終わったら、[次へ] をクリックして続けます。

フィールド名	フィールドの内容
Small icon	小さいアイコンのイメージファイル (デバイスに表示されます)。最大サイズは 16 x 16 です。
Medium icon	中サイズのアイコンのイメージファイル (デバイスに表示されます)。最大サイズは 26 x 26 です。
Large icon	大きいアイコンのイメージファイル (配布サーバーと一部のデバイスに表示されます)。最大サイズは 65 x 42 です。

9. [Define Output Settings] 画面では、BREW Simulator および対象デバイスに対してファイルをパブリッシュする場所を指定します。

これらのフィールドには、SWF ファイルの場所から抽出されたデフォルト値と、アプレット名およびフォルダ名 (Simulator とデバイス) を組み合わせたものが設定され、テストに使用するファイルおよびデバイスにアップロードするファイルを示します。デフォルト値をそのまま使用することも、別の出力場所を指定することもできます。終了したら、[次へ] をクリックして続けます。


フィールド名	フィールドの内容
For Simulator	BREW SDK の Simulator ツールを使用してアプレットをテストするために必要なファイルを格納します。想定されるファイルの場所とフォルダ構造の詳細については、BREW の Web サイトにある BREW Simulator のマニュアルを参照してください。
• MIF folder	
• Applet folder	
For Device	BREW デバイスにアップロードするファイルを格納します。対象デバイスがサポートする BREW のバージョンによるフォルダ構造の違いの詳細については、 103 ページの「BREW デバイスへのファイルのアップロード」 を参照してください。

10. [Summary of Your Selections] 画面を使用して、選択したアプレットの設定を確認します。

いずれかの設定を変更するには、[Back] ボタンを使用して前の画面に戻り、変更を行います。設定が完了したら、画面の下部にあるボックスをオンまたはオフにして、パブリッシュを行うときに再びウィザードを表示するかどうかを指定した後、[Finish] をクリックして設定を保存します。

フィールド名	フィールドの内容
Summary of Your Selections	確認のためにすべての設定の一覧が表示されます。
Do not show this wizard on publish	アプレットファイルをパブリッシュするときにウィザードを再び表示するかどうかを制御します。オフにすると、ウィザードを表示しないでパブリッシュプロセスを開始します。[パブリッシュ設定] ダイアログボックスを表示し、[ポストプロセッサ] リストの右側にある [設定] ボタンをクリックすれば、ここで設定に関係なく常にウィザードを開始できます。

11. [OK] をクリックして設定を保存した後、再び [ファイル]-[パブリッシュ設定] を選択します。[パブリッシュ設定] ダイアログボックスが表示されます。

	この手順は、特定のファイルの [ポストプロセッサ] リストから Flash Lite for BREW を初めて選択するときのみ必要です。それ以降は、[OK] を先にクリックしないでパブリッシュできます。
---	--


12. [パブリッシュ] をクリックし、ウィザードを使用して指定した場所にファイルをパブリッシュします。

BREW Simulator を使用したアプリケーションのテスト

アプリケーションをデバイスにアップロードする前に、BREW SDK の Simulator ツールを使用してテストする必要があります。このツールの詳細については、BREW の Web サイトにある BREW Simulator のマニュアルを参照してください。

BREW Simulator でアプリケーションをテストするには：

1. BREW Publisher ウィザードを使用して作成したアプレット用の Simulator フォルダに移動します。
2. Simulator フォルダを開き、すべての内容を BREW SDK のサンプル用フォルダにコピーします。

	BREW Publisher ウィザードを使用してファイルをパブリッシュしたときに、Simulator の出力ファイルに対する場所として BREW SDK のサンプル用フォルダを指定した場合は、ファイルは既に正しいフォルダに格納されているので、コピーする必要はありません。
---	--

3. [スタート]-[プログラム]-[BREW SDK v< 使用しているバージョン>] を選択し、[Simulator for BREW] を選択してアプリケーションを開きます。
4. Simulator で、[File]-[Load Device] を選択し、対象ハンドセット用のデバイスパックを選択します。
5. Simulator の [Properties] タブで、[Applet Directory] としてアプリケーションのフォルダを選択します。
6. Simulator のツールと機能を使用して、アプリケーションをテストします。

デバイスパック

BREW の認定ディベロッパーは、対象のデバイスに固有のデバイスパックを BREW の Web サイトの [ディベロッパーリソース] エリアからダウンロードできます。Simulator 用のデバイスパックは、実際のデバイスと厳密に一致しない場合があります。たとえば、Samsung SCH-A950 のデバイスパックには BACK キーがありますが、実際のデバイスでのラベルは CLR です。また、LG VX9800 用のデバイスパックはキーボードのアルファベットキーからのキー入力をサポートしていませんが、実際のデバイスはサポートしています。特定のデバイスパック、更新、または問題の詳細については、QUALCOMM に問い合わせてください。

BREW の認定ディベロッパーではない場合は、SDK のダウンロードに含まれるデフォルトのデバイスパックを使用できます。ただし、デフォルトのデバイスパックはデバイス固有のデバイスパックよりメモリが少ないので、パフォーマンスの問題やエラーを防ぐためには、最もメモリ量の多いデバイスパック (現時点では Device Pack 3) を選択するか、またはメモリを約 10,000,000 バイトに増やすことが必要になる場合があります。

1回の Simulator セッションに対してメモリを増やすには：

1. Simulator を開き、[Device Details] タブを選択します。
2. [Memory] セクションまでスクロールします。
3. [Heap Size (bytes)] の値を増やします。この値は、Simulator のバージョンによっては [RAM Size] と表示されます。

複数の Simulator セッションにわたって固定的にメモリを増やすには：

1. テキストエディタを使用して、使用するデバイスパック用の DSD ファイルを開きます (devicepack<x>, x はデバイスパックの番号です)。
2. 次のようにして適切な設定を探し、Text の値を増やします。

```
<String Id="24" Name="IDS_DD_HW_RAM_SIZE">
<Text>1048576</Text>
```

3. ファイルを保存して閉じます。

"Examples" フォルダおよびそのすべてのサブフォルダに割り当てられている領域が十分ではないためにアプリケーションが正しく機能できない場合にも、Simulator を使用する際に問題が発生する場合があります。これらのフォルダから不必要なファイルを手作業で削除したり、これらのファイルに割り当てる領域を増やしたりすることができます。

複数の Simulator セッションにわたって固定的に "Examples" ファイルのサイズを増やすには：

1. テキストエディタを使用して構成ファイルを開き、適切な値を変更します。
 - Samsung SCH-A950 の場合、構成ファイルは "SCH-A950.qsc" です。変更する値は次の例においてボールドで示されている箇所です。

```
FS_LIMITS_PER_MODULE 65535 15204352
```

- LG9800 の場合、構成ファイルは "LG-VX9800.dsd" です。変更する値は次の例においてボールドで示されている箇所です。

```
<String Id="20" Name="IDS_DD_HW_EFS_SIZE">
<Text>47000000</Text>
<Comment>50MB</Comment>
</String>
```

2. ファイルを保存して閉じます。

その他の参照情報

BREW Simulator の詳細については、以下を参照してください。

ドキュメント名と種類	場所
『BREW 開発スタートガイド』の 「About the BREW Tools Suite」 (PDF)	BREW の Web サイトにアクセスし、[ディベロッパー]- [BREW ドキュメンテーション]-[アプリケーション開発 ドキュメンテーション] を選択します。
BREW Simulator (ヘルプ)	Simulator を開き、メニューバーの [Help] をクリックして、 ヘルプを表示します。

BREW デバイスへのファイルのアップロード

ここでは、Flash Lite アプリケーションを BREW デバイスにアップロードし、ハンドセットでアプリケーションをテストする方法について説明します。

この項では、次のトピックについて説明します。

BREW 用 Flash Lite アプリケーションのアップロードについて	103
アプリケーションのアップロードのワークフロー	104
デバイスへの Flash Lite 拡張機能ファイルのアップロード (初回のみ)	105
AppLoader のヒントとベストプラクティス	105
BREW 2.x デバイスへのアプリケーションのアップロード	106
BREW 3.x デバイスへのアプリケーションのアップロード	107
デバイスでのアプリケーションのテスト	108
その他の参照情報	109

BREW 用 Flash Lite アプリケーションのアップロードについて

Flash Lite アプリケーションを作成し、Flash Lite for BREW Publisher ウィザードを使用して BREW 対応のファイルを生成したら、サポートされる BREW デバイスにアプリケーションをアップロードしてテストします。

既に、初期設計プロセスにおいて対象のデバイスを決定し、Adobe Device Central エミュレータと BREW Simulator を使用してアプリケーションのテストが済んでいるのが理想的です。ただし、その場合でも、対象デバイスでアプリケーションを十分にテストし、意図したとおりに機能することを確認することが重要です。

前提条件

アプリケーションをアップロードするには、以下のものがが必要です。

- Flash Lite for BREW をサポートするデバイス (現時点では、Samsung SCH-A950 または LG VX9800)
 - コンピュータからデバイスにアプリケーションをアップロードするためのシリアルケーブルまたは USB ケーブル (通常は、デバイスの製造元から入手できます)
 - AppLoader ツール (認定ディベロッパーは、BREW の Web サイトから BREW ツールスイートの一部としてダウンロードできます)
 - 必要なファイル形式と構造のアプリケーションファイル (Flash Lite for BREW Publisher ウィザードを使用して作成したもの)
 - appname.SIG (認定ディベロッパーは、QUALCOMM の Web ベースのテスト署名ジェネレータからダウンロードできます)
 - appname.MOD (Flash Lite Publisher for BREW ウィザードで生成されます)
 - appname.MIF (Flash Lite Publisher for BREW ウィザードで生成されます)
 - アプリケーションに必要なすべてのサウンド、ビデオ、イメージ、および BAR ファイル
- 対象デバイスに必要なファイル形式と構造の詳細については、[95 ページの「異なる BREW バージョンのデバイスファイルの構造」](#)を参照してください。

アプリケーションのアップロードのワークフロー

このワークフローでは、デバイス用の適切な USB ドライバがコンピュータに既にインストールされているものとします。詳細については、[84 ページの「BREW 用のシステムのセットアップ」](#)を参照してください。

Flash Lite アプリケーションを BREW デバイスにアップロードしてテストするには、次の作業を行います。

- ハンドセットをケーブルでコンピュータに接続します。
- コンピュータで AppLoader を開きます。
 - アプリケーションファイルをデバイスに対する適切な場所にコピーします。詳細については、[95 ページの「異なる BREW バージョンのデバイスファイルの構造」](#)を参照してください。
 - 必要に応じて、Flash Lite 拡張機能とプレーヤーのファイルをデバイスにコピーします (初回のみ)。
 - デバイスを再起動します。
- デバイスでアプリケーションをテストします。

以降では、各作業について詳細に説明します。

デバイスへの Flash Lite 拡張機能ファイルのアップロード (初回のみ)

特定のデバイスで Flash Lite for BREW アプリケーションをテストするには、先に Flash Lite 拡張機能をデバイスにコピーする必要があります。この作業を行う必要があるのは、テスト用にデバイスを初めて取得したときの1回だけです。アプリケーションのエンドユーザーは、BDS または ADS から完全なアプリケーションをダウンロードすると、その一部としてこれらのファイルを受け取りますが、ディベロッパーは手作業でファイルをコピーする必要があります。ファイルの一覧を次に示します。後で説明する手順に従って、AppLoader ツールを使用してファイルをデバイスにアップロードします。これらのファイルをコピーする場所は、BREW 2.x のデバイスと 3.x のデバイスでは異なることに注意してください。詳細については、[95 ページの「異なる BREW バージョンのデバイスファイルの構造」](#)を参照してください。

拡張機能ファイル	コピー元
flashlite_2_1.sig	BREW の Web サイトのツールを使用してディベロッパーが生成したもの。
flashlite_2_1.mif	BREW の Web サイトからダウンロード可能。
flashlite_2_1.mod	BREW の Web サイトからダウンロード可能。
brewsaplayer.sig	BREW の Web サイトのツールを使用してディベロッパーが生成したもの。
brewsaplayer.mif	BREW の Web サイトからダウンロード可能。
brewsaplayer.mod	BREW の Web サイトからダウンロード可能。

AppLoader のヒントとベストプラクティス

以下に示すのは、AppLoader ツールを使用するためのヒントです。

- デバイスにフォルダを作成するときは、バックスラッシュ (\) を使用しないでください。
- ファイルをコピーしている間は、デバイスでアプリケーションを実行しないでください。
- AppLoader を使用して変更を行った後は、デバイスを再起動してください。
- AppLoader ではファイルを上書きしないでください。代わりに、デバイス上の古いバージョンのファイルを削除し、ファイルの新しいコピーに置き換えてください。
- 名前が数字のみのファイルまたはフォルダはアップロードしないでください。一部のデバイスでは、このようなファイルやフォルダは削除できません。
- "shared" で始まるファイル名またはフォルダ名は使用しないでください。"shared" は BREW の予約語であり、名前がこの文字列で始まるファイルまたはフォルダはアップロードできません。

BREW 2.x デバイスへのアプリケーションのアップロード

以下では、BREW バージョン 2.x プラットフォームが組み込まれた Samsung SCH-A950 に BREW 用 Flash Lite アプリケーションをアップロードする方法について説明します。

この手順では、デバイス用の適切な USB ドライバがコンピュータに既にインストールされているものとします。詳細については、[84 ページの「BREW 用のシステムのセットアップ」](#)を参照してください。

また、この手順では、必要な Flash Lite 拡張機能ファイルと Flash Lite プレーヤーファイルもデバイスに既にアップロードされているものとします。これを行う必要があるのは、デバイスを初めて使用する時の1回だけです。詳細については、[105 ページの「デバイスへの Flash Lite 拡張機能ファイルのアップロード \(初回のみ\)」](#)を参照してください。

BREW 2.x デバイスにアプリケーションをアップロードするには：

1. ハンドセットの製造元から提供されるデータケーブルを使用して、BREW アプリケーションのファイルが格納されているコンピュータの COM ポートに、ハンドセットを接続します。
2. アプリケーションファイルが格納されているフォルダに移動し、ディレクトリの構造が対象デバイスに適していることを確認します。

BREW 2.x デバイスに対する構造の詳細については、[95 ページの「異なる BREW バージョンのデバイスファイルの構造」](#)を参照してください。

3. [スタート]-[プログラム]-[BREW Tools Suite<最新バージョン>]-[BREW Apploader] を選択して、AppLoader ツールを起動します。AppLoader ツールの接続ウィンドウが表示されます。
4. デバイスが接続されている COM ポートの番号を選択し (選択するポートがわからない場合は、後のメモを参照してください)、デバイスの BREW のバージョンとして 2.x を選択し、[OK] をクリックしてデバイスに接続します。

×
h

デバイスが接続されている COM ポートは、Windows のデバイスマネージャで、インストールされているデバイスに対応するポートのプロパティを見ることでわかります。常に同じデバイスモデルを同じ COM ポートに接続している場合は、この番号は変わりません。違うデバイスモデルを使用すると変わるので、新しいデバイスモデルを接続したときは、必ず COM ポート番号を確認してください。

接続すると、AppLoader ツールにデバイスの BREW ファイルシステムが表示されます。

5. アプリケーションのファイル (MOD、SIG、およびリソースファイル) をハンドセットにアップロードするには、次のいずれかを行います。
 - [File]-[Directory]-[New Directory] を選択し、アプリケーションと同じ名前で作成した後、そこにアプリケーションのファイルをコピーします。詳細については、[95 ページの「異なる BREW バージョンのデバイスファイルの構造」](#)を参照してください。
 - Windows エクスプローラから AppLoader のウィンドウに、アプリケーションフォルダをドラッグします。

6. デバイスを再起動します。

アプリケーションを実行する前に、デバイスを再起動する必要があります。デバイスを再起動するには、次のいずれかを行います。

- BREW AppLoader のメニューから、[Device]-[Reset] を選択します。
- ハンドセットの End キーを押し続けます。

7. [Get it Now]-[Get Going] に移動し、アプリケーションの名前を選択して、BREW 環境でアプリケーションを開始します。

BREW 3.x デバイスへのアプリケーションのアップロード

以下では、BREW バージョン 3.x プラットフォームが組み込まれた LG VX9800 に BREW 用 Flash Lite アプリケーションをアップロードする方法について説明します。

この手順では、デバイス用の適切な USB ドライバがコンピュータに既にインストールされているものとします。詳細については、[84 ページの「BREW 用のシステムのセットアップ」](#)を参照してください。

また、この手順では、必要な Flash Lite 拡張機能ファイルと Flash Lite プレーヤーファイルもデバイスに既にアップロードされているものとします。これを行う必要があるのは、デバイスを初めて使用するときの1回だけです。詳細については、[105 ページの「デバイスへの Flash Lite 拡張機能ファイルのアップロード \(初回のみ\)」](#)を参照してください。

BREW 3.x デバイスにアプリケーションをアップロードするには：

1. ハンドセットの製造元から提供されるケーブルを使用して、BREW アプリケーションのファイルが格納されているコンピュータの COM ポートに、ハンドセットを接続します。
2. アプリケーションファイルが格納されているフォルダに移動し、ディレクトリの構造が対象デバイスに適していることを確認します。

BREW 3.x デバイスに対する構造の詳細については、[95 ページの「異なる BREW バージョンのデバイスファイルの構造」](#)を参照してください。

3. [スタート]-[プログラム]-[BREW Tools Suite<最新バージョン>]-[BREW Apploader] を選択して、AppLoader ツールを起動します。
4. AppLoader ツールの接続ウィンドウが表示されます。

デバイスが接続されている COM ポートの番号を選択し、デバイスの BREW のバージョンとして 3.x を選択し、[OK] をクリックしてデバイスに接続します。

×
h

デバイスが接続されている COM ポートは、Windows のデバイスマネージャで、インストールされているデバイスに対応するポートのプロパティを見ることでわかります。常に同じデバイスモデルを同じ COM ポートに接続している場合は、この番号は変わりません。違うデバイスモデルを使用すると変わるので、新しいデバイスモデルを接続したときは、必ず COM ポート番号を確認してください。

接続すると、AppLoader ツールにデバイスの BREW ファイルシステムが表示されます。

5. アプリケーションのファイル (MOD、SIG、およびリソースファイル) をハンドセットにアップロードするには、次のいずれかを行います。
 - [File]-[New]-[Directory]-[New Directory] を選択し、アプリケーションと同じ名前でディレクトリを作成した後、そこにアプリケーションのファイルをコピーします。詳細については、[95 ページの「異なる BREW バージョンのデバイスファイルの構造」](#)を参照してください。
 - Windows エクスプローラから AppLoader のウィンドウに、アプリケーションフォルダをドラッグします。
6. デバイスを再起動するには、次のいずれかを行います。
 - BREW AppLoader のメニューから、[Device]-[Reset] を選択します。
 - ハンドセットの End キーを押し続けます。
7. [Get it Now]-[Get Going] に移動し、アプリケーションの名前を選択して、BREW 環境でアプリケーションを開始します。

デバイスによっては、標準の [Get Going] メニューにアプリケーションが表示されません。そのようなデバイスでアプリケーションを表示するには、次の手順を実行します。

 - a. Verizon のホーム画面で、[Select] または [OK] を押してメニューにアクセスします。
 - b. ゼロ (0) キーを押して、サービスメニューにアクセスします。
 - c. デフォルトのパスワード (6 個のゼロ (000000)) を入力します。
 - d. 9 キーを押して、[Get it Now] を表示します。
 - e. g キーを押して、Flash Lite アプリケーションが表示される画面にアクセスします。

デバイスでのアプリケーションのテスト

デバイスでアプリケーションを開始してテストを行うには、先に QUALCOMM からテスト用の署名を入手する必要があります。認定されたディベロッパーは、BREW のディベロッパーエクストラネット で提供されている Testsig という名前の Web ベースツールを使用して、テスト用の署名を生成できます。Testsig ツールを使用して署名を生成する方法の詳細については、BREW の Web サイトを参照してください。BREW ディベロッパーエクストラネットでは、BREW アプレットをテストして最終的に TBT (True BREW Testing) 基準に準拠していることを確認する方法についての包括的な情報が提供されています。

その他の参照情報

AppLoader ツールの詳細については、以下を参照してください。

ドキュメント名と種類	場所
『BREW 開発スタートガイド』の 「About the BREW Tools Suite」 (PDF)	BREW の Web サイトにアクセスし、[ディベロッパー]- [BREW ドキュメンテーション]-[アプリケーション開発ド キュメンテーション]を選択します。
BREW)_Apploader.chm (ヘルプ)	C:¥Program Files¥BREW Tools Suite < 最新バージョン >¥ BREWApploader (デフォルト)
BREW Testing Generator	BREW の Web サイトにアクセスして、SDK ダウンロード ページに移動します。

コンテンツのパフォーマンスとメモリ使用の最適化

Flash コンテンツを最適化するには、基本原則に留意する必要があります。たとえば、Flash 開発者にとって、極度に複雑なアートワーク、必要以上のトゥイーン処理、および透明の使い過ぎは避けるべきだとされていました。

以前のバージョンの Flash でも、デスクトップバージョンでは、このようなパフォーマンスの問題はほとんど解決されていますが、Adobe Flash Lite の開発者はまだモバイルデバイスの制限による課題に直面しています。たとえば、デバイスによってパフォーマンスにばらつきがあり、それが極端な場合もあります。また、モバイルオーサリングではさまざまなデバイスへのパブリッシュが必要になるので、開発者は最小公分母のオーサリングを実行しなければならないこともあります。

モバイルコンテンツを最適化するには、両立しないものどうしの妥協点を見つけなければならない場合があります。たとえば、あるテクニックを使用すると表示がきれいになり、別のテクニックを使用するとパフォーマンスが向上するとします。そのいずれを優先するかを決めるために、エミュレータでのプレビューとターゲットデバイスでのプレビューを何度も繰り返し行うことになります。

Flash 8 では、Flash Lite 2.x 用のエミュレータは Flash オーサリング環境の一部でした。Flash CS3 では、エミュレータ機能は Adobe® Device Central CS3 の一部となっています。Device Central を使用すると、Flash Lite プロジェクトをさまざまなデバイスでエミュレートできます。また、デバイスの表示、メモリ使用、およびパフォーマンスを特定のデバイスでエミュレートできます。エミュレータを使用してモバイルデバイス用の Flash Lite コンテンツを最適化する方法については、Device Central のマニュアルの「Best practices for content on mobile devices」を参照してください。

Flash Lite コンテンツのプレビュー

Adobe Flash CS3 Professional には Adobe Device Central CS3 で利用できる Adobe Flash Lite エミュレータが搭載されていますが、このエミュレータを使用すると、アプリケーションを実際のデバイスで表示して使用した場合の動作を、オーサリングツールでプレビューできます。エミュレータでのアプリケーションの動作に満足したら、アプリケーションを実際のデバイスで試すことができます。この章では、Flash CS3 Professional で使用できる Flash Lite のプレビューおよびデバッグ機能について説明します。

この章は、次のトピックを取り上げています。

Flash Lite のプレビュー機能の概要	113
エミュレータでサポートされていないプレビュー機能	114
エミュレータの使用	114
エミュレータでのデバイスビデオの再生について	123

Flash Lite のプレビュー機能の概要

Flash CS3 Professional に搭載されている Flash Lite のプレビュー機能は Adobe Device Central の一部であり、デバイスプロファイルの広範なデータベースとデバイスエミュレータの両方で構成されます。Device Central は、他にも多くの Adobe 製品 (Adobe Creative Suite®、Adobe Dreamweaver® など) に対応しています。

Adobe Device Central エミュレータでは、Flash オーサリングツール内で Flash Lite コンテンツをプレビューできます。エミュレータは、ターゲットデバイスのビヘイビアと外観に合わせて調整するように設定されています。Device Central のインターフェイスを使用して、ターゲットデバイスを選択および管理します。また、着信音、ブラウザ、またはスタンドアローンアプリケーションなど、アプリケーションのターゲットとなる Flash Lite コンテンツタイプを指定します。Device Central では、テストデバイスおよび Flash Lite コンテンツタイプの組み合わせで定義されるデバイス設定により、アプリケーションで使用可能な機能が指定されます。たとえば、サポートされるオーディオフォーマットやネットワーク接続を確立する機能などです。Flash Lite コンテンツタイプの詳細については、[117 ページの「コンテンツタイプについて」](#)を参照してください。

エミュレータの操作方法などの詳細については、Device Central のヘルプを参照してください。

エミュレータでサポートされていないプレビュー機能

Adobe Device Central エミュレータでは、標準 (デスクトップ) のプレビューウィンドウで利用できる機能のうち一部の機能がサポートされていません。デスクトップの Flash プレビューウィンドウでは利用でき、Adobe Device Central エミュレータでは利用できない機能の一覧を次に示します。

- [変数のリストアップ]([デバッグ]-[変数のリストアップ]) および [オブジェクトのリストアップ]([デバッグ]-[オブジェクトのリストアップ]) 機能
- Bandwidth プロファイラ、ストリーミング、およびフレーム毎データ量の各機能
- [表示]-[ダウンロードのシミュレート]メニューコマンド
- ActionScript デバッグ
- [表示]-[再描画する領域を表示]メニューコマンド
- [制御パネル] ツールバー ([ウィンドウ]-[ツールバー]-[制御パネル])

Flash Lite 2.1でのインラインテキストのプレビュー

現在、Adobe Device Central エミュレータでインラインテキスト機能をプレビューすることはできません。この機能はデバイスでプレビューする必要があります。エミュレータでプレビューする場合、Flash Lite コンテンツ上に表示されるモーダルダイアログボックスを使用して入力テキストフィールドの内容を編集する必要があります (つまり、エミュレータは Flash Lite 1.x および Flash Lite 2.x の場合と同様に機能します)。モーダルダイアログボックスの動作の詳細については、[39 ページの「テキスト入力フィールドの使用」](#)を参照してください。

エミュレータの使用

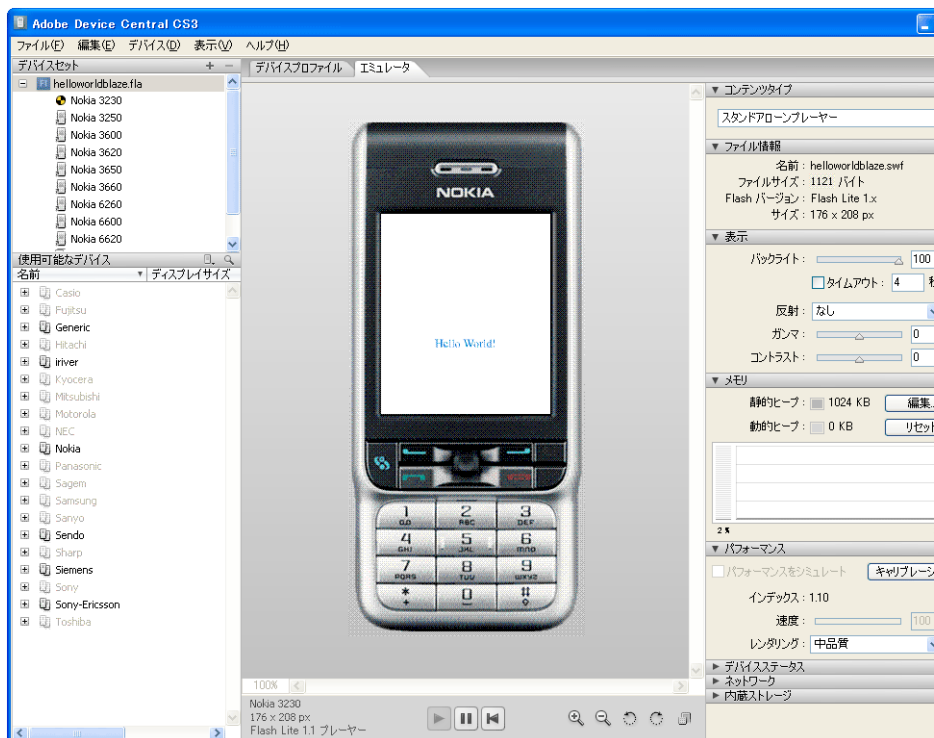
エミュレータを起動するには、Flash デスクトップコンテンツをプレビューするときと同じように、Flash オーサリングツールから [制御]-[ムービープレビュー] を選択します。ただし、Adobe Device Central と Flash デスクトップコンテンツのプレビューウィンドウでは、外観と機能が異なります。

この項では、次のトピックについて説明します。

- [115 ページの「Adobe Device Central の呼び出し」](#)
- [116 ページの「エミュレータのデバッグオプションの設定」](#)
- [120 ページの「エミュレータの Flash Lite 固有の情報」](#)
- [114 ページの「エミュレータでサポートされていないプレビュー機能」](#)
- [122 ページの「エミュレータでサポートされていない Flash Lite 機能」](#)
- [123 ページの「エラー」](#)

Adobe Device Central の呼び出し

[制御]-[ムービープレビュー]を選択するかCtrl+Enterを押してDevice Centralを開始すると、FlashからDevice CentralにSWFが書き出されていることを示すプログレスバーが表示されます。書き出しが完了すると、Device Centralが起動し、エミュレータにフォーカスが設定され、SWFが読み込まれます。



Device Central の [Device Sets] リストには、Flash でアプリケーションに保存されたすべてのターゲットデバイスが表示されます。デフォルトでは、セットの先頭のデバイスがエミュレーションの対象として選択されています。選択されているコンテンツタイプは、アプリケーションの作成時に保存したコンテンツタイプです (Flash Lite のコンテンツタイプの詳細については、[117 ページの「コンテンツタイプについて」](#)を参照してください)。

アプリケーションは、さまざまなコンテンツタイプとデバイスでテストできます。コンテンツタイプを変更したり、デバイスセットからデバイスを追加したり削除すると、Flash のデバイス設定も自動的に変更されます。

別のデバイスでコンテンツがどのように表示されるかをテストするには、上部または下部の一覧で別のデバイスをダブルクリックします。テスト中のデバイスの隣に回転アイコンが表示され、選択したデバイスでのデバイス実行シミュレーションに変わります。

エミュレータのデバッグオプションの設定

Adobe Device Central エミュレータでは、コンテンツの実行中にデバッグメッセージを Flash の [出力] パネルに出力できます。エミュレータでも、ポップアップ形式の [出力] パネルに同じメッセージが表示されます。

エミュレータによって次の種類の情報が出力されます。

トレースメッセージ。Flash Lite アプリケーション内で `trace()` 関数呼び出しによって生成されます。`trace()` の使用の詳細については、『Flash Lite 2.x ActionScript リファレンスガイド』の「`trace()`」を参照してください。

情報メッセージ。選択したテストデバイスや SWF ファイルサイズなどに関する一般的な情報です。これらのメッセージは、エミュレータの [警告] パネルに表示されます。

警告メッセージ。再生に影響する可能性がある、Flash Lite コンテンツの問題に関する情報です。エミュレータで生成する情報の種類は、次の方法で絞り込むことができます。

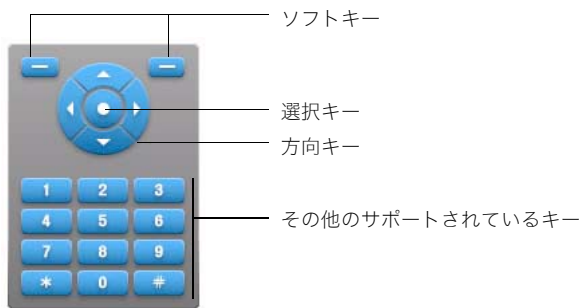
Flash Lite の出力オプションを設定するには：

1. [制御]-[ムービープレビュー] を選択します。アプリケーションが Adobe Device Central に書き出され、エミュレータに表示されます。
2. [表示]-[Flash 出力]-[Show in Device Central] を選択します。
 - [トレース] オプションを選択または選択解除します。
 - [情報] オプションを選択または選択解除します。
 - [警告] オプションを選択または選択解除します。

エミュレータの操作

コンピュータのマウスまたはキーボードショートカットを使用して、エミュレータのキーを操作できます。エミュレータでは次のキーを操作できます。

- 数値キー (0 ～ 9)、アスタリスク (*) キー、およびシャープ (#) キー
- 方向キー (上、下、左、右、および選択)
- 左右のソフトキー



マウスを使用してエミュレータのキーを直接クリックするか、次の対応するキーボードショートカットを使用することができます。

- キーボードの矢印キー(左、右、上、下) は、エミュレータの方向キーにそれぞれ対応します。
- Enter キーまたは Return キーは、エミュレータの選択キーに対応します。
- PageUp キーと PageDown キーは、それぞれエミュレータの左右のソフトキーに対応します。
- キーボードの数値キーは、エミュレータの数値キーと対応します。

アプリケーションをテストする際のエミュレータの操作方法の詳細については、Adobe Device Central のオンラインヘルプを参照してください。

コンテンツタイプについて

Flash Lite プレーヤーはさまざまなデバイスにインストールされます。インストールされた Flash Lite は1つまたは複数のアプリケーションモード、つまりコンテンツタイプをサポートします。たとえば、一部のデバイスでは Flash Lite を使ってスクリーンセーバーや着信アニメーションを有効にします。また、モバイル Web ページに埋め込まれた Flash コンテンツをレンダリングするデバイスもあります。

次の表は、このマニュアルの作成時点で利用可能なすべての Flash Lite コンテンツタイプとその説明の一覧です。利用可能なその他の Flash Lite コンテンツタイプに関する最新情報については、www.adobe.com/go/mobile_supported_devices_jp/ にある「Flash サポート対象の携帯デバイス」のページを参照してください。



本マニュアルの作成時点では、Flash Lite 2.0 がサポートしているのはスタンドアローンプレーヤーのコンテンツタイプのみです。

Flash Lite は次のコンテンツタイプをサポートしています。

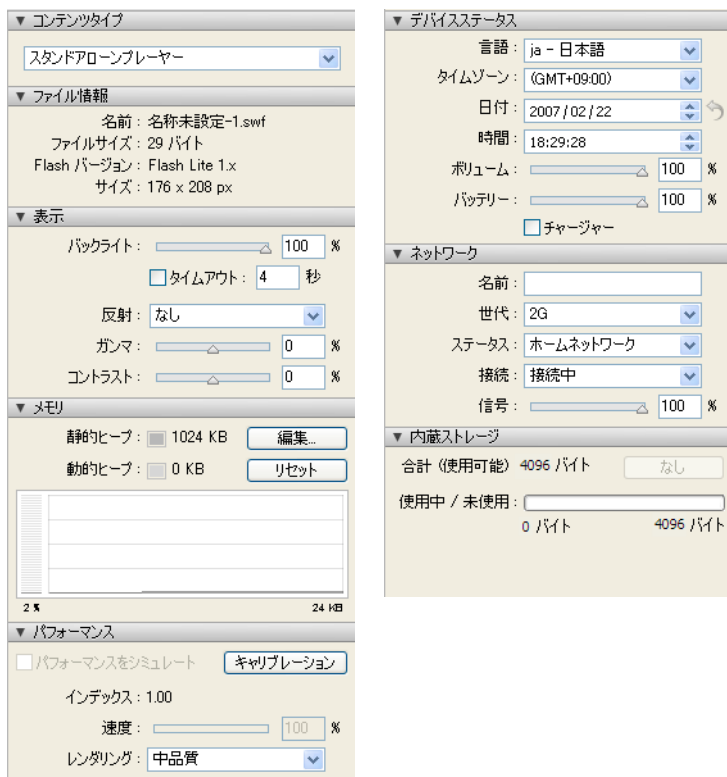
Flash Lite コンテンツタイプ	説明	利用状況
アドレス帳	デバイスのアドレス帳アプリケーションの項目と SWF ファイルを関連付けることができます。	DoCoMo およびボーダフォン (日本のみ)
アラーム	デバイスのアラームとして再生する SWF ファイルを選択できます。	KDDI およびソフトバンク (日本のみ)
ブラウザ	モバイル Web ページに埋め込んでデバイスの Web ブラウザで表示する Flash コンテンツをレンダリングします。	DoCoMo、KDDI、およびソフトバンク (日本のみ)

Flash Lite コンテンツタイプ	説明	利用状況
呼び出し履歴	ユーザーのアドレス帳の各エントリに関連付けられているイメージやアニメーションを、名前や電話番号と共に表示します。	KDDI (カシオ製の機種のみ)
呼び出し画面	発着信時にアニメーションを表示します。	DoCoMo および KDDI (日本のみ)
着 Flash	着信音として再生する SWF ファイルを選択できます。	KDDI (日本のみ)
データボックス	デバイスのデータボックスアプリケーションで Flash コンテンツをレンダリングします。これにより、ユーザーはデバイス上のマルチメディアファイルを管理およびプレビューできます。	DoCoMo、KDDI、およびソフトバンク (日本のみ)
データフォルダ	デバイスのデータフォルダアプリケーションで Flash コンテンツをレンダリングします。これにより、ユーザーはデバイス上のマルチメディアファイルを管理およびプレビューできます。	KDDI (日本のみ)
アイコンメニュー	ユーザーがデバイスのランチャーアプリケーション (UILauncher コンテンツタイプで使用) のカスタムアイコンメニューを選択できるようにします。	KDDI (カシオ製の機種のみ)
イメージビューア	イメージビューアアプリケーションを使用して、ユーザーは SWF ファイルを含む、デバイス上のマルチメディアファイルを管理およびプレビューできます。	DoCoMo (日本のみ)
着信コール	着信時にアニメーションを表示します。	DoCoMo、KDDI、およびソフトバンク (日本のみ)
メーラー	電子メールメッセージの送受信時にアニメーションを表示します。	ボーダフォン (日本のみ)
マルチメディア	SWF ファイル (およびその他のマルチメディア形式) をプレビューします。	KDDI (日本のみ)

Flash Lite コンテンツタイプ	説明	利用状況
マイピクチャ	マイピクチャアプリケーションを使用して、ユーザーはデバイス上の SWF ファイルおよびその他のイメージ形式を管理およびプレビューできます。	DoCoMo (日本のみ)
OpenEMIRO	デバイスがスタンバイモードから復帰したときに Flash Lite コンテンツを表示します。これは他のデバイスのウェイクアップ画面のコンテンツタイプに似ています。	KDDI (カシオ製の機種のみ)
スクリーンセーバー	デバイスのスクリーンセーバーを表示します。	KDDI およびソフトバンク (日本のみ)
SMIL プレーヤー	SWF ファイル (およびその他のマルチメディア形式) をプレビューします。	KDDI (日本のみ)
スタンドアローンプレーヤー	Flash Lite をスタンドアローンアプリケーションとして使用して、ユーザーがデバイスに保存されている SWF ファイルや受信ボックスで受信した SWF ファイルを起動して表示できるようにします。	Symbian Series 60 および UIQ の限定デバイスでグローバルに使用可能
待受画面	デバイスの待受画面 (または壁紙画面) を表示します。	DoCoMo および KDDI (日本のみ)
サブ LCD	一部の折りたたみ型携帯端末の外側にあるサブ画面にコンテンツを表示します。	KDDI (日本のみ)
UILauncher	デバイスのランチャーアプリケーションに使用します。	デバイスのランチャーアプリケーション (ユーザーが他のアプリケーションを起動するのを可能にするアプリケーション) を表示します。
ウェイクアップ画面	電話機の起動時にアニメーションを表示します。	DoCoMo (日本のみ)

エミュレータの Flash Lite 固有の情報

エミュレータには、Flash Lite アプリケーション固有の情報が表示されるパネルがいくつかあります。各パネルはウィンドウの右側に配置され、Flash の場合と同様に開いたり閉じたりすることができます。



- [コンテンツタイプ] パネルには、アプリケーションのデフォルトのコンテンツタイプが表示されますが、対応する別のコンテンツタイプを選択することもできます。
- [ファイル情報] パネルには、ファイル名、Flash のファイルバージョン、寸法、サイズ (キロバイト) が表示されます。
- [警告] パネルは問題が発生すると開き、警告メッセージが表示されます。
- [メモリ] と [パフォーマンス] パネルには、スタティックメモリとダイナミックメモリのサイズが表示され、さまざまなパラメータを調整してアプリケーションのパフォーマンスを向上させることができます。
- [デバイスのステータス] パネルには、タイムゾーンやバッテリーレベルなど、さまざまなプラットフォーム固有の (`fscommand()`) 設定が表示されます。
- [ネットワーク] パネルには、ネットワーク接続に関する情報が表示されます。

- [Persistent Storage] パネルには、デバイスの記憶域で使用された容量が表示されます。この値はデバイスごとに異なります。つまり、複数のアプリケーションが固定記憶域に書き込む場合、値はすべてのデータの合計となります。デバイスで実行していたすべてのコンテンツの固定オブジェクトを削除するには、[Empty] をクリックしてそのデバイスの記憶域を消去します。

各パネルの詳細と使用方法については、Adobe Device Central のオンラインヘルプを参照してください。

スクリーンサイズと利用可能なステージサイズ

ターゲットデバイスと Flash Lite コンテンツタイプの組み合わせによって、Flash Lite アプリケーションで利用できるスクリーンの領域が決まります。使用可能なステージ領域は、デバイスのフルスクリーンサイズ以下です (同サイズを含む)。

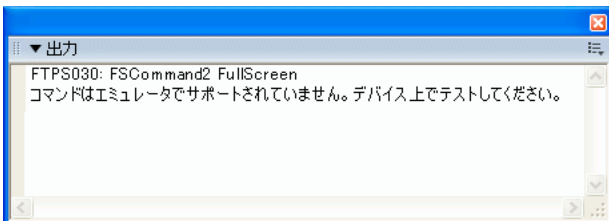
たとえば、Nokia Series 60 デバイスのスタンドアローンプレーヤーで、SWF ファイルをフルスクリーンモードで実行する場合、使用可能なステージ領域はデバイスのフルスクリーンサイズ (176 x 208 ピクセル) と同じです。その他のデバイス (日本で販売されているデバイスなど) では、特殊なコンテンツタイプ (アドレス帳やスクリーンセーバーなど) で実行される SWF ファイルで使用可能なステージ領域は、デバイスのフルスクリーンサイズよりも小さい場合があります。たとえば、Fujitsu 700i のスクリーンサイズは 240 x 320 ピクセルですが、このデバイスのアドレス帳アプリケーションで実行される SWF ファイルで使用可能なステージ領域は 96 x 72 ピクセルです。

SWF ファイルのステージサイズが使用可能なステージサイズと異なる場合は、使用可能なステージ領域に収まるように、Flash Lite プレーヤーによってコンテンツが (縦横比を保って) 伸縮されます。エミュレータでコンテンツをプレビューする場合、アプリケーションのステージサイズが使用可能なステージ領域と異なるときにも警告が表示されます。

意図しない伸縮の問題を回避するために、Flash ドキュメントのステージサイズを、選択したテストデバイスやコンテンツタイプで使用可能なステージ領域と一致させることをお勧めします。

エミュレータでサポートされていない Flash Lite 機能

エミュレータは、実際のデバイスで実行される Flash Lite アプリケーションで使用可能なすべての ActionScript コマンドやプレーヤー機能をサポートしているわけではありません。たとえば、エミュレータは電話をかけたり、SMS メッセージを送信したりする機能をサポートしていません。エミュレータでサポートされていないコマンドや機能を実行しようとすると、次の図のように、[出力] パネルにメッセージが表示されます。



実際のデバイスで SWF ファイルをプレビューし、これらの機能が、意図したとおりに機能するかどうかを確認する必要があります。

サポートされていない ActionScript コマンド

Adobe Device Central エミュレータは、次の `fscommand()` および `fscommand2()` コマンドをサポートしていません。

- `FullScreen`
- `GetFreePlayerMemory`
- `GetTotalPlayerMemory`
- `Launch`
- `Quit`
- `StartVibrate`
- `GetNetworkConnectStatus`
- `GetNetworkRequestStatus`
- `GetNetworkStatus`

エミュレータでのデバイスビデオの再生について

Flash Lite プレーヤーは、SWF ファイル内でビデオコンテンツを再生するのに、ビデオをネイティブにデコードするのではなく、デバイスのデフォルトビデオハンドラアプリケーションを使用します。これにより Flash Lite の開発者は、3GPP、MPEG、AVI など、ターゲットデバイスがサポートしている任意のビデオ形式を使用することができます。Flash Lite でのビデオ使用の詳細については、[66 ページの「デバイスビデオの使用」](#)を参照してください。

Adobe Device Central エミュレータは、Flash オーサリングツールでプレビューする際、QuickTime Player を使用してデバイスビデオをレンダリングします。QuickTime Player の最新バージョン (現時点ではバージョン 7) は、3GPP などのさまざまなデバイスビデオ形式の再生をサポートしています。ただし、デフォルトでは、実際のデバイスでサポートされているビデオ形式の再生が QuickTime ではサポートされていない場合があり、これらのビデオ形式は Adobe Device Central エミュレータでも再生されません。したがって、常に実際のデバイスでコンテンツをテストする必要があります。

デフォルト設定では QuickTime Player でデバイスビデオが再生されない場合は、次のことを試してください。

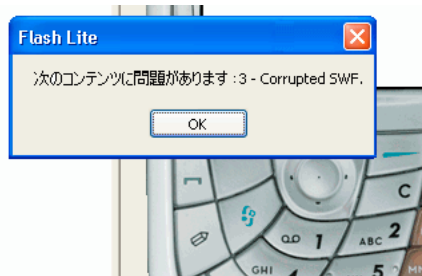
- 最新バージョンの QuickTime Player にアップグレードします。
- 利用可能であれば、使用するビデオ形式をサポートしているサードパーティのビデオコーデック (compressor-decompressor の略) をインストールします。

エラー

Adobe Device Central では、コンテンツのプレビュー中に警告メッセージが生成されます。ある種類の警告メッセージはエミュレータにのみ表示され、実際に発生したエラーまたは発生する可能性があるエラーの情報を提供します。別の種類の警告メッセージは、エミュレータと実際のデバイスの両方に表示されます。

最初の種類の警告メッセージは、SWF ファイルに関するデバッグ情報です。たとえば、SWF ファイルに Flash Lite (または現在選択しているテストデバイスで使用可能な Flash Lite のバージョン) でサポートされていない ActionScript が含まれている場合は、警告メッセージが表示されます。

エミュレータに表示されるもう1つの種類のメッセージは、実際のデバイスにも表示されます。この種類のエラーはエラーダイアログボックスに表示され、アプリケーションを続行するためにユーザーはエラーダイアログボックスを閉じる必要があります。次の図は、エミュレータに表示されるエラーダイアログボックスの例を示しています。



デバイスでは、エラーダイアログボックスには、" 次のコンテンツに問題があります " という文字列に続けてエラー番号が表示されます。エミュレータのエラーダイアログボックスには、簡単なエラーの文字列も表示されます。エミュレータの [出力] パネルには、エラーの詳しい説明が表示されます。次の表は、Flash Lite プレーヤーで発生するすべてのエラーと、エラー番号、エラーダイアログボックスに表示される簡単な説明、および [出力] パネルに表示される詳しい説明の一覧です。

エラー番号	エラーストリング	説明および考えられる原因
1	メモリ不足です。	エミュレータのヒープメモリが不足しています。特に指定しない限り、エミュレータは SWF ファイルで使用するために 1MB のメモリを割り当てます。
2	スタックの限界に達しました。	エミュレータは、スタックの限界に到達したか、限界を超過したことを検出しました。これには、複数レベルのネストされたムービークリップや複雑なベクターの描画など、さまざまな原因が考えられます。
3	SWF が破損しています。	エミュレータは SWF ファイルデータが壊れていることを検出しました。
4	ActionScript が停止しました。	エミュレータは、SWF ファイル内の特定の ActionScript コードの実行に時間がかかりすぎていることを検出しました。その結果、エミュレータは ActionScript コードの実行を停止しました。
5	該当なし	該当なし
6	不正なイメージデータです。	SWF ファイルには、Flash Lite またはプラットフォームのネイティブイメージデコードのいずれかでデコードできなかったイメージが含まれています。

エラー番号	エラーメッセージ	説明および考えられる原因
7	不正なサウンドデータです。	SWF ファイルが、サポートされていない形式のサウンドをロードしようとしたか、サウンドデータが破損しています。
8	ルートムービーがアンロードされました。	このエラーは、ルート (レベル 0) の SWF ファイルが別の SWF ファイルに置き換えられるときに発生します。
9	該当なし	該当なし
10	getURL ストリングが長すぎます。	getURL 呼び出しの URL ストリングが長すぎます。
11	イメージを解凍するための十分なメモリがありません。	Flash Lite に、SWF ファイル内のイメージをデコードするのに十分なメモリがありません。
12	不正な SVG データです。	Flash Lite が、破損した SVG データのロードを試みました。
13	ストリームローディングでメモリ不足が発生しました。	Flash Lite に、URL からストリームするデータを処理するのに十分なメモリがありません。たとえば、Flash Lite が一度に処理するには大きすぎる XML ファイルをネットワーク経由でロードしようとする、このエラーが発生します。可能であれば、データファイルをいくつかの小さなファイルに分割してから、それぞれのファイルを個別にロードすることを試みてください。

警告メッセージとエラーメッセージ

この付録では、Adobe Device Central のエミュレータで Flash Lite アプリケーションをテストするときに生成される情報および警告メッセージの一覧を示します。メッセージは、エミュレータの [出力] ポップアップウィンドウに表示され、Flash でも [出力] パネルで通知されます。

エミュレータと実際のデバイスの両方で出力される可能性があるエラーについては、[123 ページの「エラー」](#)を参照してください。

Adobe Device Central エミュレータのエラーメッセージと警告メッセージ

次の表は、Adobe Device Central エミュレータで表示されるすべての情報メッセージの一覧です。

エラーコード	メッセージ	説明
FTPA002	FSCommand は無視されます。	エミュレータは fscommand() 関数呼び出しを検知しました。この関数呼び出しは、選択したテストデバイスではサポートされません。デバイス固有の SWF ファイルは変更されません。これは単なる警告です。
FTPA003	loadVariables は無視されます。	エミュレータは loadVariables() 関数呼び出しを検知しました。この関数呼び出しは、選択したテストデバイスおよびコンテンツタイプではサポートされません。デバイス固有の SWF ファイルは変更されません。これは単なる警告です。
FTPA004	loadMovie は無視されます。	エミュレータは loadMovie() 関数呼び出しを検知しました。この関数呼び出しは、選択したテストデバイスおよびコンテンツタイプではサポートされません。デバイス固有の SWF ファイルは変更されません。これは単なる警告です。

エラーコード	メッセージ	説明
FTPA005	キー入力ごとに複数の要求があったため、<URL>に対する GetURL の呼び出しは無視されました。	Flash Lite ではキー押下ごとに許可される getURL() 関数呼び出しは1つだけです。エミュレータは複数の getURL() が存在することを検知したので、最初のコマンドだけが処理されます。その他のコマンドは無視されます。
FTPA006	キー入力に関連付けられていなかったため、<URL>に対する GetURL の呼び出しは無視されました。	現在選択しているテストデバイスおよびコンテンツタイプは、ユーザーがデバイスのキーを押した結果として実行される getURL() 関数呼び出しのみを処理します。エミュレータは、キー押下に関連付けられていない getURL() 呼び出しをアプリケーションが行ったことを検知しました。
FTPA007	getProperty または setProperty は、movieclip プロパティではサポートされていません。	Flash Lite は指定されたムービークリッププロパティをサポートしていません。
FTPA008	getProperty または setProperty は、movieclip プロパティでは完全にはサポートされていません。	Flash Lite は指定されたムービークリッププロパティを完全にはサポートしていません。詳細については、『Flash Lite 2.x ActionScript リファレンスガイド』で、指定したプロパティのエントリを参照してください。
FTPA009	startDrag および stopDrag はサポートされません。	エミュレータは startDrag() または stopDrag() 関数呼び出しを検知しました。Flash Lite はこの関数呼び出しをサポートしていません。
FTPA014	getURL は無視されます。	エミュレータは getURL() 関数呼び出しを検知しました。この関数呼び出しは、選択したテストデバイスおよびコンテンツタイプではサポートされません。デバイス固有の SWF ファイルは変更されません。これは単なる警告です。
FTPA015	キー入力ごとに複数の要求があったため、<URL>に対する loadMovie の呼び出しは無視されました。	Flash Lite ではキー押下ごとに許可される loadMovie() 関数呼び出しは1つだけです。エミュレータは複数の loadMovie() が存在することを検知したので、最初のコマンドだけが処理されます。その他のコマンドは無視されます。

エラーコード	メッセージ	説明
FTPA016	キー入力に関連付けられていなかったため、<URL> に対する loadMovie の呼び出しは無視されました。	現在選択しているテストデバイスおよびコンテンツタイプは、ユーザーがデバイスのキーを押した結果として実行される loadMovie() 関数呼び出しのみを処理します。エミュレータは、キー押下に関連付けられていない loadMovie() 呼び出しをアプリケーションが行ったことを検知しました。
FTPA017	キー入力ごとに複数の要求があったため、<URL> に対する loadVariables の呼び出しは無視されました。	アプリケーションは、1 回のキー押下イベントの間に複数の loadVariables() 関数呼び出しを行いました。Flash Lite ではキー押下ごとに許可される loadVariables() コマンドは 1 つだけであるため、最初のコマンドだけが処理されます。その他のコマンドは無視されます。
FTPA018	キー入力に関連付けられていなかったため、<URL> に対する loadVariables の呼び出しは無視されました。	現在選択しているテストデバイスおよびコンテンツタイプは、ユーザーがデバイスのキーを押した結果として実行される loadVariables() 関数呼び出しのみを処理します。エミュレータは、キー押下に関連付けられていない loadVariables() 呼び出しをアプリケーションが行ったことを検知しました。
FTPA019	キー入力ごとに複数の要求があったため、パラメータ<コマンド引数>を使用した FSCommand の呼び出しは無視されました。	Flash Lite ではキー押下ごとに許可される fscommand() 関数呼び出しは 1 つだけです。エミュレータは複数の fscommand() が存在することを検知したので、最初のコマンドだけが処理されます。その他のコマンドは無視されます。
FTPA020	キー入力に関連付けられていなかったため、パラメータ<コマンド引数>を使用した FSCommand の呼び出しは無視されました。	現在選択しているテストデバイスおよびコンテンツタイプは、ユーザーがデバイスのキーを押した結果として実行される fscommand() 関数呼び出しのみを処理します。エミュレータは、キー押下に関連付けられていない fscommand() 呼び出しをアプリケーションが行ったことを検知しました。
FTPE001	キーは処理されません : <i>keyname</i> ASCII 値 : <i>value</i>	エミュレータは、Flash Lite でサポートされていないデバイスキーが押されたことを検知しました。このキーの押下は無視されます。

エラーコード	メッセージ	説明
FTPE013	テキスト入力フィールドは、このデバイスの選択されたコンテンツタイプではサポートされません。	現在のテストデバイスおよびコンテンツタイプはテキスト入力フィールドをサポートしません。
FTPS010	ストリーミングサウンドはサポートされません。	選択したテストデバイスおよびコンテンツタイプはストリーミングサウンドをサポートしません。
FTPS011	一度に再生できるのは、1つのサウンドだけです (ミキシングなし)。	エミュレータは、SWF ファイル内の複数のサウンドが同時に再生されていることを検知しました。同時再生は Flash Lite でサポートされていません。
FTPS012	キー入力に関連付けられていなかったため、イベントサウンドは無視されました。	Flash Lite 1.0 では、ユーザーがデバイスのキーを押した場合にのみサウンドを再生できます。この制限は、Flash Lite 2.0 には適用されません。
FTPS021	サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	選択したテストデバイスおよびコンテンツタイプはサウンドをサポートしません。
FTPS022	ADPCM サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは ADPCM 圧縮方式で圧縮されたネイティブ (非デバイス) サウンドを検知しました。ADPCM 圧縮方式は、このデバイスの選択されたコンテンツタイプではサポートされません。デバイス固有の SWF ファイルは変更されません。これは単なる警告です。
FTPS023	MP3 サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは MP3 圧縮方式で圧縮されたネイティブ (非デバイス) サウンドを検知しました。MP3 圧縮方式は、このデバイスの選択されたコンテンツタイプではサポートされません。デバイス固有の SWF ファイルは変更されません。これは単なる警告です。
FTPS024	MIDI サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは MIDI サウンドを検知しました。MIDI サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。
FTPS025	PCM サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは PCM 圧縮方式で圧縮された Flash サウンドを検知しました。PCM 圧縮方式は、このデバイスの選択されたコンテンツタイプではサポートされません。

エラーコード	メッセージ	説明
FTPS026	ムービーのデバッグは指定したムービープレビュープレーヤーではサポートされていません。	Adobe Device Central エミュレータは [制御]-[ムービーのデバッグ] メニューコマンドをサポートしません。
FTPS027	サウンドバンドルが見つかりました。	エミュレータは、SWF ファイルにサウンドバンドルファイルが含まれていることを検知しました。
FTPS028	無効な FSCommand2<コマンド名> コマンドが見つかりました。	指定した fscommand2() コマンドは有効なコマンド文字列ではありません。有効な fscommand2() コマンドの一覧については、『Flash Lite 2.x ActionScript リファレンスガイド』の「fscommand2 関数」を参照してください。
FTPS029	FSCommand2<コマンド名> コマンドが見つかりました。	エミュレータは指定した fscommand2() コマンドを検知しました。
FTPS030	FSCommand2<コマンド名> コマンドはエミュレータでサポートされていません。デバイス上でテストしてください。	エミュレータは指定した fscommand2() コマンドをサポートしていません。指定したコマンドが意図したとおりに機能するかどうかを確認するには、Flash Lite がインストールされたデバイスでこの SWF ファイルをテストする必要があります。
FTPS031	URL リクエストを呼び出すインスタンスが複数見つかりました。キー入力 / フレームごとに許可される呼び出しは1つだけです。	Flash Lite ではキー押下またはフレームごとに許可される getURL() 関数呼び出しは1つだけです。エミュレータは複数の getURL() が存在することを検知したので、最初のコマンドだけが処理されます。その他のコマンドは無視されます。
FTPS032	GetURL(<URL>) 呼び出しが見つかりました。制限が適用される場合があります。	エミュレータは getURL() 関数呼び出しを検知しました。この関数呼び出しを選択したデバイスで実行したときに、実行時の制限が適用される場合があります。コマンドが意図したとおりに機能するかどうかを確認するには、実際のデバイスで SWF ファイルをテストします。

エラーコード	メッセージ	説明
FTPS033	loadVariables(<URL>) の呼び出しが見つかりました。制限が適用される場合があります。	エミュレータは loadVariables() 関数呼び出しを検知しました。この関数呼び出しを選択したデバイスで実行したときに、実行時の制限が適用される場合があります。コマンドが意図したとおりに機能するかどうかを確認するには、デバイスで SWF ファイルをテストします。
FTPS034	FSCommand(< コマンド名 >) 呼び出しが見つかりました。制限が適用される場合があります。	これは、すべてのデバイスおよび Flash Lite コンテンツタイプで、アプリケーションの fscommand() がサポートされているわけではないという警告です。コマンドが意図したとおりに機能するかどうかを確認するには、デバイスで SWF ファイルをテストします。
FTPS035	loadMovie(<URL>) の呼び出しが見つかりました。制限が適用される場合があります。	エミュレータは loadMovie() 関数呼び出しを検知しました。この関数呼び出しを選択したデバイスで実行したときに、実行時の制限が適用される場合があります。コマンドが意図したとおりに機能するかどうかを確認するには、デバイスで SWF ファイルをテストします。
FTPS036	サウンドバンドルに <N> キロバイトの <デバイスサウンド> サウンドが見つかりました。	サウンドバンドルのサウンドごとに、エミュレータは種類 (MIDI や SMAF など) とサイズを報告します。
FTPS037	SMAF サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは SMAF デバイスサウンドを検知しました。このデバイスの選択されたコンテンツタイプではサポートされないサウンドです。
FTPS038	1つのフレームまたはイベントに対し複数の要求があるため、StartVibrate 呼び出しは無視されました。	Flash Lite ではキー押下またはフレームごとに許可される fscommand2("StartVibrate") 呼び出しは1つだけです。エミュレータは複数の呼び出しを検知したので、最初のコマンドだけが処理されます。その他のコマンドは無視されます。
FTPS039	FSCommand2 SetInputTextType (< コマンド引数 >) が見つかりましたが、エミュレータでサポートされていません。デバイス上でテストしてください。	SetInputTextType コマンドはエミュレータではサポートされていません。実際のデバイスでテストする必要があります。

エラーコード	メッセージ	説明
FTPS048	4 方向ナビゲーションは、このデバイスではサポートされていません。	選択したテストデバイスおよびコンテンツタイプは 2 方向ナビゲーションをサポートしています。エミュレータの 5 方向キーの左または右矢印キーを押しましたが、2 方向ナビゲーションではサポートされていません。詳細については、 16 ページの「デフォルトのナビゲーションモード」 を参照してください。
FTPS049	このデバイスでは、折り返しつきの 4 方向ナビゲーションをサポートしていません。	選択したテストデバイスおよびコンテンツタイプは 4 方向ナビゲーションをサポートしています。デバイスのいずれかの矢印キーを押しましたが、ステージ上で押した矢印キーの方向にはフォーカスを受け取るオブジェクトがありませんでした。詳細については、 16 ページの「デフォルトのナビゲーションモード」 を参照してください。
FTPS050	汎用 MFi サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは汎用 MFi デバイスサウンドを検知しました。汎用 MFi デバイスサウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。
FTPS051	サポートされていないマウスイベント (イベント名) が見つかりました。	指定したマウスイベントは、選択したテストデバイスおよびコンテンツタイプではサポートされていません。
FTPS067	SMAF(MA-2) サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは SMAF (MA-2) デバイスサウンドを検知しました。このデバイスの選択されたコンテンツタイプではサポートされないサウンドです。
FTPS068	SMAF(MA-3) サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは SMAF (MA-3) デバイスサウンドを検知しました。このデバイスの選択されたコンテンツタイプではサポートされないサウンドです。
FTPS069	SMAF (MA-5) サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは SMAF (MA-5) デバイスサウンドを検知しました。このデバイスの選択されたコンテンツタイプではサポートされないサウンドです。
FTPS070	Fujitsu 拡張形式の MFi サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは Fujitsu 拡張形式の MFi デバイスサウンドを検知しました。このデバイスの選択されたコンテンツタイプではサポートされないサウンドです。

エラーコード	メッセージ	説明
FTPS071	Mitsubishi 拡張形式の MFI サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは Mitsubishi 拡張形式の MFi デバイスサウンドを検知しました。このデバイスの選択されたコンテンツタイプではサポートされないサウンドです。
FTPS072	NEC 拡張形式の MFI サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは NEC 拡張形式の MFi デバイスサウンドを検知しました。このデバイスの選択されたコンテンツタイプではサポートされないサウンドです。
FTPS073	Panasonic 拡張形式の MFI サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは Panasonic 拡張形式の MFi デバイスサウンドを検知しました。このデバイスの選択されたコンテンツタイプではサポートされないサウンドです。
FTPS074	Sharp 拡張形式の MFI サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは Sharp 拡張形式の MFi デバイスサウンドを検知しました。このデバイスの選択されたコンテンツタイプではサポートされないサウンドです。
FTPS075	Sony 拡張形式の MFI サウンドは、このデバイスの選択されたコンテンツタイプではサポートされません。	エミュレータは Sony 拡張形式の MFi デバイスサウンドを検知しました。このデバイスの選択されたコンテンツタイプではサポートされないサウンドです。
FTPS099	プリントコマンドはサポートされていません。	アプリケーションに ActionScript プリントコマンドのいずれか (print() または printAsBitmap() など) が含まれていますが、Flash Lite ではサポートされません。
FTPS100	〈デバイスサウンド〉サウンドはサウンドバンドルで選択されています。	エミュレータが再生したサウンドバンドル内のデバイスサウンド名を示します。
FTPS101	サウンドバンドルのいずれの形式も、このデバイスではサポートされていません。	このデバイスの選択されているコンテンツタイプでサポートされているサウンドバンドル内のデバイスサウンドはないことを示します。
FTPS102	SMAF サウンド再生に失敗しました。	エミュレータは SMAF サウンドを再生できませんでした。
FTPS105	この SWF は Flash Lite 形式ではありません。	アプリケーションが Flash Lite 形式ではないバージョンの SWF ファイルをロードしようとした。Flash Lite でロードできるのは、他の Flash Lite SWF ファイルまたは Flash 4 形式の SWF ファイルだけです。

エラーコード	メッセージ	説明
FTPS106	Keypress によってトリガされていないため、マウスイベント (イベント名) は無視されました。	エミュレータは、Flash Lite アプリケーションのボタン上でマウスイベントを検知しました。現在のテストデバイスはペンまたはタッチスクリーンインターフェイスをサポートしていないので、スクリーン上のボタンの操作に使用できるのは、エミュレータのキーまたは相当するキーボードショートカットだけです。
FTPS107	キーは処理されません : device-key。 このキーを有効にするには FSCommand2 SetSoftKeys を使用 します。	最初に SetSoftKeys コマンドを呼び出さずに、エミュレータでいずれかのソフトキーを押しました。詳細については、 32 ページの「ソフトキーの使用」 を参照してください。
FTPS108	無効な FSCommand (コマンド名) が見つかりました。	指定した fscommand() コマンドは有効なコマンド文字列ではありません。
FTPS109	FSCommand (コマンド名) はエミュレータでサポートされていません。デバイス上でテストしてください。	エミュレータは指定した fscommand() コマンドをサポートしていません。指定したコマンドが意図したとおりに機能するかどうかを確認するには、Flash Lite がインストールされたデバイスでこの SWF ファイルをテストする必要があります。
FTPS110	ソフトキーは、Flash Lite 1.0 プレーヤーではサポートされません。	エミュレータはいずれかのソフトキーが押されたことを検知しましたが、ドキュメントの SWF ファイルのパブリッシュ設定でバージョンが Flash Lite 1.0 に設定されています。Flash Lite 1.0 はソフトキーをサポートしていません。

索引

A

- ActionScript の fs コマンド (BREW) 83
- Adobe Device Central エミュレータ
 - BREW Simulator との比較 88
 - デバッグオプション 116
- Adobe Device Central エミュレータ (BREW)
 - テスト 90
- ADS、定義 80
- Applet Icon 画面 (ウィザード) 99
- AppLoader ツールのヒント 105

B

- BAR ファイル
 - コピー 95
 - セキュリティ 95
 - 定義 95
- BDS、定義 80
- Binary Runtime Environment for Wireless
 - (「BREW」を参照)
- BREW
 - オーサリングとパブリッシュのセットアップ 84
 - オーサリングの説明 87
 - デバイスへのファイルのアップロード 103
 - ファイル形式 92
 - ファイルのパブリッシュ 91
- BREW 2.x
 - デバイスファイルの構造 95
 - デバイスへのファイルのアップロード 106
- BREW 3.x
 - デバイスファイルの構造 95
 - デバイスへのファイルのアップロード 107
- BREW Delivery System (「BDS」を参照) 80
- BREW SDK
 - インストール 85
 - サポートされるバージョン 83

BREW Simulator

- Adobe Device Central エミュレータとの比較 88
- テスト 101
- BREW ツールスイートのインストール 85
- BREW ディベロッパーとしての登録 85
- BREW デバイスでのアプリケーションのテスト 108
- BREW デバイスへのファイルのアップロード
 - ワークフロー 104

D

- Define Output Settings 画面 (ウィザード) 100
- DLL ファイル
 - 定義 92
 - 命名規約 92

F

- Flash Lite 2.1 for BREW
 - 定義 79
- Flash Lite Publisher for BREW
 - ウィザードの使用 97
 - 定義 79
- Flash Lite エミュレータ
 - アプリケーションのプレビュー 114
 - 警告メッセージとエラーメッセージ 123
 - サポートされていない機能 114
 - 操作 116
- Flash Lite のアプリケーションモード 117
- Flash Lite の機能 (BREW)
 - サポートされない 82
 - サポートされる 82
- Flash Lite のコンテンツタイプ、説明 117

G

getURL() (BREW)

制限 89

説明 83

I

Identify Applet 画面 (ウィザード) 98

Include Applet Information 画面 (ウィザード) 98

M

MIF ファイル

定義 92

命名規約 92

MMS 83

MOD ファイル

定義 92

命名規約 92

N

National Software Testing Laboratories (「NSTL」
を参照)

NSTL、定義 80

P

Publisher ウィザードの使用 (BREW) 97

S

SDK (BREW)

インストール 85

サポートされるバージョン 83

SIG ファイル

取得 94

定義 92

命名規約 92

Simulator とエミュレータ (BREW) 88

SMS 83

Summary of Your Selections 画面 (ウィザード) 100

T

TBT、定義 80

True BREW Testing (「TBT」を参照)

U

USB ドライバのインストール (BREW) 86

X

XML (BREW)

ソケット 82

データ処理 82

XML.Send() の制限 (BREW) 89

あ

アイコン (BREW)

サイズ 99

ファイル形式 99

アップロード、BREW デバイスにファイルを
BREW 2.x 106

アップロード、ファイルを BREW デバイスに
BREW 3.x 107

説明 103

アプリケーション (BREW)

開発の対象 88

コンテンツタイプとして指定 99

アプリケーションダウンロードサーバー
(「ADS」を参照) 80

アプレットのドメイン

指定 99

セキュリティ 99

い

インストール

BREW SDK とツールスイート 85

Flash Lite 2.1 for BREW 拡張機能 105

USB ドライバ 86

インタラクティブ機能

キー押下イベントの処理 23

キー押下動作の検出 31

作成、ボタン 24

タブナビゲーション 15

ボタンを使用したメニューの作成 26

インラインテキスト (BREW) 82

う

ウィザード (BREW)

- Applet Icon 画面 99
- Define Output Settings 画面 100
- Identify Applet 画面 98
- Include Applet Information 画面 98
- Summary of Your Selections 画面 100
- 使用 97

え

- 永続データ (BREW) 82
- エミュレータと Simulator (BREW) 88

お

- オーサリング、BREW 用についての説明 87

か

- 外部イメージ、ActionScript を使用したロード 77
- 外部イメージ、サポートされている形式の確認 77
- 拡張機能、デバイスへのインストール (BREW) 105
- 確認、サポートされているオーディオファイル形式 62
- 壁紙 (BREW) 83

き

- キー押下イベント
 - ActionScript キーコード 14
 - ActionScript による処理 23
 - キーハンドラスクリプトの作成 14
 - キーリスナーの作成 31
 - サポートされるキー 14
- 共有メディア (BREW) 96

こ

- コピー、BAR ファイル 95
- コンテンツタイプ (BREW)
 - アプリケーション 88
 - スクリーンセーバー 88

さ

- サイズ、アイコン 99
- サウンド (BREW)
 - ストリーミング 82
 - デコード 82
- サウンド、デバイスとネイティブについて 55
- サポートされない Flash Lite の機能 (BREW) 82
- サポートされる
 - BREW SDK のバージョン 83
- サポートされる (BREW)
 - Flash Lite の機能 82
 - デバイス 84

す

- スクリーンセーバー (BREW)
 - 開発の対象 88
 - コンテンツタイプとして指定 99
- スクロールテキスト 53
- ステージ、スクリーンのサイズと使用可能なステージ領域 121
- ストリーミング (BREW)
 - サウンド 82
 - ビデオ 82

せ

- 制限 (BREW)
 - getURL() 89
 - loadVars() 89
 - XMLSend() 89
- セキュリティと BAR ファイル 95
- セットアップ、BREW のオーサリングとパブリッシュの説明 84
- 説明
 - BREW デバイスへのファイルのアップロード 103
 - BREW のオーサリングとパブリッシュのセットアップ 84
 - BREW 用ファイルのパブリッシュ 91
 - オーサリング、BREW 用の 87
- 前提条件、デバイスへのファイルのアップロードに対する (BREW) 104

そ

- ソフトウェア要件 (BREW) 81

た

ダイアログボックス (BREW)
 パブリッシュ設定 97
対象、開発のデバイス (BREW) 88
タブナビゲーション
 アプリケーション例 26
 折り返し付きの 4 方向 17
 ガイドライン 20
 説明 15
 フォーカス矩形 19
 モード 16

ち

着信アニメーション (BREW) 83

つ

ツールスイート (BREW) のインストール 85

て

定義
 ADS 80
 BAR ファイル 95
 BDS 80
 DLL ファイル 92
 Flash Lite 2.1 for BREW 79
 Flash Lite Publisher for BREW 79
 MIF ファイル 92
 MOD ファイル 92
 NSTL 80
 SIG ファイル 92
 TBT 80
データ、永続 (BREW) 82
テキスト入力フィールド
 サンプルアプリケーション 51
 文字の制限 45
テキストフィールド
 スクロールテキストの作成 53
 テキスト入力フィールドの文字の制限 45
 テキスト入力フィールド、使用 39
 レンダリング品質 49
テキスト、インライン (BREW) 82
テスト (BREW)
 Adobe Device Central エミュレータの使用 90
 BREW Simulator 101
 BREW デバイスでの 108
 Simulator とエミュレータ 88

デバイス (BREW)

 BREW 2.x のファイル構造 95
 BREW 3.x のファイル構造 95
 開発の対象 88
 サポートされる 84
 ファイルのアップロードに対する前提条件 104
デバイスサウンド 62
デバイスサウンド、_forceframerate プロパティ 62
デバイスサウンド、アニメーションと同期 62
デバイスバック (BREW)
 一般と固有 90、101
 ダウンロード 90、101
デバイスビデオ
 エミュレータ 123
 外部ビデオの再生 73
 説明 66
 バンドル 67
 読み込み 67
 ライブラリからの再生 71
デバイスファイルの構造、BREW 2.x 95
デバイスファイルの構造、BREW 3.x 95

と

ドメイン (アプレット)
 指定 99
 セキュリティ 99
ドライバ (USB) のインストール 86

な

ナビゲーション、「タブナビゲーション」を参照

ね

ネイティブサウンド
 8kHz でのリサンプリング 63
 説明 63

は

ハードウェア要件 (BREW) 81
パブリッシュ設定 (BREW)
 ダイアログボックス 97
 ポストプロセッサリスト 97
パブリッシュ、BREW 用のファイルの説明 91

ひ

未サポートの Flash Lite の機能 (BREW) 82

ビデオのストリーミング (BREW) 82

ヒント、AppLoader ツールの使用 105

ふ

ファイル形式

BREW 92

アイコン 99

命名規約 92

フォントアウトラインの埋め込み、説明 50

フォントアウトライン、SWF ファイルへの埋め込み

50

フォントのレンダリング方法、テキストフィールドへの
適用 47

ほ

ポストプロセッサリスト、パブリッシュ設定ダイアログ

ボックス (BREW) 97

ボタンイベント 24

め

命名規約、ファイル形式 (BREW) 92

メディア、共有 (BREW) 96

メニュー、ボタンを使用して作成 26

よ

要件 (BREW)

ソフトウェア 81

ハードウェア 81

れ

レンダリング品質、デフォルトの画質 49

