

# ADOBE FLASH MEDIA SERVER 4.5

## 技術概要

## **法律上の注意**

法律上の注意については、[http://help.adobe.com/ja\\_JP/legalnotices/index.html](http://help.adobe.com/ja_JP/legalnotices/index.html) を参照してください。

# 目次

## 第1章：はじめに

サーバーのエディション .....	1
必要システム構成 .....	1
Flash Media Server 4.5 の新機能 .....	1
サーバーの一般的な使用法 .....	4

## 第2章：アーキテクチャ

Flash Media Server アーキテクチャ .....	8
ストリーミングメディア .....	12
言語ライブラリ .....	17
サーバーの規模の調整 .....	20
サーバーの設定 .....	22
サーバーの管理 .....	22

## 第3章：セキュリティ

コンテンツのセキュアなストリーミング .....	24
コンテンツの保護 .....	25
サーバーのセキュアな設定 .....	27
サーバーサイドスクリプトのセキュリティ .....	27
セキュアなアプリケーションシナリオ .....	28

# 第 1 章：はじめに

Adobe® Flash® Media Server は、アダプティブビットレートのビデオオンデマンド（VOD）およびライブビデオを Adobe® Flash® Player、Adobe® AIR™、Adobe® Flash® Lite™、Apple® iOS および Mac OS® に配信するリアルタイムメディアサーバーです。

また、Flash Media Server は、ストリーミングミュージック、ビデオのブログ、ビデオメッセージング、マルチメディアチャット環境、リアルタイムデータキャストおよびマルチユーザーのゲームも提供します。コンテンツアプリケーションとメディアを提供する Flash Media Server は、複数のプラットフォームとブラウザで一環して動作します。

Flash Media Server 4.5 には、Adobe Flash Media Enterprise Server、Adobe Flash Media Interactive Server、Adobe Flash Media Streaming Server、および Adobe Flash Media Development Server という 4 つのエディションがあります。

## サーバーのエディション

サーバー エディションの機能比較については、「[www.adobe.com/jp/products/flashmediaserver/compare/](http://www.adobe.com/jp/products/flashmediaserver/compare/)」を参照してください。

## 必要システム構成

最新の必要システム構成については、[www.adobe.com/go/learn\\_fms\\_sysreqs\\_jp](http://www.adobe.com/go/learn_fms_sysreqs_jp) を参照してください。

## Flash Media Server 4.5 の新機能

### Adobe HTTP Dynamic Streaming および Apple HTTP Live Streaming を使用したコンテンツの提供

HTTP 経由でアダプティブビットレートのオンデマンドおよびライブコンテンツを Flash Player、AIR、iOS および Mac OS に提供します。サーバーはクライアントが要求する場合には、リアルタイムにライブおよびオンデマンドコンテンツをパッケージングします。HTTP Dynamic Streaming および HTTP Live Streaming はともに DVR をサポートしています。

**注意：**Flash Media Server 4.0 では、HTTP Dynamic Streaming をサポートしていましたが、Flash Media Server 4.5 では、オンデマンドコンテンツのジャストインタイムパッケージングのサポートも追加されています。

詳しくは、以下のトピックを参照してください。

- 『Flash Media Server デベロッパーズガイド』の「ライブメディアのストリーミング（HTTP）」および「オンデマンドメディアのストリーミング（HTTP）」
- 『Flash Media Server デベロッパーズガイド』の「Adobe HTTP Dynamic Streaming および Apple HTTP Live Streaming の設定」
- 『Flash Media Server 設定および管理ガイド』の「HTTP Streaming 用のポートの設定」
- 「Flash Media Server 4.0 と Flash Media Server 4.5 での HTTP Dynamic Streaming の相違点」

## 保護された HTTP Dynamic Streaming および HTTP Live Streaming

保護されたライブおよびオンデマンドマルチビットレートビデオを DRM ライセンスサーバーを使用しないで、Flash Player、AIR、iOS および Mac OS に提供します。保護された HTTP Dynamic Streaming も SWF 検証を差ポートしています。

詳細については、『[Flash Media Server デベロッパーズガイド](#)』の次のセクションを参照してください。

- [Protected HTTP Dynamic Streaming \(PHDS\) の設定](#)
- [Protected HTTP Live Streaming \(PHLS\) の設定](#)
- PHDS と PHLS の連携使用

## HTTP ストリーミング用オーディオのみのストリームのパブリッシュ

Apple HTTP Live Streaming では、セルラーネットワーク経由でコンテンツを配信するには、マルチビットレートセットのストリームの 1 つがオーディオのみである必要があります。

「[オーディオのみのストリームのパブリッシュ \(HLS\)](#)」を参照してください。

## セットレベル F4M/M3U8 ファイル

マルチビットレートストリームのセットを示すマニフェストファイルを作成します。これらのファイルは、「セットレベルマニフェストファイル」と呼ばれています。HTTP Packager は、コンテンツが要求されたときにリアルタイムにストリームレベルのマニフェストファイルを生成します。

**注意：** Adobe HTTP Dynamic Streaming の場合、マニフェストファイルは F4M ファイルです。Apple HTTP Live Streaming の場合、同等なファイルは「バリエーション再生リスト」と呼ばれ、ファイル名の拡張子は M3U8 です。ドキュメントでは、両方のファイルタイプを参照するため、「マニフェストファイル」という一般的な用語を使用します。

セットレベル F4M/M3U ファイルには、コンテンツのセットに関するビットレート情報が含まれています。ストリームレベル F4M/M3U8 ファイルには、ブートストラップ情報および DRM メタデータが含まれています。Flash Media Server 4.5 には、HTTP Dynamic Streaming 用のセットレベル F4M v2.0 ファイルおよび HTTP Live Streaming 用のセットレベル M3U8 バリエーション再生リストを作成する Set-Level File Generator ツールが含まれています。

「[セットレベルマニフェストファイル](#)」を参照してください。

## File プラグインを使用した HTTP ライブストリーミングコンテンツの管理

File プラグインを使用し、非同期ファイル IO の操作を含む、HTTP ライブストリーミングコンテンツの管理を行います。Live Packager アプリケーション (livepkg) は、ライブストリームを取り込んでフラグメント (F4F ファイル) と追加ヘルパーファイルにパッケージ化します。これらのファイルを記録するサーバー操作は、File プラグインを介してルーティン化されます。

「[File プラグインを使用した HTTP ライブストリーミングコンテンツの管理](#)」を参照してください。

## HTTP ファイルシステムのパフォーマンスの向上

IO バッファにより、HTTP ストリーミング (HTTP Dynamic Streaming および HTTP Live Streaming 用) の読み取りおよび書き込みのパフォーマンスが向上します。IO バッファは、ディスクファイルをメモリ内のバッファにロードします。システムコールを作成するのではなく、メモリ内のバッファに対する読み取りと書き込みを行います。

「[IO バッファのサイズの設定](#)」を参照してください。

## ディスク管理により年中無休のライブストリーミングが可能

サーバーがディスク上に格納するライブコンテンツ量を設定します。デフォルトでは、サーバーは3時間相当のコンテンツを格納します。この機能により、ディスクをいっぱいにすることなく、1日24時間年中無休でライブストリームを配信することができます。また、長時間のライブイベントにはDVRを使用することもできます。

「ディスク管理」を参照してください。

## サーバー間のRTMFPピア紹介の配信

Flash Media Serverは、RTMFPクライアントを互いに紹介し、クライアントが互いに直接接続できるようにします。この直接接続は、ピアツーピア接続と呼ばれます。Flash Media Server 4.0は、クライアントが1台のサーバーに接続している場合のみ、クライアントを互いに紹介することができます。クライアントが個別のサーバーに接続されている場合でも、Flash Media Server 4.5に追加されているサーバーサイド ActionScript API を使用して、クライアントを互いに紹介します。サーバー間に紹介を配信すると、ピアアシストネットワークングアプリケーションの規模を調整することができます。

「サーバー間のピア紹介の配信」を参照してください。

## マルチキャストストリームの取り込み

サーバーサイド ActionScript を使用して、マルチキャスト RTMFP ストリームを取り込みます。サーバーがマルチキャストストリームを取り込んだら、スクリプトを記述して次の操作を行います。

- マルチキャストストリームを Stream オブジェクトに変換します。
- livepkgr アプリケーションに接続し、HTTP Dynamic Streaming と HTTP Live Streaming を使用した配信のために Stream をパッケージ化します。
- Stream オブジェクトを記録します。
- RTMP/T/S/E を使用して Stream オブジェクトをクライアントに配信します。

「マルチキャストストリームの取り込み」を参照してください。

## 管理者パスワードの安全な格納

Flash Media Administration Console 用の管理者パスワードは、テキスト形式ではなく、安全に格納されます。「管理者アカウントの削除とパスワードのリセット」を参照してください。

## 設定ファイルの更新

新しい機能とセキュリティ強化を図るため、一部の設定ファイルが変更されています。Flash Media Server 4.0 から 4.5 にアップグレードする場合は、ドキュメントを利用して、旧バージョンからの移行に役立ててください。

XML ファイルの相違点の詳細については、「4.0 から 4.5 への設定ファイルの変更」を参照してください。

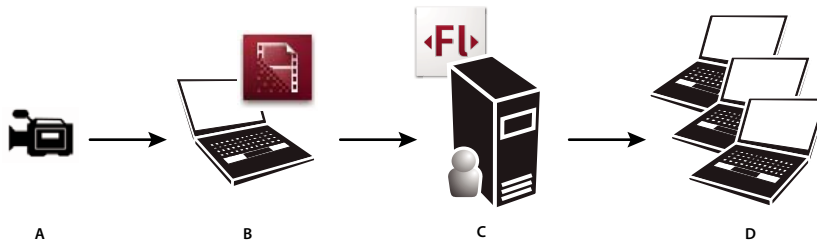
## サーバーの一般的な使用法

### ライブビデオのキャプチャとブロードキャスト

Adobe Flash Media Live Encoder では、オーディオとビデオを Flash Media Server にストリーミングすると同時にキャプチャできます。PC カメラから高性能のデジタルビデオ カメラに至るさまざまなカメラを使用して、ビデオをキャプチャすることができます。HTTP 経由でビデオを Flash Player、AIR、iOS および Mac OS に配信します。RTMP 経由でビデオを Flash Player および AIR に配信します。

多くの視聴者にメディアをブロードキャストするには、マルチポイントパブリッシュを使用します。この機能を使用すると、ライブビデオをカメラからパブリッシュサーバーにストリーミングしてから、ブロードキャストサーバーにストリーミングすることができます。ブロードキャストサーバーは、多くの場合、コンテンツ配信ネットワークです。

**注意:** ライブビデオは Flash Media Streaming Server へおよび Flash Media Streaming Server からストリーミングすることができます。ただし、Flash Media Streaming Server では、サーバーサイドのスク립ティングが必要になるので、マルチポイントパブリッシュはサポートされません。



A. ライブビデオ B. Flash Media Live Encoder (またはカスタム構築された Flash Player または AIR ソリューション) C. Flash Media Server D. Flash Player および AIR クライアント

マルチポイントパブリッシュは、ライブストリームにメタデータを挿入するために使用することができます。例えば、インターネットテレビ局を作成し、Flash Media Development Server にストリームをパブリッシュできます。開発サーバーでは、何百万ものユーザーにストリームをプッシュ配信できる CDN に、ストリームをパブリッシュします。

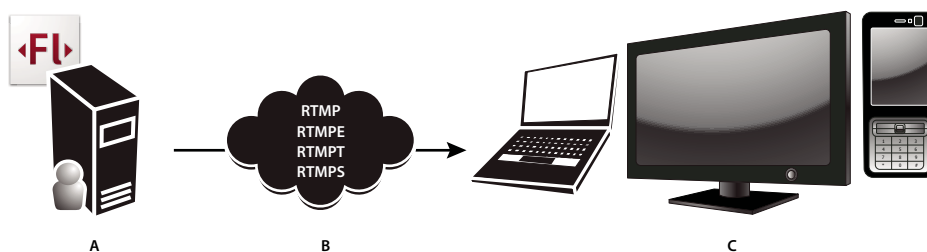
Flash Media Live Encoder の詳細については、[http://www.adobe.com/go/learn\\_fms\\_fme\\_jp](http://www.adobe.com/go/learn_fms_fme_jp) を参照してください。

### 録画されたビデオのブロードキャスト

録画されたビデオ、または「オンデマンド」ビデオを配信するために、Strobe Media Playback というビデオプレーヤーを提供しています。このビデオプレーヤーは、FLV ファイルおよび MP4/F4V ファイルの再生をサポートしています。サーバーはこのビデオプレーヤーを `rootinstall/samples/videoPlayer` にインストールします。ユーザー自身のビデオ Player を開発することもできます。ビデオ Player は Flash Player または AIR で実行され、Flash Media Server からの記録済みビデオまたはライブビデオをストリーミングします。メディアを HTTP 経由で Flash Player、AIR、iOS デバイスおよび Mac OS にブロードキャストできます。

以下に、ストリーミングできるコンテンツの種類の例を示します。

- コマーシャルなど、30 秒までの短いビデオクリップ
- ユーザーが生成したビデオなど、30 分までの長いビデオクリップ
- 数時間までの録画されたテレビ番組や映画
- クライアントサイドまたはサーバーサイドの再生リストでは、ストリームの種類がライブストリーム、記録されたストリームまたはそれらの組み合わせのどれであっても、リストにあるストリームを順番に再生できます。再生リストはクライアントサイドスクリプトでも、Flash Media Streaming Server を除くすべてのサーバーエディション上のサーバーサイドスクリプトでも作成できます。

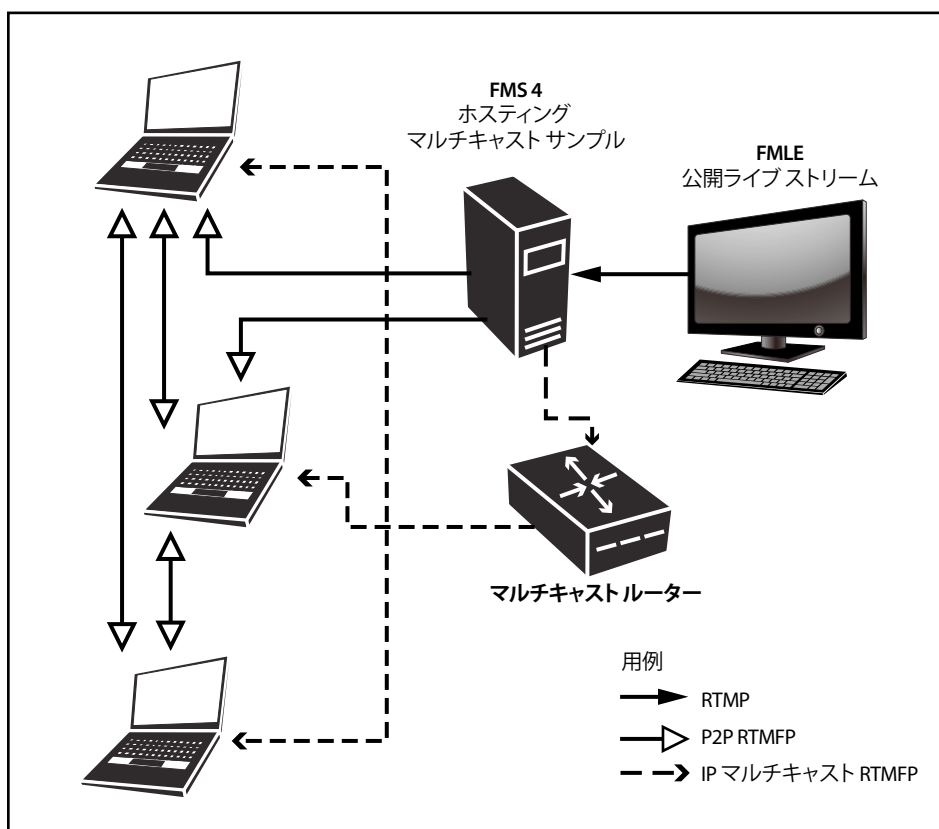


A. Flash Media Server は、記録されたメディアをクライアントにストリーミングします。B. インターネット (RTMP、HTTP) C. Flash Player、AIR、iOS デバイスおよび Mac OS がビデオをストリーミングします。

## 社内でのライブビデオのマルチキャスト

ピアアシストネットワーク、IP レベルマルチキャスト (または両方) を使用して、Flash Player と AIR クライアントにライブビデオをマルチキャストします。両方の技術を併用するとき、この技法を「マルチキャストフュージョン」と呼びます。

ビデオがブロードキャストされると、サーバーはすべてのクライアントにストリーミングを送信します。ビデオをマルチキャストすると、マルチキャスト対応ルーターがビデオをクライアントに送信するか、クライアントがビデオを相互に送信します。

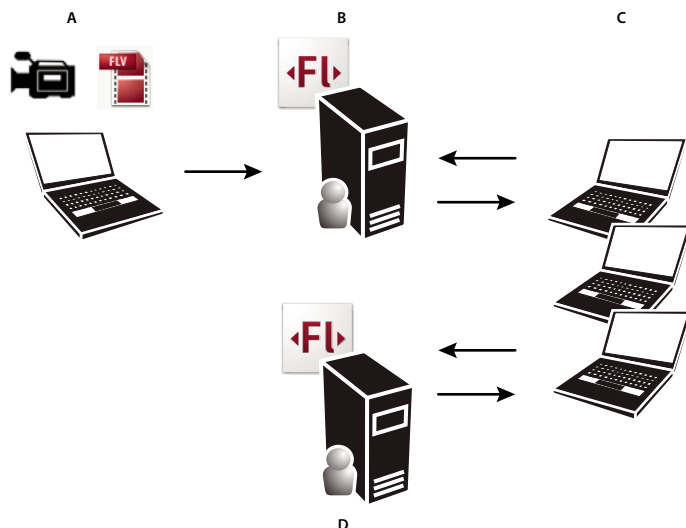


マルチキャストフュージョン

**注意:** また、オンデマンドビデオもマルチキャストできますが、使用例は一般的ではありません。

## 広告付きブロードキャストビデオ

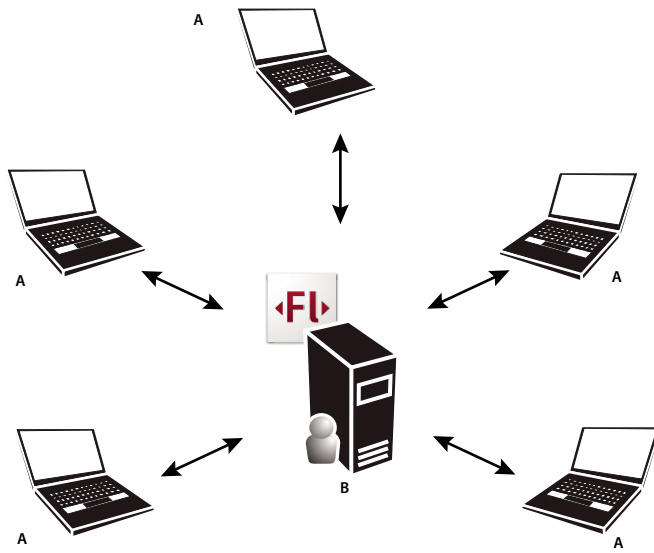
ストリーミングビデオアプリケーションは、録画されたテレビ番組やライブビデオの前に再生される短いコマーシャルなど、様々なポイントに広告を挿入できます。多くの場合、1つのサーバーから広告がストリーミングされ、別のサーバーまたはコンテンツ配信ネットワークからコンテンツがストリーミングされます。広告付きビデオのアプリケーションは通常、広告サーバーに接続し、広告をストリーミングしてから広告サーバーとの接続を閉じます。次にコンテンツサーバーに接続し、コンテンツをストリーミングしてからその接続を閉じます。ビデオがストリーミングされるたびにこの処理の流れを繰り返します。



A. ライブビデオ B. Flash Media Server (記録されたコンテンツおよびライブコンテンツを提供) C. Flash Player および AIR クライアント HTTP 経由の場合は、同様に iOS および Mac OS クライアント D. Flash Media Server (広告を提供)

## ビデオとインタラクティブアプリケーションとの統合

Flash Media Server アプリケーションは、ビデオ共有、オンラインチャット、Web 会議およびその他のコミュニティ構築機能を通じて、ユーザーを引きつけることができます。ユーザーは、テキストメッセージはもちろんオーディオやビデオもサーバーに送信することができます。サーバーはそれらのデータを、接続されているすべてのユーザーにストリーミングします。サーバーは、ビデオメッセージングアプリケーション使用時のように、後で再生するためにメディアを記録することもできます。



**A.** クライアントはオーディオ、ビデオおよびデータメッセージを送受信できます。**B.** Flash Media Server は接続しているすべてのユーザーにメディアとデータをブロードキャストします。

## 第 2 章：アーキテクチャ

### Flash Media Server アーキテクチャ

#### クライアントサーバーアーキテクチャ

Adobe Flash Media Server アプリケーションはクライアントサーバーアーキテクチャを採用しています。クライアントコードは ActionScript (1、2、または 3) で記述され、Adobe Flash Player 6 以降および Adobe AIR 1 以降で実行されます。Flash Media Server はメディアを iOS デバイスなどの Apple HTTP Live Streaming クライアントにストリーミングすることもできます。

サーバーコードはサーバーサイド ActionScript で記述されます。これは、ActionScript 1.0 に類似しています。

#### リアルタイムメディアサーバー

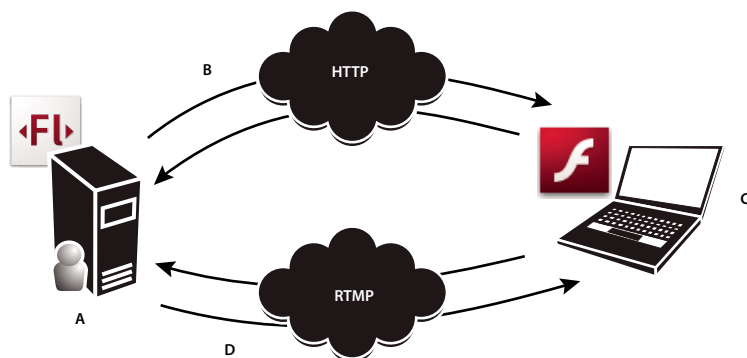
Flash Media Streaming Server は、ライブ、VOD (ビデオオンデマンド)、livepkgr (HTTP ストリーミング) およびマルチキャスト (RTMFP) の 4 種類のストリーミングサービスを提供します。ストリーミングサービスは、構築済みのサーバーサイドアプリケーションです。各ストリーミングサービスには、開発者が独自のクライアントを記述するために使用できるクライアント SDK だけでなく、構築済みのサンプルクライアントが用意されています。

ストリーミングを素早く理解するチュートリアルについては、「メディアのストリーミングの概要」を参照してください。

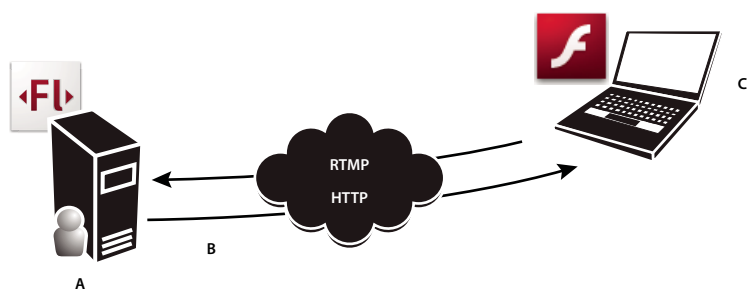
#### リアルタイムコラボレーションアプリケーションサーバー

リアルタイムコラボレーションアプリケーションを構築するには、RTMP および クラシッククライアントサーバーモデルを使用します。または RTMFP を使用してピアアシストネットワークングアプリケーションを構築します。Flash Player 10.1 および AIR 2 は、Real-Time Media Flow Protocol (RTMFP) および RTMFP グループをサポートしています。RTMFP は、User Datagram Protocol (UDP) 上に構築されています。RTMP は、Transmission Control Protocol (TCP) 上に構築されています。UDP では、TCP よりも遅延が小さくなります。また、エンドツーエンドのピアリング (つまり、2 つのクライアント間の直接データ送信) も可能です。クライアントは Flash Media Server に最初の接続を行います。サーバーはクライアントをグループに接続するイントロデューサーとして機能します。グループに接続されると、グループのメンバーはサーバーにデータを戻すことなく、グループ間でデータを交換します。詳細については、「ピアアシストネットワークングアプリケーションの構築」を参照してください。

サーバーとクライアントは永続的な接続を介して、Real-Time Messaging Protocol (RTMP) を使用して通信することができます。RTMP は、ストリーミングとデータサービスのための、信頼性が高い TCP/IP プロトコルです。一般的には、Web サーバーが HTTP 経由でクライアントを配信します。クライアントは RTMP を使用して Flash Media Server へのソケット接続を作成します。この接続によって、クライアントとサーバーの間でリアルタイムにデータをストリーミングできます。Flash Media Server は、デフォルトで Apache Web サーバーと共にインストールされます。この Web サーバーから HTTP コンテンツを配信することができます。または、Flash Media Server のインストールから Apache を除外して、SWF および HTML コンテンツを外部の任意の Web サーバーから配信することもできます。



Flash Media Server アプリケーションは、連携して動作するクライアントコンポーネントとサーバーコンポーネントから構成されます。  
A. Flash Media Server B. Web サーバーが SWF ファイルを送信します。 C. Flash Player および AIR クライアントが SWF ファイルを再生します。  
D. SWF ファイルが Flash Media Server 上のアプリケーションに接続します。サーバーは、永続的な接続を介してデータをストリーミングします。



A. Flash Media Server B. Web サーバーが SWF ファイルを送信します。 C. Flash Player または AIR クライアントが SWF ファイルを再生します。

Flash Media Enterprise Server、Flash Media Interactive Server、および Flash Media Development Server には、Flash Media Streaming Server と同じストリーミングサービスが含まれています。さらに、開発者がクライアントサイドとサーバーサイドの両方のメディアアプリケーションのコンポーネントを記述し、双方向のインタラクティブアプリケーションを作成できる SDK が提供されています。これらのサーバーエディションにはプラグイン SDK も付属します。この SDK により、開発者はサーバーのコア機能を拡張するプラグインを C++ で記述することができます。

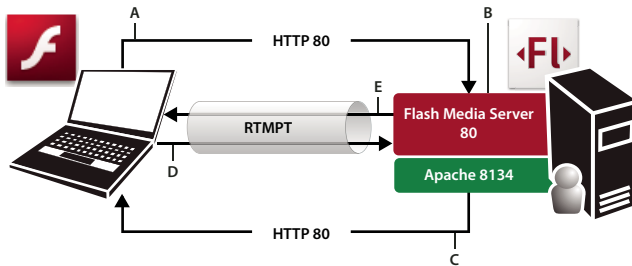
**注意：**Flash Media Streaming Server がサポートしているのは、Access プラグインのみです。

### ビルトイン Apache HTTP Server

Flash Media Server 3.5 以降のすべてのバージョンに Apache 2.2 HTTP Server が含まれています。Apache をインストールして有効にすると、クライアント SWF ファイル、コンテナ HTML ファイルおよびすべてのメディアアセットを同じサーバーから配信することができます。

Flash Media Server 4.5 と Apache 2.2 HTTP Server を使用して、Adobe HTTP Dynamic Streaming と Apple HTTP Live Streaming を使用したコンテンツを配信します。

デフォルトでは、次の図に示すように、Flash Media Server は HTTP 要求をポート 80 からポート 8134 にプロキシします。



A. クライアントが HTTP 経由でコンテンツを要求します。B. Flash Media Server がポート 8134 に要求をプロキシします。C. Apache Web サーバーがプログレッシブダウンロード用にコンテンツを配信します。D. クライアントが RTMPT 経由でコンテンツを要求します。E. Flash Media Server がコンテンツをクライアントにストリーミングします。

詳細については、「[HTTP Streaming 用のポートの設定](#)」を参照してください。

プロキシは `fms.ini` ファイルで設定することができます。詳細については、「[Apache HTTP Server の設定](#)」を参照してください。

## 複数のアプリケーションのホスティング

Flash Media Enterprise Server、Flash Media Interactive Server、および Flash Media Development Server がホスティングできるアプリケーションの数に制限はありません。Flash Media Streaming Server がホスティングできるライブサービスと VOD サービスのインスタンスの数に制限はありません。

**注意：**Flash Media Streaming Server では、実行するサービスはアドビ システムズ社が提供するものに制限されています。

例えば、Flash Media Interactive Server は、Web 会議アプリケーション、ビデオブログアプリケーション、ビデオチャットアプリケーションおよびマルチプレーヤーゲームをホストすることができます。サーバーでは、ライブサービスと VOD サービスをホストすることもできます。これらのアプリケーションそれぞれについて、複数のインスタンスを作成できます。インスタンスを利用することで、グループ同士はやり取りせず、ユーザーのグループが同じアプリケーションにアクセスすることができます。例えば、トピックごとに部屋を設けるビデオチャットアプリケーションを作成することができます。

アプリケーションを作成するには、サーバー上に、アプリケーションの名前を持つフォルダーを作成します。サーバーに接続するには、クライアント内で `NetConnection.connect()` メソッドを呼び出すコードを記述して、アプリケーションの名前を渡します。

サーバーには、デフォルトで `RootInstall/applications` に格納されるルートアプリケーションフォルダーがあります。アプリケーションを作成するには、ルートアプリケーションフォルダー内にサブフォルダーを作成します。例えば、`RootInstall/applications/mySampleApp` は、「mySampleApp」というアプリケーションを作成します。mySampleApp に接続するには、クライアントから `myNetConnection.connect("rtmp://serverName/mySampleApp")` を呼び出します。

アプリケーションのインスタンスを作成するには、アプリケーションのフォルダー内にサブフォルダーを作成します。例えば、`RootInstall/applications/mySampleApp/instance1` は、「mySampleApp」というアプリケーションのインスタンスを作成します。このインスタンスに接続するには、クライアントから `myNetConnection.connect("rtmp://serverName.mySampleApp/instance1")` を呼び出します。

サーバー管理者はルートアプリケーションフォルダーの場所を変更できます。また、サーバーを複数のアダプターと仮想ホストに分割することもできます。このようにすると、仮想ホストごとに専用のアプリケーションフォルダーを持つことができます。

詳細情報

[仮想ホスト用のサーバーの設定](#)

[コンテンツの格納場所の設定](#)

[サーバーへの接続](#)

## データモデル

Flash Media Server アプリケーションは、簡単であっても強力な、共有オブジェクトに基づく分散データモデルを使用します。クライアントサイド ActionScript とサーバーサイド ActionScript のどちらにも SharedObject クラスがあり、開発者はこのクラスを使用して、サーバーに接続されたクライアント間でデータを共有することができます。

共有オブジェクトには、ローカルとリモートの 2 つの種類があります。ローカル共有オブジェクトはクライアントコンピュータに格納され、リモート共有オブジェクトはサーバーに格納されます。ローカルとリモートのどちらの共有オブジェクトも、一時的または永続的のいずれかにすることができます。

ローカル共有オブジェクトは cookie に似ており、オフラインアクセスやユーザー設定を保存するために、ユーザーのコンピュータにデータを保存します。ローカル共有オブジェクトは Flash Player の機能であり、その動作に Flash Media Server は必要ありません。

リモート共有オブジェクトは、サーバーによって管理と格納が行われます。開発者は、メッセージング、データ同期およびデータの格納のためにリモート共有オブジェクトを使用できます。クライアントはリモート共有オブジェクトに接続し、共有オブジェクトに変更があった場合には更新を受け取ります。リモート共有オブジェクトに接続しているすべてのクライアントにメッセージを送信できます。

**注意：**リモート共有オブジェクトは Flash Media Streaming Server ではサポートされていません。

詳細については、サーバーサイド ActionScript の「[SharedObject クラス](#)」、クライアントサイド ActionScript の「[SharedObject クラス](#)」および「[ソーシャルアプリケーションの開発](#)」を参照してください。

## リモートメソッドの呼び出し

Flash Media Enterprise Server、Flash Media Interactive Server、および Flash Media Development Server は、双方向で非同期に行われるリモートメソッドの呼び出しをサポートしています。クライアントはサーバーで定義されているメソッドを呼び出すことができ、サーバーは自身に接続されているクライアントのメソッドを呼び出すことができます。クライアントサイドスクリプトでは、NetConnection.call() メソッドを呼び出してサーバーサイドの Client オブジェクトで定義されているメソッドを呼び出します。サーバーサイドスクリプトでは、Client.call() メソッドを呼び出してクライアントサイドの NetConnection オブジェクトで定義されているメソッドを呼び出します。サーバーサイドスクリプトで NetConnection.call() を呼び出す場合も、リモートサーバーのメソッドを呼び出すことができます。

詳細については、クライアントサイドの「[NetConnection.call\(\)](#)」メソッドおよびサーバーサイドの「[Client.call\(\)](#)」メソッド、「[Client.remoteMethod\(\)](#)」メソッド、「[NetConnection.call\(\)](#)」メソッドを参照してください。

## 外部ソースへの接続

**注意：**Flash Media Streaming Server はこの機能をサポートしていません。

Flash Media Server は、Web サービスおよびリレーショナルデータベースなどの外部データソースや、他の Flash Media Server アプリケーションとのやり取りができます。例えば、Web サービスに接続する、あるいは ColdFusion® アプリケーションに接続するサーバーサイド ActionScript を記述して、名前や電話番号のリストを取得します。クエリの結果を共有オブジェクトに入れます。

詳細については、『[サーバーサイド ActionScript リファレンスガイド](#)』を参照してください。

アクションメッセージ形式 (AMF) は、シリアル化された ActionScript オブジェクトを表すバイナリファイル形式です。ActionScript オブジェクトをインターネット経由で送信するために使用します。AMF には、AMF 0 (ActionScript 1.0 と 2.0) および AMF 3 (ActionScript 3.0) の 2 つのバージョンがあります。次の表に、上記の 2 つの AMF バージョンのサポートに追加されたサーバーのバージョンとプレーヤーのバージョンを示します。

AMF バージョン	Flash Media Server バージョン	Flash Player バージョン
AMF 0	1	6 ; Flash Lite 3

AMF バージョン	Flash Media Server バージョン	Flash Player バージョン
AMF3	3	8

**注意：**Flash Player、AIR および Flash Lite クライアントは、外部ソースとのやり取りに、サポートされているすべてのクライアントサイド ActionScript API を使用することもできます。Flash Media Server には、この機能に関する制限はありません。詳細については、「[Adobe Flash Platform 用 ActionScript 3.0 リファレンスガイド](#)」を参照してください。

## ストリーミングメディア

### ストリームメディア

メディア（オーディオ、ビデオおよびデータを含む場合があります）はクライアントと Flash Media Server の間でリアルタイムに送信され、到着すると表示されます。このようなデータ送信は「ストリーミング」と呼ばれます。クライアントと Flash Media Server の間でストリーミングされるメディアは「ストリーム」と呼ばれます。ストリームはパブリッシュ/サブスクライブモデルを使用します。「サブスクライブ」とは再生を意味します。クライアントとサーバーのどちらでもストリームをパブリッシュできますが、ストリームを再生できるのはクライアントだけです。メディアは RTMP または HTTP 経由でストリーミングできます。

例えば、作成者が Adobe Flash Media Live Encoder を使用して、基調講演のライブのオーディオとビデオをキャプチャしてエンコードし、サーバーにパブリッシュしたとします。ユーザーはこの講演を、ストリームにサブスクライブする Flash Player または AIR クライアントで見ることができます。

別のシナリオでは、ソーシャルメディア Web サイトのユーザーは、ライブストリームをビデオチャットアプリケーションでパブリッシュできます。この場合、ユーザーの PC カメラからビデオを Flash Player でキャプチャすることになります。ソーシャルメディアサイトの他のユーザーは、ストリームをライブで見ることができます。また、開発者が機能を追加した場合は、記録されたストリームを後で再生することができます。

**注意：**Flash Media Streaming Server を除くサーバーのすべてのエディションでは、ライブストリームを記録して後で再生することができます。

サーバーはストリームを、クライアントまたは他のサーバーに対してパブリッシュできます。例えば、XML データをサーバーサイドスクリプトに取り込んで再生リストを作成し、それをクライアントにストリームとしてパブリッシュすることもできます。より多くのクライアントをサポートするため、あるサーバーから他のサーバーにストリームをパブリッシュし、ライブブロードキャストアプリケーションの規模を調整することができます。

### メディアのマルチキャスト

マルチキャストリングは、オーディオおよびビデオデータを、RTMFP グループのメンバー間で配信します。サーバーはデータを各クライアントに送信しません。データはピア間で配信されます。マルチキャストリングにより、少ないパブリッシュ元が大量のデータを送信できます。メディアをマルチキャストするには、GroupSpecifier クラスを使用して、RTMFP グループを示す不透明な groupspec 文字列を構築します。groupspec を NetStream コンストラクターに渡します。マルチキャストデータに対し、NetStream.publish() または NetStream.play() を呼び出します。データをマルチキャストするには、クライアントサイド NetStream クラスとサーバーサイド NetStream クラスを使用できます。「マルチキャストリング」を参照してください。

## サポートされるプロトコル

### RTMP (Real-Time Messaging Protocol)

すべてのサーバーエディションは Flash Player および AIR と、RTMP (Real-Time Messaging Protocol) を介して通信します。RTMP は負担が大きいストリームをリアルタイムに配信できるよう最適化されています。1つの RTMP 接続で、任意の数のストリームのマルチプレックス接続を実行できます。各ストリームには、同期されたオーディオ、ビデオおよびデータのチャンネルが含まれます。リモートメソッドの呼び出しと共有オブジェクトメッセージは、データ専用ストリームで伝達されます。

Flash Media Server は、次の種類の RTMP 接続をサポートしています。

**RTMP** これは標準の、暗号化されない RTMP です。デフォルトのポートは 1935 です。ポートが指定されていない場合、クライアントは 1935、443、80 (RTMP)、80 (RTMPT) という順序でポートへの接続を試みます。

**RTMPT** このプロトコルは HTTP を介してトンネリングされた RTMP です。RTMP データは有効な HTTP としてカプセル化されます。デフォルト値は 80 です。

**RTMPS** このプロトコルは SSL を介した RTMP です。SSL は、TCP/IP 上での通信を保護するプロトコルです。Flash Media Server では、外部からと外部への両方の SSL 接続をネイティブでサポートしています。デフォルト値は 443 です。

**RTMPE** このプロトコルは、RTMP の暗号化されたバージョンです。RTMPE は SSL より高速で、証明書の管理を必要とせず、サーバーの `Adaptor.xml` ファイルで有効にされます。ポートを明示的に指定しないで RTMPE を指定すると、Flash Player は RTMP の場合と同様に 1935 (RTMPE)、443 (RTMPE)、80 (RTMPE)、80 (RTMPT) という順序でポートをスキャンします。

**RTMPTE** このプロトコルは、暗号化されたトンネリング接続を使用する RTMPE です。デフォルト値は 80 です。

### RTMFP (Real-Time Media Flow Protocol)

#### Flash Player 10、AIR 1.5

このプロトコルは、UDP ベースであり、Flash Player または AIR と Flash Media Server 間のセキュアなネットワーク伝送です。UDP は RTMP より送信の損失があり、待ち時間が少なく、優先順位付けが優れています。RTMFP は、クライアントとサーバー間およびピアツーピアの通信に使用できます。RTMFP を使用すると、クライアントはサーバーを介してメディアをリレーすることなくオーディオおよびビデオを相互にパブリッシュおよび再生することができます。クライアントが相互にデータを渡すためには、Flash Media Server に接続する必要があります。RTMFP を使用して、ピアアシストネットワークワーキングアプリケーションを構築できます。また、RTMFP は従来のブロードキャストアプリケーションにも使用できます。

プロトコルの詳細については、詳細については、アドビシステムズ社のコンピューターサイエンティスト Jozsef Vass の記事「[Benefits of RTMFP](#)」を参照してください。

RTMFP の使用の詳細については、「ピアアシストネットワークワーキングアプリケーションの構築」を参照してください。

RTMFP を使用するためのサーバーの設定の詳細については、「ピアアシストネットワークワーキングの IP アドレスとポートの設定」を参照してください。

### HTTP (Hypertext Transfer Protocol)

#### プログレッシブダウンロード

#### Flash Media Server 3.5

Apache HTTP Server を Flash Media Server とともにインストールする場合は、クライアント SWF ファイル、コンテンツ HTML ページ、および追加のアセットを HTTP 経由で配信できます。

さらに、クライアントサイド **ActionScript** を記述することができます。これにより、**Apache** では、**RTMP** ストリーミングに失敗した場合、**HTTP** プログレッシブダウンロードを通じてメディアアセットを配信することができます。例えば、クライアントで **RTMP** 経由でビデオをストリーミングして失敗した場合、サーバーでは **HTTP** 経由で **RTMP** のトンネリングが試みられます。この試行に失敗した場合、サーバーは **HTTP** 経由でビデオを配信します。

「**Apache HTTP Server** の設定」を参照してください。

## Adobe HTTP Dynamic Streaming および Apple HTTP Live Streaming Flash Media Server 4.5

**HTTP** 経由でコンテンツを配信することを、「**プログレッシブダウンロード**」といいます。コンテンツは、サーバーからクライアントに、ファイルの最初から最後まで連続して転送する必要があります。クライアントが前方の場所をシークするには、その場所およびそれより前のすべてのデータがダウンロードされている必要があります。

**RTMP** 経由でコンテンツを配信することを「**ストリーミング**」といいます。(Flash Media Server などの) メディアサーバーはクライアントに対するソケット接続を作成し、それを通してコンテンツが連続したストリームで送信されます。クライアントは、転送が済んだデータの量に関係なく、コンテンツ内の任意の点をすぐにシークできます。

**HTTP Dynamic Streaming** はこれらの手法を組み合わせます。**HTTP Dynamic Streaming** は、Flash Player クライアントがファイル全体をダウンロードせずにすぐアクセスできるフラグメントに、メディアファイルをパッケージ化します。**Adobe HTTP Dynamic Streaming** に含まれる複数のコンポーネントが連携して、メディアをパッケージ化し、**HTTP** 経由で Flash Player にストリーミングします。これらのコンポーネントでは **Flash Media Server** もインストールされます。

**Flash Media Server 4.0** は、**HTTP Dynamic Streaming** のライブリアルタイムパッケージングをサポートしています。

**Flash Media Server 4.5** では、**Apple HTTP Live Streaming** のライブおよびオンデマンドリアルタイムパッケージングのサポートが追加されています。**HTTP** 上で **Flash Player**、**AIR** および **iOS** デバイスにオンデマンドおよびライブメディアを配信します。

**Flash Media Server 4.5** では、**Adobe HTTP Dynamic Streaming** のオンデマンドリアルタイムパッケージングのサポートも追加されています。

「**Adobe HTTP Dynamic Streaming** および **Apple HTTP Live Streaming** 用の **Apache** の設定」、「**ライブメディアのストリーミング (HTTP)**」および「**オンデマンドメディアのストリーミング (HTTP)**」を参照してください。

## サポートされているファイル形式とコーデック

ファイル形式とコーデックの違いを理解しておくことは重要です。ファイル形式は、1つ以上のメディアが格納される**コンテナ**です。**F4V** はファイル形式です。**コーデック**は、メディアの形式を示します。**H.264** はコーデックです。コンテナには複数のメディアを含めることができます。例えば、**F4V** ファイルには **AAC** コーデックを使用するオーディオと **H.264** コーデックを使用するビデオを含めることができます。

### ファイル形式

すべてのエディションの **Flash Media Server** が次のファイル形式をストリーミングします。

**FLV** すべてのバージョンの **Flash Media Server** が **FLV** ファイル形式の再生と記録をサポートしています。オンデマンド、リアルタイム **HTTP Dynamic Streaming** は **FLV** 形式をサポートしません。

**F4V (MPEG-4 互換)** **Flash Media Server** バージョン 3 以降は、**MPEG-4 Part 12** のコンテナ形式に格納されている、**Flash Media Server** でサポートされるすべてのオーディオおよびビデオコーデックの再生をサポートしています。**MPEG-4** 形式でサポートされるファイル形式には、**F4V**、**MP4**、**M4A**、**MOV**、**MP4V**、**3GP** および **3G2** があります。

**Flash Media Server 3.5** では、**MPEG-4** 形式での記録に対するサポートが導入されています。**Flash Media Server** でサポートされるコーデックを含むライブビデオストリームおよびライブオーディオストリームは、サーバー上で **F4V** 形式で記録することができます。

**重要:** ファイル内に、Flash Media Server でコーデックがサポートされていないオーディオやビデオの一部が含まれている場合、ストリームのそれらの部分は再生されません。

**MP3** MPEG 層 3 (mp3) ファイル形式は、データ損失圧縮を使用するオーディオファイル形式です。

**RAW** Record and Watch (RAW) ファイル形式では、メディアを、どのバージョンの Flash Player にもストリーミングされる設定可能なブロックに記録します。RAW ファイル形式を使用して、パフォーマンスに関する問題を引き起こさずに、長い複数のビットレートに対応する DVR ストリームを提供します。RAW ファイル形式は、H.264 ビデオ、データのみ、オーディオのみなどを含む Flash Media Server がサポートするすべてのストリームを記録および再生します。

**F4M** Flash Media マニフェストファイルフォーマットでは、Flash メディアアセットについての情報が含まれます。マニフェスト情報には、メディアの場所、コーデック、解像度、複数のビットレートでエンコードされたファイルの可用性、DRM 認証情報などが含まれます。HTTP Dynamic Streaming 向けの F4M ファイル形式を使用します。

**F4F** F4F ファイル形式では、メディアコンテンツをセグメントとフラグメントに分割する方法が説明されています。各フラグメントには、キャッシュ管理と高速シークを提供する固有のブートストラップ情報があります。HTTP Dynamic Streaming 向けの F4M ファイル形式を使用します。

コーデックがサポートするファイル形式、サーバーバージョン、および Flash Player のバージョンを確認するには、次の表を使用します。

ファイル形式	コーデック	Flash Media Server バージョン	Flash Player バージョン
FLV	Sorenson Spark	1	6; AIR 1; Flash Lite 3
FLV	Nellymoser	1	6; AIR 1; Flash Lite 3
FLV	On2 VP6	1	8; AIR 1; Flash Lite 3
FLV	Speex	3	10; AIR 1.5
MPEG-4 : F4V、MP4、M4V、3GPP	H.264*	再生 : 3、記録 : 3.5	9,0,115,0; AIR 1
MPEG-4 : F4V、MP4、M4V、3GPP	AAC+/HE-AAC/AAC v1/AAC v2	再生 : 3、記録 : 3.5	9,0,115,0; AIR 1
MPEG-4 : F4V、MP4、M4V、3GPP	MP3	再生 : 3、記録 : 3.5	9,0,115,0; AIR 1
MPEG-4 : F4V、MP4、M4V、3GPP	On2 VP6	再生 : 3、記録 : 3.5	9,0,115,0; AIR 1
mp3	mp3	1	6; AIR 1; Flash Lite 3
RAW	Flash Player でサポートされるすべてのコーデック。	3.5.3	6; AIR 1; Flash Lite 3
F4M	N/A	4.0	10.1, AIR 2
F4F	Flash Player (mp3 と Speex を除く) でサポートされるすべてのコーデック。	4.0	10.1, AIR 2

**注意:** Flash Player での H.264 再生では、Base、Main、HiP などのほとんどのプロファイルがサポートされます。F4V 形式は、MPEG-4 ISO 14496-10 および AAC+ (ISO 14496-3) のサブセットです。

#### コーデック

Flash Media Server では、オーディオおよびビデオ情報をエンコードしたり、デコードしたりしません。Flash Media Server では、既にエンコードされたメディアをストリーミングします。

オンデマンドメディアをエンコードするには、ターゲットの Flash Player または AIR のバージョンをサポートするコーデックを使用します。

ライブビデオのキャプチャ、エンコードおよびストリーミングを行う場合は、Flash Media Live Encoder を使用するか、ActionScript を使用して独自の Flash Player または AIR クライアントを構築することができます。Flash Media Live Encoder 2.5 以降では、ビデオを On2 VP6 コーデック または H.264 コーデックでエンコードします。Flash Player と AIR は、Sorenson Spark コーデックを使用してライブビデオをエンコードし、独自仕様の Nellymoser のコーデックを使用してライブオーディオをエンコードします。

次の表に、サポートするコーデックと、必要とされる最も古い SWF ファイル形式および Flash Player のバージョンを示します。

コーデック	オーディオまたはビデオ	SWF ファイル形式のバージョン (サポートされる最も古いパブリッシュバージョン)	Flash Player のバージョン (再生に必要なバージョン)
Sorenson Spark	ビデオ	6	6、Flash Lite 3, AIR 1
Nellymoser	オーディオ	6	6、Flash Lite 3, AIR 1
mp3	オーディオ	6	6、Flash Lite 3, AIR 1
On2 VP6	ビデオ	8	8、Flash Lite 3, AIR 1
H.264 (MPEG-4 Part 10)	ビデオ	9	9 Update 3、AIR1
AAC (MPEG-4 Part 3)	オーディオ	9	9 Update 3、AIR1
Speex	オーディオ	10	10、AIR 1.5

**注意：**Flash Player と AIR がアルファチャンネルをサポートするのは On2 VP6 コーデックの場合のみです。

Flash Player 9 Update 3 (9, 0, 115, 0) は、次の MPEG-4 ファイル形式標準のサブセットの再生をサポートしています。

MPEG-4 標準	Flash Player 9 アップデート 3/AIR のサポート
ISO/IEC 14496-3 (Audio AAC)	AAC Main、AAC LC、SBR。これらのコーデックは HE-AAC としても知られています。
ISO/IEC 14496-10 (Video AVC)	Base (BP)、Main (MP)、High (HiP)。すべてのレベルがサポートされます。
ISO/IEC 14496-12 (コンテナ)	オーディオトラック 1 つ、ビデオトラック 1 つ
3GPP TS 26.245 (Timed Text Format)	フルサポート

## サポートされるメタデータ形式

### Flash Media Server 3.5

Adobe Extensible Metadata Platform (XMP) メタデータを埋め込んだビデオストリーミングを、Flash Media Server を通じて Flash Player に配信するようになってきました。XMP は、ファイルに関するデータをファイル自体に埋め込むことができるラベリング技術です。XMP メタデータは、重要なメディア情報を、メディアが作成された位置からメディアが表示される位置まで通信するシステムです。XMP により、アプリケーションやパブリッシュシステムでメタデータをキャプチャして共有し、活用することができます。Flash Media Server では、メディアファイル内に埋め込まれているすべてのメ

タデータが正確に配信されます。さらに、ファイル内に埋め込まれ、Adobe Media Encoder CS4 からエンコードされた speech-to-text メタデータも配信することができます。XMP の詳細については、[www.adobe.com/go/learn\\_fms\\_xmp\\_jp](http://www.adobe.com/go/learn_fms_xmp_jp) (英語) を参照してください。

## 言語ライブラリ

### クライアントサイド ActionScript API

クライアントサイドスクリプトは、Flash Player、AIR または Flash Lite で実行されます。Adobe Flash または Adobe Flex を使用して、ActionScript でサーバーリソースにアクセスするクライアントサイドスクリプトを記述することができます。例えば、ライブのオーディオとビデオをキャプチャし、それをサーバーにストリーミングして、そのサーバーから、接続されているすべてのクライアントにストリーミングします。

開発者はクライアントサイドスクリプトで、任意の Flash Player クラスを使用できます。Flash Media Server リソースにアクセスするには、次のクラスを使用します。

**Camera** Flash Player または AIR を実行するコンピューターに接続されているカメラからビデオをキャプチャします。NetConnection クラスおよび NetStream クラスは、Flash Media Server にビデオを送信するために使用します。Flash Media Server は、他のサーバーにビデオを送信し、Flash Player、AIR または Flash Lite を実行する他のクライアントにブロードキャストできます。

**GroupSpecifier** GroupSpecifier クラスは、NetStream および NetGroup コンストラクターに渡される不透明な groupspec ストリングを構築するために使用されます。groupspec は、groupspec を使用するメンバーの機能、制限、および権限を含めた RTMFP ピアツーピアグループを指定します。

**Microphone** Flash Player または AIR を実行するコンピューターに接続されているマイクからオーディオをキャプチャします。NetConnection および NetStream クラスは、Flash Media Server にオーディオを送信するために使用します。Flash Media Server は、他のサーバーにオーディオを送信し、Flash Player、AIR または Flash Lite を実行する他のクライアントにブロードキャストできます。

**NetConnection** クライアントと Flash Media Server の間、または Flash Media Server と Flash Remoting の間に開かれる双方向の接続を表します。

**NetGroup** RTMFP グループ内のメンバーシップを表す NetGroup クラスのインスタンス。

**NetStream** NetConnection オブジェクトによって使用可能になった接続を介して、クライアントと Flash Media Server の間に開かれる単方向のストリームを表します。

**SharedObject** 複数の SWF ファイル間でデータを共有します。Flash Media Server の複数のアプリケーションインスタンス間でもデータを共有できます。

**Video** Video クラスは、ビデオを SWF ファイルに埋め込むことなく、ライブビデオまたは録画ビデオをアプリケーションに表示します。Video オブジェクトは、アプリケーションの表示リスト上の表示オブジェクトであり、ユーザーインターフェイスで実行されるビデオのビジュアルスペースを表します。

詳細については、「[Adobe Flash Platform 用 ActionScript 3.0 リファレンスガイド](#)」を参照してください。

### サーバーサイド ActionScript API

Flash Media Enterprise Server、Flash Media Interactive Server、および Flash Media Development Server は、サーバーサイド ActionScript へのアクセスを提供します。サーバーサイドのコードを記述すると、ログインプロシージャの制御、他のサーバーへのコンテンツの再パブリッシュ、サーバーリソースへのユーザーアクセスの許可や拒否、ユーザーによる情報の更新と共有の許可などを行うことができます。

サーバーサイド **ActionScript** は、JavaScript 1.5 のアドビ システムズ社での名前で、ActionScript 1.0 に似ていますが、同じものではありません。どちらの言語も ECMAScript Language Specification Edition 3 に基づいていますが、ActionScript 1.0 には仕様が正確に実装されていませんでした。サーバーサイド ActionScript は、Flash Media Server に埋め込まれている Mozilla SpiderMonkey エンジンで実行されます。

サーバーサイド ActionScript には次のクラスがあります。

**Application** サーバーサイドスクリプト内でアプリケーションのインスタンスを表す Singleton クラスです。Application クラスのイベントハンドラーを使用して、アプリケーション内のコードのフローを制御し、接続の受諾、拒否またはダイレクトを行います。

**ByteArray** バイナリデータの読み取り、書き込み、および操作を最適化するメソッドおよびプロパティを提供します。

**Client** アプリケーションに接続されたクライアントを表します。サーバーはクライアントが接続するたびに Client オブジェクトを作成します。このクラスは、クライアントに関する情報の取得、サーバーリソースに対する読み取りおよび書き込みアクセスの設定およびリモートメソッドの呼び出しのために使用します。

**File** アプリケーションがサーバーのファイルシステムに書き込めるようにします。このクラスは、データベースを使用しない情報の格納、デバッグで使用するログファイルの作成または使用状況の追跡のために使用します。

**GroupSpecifier** RTMFP ピアツーピアグループの機能、制限事項、権限を定義します。GroupSpecifier オブジェクトを NetGroup と NetStream コンストラクターに渡し、ピアツーピアグループを作成してオーディオとビデオをマルチキャストします。

**GroupControl** GroupControl クラスは、レポートピアと参加したグループの間の関連付けを表します。

**LoadVars** このクラスは、HTTP を介して、オブジェクト内の変数を指定の URL に送信したり、指定した URL にある変数をオブジェクトにロードしたりするために使用します。

**Log** このクラスは、Web サービスの使用時にログファイルを作成するために使用します。

**MulticastStreamInfo** NetStream オブジェクトの基盤となる RTMFP ピアツーピアおよび IP マルチキャストストリームトランスポートに関連する様々な Quality of Service (QoS) を指定します。

**MulticastStreamIngest** MulticastStreamIngest クラスは、RTMFP マルチキャストストリームを取り込み、マルチキャストデータユニットを Stream オブジェクトが使用できるメッセージに変換するために使用します。

**NetConnection** 同じサーバー上の Flash Media Server とアプリケーションサーバー、別の Flash Media Server、またはアプリケーションインスタンス間の双方向の接続を作成します。

**NetGroup** RTMFP グループ。このオブジェクトを使用して、グループへのメッセージの送信、グループメンバーへの直接のメッセージのルーティング、グループ間でのオブジェクトの複製を行います。同様に、いくつかの NetGroup ヘルパークラスがあります。

**NetStream** NetConnection オブジェクトを使用して、2つの場所にインストールされている Flash Media Server 間に単方向のストリームを開きます。このクラスを使用すると、マルチキャストストリームとして RTMFP グループにソースストリームをパブリッシュすることもできます。

**ProxyStream** サーバーサイド NetConnection オブジェクト上の Flash Media Server から別の Flash Media Server へのストリーミングをプロキシします。DVR 機能を使用する大規模なアプリケーションを構築するには、ProxyStream クラスを使用します。

**SharedObject** データをサーバーに格納し、複数のクライアントアプリケーション間でリアルタイムにデータを共有します。

**SHA256** FIPS 180-2 (Federal Information Processing Standards Publication 180-2) で説明されている SHA-256 (安全なハッシュアルゴリズム) の実装このクラスを使用して、バイナリデータにダイジェスト、または署名を生成します。データの受信者は、自身のダイジェストを計算して元のダイジェスト値と比較し、バイナリデータが改ざんされていないことを確認します。

**SOAPCall** すべての Web サービス呼び出しから返されるオブジェクトタイプです。

**SOAPFault** WebService.onFault() や SOAPCall.onFault() に返されるエラーオブジェクトのオブジェクトタイプです。

**Stream** このクラスは、ストリームの管理や再パブリッシュのために使用します。例えば、**Stream** クラスを使用してサーバーサイドの再生リストを作成します。このクラスを使用すると、ストリームをサブスクライブしたクライアントにデータを送信したり、ライブストリームにメタデータを追加したりすることもできます。

**WebService** このクラスは、WSDL/SOAP Web サービスの作成と同 Web サービスへのアクセスのために使用します。

**XML** このクラスは、XML ドキュメントツリーのロード、解析、送信、作成および操作を行うために使用します。

**XMLSocket** このクラスは、Flash Media Server が IP アドレスまたはドメイン名で識別される別のサーバーと通信できるようにするクライアントソケットを実装するために使用します。

**XMLStreams** データを断片的に送受信する XMLSocket クラスの一種です。

詳細については、『[サーバーサイド ActionScript リファレンスガイド](#)』を参照してください。

## プラグイン API

この Flash Media Server Plug-in API により、開発者はサーバーの機能を拡張するプラグインを C++ で記述することができます。

**Access プラグイン** サーバーがクライアントからの接続要求を受諾する前に、セキュリティの層を追加します。**Access** プラグインは、各接続要求がサーバーのスクリプト層に到達する前に、それらの要求を検査します。このプラグインでは、サーバーで要求を受諾するか拒否するかが、現在の接続数などのサーバーの統計に基づいて決定されます。**Access** プラグインをコーディングして、ストリームや共有オブジェクトなどのサーバーリソースに対する権限を設定することができます。また、**Access** プラグインをコーディングして、アプリケーションへのクライアント接続を特定のコアプロセスに割り当てることもできます。

**Authorization プラグイン** サーバー上で発生するイベントに対するクライアントのアクセスを許可します。

**Authorization** プラグインを使用して、サーバーへの接続、ストリームの再生およびストリームのパブリッシュを許可します。論理 URL の物理的な場所へのマッピングを行うこともできます。**Authorization** プラグインは、クライアントの地理的な場所、サブスクリプションのレベル、またはクライアントがストリームを表示している時間に従ってコンテンツを配信することができます。Flash Media Server 3.5 以降では、送受信されたバイト数、送受信された RTMP メッセージ数、ドロップされた RTMP メッセージ数などのクライアントの統計にアクセスすることができます。

**File プラグイン** ファイル I/O メカニズムとサーバーの間のインターフェイスを提供します。開発者はデフォルトのオペレーティングシステムに基づくファイルシステムの代わりに、代替のファイル I/O システムを実装することができます。**File** プラグインインターフェイスは、ファイル操作の完全に非同期的なモデルをサポートしているため、ネットワークベースのファイル I/O を簡単に実装できます。Flash Media Server 3.5 以降では、**File** プラグインがストリームおよび SWF ファイルへの操作をサポートしています。

詳細については、「[プラグインの開発](#)」および「[Adobe Flash Media Server プラグイン API 参照](#)」を参照してください。

## Administration API

**Administration API** を使用すると、RTMP 経由では Flash Player または AIR クライアントから、HTTP 経由では Web クライアントからサーバーの監視、管理および設定を行うことができます。**Administration API** では以下のことが可能です。

- サーバーと、そのアプリケーションおよびサービスを監視します。
- 管理ユーザーの追加、サーバー、仮想ホストおよびアプリケーションの起動と終了など、管理タスクを実行します。
- サーバー設定キーの値を表示して設定します。

一部のメソッドはサーバー管理者のみが使用できます。仮想ホストの管理者はそれらのメソッドを使用できません。ただし、仮想ホスト管理者でも制限付きでメソッドを使用できる場合もあります。このような制限については、各メソッドの項目で説明しています。詳細については、『[Adobe Flash Media Server Administration API リファレンス](#)』を参照してください。

## サーバーの規模の調整

### サーバーのクラスタのデプロイメント

Flash Media Server の規模を調整すれば、サポートの対象となる接続数が増加し、ストリーミング用と、多方向アプリケーション用のハードウェア容量が増加します。Flash Media Server の規模を調整するため、オリジンサーバーのクラスタをデプロイすることができます。Flash Media Streaming Server を除くすべてのエディションでは、セクション 20 ページの「[エッジサーバーのデプロイメント](#)」で説明しているようにエッジオリジンサーバークラスタを設定することもできます。

クラスタ化では、デプロイメントを簡単に行えるように、アセットをローカルで管理することによって 1 台のサーバーのパフォーマンスを向上させます。クラスタ内では、クライアントは共通の URL から接続を要求し、ハードウェアロードバランサまたはサーバーサイドスクリプトのいずれかによって、接続を Flash Media Server に送ります。

詳細については、「[クラスタへのサーバーのデプロイメント](#)」を参照してください。

#### ロードバランスへのハードウェアの使用

クライアントが Flash Media Server アプリケーションへの接続を要求したときに、ハードウェアロードバランサは接続をサーバーに送ることができます。ハードウェアロードバランサは、クライアントの負荷を複数のサーバーに分散するためのさまざまな方法を提供します。この方法は、クライアントが相互に通信する必要がない場合に最適です（例えば、テキストやビデオのチャットアプリケーションで通信するような場合）。

#### ロードバランスへのサーバーサイド ActionScript の使用

**注意：**Flash Media Streaming Server では、サーバーサイド ActionScript にアクセスできません。

例えば、接続を特定のサーバーにルーティングしたり、ロードバランシングを実行したりする必要がある多方向アプリケーションなどで、サーバーサイドスクリプトを記述し、外部からの接続をクラスタ内の特定のサーバーにリダイレクトすることができます。クライアントをサーバーに割り当てる方法をいくつかの中から選択することができます。例えば、クライアントをランダムに割り当てるか、Administration API の `getServerStats()` メソッドを呼び出して負荷の最も低いサーバーを特定することができます。

### エッジサーバーのデプロイメント

**注意：**Flash Media Streaming Server をエッジサーバーとして設定することはできません。

どのエディションのサーバーにも接続や帯域幅の制限はありませんが、すべてのサーバーエディションがハードウェア容量の制限を受けます。オリジンサーバーへの接続が発生するたびに、その接続を通るデータフローとは別に、リソースが消費されます。接続の数が増えれば、このような負荷が非常に大きくなり、結果としてサーバーのパフォーマンスを低下させることとなります。

この制限を克服し、サーバーに同時に接続できるユーザーの数を増やすため、サーバーをリバースプロキシまたはエッジサーバーとして実行されるように設定できます。エッジサーバーを利用することによって、クライアントは、オリジンサーバーに接続する代わりに、エッジサーバーに接続します。エッジサーバーは多数のクライアントからの要求を集約し、それらをオリジンサーバーに送信します。

エッジサーバーは、単方向のビデオオンデマンド (VOD)、単方向のライブイベントおよび 24 時間 365 日停止しない単方向のライブストリームに適したソリューションです。エッジサーバーは大容量の単方向ストリーミングのために、セキュリティを強化し、接続要求の負荷を分散させて、帯域幅とシステムリソースを節約します。

エッジサーバーは、接続の管理、コンテンツのキャッシュへの格納およびクライアントへのデータのプッシュ送信を行います。エッジサーバーにキャッシュされたアセットがあれば、サーバーが保存場所にアクセスする必要性が低下し、ビデオはクライアントによりすばやく配信されます。保存場所へのアクセスは、大規模なメディア配信のボトルネックになる可能性がある処理です。コンテンツは、メモリ内やローカルの保存場所にもキャッシュできます。サーバー管理者は、キャッシュ

が使用するメモリ量に制限を設定することで、キャッシュのサイズを制御できます。また、クリアされるキャッシュファイル数やファイル数などの LRU (Least-Recently-Used) 設定を指定することで、キャッシュをクリアする方法を制御することもできます。

**注意：**サーバーサイド ActionScript はエッジサーバーではなくオリジンサーバーで実行されます。

詳細については、「エッジサーバーのデプロイメント」を参照してください。

### 地域別のエッジサーバーのデプロイメント

エッジサーバーは個別にデプロイすることも、クラスタ内にデプロイすることもできます。また、エッジサーバーをチェーン構成にし、あるエッジサーバーが他のエッジサーバーとそれらのクライアントからの接続要求を集約してからオリジンサーバーにその要求を送るという方式にすることもできます。

ネットワークリソースとシステムリソースの需要を分散させるため、管理者はある地域に属するクライアントを特定のエッジサーバーに割り当てることができます。例えば、あるエッジサーバーは東京のクライアントからの要求を集約して転送し、別のエッジサーバーはパリからの要求を集約して転送するというように設定できます。パリと東京のエッジサーバーはそれぞれのクライアントからの要求を集約し、シカゴなどの別の場所に置かれているオリジンサーバーに、集約した要求を転送します。

パリと東京のクライアントは、それぞれの地域に割り当てられたエッジサーバーを必ず経由して、オリジンサーバーにアクセスします。これらのエッジサーバーはオリジンサーバーからの応答を受信して、対応する各地域、つまりパリや東京のクライアントに応答を配信します。さらに、エッジサーバーは、オリジンサーバーから受信したデータをキャッシュに格納し、そのエッジサーバーに接続している他のクライアントもそのデータを使用できるようにします。このようなデータの再利用も、エッジサーバーで効率的にリソースを使用する方法として挙げることができます。この方法では、コンテンツをキャッシュすることで、オリジンサーバー全体の負荷が低下します。

### 2層でのエッジサーバーのデプロイメント

さらに負荷を分散させるため、エッジサーバーをホスト層と配布層の2層に重ねることができます。例えば、全世界にコンテンツをブロードキャストするオリジンサーバーがロンドンにあるとします。オリジンサーバーは、例えばニューヨーク、ミュンヘン、カラカスおよび東京にあるホストエッジサーバーの層にデータをプッシュ配信します。それぞれのホストエッジサーバーは、配布層を構成するエッジサーバーのクラスタに順次接続します。

## ピアアシストネットワークングアプリケーションの規模の調整

### Flash Media Server 4.5

Flash Media Server は、RTMFP クライアントを互いに紹介し、クライアントが互いに直接接続できるようにします。この直接接続は、ピアツーピア接続と呼ばれます。Flash Media Server 4.0 は、クライアントが1台のサーバーに接続している場合のみ、クライアントを互いに紹介することができます。クライアントが個別のサーバーに接続されている場合でも、Flash Media Server 4.5 に追加されているサーバーサイド ActionScript API を使用して、クライアントを互いに紹介します。サーバー間に紹介を配信すると、ピアアシストネットワークングアプリケーションの規模を調整することができます。

サーバー間のピア紹介の配信を参照してください。

## サーバーの設定

### 仮想ホスト用のサーバーの設定

サーバーは、サーバー、アダプター、仮想ホスト（「**vhost**」とも呼ばれます）およびアプリケーションという階層レベルに分けられます。サーバーは最上位のレベルにあり、1つ以上のアダプターを含む場合があります。各アダプターは1つ以上の仮想ホストを含みます。仮想ホストはそれぞれ、1つ以上のアプリケーションを含む場合があります。アダプターや仮想ホストを追加すると、複数のアプリケーションやサイトのホスティングのためにサーバーを編成できます。

各階層レベル（サーバー、アダプター、仮想ホストおよびアプリケーション）には個別の設定パラメーターがあり、それぞれ `Server.xml`、`Adaptor.xml`、`Vhost.xml`、`Application.xml` という名前の XML ファイルに格納されています。最も重要な設定パラメーターは `fms.ini` ファイルに取り込まれているため、1つのファイルを使用してサーバーを設定することができます。サーバーのレベルをそれぞれ調整して、サーバーがホスティングするアプリケーションごとに最も高品質のサービスを提供できます。

詳細については、「[仮想ホスト用のサーバーの設定](#)」を参照してください。

### サーバーのパフォーマンスチューニング

パフォーマンスをチューニングするには、`RootInstall/conf` に格納されている XML 設定ファイルを編集します。サーバーをチューニングして、配信するアプリケーションの種類で可能な最大限のパフォーマンスを実現することができます。サーバー、アダプター、仮想ホストなどの各サーバーレベルをチューニングすることもできます。記録済みメディアのキャッシュのサイズ、ストリームチャンクのサイズ、オーディオサンプルの結合方法、接続制限の有無などを設定します。

詳細については、「[パフォーマンス機能の設定](#)」を参照してください。

### コンテンツの管理

次のいずれかの方法で、Flash Media Server によって配信されるコンテンツを管理することができます。

- 「[ネットワークドライブへのディレクトリのマッピング](#)」。
- アプリケーションファイルの場所の設定
- 「[記録されたストリームと共有オブジェクトの場所の設定](#)」。
- `Authorization` プラグインを使用して、物理的な場所に論理 URL をマップします。「[Authorization プラグインのコードの記述](#)」を参照してください。
- ストリームと共有オブジェクトを格納するための仮想ディレクトリを作成する。「[仮想ディレクトリと物理ディレクトリ間のマッピング](#)」を参照してください。
- `File` プラグインを使用して HTTP 経由でコンテンツにアクセスします。「[File プラグインの開発](#)」を参照してください。
- `File` プラグインを使用してカスタムのファイル I/O システムを作成します。「[File プラグインの開発](#)」を参照してください。

## サーバーの管理

### Administration Console

Administration Console（Administration API を使用して構築されています）は、管理者の管理、サーバーの監視およびサーバー上で実行されているアプリケーションに関する詳細情報の表示を行うための、使いやすいインターフェイスを備えています。サーバー管理者と仮想ホスト管理者という2つのレベルの Administration Console ユーザーが存在します。こ

れにより、ホスティングを提供する側の組織は、ホスティングサービスを利用している企業に提供する管理上の権利をより少なくすることができます。アプリケーション開発者が **Administration Console** を使用して、アプリケーションをデバッグします。

詳細については、「[Administration Console の使用](#)」を参照してください。

## コマンドラインツール

アドビ システムズ社は、サーバーの管理に役立つ **FMSCheck** と **FLVCheck** という 2 つのコマンドラインツールを提供しています。**FMSCheck** は、サーバーが実行されているかどうか、応答時間はどのくらいか、どの **fmscore** プロセスが応答していないかなどの情報を提供します。**FLVCheck** では、ビデオファイルが **Flash Media Server** で正しく実行されることを検証できます。**FLVCheck** ツールは MP4 ファイルと FLV ファイルをサポートしています。

「サーバーステータスの確認」と「ビデオファイルの確認」を参照してください。

## ログファイル

**Flash Media Server** のログでは、リアルタイムのサーバー監視が行われるので、サーバーの管理と問題のトラブルシューティングに役立ちます。ログファイルでは、トラフィック全般やサーバー負荷などの状況、サーバーにアクセスしているユーザー、クライアントの動作とやり取りおよび全般的な診断が追跡されます。ログファイルは、W3C 形式で記述されます。ログファイルの解析には標準的な解析ツールを使用できます。ログのデータは **Administration Console** でも表示できます。

**Flash Media Server** では、**RootInstall/logs** フォルダー内の次のログが管理されます。

**access.xx.log** サーバーのアクセスに関する情報を追跡します。例えば、クライアントがサーバーに接続した日付と時間、セッション中に消費された帯域幅の合計などです。接続によってアクセスされたストリームや、クライアントがストリームをパブリッシュしたかどうかも追跡します。さらに、クライアントが記録済みストリーム内の新しい場所にジャンプしたかどうかも追跡します。

**application.xx.log** アプリケーションインスタンスの動作に関する情報を追跡します。例えば、イベントの日付と時間や、イベントのサーバープロセスの ID などです。また、イベントステータスレベル（警告、エラー、情報、デバッグなど）も追跡します。

**httperror.xx.log** Apache HTTP Server エラーに関する情報を追跡します。このファイルは **RootInstall/Apache2.2/logs** ディレクトリにあります。

**httpaccess.xx.log** Apache HTTP Server アクセスに関する情報を追跡します。このファイルは **RootInstall/Apache2.2/logs** ディレクトリにあります。

**診断ログ** サーバーの動作に関する情報が追跡されます。例えば、ストリームイベント、アプリケーションインスタンス、仮想ホスト、エッジサーバーとオリジンサーバーの問題に関する情報などです。デフォルトでは、サーバーはサーバープロセスのタイプごとに 1 つの診断ログを作成するように設定されています。デフォルトの診断ログは、**master.xx.log**、**edge.xx.log**、**core.xx.log**、**admin.xx.log** および **httpcache.xx.log** です。

詳細については、「[ログファイルの監視と管理](#)」を参照してください。

## 第3章：セキュリティ

すべてのエディションの Adobe Flash Media Server は、コンテンツの盗難や誤用を防止する機能をサポートしています。真のストリーミングなどの一部の機能はサーバーにもともと備わっており、設定を行う必要はありません。拡張 RTMP (RTMPE) などの他の機能は、XML 設定ファイルを使用して設定したり、無効にしたりすることができます。特定のサーバーフォルダーに対する読み取りおよび書き込みアクセスの制御など、他の機能をクライアントサイド ActionScript とサーバーサイド ActionScript を使用して独自に構築することもできます。

**重要：**Flash Media Server セキュリティへの最終ガイドについては、Adobe Developer Center の「[Hardening guide for Flash Media Server](#)」を参照してください。このガイドの著者は、Adobe Flash Media Server エスカレーションとセキュリティエンジニアリングマネージャーの Asa Whillock です。

### コンテンツのセキュアなストリーミング

#### 真のストリーミング

アプリケーションがプログレッシブダウンロード（一般「プログレッシブストリーミング」と呼ばれます）を使用している場合、コンテンツは、悪意のあるエージェントがビデオをキャプチャして再配布する可能性があるクライアントのハードドライブにダウンロードされます。Flash Media Server は、プログレッシブダウンロードではなく真のストリーミングを使用します。これは、Flash Media Server から Adobe Flash Player にストリーミングされるメディアが、クライアントのキャッシュやクライアントのハードドライブ上のどこにも、ローカルでは保存されないことを意味します。設定を行う必要はなく、Flash Media Server は常に真のストリーミングテクノロジーを使用してコンテンツを保護します。

#### セキュアなネットワークプロトコル

Flash Media Server は、異なるレベルのセキュリティを提供する 2 つのネットワークプロトコルをサポートしています。組織とアプリケーションのニーズに最適なネットワークプロトコルを選択してください。

**RTMPE** クライアントとサーバーの間のデータについて、128 ビットの暗号化チャンネルが使用されます。証明書の管理は使用されません。エンドポイントの認証を必要とせず、SSL で可能なレベルより高いパフォーマンスと速い速度が必要なアプリケーションに最適です。RTMPE は、RTMP より 15% 高い処理能力を必要とします。暗号化チャンネルを指定するには、その接続 URI で、例えば次のように RTMPE プロトコルを使用します。  
`nc.connect("rtmpe://www.example.com/mediaApplication")`

RTMPE を介して渡されるデータは、よく知られたストリーム暗号化アルゴリズムで暗号化されています。このアルゴリズムは、受動的観察を防ぎ、Perfect Forward Secrecy を提供するよく知られた公開鍵交換アルゴリズムを使用して鍵をかけます。

**注意：**Flash Player 9 Update 3 以降および AIR 1.0 以降は、RTMPE をサポートしています。

**RTMPS** TCP/IP 上でセキュアな通信を可能にするプロトコルです。Flash Media Server では RTMPS プロトコルを使用して、外部からと外部への両方の SSL 接続をネイティブでサポートしています。SSL によりドメインの偽装から保護できます。また、必要な暗号化のレベルを選択することができます。SSL は、最高レベルのセキュリティを提供することも可能ですが、RTMP より約 50% 高い処理能力が必要になります。RTMPS 接続を指定するには、その接続 URI で、例えば次のように RTMPS プロトコルを使用します。`nc.connect("rtmps://www.example.com/mediaapplication")`

RTMPS を介して渡されるデータは、SSL の保証により保護されています。サーバー証明書確認では、サーバーが有効であることを認証しますが、クライアントが有効な Flash Player または AIR クライアントであることは認証しません。「[SSL の設定](#)」を参照してください。

**RTMFP** 128 ビットの暗号化を使用します。RTMFP を介して渡されるデータは、よく知られたブロック暗号化アルゴリズムで暗号化されています。アルゴリズムは、受動的観察を防ぎ、**Perfect Forward Secrecy** を提供するよく知られた公開鍵交換アルゴリズムを使用して鍵をかけます。セッションノンスが提供されて鍵交換に関連付けられるので、交換された値の上部でクライアントとサーバーの認証を構築できます。

ピアツーピア接続を確立するには、サブスクリブするピアがパブリッシュするピアのピア ID を認識する必要があります。ピア ID は、Flash Player の各インスタンスに対して一意である必要があります。ピア ID は、ランダムな暗号のソースから作成される 256 ビットの値です。ピア ID を使用して RTMFP ピアが別の RTMFP ピアに接続すると、公開鍵交換アルゴリズムがそのピア ID に関連付けられるので、**man-in-the-middle** 攻撃を行うことが不可能になります。RTMFP によって提供されるその他の保証もすべて整っています。

RTMFP 接続を指定するには、その接続 URI で、例えば次のように RTMFP プロトコルを使用します。  
`nc.connect("rtmfp://www.example.com/mediaapplication")`

**セッションノンス** Flash Player 10、Flash Media Server 3.0.3 および 3.5.1 では、RTMFP および RTMPE 用のセッションノンスのサポートが追加されました。ノンスは鍵交換に関連付けられるので、交換された値の上部でクライアントとサーバーの認証を構築できます。

Flash Player では、NetConnection オブジェクト (nearNonce および farNonce) のプロパティとしてノンスを使用できます。Flash Media Server では、サーバーサイド ActionScript の Client オブジェクト (nearNonce および farNonce)、サーバーサイド ActionScript の NetConnection オブジェクト (nearNonce および farNonce)、Access プラグイン (CONNECT イベント上の nearNonce および farNonce フィールド)、Authorization プラグイン (Client オブジェクトを持つ任意の許可イベントの F\_CLIENT\_NEAR\_NONCE フィールドおよび F\_CLIENT\_FAR\_NONCE フィールド) でノンスを使用できます。

## コンテンツの保護

### 保護された HTTP Dynamic Streaming および HTTP Live Streaming

Flash Media Server 4.5

Protected HTTP Dynamic Streaming を使用して、DRM License Server を使用せずに、ライブおよびオンデマンドの保護されたコンテンツを、HTTP 経由で Flash Player と AIR に配信します。保護された HTTP Dynamic Streaming も SWF 検証を差ポートしています。Flash Media Server は、コンテンツをパッケージ化すると、ライセンスを生成し、そのライセンスをコンテンツストリームのメタデータに埋め込みます。

Protected HTTP Live Streaming を使用して、DRM License Server を使用せずに、ライブおよびオンデマンドの保護されたコンテンツを、iOS と MacOS に配信します。

詳細については、「[Protected HTTP Dynamic Streaming](#)」および「[Protected HTTP Live Streaming](#)」を参照してください。

### Adobe Flash Access の使用

Flash Media Server は Flash Access によって暗号化されたコンテンツを Flash Player および AIR アプリケーションにストリーミングすることができます。Flash Media Server は、暗号化された FLV ファイル、暗号化された MP4/F4V ファイル、および暗号化された HTTP Dynamic Streaming ファイルをストリーミングすることができます。暗号化されたファイルは、サポートされているどのプロトコルを使用してもストリーミングできます。

Flash Media Server が Flash Access で動作するのに特定の設定は不要です。

詳細については、「[Adobe Flash Access](#)」を参照してください。

## クライアントの検証

Flash Media Server では、許可されていないクライアントがサーバーにストリームを送信したり、アクセスを許可されていないストリームを再生したりしないようにする機能をサポートしています。

すべてのサーバーエディションで次の機能をサポートしています。

- SWF ファイルを検証します。  
アプリケーションへの接続を許可する前にクライアントの SWF ファイルを検証するよう、Flash Media Server を設定することができます。この機能により、ユーザーが作成した SWF ファイルだけが特定のサーバーにアクセスできるようになります。リソースをストリーミングしようとする SWF ファイルが第三者によって独自に作成されないようにします。詳細については、「[SWF ファイルの検証](#)」を参照してください。
- 指定したドメインからの接続を許可または拒否します。  
詳細については、「[仮想ホストへの接続を許可するドメインの制限](#)」を参照してください。

次の機能は、Flash Media Streaming Server を除くすべてのエディションでサポートされます。

- SWF ファイルを検証するには、File プラグインを使用します。「[検証のための外部 SWF ファイルの取得](#)」を参照してください。
- サーバーに対して検証される一意の鍵を生成するか、アプリケーションサーバーとの間で暗号化されたトークンの要求と受信を行います。
- サーバーサイド ActionScript Client オブジェクトを使用します。このオブジェクトは、接続要求を受諾または拒否するために使用できる、クライアントに関する情報を提供します。Client.referrer プロパティを使用して、SWF ファイルの URL またはクライアント接続の作成元サーバーの URL をチェックします。Client.ip プロパティを使用して、クライアントの IP アドレスをチェックします。
- Flash Player のバージョンを確認します。

これらの機能の詳細については、『[Adobe Flash Media Server デベロッパーズガイド](#)』の「[アプリケーションのセキュリティ確保](#)」を参照してください。

## サーバーアクセスの制御

開発者と管理者は以下の方法で、クライアントがアクセスできるデータを制御することができます。

**注意：** 次の機能は Flash Media Streaming Server ではサポートされていません。

- サーバーサイド ActionScript の Client.readAccess プロパティと Client.writeAccess プロパティを使用して、共有オブジェクトやストリームなどのサーバーリソースに対するクライアントの読み取りおよび書き込みの権限を指定します。
- サーバーサイド ActionScript の Client.audioSampleAccess プロパティと Client.videoSampleAccess プロパティを使用して、指定したフォルダーのストリームに含まれる圧縮されていない生のデータにクライアントがアクセスできるようにします。
- Authorization プラグインを使用して、ストリームの再生やパブリッシュなど、サーバー上のイベントへのアクセスを許可します。「[Authorization プラグインの開発](#)」を参照してください。

**注意：** 次の機能はすべてのサーバーエディションでサポートされています。

- ストリーム、共有オブジェクトおよびファイル用の仮想保存フォルダーを設定します。詳細については、「[コンテンツの格納場所の設定](#)」を参照してください。
- Access プラグインを使用して、サーバーのスクリプト層に到達する前に接続を受諾、拒否またはリダイレクトします。詳細については、「[Access プラグインの開発](#)」を参照してください。

## ユーザーの認証

ユーザー名やパスワードなどの証明書をクライアントサイドの `NetConnection.connect()` 呼び出しで渡すと、データベース、LDAP サーバーまたはその他のアクセス許可サービスなどの外部リソースに基づいて証明書を検証することができます。詳細については、「[ユーザーの認証](#)」を参照してください。

## サーバーのセキュアな設定

### Administration Console のセキュリティの確保

管理者は以下の方法を使用して、Administration Console に対するアクセスをセキュアなものにします。

- 管理者であるユーザーとそのパスワードを定義します。
- 管理者であるユーザーが必要とする、または管理者であるユーザーに対して許可しない、IP アドレスの範囲またはドメイン名を定義します。「AdminServer」(XML 設定ファイルの章) を参照してください。
- ログイン試行許可数に制限を設定します。「[LoginLimits](#)」を参照してください。
- 管理者用の接続ポートを定義し、ファイアウォールの内側にいるユーザーのみにアクセスを許可します。「[管理者の管理](#)」を参照してください。

### サーバーのパフォーマンスの確保

管理者は以下の方法を使用して、パフォーマンスレベルが維持されるようにサーバーを設定できます。

- サーバーのアクセスとアプリケーションイベントをログに記録します。
- スレッド数の制限、ガベージコレクションの間隔、ストリームの割り当てサイズなどのパフォーマンスパラメーターを設定します。
- アプリケーションインスタンス、ストリームおよび共有オブジェクトに対して制限を設定し、アプリケーションがコンピューター上のすべてのディスクスペースを消費しないようにします。
- アプリケーションの帯域幅の容量を制限し、1 つのアプリケーションがコンピューター上のすべてのネットワーク帯域幅を消費しないようにします。
- アプリケーションに接続するクライアントのデフォルトの帯域幅を設定します。

詳細については、「[パフォーマンス機能の設定](#)」を参照してください。

## サーバーサイドスクリプトのセキュリティ

### スクリプトのメモリ使用量の制限

Application.xml 設定ファイルの ScriptEngine セクションでは、仮想ホストのサーバーサイド ActionScript が使用できるメモリ容量を制限できます。

ガベージコレクションの実行頻度、関数の実行に使用できる最大時間、スクリプトライブラリのパスなど、JavaScript (サーバーサイド ActionScript) エンジン以外の面についても設定が可能です。

設定ファイルの編集の詳細については、「[設定ファイルの操作](#)」を参照してください。スクリプトエンジンの設定の詳細については、「[ScriptEngine](#)」を参照してください。

## スクリプトのセキュアなロード

Flash Media Server のスクリプトセキュリティモデルでは、悪意がある、またはバグを生じる可能性があるサードパーティ製のコードがサーバーサイドに含まれている場合に、それらのコードとの接触を制限することができます。このスクリプトセキュリティモデルでは、サードパーティ製のコード内の無限ループなどのエラー状態を検出したり、その実行を防止したりすることはできません。しかし、任意に接続を作成したり、ファイルオブジェクトに対して読み取りや書き込みを行ったりするような、潜在的な危険のある機能を阻止したり制限したりする場合に役立ちます。

スクリプトのセキュリティは、外部ソースからコードをロードして評価を行う、動的に拡張可能なアプリケーションを構築する場合に非常に重要です。

アプリケーションが開始されると、サーバーはアプリケーションのフォルダーで `secure.asc` ファイルを探します。ファイルが存在する場合は、サーバーによってロードされます。この間に、`protectObject()` メソッドと `getGlobal()` メソッドが使用可能になります。これらのメソッドを使用して、通常のアプリケーションの実行中には不可能な方法で、グローバルな関数、クラスおよびオブジェクトを操作します。`getGlobal()` メソッドは、`secure.asc` ファイルのロード中に、このファイルからグローバルオブジェクトへのアクセスを提供します。`protectObject()` メソッドは、アプリケーションコードがオブジェクトのメソッドに直接アクセスしたり、メソッドを調べたりしないようにします。`protectObject()` メソッドは、`secure.asc` ファイル内でのみ使用できます。サーバーが `secure.asc` ファイルのロードを完了すると、これらのメソッドは使用できなくなります。サーバーは次に、`main.asc` ファイルと他のスクリプトを通常の方法でロードします。

詳細については、「[Adobe サーバーサイド ActionScript リファレンスガイド](#)」を参照してください。

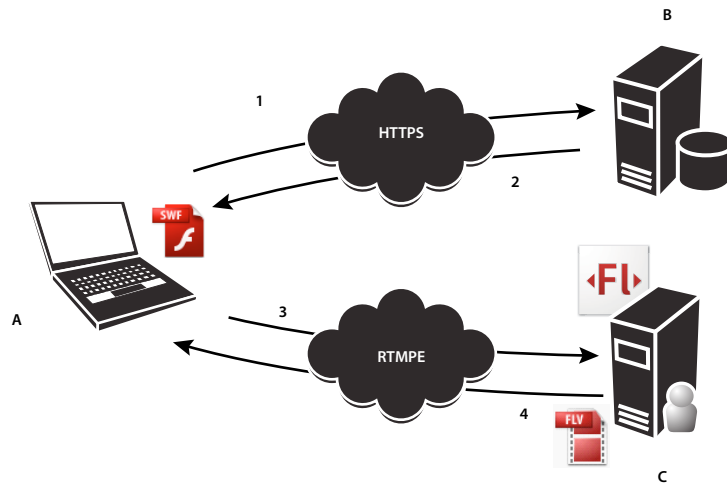
## セキュアなアプリケーションシナリオ

すべてのサーバーエディションに、ビデオストリームを保護するための多数の方法が用意されています。このシナリオの例では、外部認証、暗号化された Real-Time Messaging Protocol (RTMPE) および接続する SWF ファイルを検証する機能を使用しています。

**外部認証** 外部のアプリケーションサーバー (J2EE ベースのサーバーなど) とデータベースに基づいてクライアントを認証します。

**SWF ファイルの検証** 接続する SWF ファイルと、サーバーに格納されている SWF ファイルのコピーを比較して、SWF ファイルを検証します。検証済みの SWF ファイルのみがサーバー上のコンテンツにアクセスできます。

**RTMPE** RTMPE はクライアントとサーバーの間で、128 ビットの暗号化されたデータを送信します。クライアントで実行されている検証済みの SWF が RTMPE 経由で Flash Media Server に要求を送信すると、Flash Media Server がライブコンテンツまたは記録されたコンテンツをストリーミングします。送信中、ストリームは暗号化されます。



このセキュアなビデオアプリケーションは、サーバーにあるコンテンツの要求をクライアントに許可する前に、アプリケーションサーバーを使用してクライアントを認証します。

A. Flash Player または AIR クライアント B. データベースがあるアプリケーションサーバー C. Flash Media Server 1. HTTPS 認証要求 2. 認証の応答 3. コンテンツの RTMPE 要求による SWF 検証のトリガー 4. クライアントへのコンテンツストリーム

**注意：**ビデオをパブリッシュするときは、非常に多くの視聴者がいる可能性があり、ユーザー数が数百万に達する可能性もあります。このような場合、アプリケーションサーバーとデータベースによってユーザーを認証するのは現実的ではありません。パブリッシュを行うときは、RTMPE 経由の接続を許可する前に、外部認証を省き、SWF ファイルの検証のみを行うことができます。