

ADOBE® FLASH® MEDIA SERVER 3.5

Administration API リファレンスガイド

© 2009 Adobe Systems Incorporated. All rights reserved.

Adobe® Flash® Media Server 3.5 Administration API リファレンスガイド

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つ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、実在の組織を示すものではありません。

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

Microsoft, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. All other trademarks are the property of their respective owners.

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

Portions include software under the following terms:

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

Speech compression and decompression technology licensed from Nellymoser, Inc. (www.nellymoser.com)

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

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 Unpublished-rights reserved under the copyright laws of the United States.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

目次

Administration API リファレンス

addAdmin()	1
addApp()	2
addVHostAlias()	3
approveDebugSession()	4
broadcastMsg()	5
changePswd()	6
gc()	8
getActiveInstances()	8
getActiveVHosts()	9
getActiveVHostStats()	10
getAdaptors()	13
getAdminContext()	14
getAdmins()	15
getApps()	16
getAppStats()	17
getConfig()	21
getConfig2()	23
getFileCacheStats()	26
getGroupMembers()	29
getGroupStats()	30
getGroups()	32
getInstanceStats()	33
getIOStats()	36
getLicenseInfo()	38
getLiveStreams()	41
getLiveStreamStats()	42
getMsgCacheStats()	44
getNetStreams()	46
getNetStreamStats()	47
getRecordedStreams()	48
getRecordedStreamStats()	49
getScriptStats()	51
getServerStats()	52
getSharedObjects()	54
getSharedObjectStats()	56
getUsers()	57
getUserStats()	58

getVHosts()	60
getVHostStats()	61
ping()	63
reloadApp()	64
removeAdmin()	65
removeApp()	66
removeVHostAlias()	68
restartVHost()	69
setConfig()	70
setConfig2()	73
startServer()	76
startVHost()	77
stopServer()	78
stopVHost()	79
unloadApp()	80

Administration API リファレンス

Administration API は、Adobe® Flash® Media Server の監視および管理のためのツールの作成に使用します。

注意：以前のバージョンの Flash Media Server では、Administration API は「Server Management API」と呼ばれていました。

Administration API メソッドは、クライアントサイド Adobe® ActionScript® 1.0、ActionScript 2.0 または ActionScript 3.0 から呼び出すことができます。HTTP 経由でメソッドを呼び出すには、Users.xml 設定ファイルに権限を追加し、HTTP 要求を作成します。RTMP または RTMPE 経由でメソッドを呼び出すには、クライアントサイド NetConnection.call() メソッドを呼び出し、Administration API メソッド、応答オブジェクトおよび任意の追加パラメータを渡します。

Administration API の使用方法の詳細については、[Adobe Flash Media Server 設定および管理ガイド](#)の「Administration API の使用」を参照してください。

エントリはアルファベット順に並んでいます。このドキュメントで「RTMP/E」と記述されている場合、RTMP または RTMPE を意味します。

addAdmin()

RTMP/E

```
addAdmin(username:String, password:String [,scope:String]): Object
```

HTTP

```
http://www.example.com:1111/admin/addAdmin?auser=aUsername&apswd=aPassword&user|username=username&pass|password=password[&scope=scope]
```

管理者をシステムに追加します。追加する管理者は、渡すパラメータによってサーバー管理者または仮想ホスト管理者になります。管理者が既に存在している場合、このメソッドは失敗します。存在しない場合は、管理者が追加され、Users.xml ファイルは新しい値で更新されます。仮想ホスト管理者は、特定の仮想ホストフォルダにある Users.xml ファイルに追加されます。

サーバー管理者のみが、サーバー管理者を追加したり、現在接続されている仮想ホスト以外の仮想ホストの管理者を追加したりすることができます。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

user|username 追加する管理者のユーザー名を示す文字列。

pass|password 追加する管理者のパスワードを示す文字列。パスワードは Users.xml 設定ファイルに書き込まれる前にエンコーディングされます。scope パラメータに仮想ホストを指定すると、新しい管理者はその仮想ホストの Users.xml ファイルに追加されます。

aUsername この呼び出しを行っている管理者のユーザー名を示す文字列。

注意：RTMP/E 呼び出しでは、呼び出しを行っている管理者のユーザー名とパスワードは NetConnection.connect() 呼び出しで渡されます。

aPassword この呼び出しを行っている管理者のパスワードを示す文字列。

scope 管理者がサーバー管理者であるか、または仮想ホスト管理者であるか（その場合は、どの仮想ホスト管理者であるか）を示す文字列。サーバー管理者を追加するには、「server」と指定します。接続している仮想ホストに管理者を追加するには、このパラメータを省略します。別の仮想ホストに仮想ホスト管理者を追加するには、追加先の仮想ホストを `adaptor_name/virtual_host_name` の形式で指定します。

注意：パラメータのデータタイプは RTMP/E 呼び出しの場合のみ関連します。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは `level` プロパティの `status` および `code` プロパティの `NetConnection.Call.Success` を含む情報オブジェクトを送信します。呼び出しが失敗すると、サーバーは、`level` プロパティの `error` および `code` プロパティの `NetConnection.Admin.CommandFailed`、またはより具体的な値（存在する場合）を含む情報オブジェクトを送信します。この情報オブジェクトには、呼び出しの日時を示す `timestamp` プロパティも含まれます。

HTTP 呼び出しは次の構造の XML を返します。

```
<result>
  <level>
  <code>
  <timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される `Object` のプロパティと同じ情報が含まれます。

注意：HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

次の 3 つの例は、RTMP 経由の `addAdmin()` メソッドの呼び出しでのパラメータの指定方法を示しています。

```
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "JLee", "x52z49ab");

/* Adds a server administrator named "GLee" with password "boat4907" */
nc_admin.call("addAdmin", new onAddAdmin(), "GLee", "boat4907", "server");

/* Adds a virtual host administrator named "ChrisM" with password "tree2981" */
nc_admin.call("addAdmin", new onAddAdmin(), "ChrisM", "tree2981");

/* Adds a virtual host administrator "DHong" with password "wate3235" */
/* for vhost tree.oak.com */
nc_admin.call("addAdmin", new onAddAdmin(), "DHong", "wate3235", "_defaultRoot/_ tree.oak.com");
```

addApp()

RTMP/E

`addApp(appname:String) : Object`

HTTP

`http://www.example.com:1111/admin/addApp?user=username&pswd=password&app|appname=applicationname`

ディレクトリのツリーに新しいアプリケーションに必要なディレクトリを作成することによって、接続している仮想ホストに新しいアプリケーションを追加します。この新しいアプリケーションのディレクトリが作成されると、ユーザー（または、ファイルシステムへのアクセス権を持つ他の管理者）は、ディレクトリに必要なサーバーサイドスクリプトを追加できます。クライアントサイドのコードでは、この新しいアプリケーションのディレクトリを `NetConnection.connect` の呼び出しの URI パラメータに使用します。

仮想ホストは複数のアプリケーションディレクトリを持つ場合があります。このアプリケーションは、XML 設定ファイルで指定された最初のアプリケーションディレクトリに作成されます。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

app|appname 追加するアプリケーションの名前を示す文字列。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは **level** プロパティの **status** および **code** プロパティの `NetConnection.Call.Success` を含む **Object** を返します。

呼び出しが失敗すると、サーバーは、**level** プロパティの **error** および **code** プロパティの `NetConnection.Admin.CommandFailed`、またはより具体的な値（存在する場合）を含む **Object** を返します。失敗の原因について説明する文字列を含む **description** プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しは次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される **Object** のプロパティと同じ情報が含まれます。存在する場合は、XML に **description** Element が含まれることがあります。

注意：HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

次の例では、接続中の仮想ホストに ChatApp アプリケーションを追加します。

```
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "JLee", "x52z49ab");
nc_admin.call("addApp", new onAddApp(), "ChatApp");
```

addVHostAlias()

RTMP/E

`addVHostAlias(vhost:String, alias:String, persist:Boolean) : Object`

HTTP

`http://www.example.com:1111/admin/addVHostAlias?auser=username&apswd=password&vhost=hostname&alias=alias&persist=value`

仮想ホストにエイリアスを追加します。エイリアスは、Flash Media Server 接続が確立されるときにターゲットとして使用される仮想ホストの代替名です。エイリアスを削除すると、接続が確立されるときにエイリアスを使用できなくなります。

使用できるバージョン

Flash Media Server 2.0

パラメータ

vhost エイリアスを追加する仮想ホストを示す文字列。

alias 指定した仮想ホストに追加するエイリアスを示す文字列。

persist エイリアスの変更を設定ファイルに書き込むかどうかを示すブール値。「true」に設定すると、仮想ホストが次回再起動した後もエイリアスが保持されます。「false」に設定すると、仮想ホストが再起動したときにエイリアスは保持されません。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは `level` プロパティの `status` および `code` プロパティの `NetConnection.Call.Success` を含む `Object` を返します。呼び出しが失敗すると、サーバーは、`level` プロパティの `error` および `code` プロパティの `NetConnection.Admin.CommandFailed` を含む `Object` を返します。

HTTP 呼び出しは次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される `Object` のプロパティと同じ情報が含まれます。

注意：HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

```
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "JLee", "x52z49ab");
nc_admin.call("addVHostAlias", new Result(), "myvhost", "myalias", true);
```

関連項目

[removeVHostAlias\(\)](#)

approveDebugSession()

RTMP/E

`approveDebugSession(appInst:String, PIN:Number) : Object`

注意：このメソッドは HTTP 経由ではサポートされません。

保留のデバッグセッション要求を許可して、選択されたアプリケーションに接続できるようにします。許可されたデバッグセッションは、アプリケーションに接続し、通常のクライアントとして動作できるようになります。デバッグセッションを使用してアプリケーションに接続することにより、Administration Console からアプリケーションのストリームや共有オブジェクトを表示できます。

使用できるバージョン

Flash Media Server 2.0

パラメータ

`applnst` 承認のためにデバッグ接続が保留になっているアプリケーションおよびインスタンスの名前を示す文字列。

`PIN` デバッグセッションの個人識別番号 (PIN) を示す数字。各デバッグ接続では、アプリケーションへの接続がキューに格納されるときにデバッグ番号が発行されます。この番号は、この API に含まれています。この API が処理されるとき、`PIN` が照合され、対応する接続が許可されます。これは、許可されないユーザーがデバッグ接続を実行することを防ぐためのセキュリティ上の対策です。

0 は、有効な `PIN` の値ではありません。セキュリティを確保するため、容易に推測できる数字を `PIN` に設定しないようにします。`PIN` はゼロ以外の数字であり、さらに $\pm (2^{31} - 1)$ の範囲内 (プラス 2147483648 から マイナス 2147483648 の間) である必要があります。特定の `PIN` を持つ接続が保留中となっているときに、同一の `PIN` を持つ別の接続を受信した場合、2 番目の接続はセキュリティ対策のために拒否されます。

戻り値

`RTMP/E` 呼び出しが成功すると、サーバーは `level` プロパティの `status` および `code` プロパティの `NetConnection.Call.Success` を含む `Object` を返します。呼び出しが失敗すると、サーバーは、`level` プロパティの `error` および失敗について説明する `code` プロパティを含む `Object` を返します。

例

有効なコマンドの例を次に示します。

```
nc_admin.connect("rtmp://serverName/appName%3F%5Ffcs%5Fdebugreq%5F%3D1234");  
// The original string is _fcs_debugreq=1234.
```

次に、デバッグ許可要求の例を示します。

```
nc_admin.call("approveDebugSession", null, "appName/instName", 1234);
```

broadcastMsg()

RTMP/E

```
broadcastMsg(scope:String, method:String [, arg0, ..., argN])
```

HTTP

```
http://localhost:1111/admin/broadcastMsg?user=admin&passwd=admin&scope="App:accept"&method="bounceback"&arg0="hello"&arg1="world"
```

特定のクライアント (サーバー、アダプタ、`vhost`、アプリケーション、アプリケーションインスタンス全体など) にメッセージをブロードキャストします。

使用できるバージョン

Flash Media Server 3.5

パラメータ

`scope` ブロードキャスト対象を示す文字列 c 次のいずれかになります。

- "Server" (サーバー全体へのブロードキャスト)
- "VHost" (現在の `vhost` 全体へのブロードキャスト)
- "App:<name>" (現在の `vhost` 上のアプリケーション全体へのブロードキャスト)

- "App:<name>/<instance>" (現在の vhost 上のインスタンス全体へのブロードキャスト)
- "Adaptor[:<name>]" (アダプタ全体への特定のブロードキャスト)
- "Adaptor[:<name>]/VHost[:<name>]" (指定したアダプタ上の vhost 全体への特定のブロードキャスト)
- "Adaptor[:<name>]/VHost[:<name>]/App:<name>" (指定したアダプタおよび vhost 上のアプリケーション全体へのブロードキャスト)
- "Adaptor[:<name>]/VHost[:<name>]/App:<name>/<instance>" (指定したアダプタ、vhost、アプリケーション上のインスタンス全体へのブロードキャスト)

method ブロードキャストのすべての受信先で呼び出されるリモートメソッドを示す文字列。

arg0, ..., argN 位置引数 (arg0、arg1、arg2 など) リモートメソッドの呼び出し引数として渡されます。複数の引数がある場合でも、すべて argN の形式を取る必要があります。N は引数の位置です。これらの引数は番号が連続している必要があります。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは level プロパティの status および code プロパティの NetConnection.Call.Success を含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Admin.CommandFailed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。指定した管理者が存在しない場合、このメソッドは失敗します。

HTTP 呼び出しは次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

例

```
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "JLee", "x52z49ab");
nc_admin.call("broadcastMsg", new Result(), "App:test", "testMethod", "Hello", "World");
```

changePswd()

RTMP/E

```
changePswd(admin_name:String, password:String [,scope:String]) : Object
```

HTTP

```
http://www.example.com:1111/admin/changePswd?auser=username&apswd=password&username=name
&password=password[&scope=scope]
```

指定した管理者のパスワードを変更します。パスワードは Users.xml 設定ファイルに書き込まれる前にエンコーディングされます。

仮想ホスト管理者は、自分のパスワード以外は変更できません。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

admin_name パスワードを変更する管理者の名前を示す文字列。

Username パスワードを変更する管理者の名前を示す文字列。

password 管理者の新しいパスワードを示す文字列。

scope 管理者がサーバー管理者であるか、または仮想ホスト管理者であるか（その場合は、どの仮想ホスト管理者であるか）を示す文字列。

接続中の仮想ホスト上の管理者のパスワードを変更するには、このパラメータを削除します。異なる仮想ホスト上の管理者のパスワードを変更するには、**adaptor_name/virtual_hostname** の形式で異なる仮想ホストを指定します。

サーバー管理者のパスワードを変更するには、「server」と指定します。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは **level** プロパティの **status** および **code** プロパティの **NetConnection.Call.Success** を含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、**level** プロパティの **error** および **code** プロパティの **NetConnection.Admin.CommandFailed**、またはより具体的な値（存在する場合）を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む **description** プロパティが含まれているオブジェクトもあります。

指定した管理者が存在しない場合、このメソッドは失敗します。

HTTP 呼び出しは次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される **Object** のプロパティと同じ情報が含まれます。

注意：HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

```
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "JLee", "x52z49ab");

/* Change password for server administrator named "ASilva" to "cbx5978y" */
nc_admin.call("changePswd", new onChangePswd(), "ASilva", "cbx5978y", "server");

/* Change password for virtual host administrator "JLee" to "kbat3786" on */
/* virtual host "tree.oak.com" */
nc_admin.call("changePswd", new onChangePswd(), "JLee", "kbat3786", "_defaultRoot/_ tree.oak.com");
```

gc()

RTMP/E

gc() : Object

HTTP

http://www.example.com:1111/admin/gc?auser=username&apswd=password

閉じられたストリーム、アプリケーションインスタンス、非永続的共有オブジェクトなど、使用されなくなったすべてのサーバーリソースを収集し、削除します。呼び出しでこの処理に要する時間は、約 1 秒です。

この操作を実行できるのは、サーバー管理者のみです。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは level プロパティの status および code プロパティの NetConnection.Call.Success を含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Admin.CommandFailed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しは次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。失敗の原因について説明する文字列を含む description Element が含まれている XML もあります。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

getActiveInstances()

RTMP/E

getActiveInstances (processID:Number) : Object

HTTP

http://www.example.com:1111/admin/getActiveInstances?auser=username&apswd=password&pid|processID=number

processID パラメータで指定した FMSCore プロセスで実行中のすべてのアプリケーションインスタンスの名前が含まれる文字列の配列を返します。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

pid|processID FMSCore プロセスのプロセス識別子を示す数字。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびサーバーまたは指定されたプロセスで実行中のすべてのアプリケーションインスタンスの名前を含む文字列の配列である data プロパティを含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Admin.CommandFailed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

```
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "JLee", "x52z49ab");
nc_admin.call("getActiveInstances", new Result());
```

getActiveVHosts()

RTMP/E

```
getActiveVHosts([adaptor]);
```

HTTP

```
http://www.example.com:1111/admin/getActiveVHosts?auser=username&apswd=password[&adaptor=name]
```

指定したアダプタに対して定義されているアクティブな仮想ホストの配列を返します。このメソッドを呼び出すことができるのは、サーバー管理者のみです。仮想ホストは、その上で実行中の少なくとも 1 つ以上のアプリケーションが接続されている場合はアクティブです。

使用できるバージョン

Flash Media Server 3

パラメータ

adaptor アダプタのユーザー名を示す文字列。指定しない場合は、`_defaultRoot_` が使用されます。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、`level` プロパティの `status`、`code` プロパティの `NetConnection.Call.Success` およびすべてのアクティブな仮想ホストの名前を含む文字列の配列である `data` プロパティを含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、`level` プロパティの `error` および `code` プロパティの `NetConnection.Call.Failed`、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む `description` プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次のような XML を返します。

```
<result>
  <level>status</level>
  <code>NetConnection.Call.Success</code>
  <timestamp>9/23/2007 6:16:40 PM</timestamp>
  <data>
    <_0>
      <vhost_name>vhost1</vhost_name>
    </_0>
    <_1>
      <vhost_name>_defaultVHost</vhost_name>
    </_1>
  </data>
</result>
```

注意： HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

getActiveVHostStats()

RTMP/E

`getActiveVHostStats([verbose]:Boolean) : Object`

HTTP

`http://www.example.com:1111/admin/getActiveVHostStats?auser=username&apswd=password[&verbose=value]`

いずれのアクティブなコアプロセス上にあるアクティブな仮想ホストに対して、すべてのアプリケーションのすべてのインスタンスを集約したパフォーマンスデータを返します。仮想ホストが複数のコア間で機能する場合はいつでも、このメソッドはアクティブなコアプロセスの統計を表示します。このメソッドを呼び出すことができるのはサーバー管理者のみです。

使用できるバージョン

Flash Media Server 3

パラメータ

verbose ブール値。true の場合、アクティブな仮想ホストで実行中のアプリケーションの統計および仮想ホストの集約した統計が表示されます。false の場合、アクティブな仮想ホストの集約した統計のみが表示され、仮想ホストごとの統計は表示されません。デフォルト値は true です。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびアクティブな仮想ホストのパフォーマンスデータを含むオブジェクトである data プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
msg_in	仮想ホストが処理したメッセージの合計数を示す数字。
msg_out	仮想ホストが送信したメッセージの合計数を示す数字。
msg_dropped	仮想ホストによって欠落したメッセージの合計数を示す数字。
bytes_in	仮想ホストが読み取った合計バイト数を示す数字。
bytes_out	仮想ホストが書き込んだ合計バイト数を示す数字。
accepted	仮想ホストが許可した接続の合計数を示す数字。
rejected	仮想ホストが拒否した接続要求の合計数を示す数字。
connected	現在アクティブな接続の合計数を示す数字。
total_apps	インスタンスが作成されたアプリケーションの合計数を示す数字。
total_connects	サーバーへの接続の合計数を示す数字。
total_disconnects	サーバーの接続解除の合計数を示す数字。
total_instances_loaded	ロードされたインスタンスの合計数を示す数字。 このプロパティは、ロードされたアクティブなインスタンスの合計数ではありません。ロードされたアクティブなインスタンスの数を取得するには、total_instances_loaded の値から total_instances_unloaded の値を引きます。
total_instances_unloaded	アンロードされたインスタンスの合計数を示す数字。
bw_in	現在の入力帯域幅 (bps) を示す数字。
bw_out	現在の出力帯域幅 (bps) を示す数字。
server_bytes_in	Number : SSAS NetConnection によって読み込まれる rtmp バイト数の合計。
server_bytes_out	Number : SSAS NetConnection によって送信される rtmp バイト数の合計。
tunnel_bytes_in	トンネルで読み取られた合計バイト数を示す数字。
tunnel_bytes_out	トンネルで書き込まれた合計バイト数を示す数字。
tunnel_requests	現在の要求数を示す数字。
tunnel_idle_requests	現在アイドル状態の要求数を示す数字。

プロパティ	説明
tunnel_idle_responses	現在アイドル状態の応答数を示す数字。
normal_connects	通常の接続の合計数を示す数字。
virtual_connects	リモートエッジによる接続の合計数を示す数字。
group_connects	接続されるリモートエッジの合計数を示す数字。
service_connects	サービス接続の合計数を示す数字。
service_requests	要求されたサービスの合計数を示す数字。
admin_connects	管理者による接続の合計数を示す数字。
debug_connects	デバッグ接続の合計数を示す数字。
swf_verification_attempts	SWF 検証を試みた回数を示すカウンタ。確認のためにサーバーに渡された SWF 検証の証明書の合計を表します。接続ごとに複数の証明書が提示される場合があります。
swf_verification_exceptions	SWF 検証で発生した例外の数を示すカウンタ。Application.xml ファイルでの明示的な設定により、特定のユーザーエージェントが SWF 検証要求を省略できるようにする例外が許可されます。例外として許可されるすべての接続がここにカウントされます。
swf_verification_failures	SWF 検証が失敗した回数を示すカウンタ。提示された SWF 検証の証明書がロード済みの証明書と一致しないことが検出されると、失敗になります。失敗はそれぞれ接続の切断に対応します。
swf_verification_unsupported_rejects	拒否された SWF 検証のうち、サポートされない拒否の回数を示すカウンタ。SWF 検証をサポートしないバージョンの Flash Player が、SWF 検証を必要とするアプリケーションに接続すると、サポートされない拒否の数が増えます。また、接続中のクライアントが切断されます。
swf_verification_matches	一致の合計数を示すカウンタ。信頼できる SWF 検証の証明書が提示されると、この数は増加します。接続ごとに複数の一致がある場合があります。
swf_verification_remote_misses	プロキシ/リモートサーバープロセスにおいて、SWF 検証の試行が成功しなかった回数を示すカウンタ。プロキシサーバーまたはリモートサーバーは、SWF 検証の試行を受け取ると、ローカルキャッシュに有効な SWF 検証がないか探します。一致するものが見つからないと、リモートミスを記録し、検証の試行の応答をオリジンサーバーに任せます。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Call.Failed、またはより具体的な値（存在する場合）を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E の戻り値セクションに記載された各プロパティを含むエレメントがネストされます。

呼び出しが失敗すると、次のような内容を返します。

```
<result>
  <level>error</level>
  <code>Admin.API.InvalidMethod</code>
  <description>getactivevhoststatss - No such method found!</description>
  <timestamp>9/13/2007 7:55:58 PM</timestamp>
</result>
```

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

getAdaptors()

RTMP/E

getAdaptors() : Object

HTTP

http://www.example.com:1111/admin/getAdaptors?user=username&apswd=password

定義されているアダプタの配列を返します。このメソッドを呼び出すことができるのは、サーバー管理者のみです。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

user 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびすべてのアダプタの名前を含む文字列の配列である data プロパティを含む情報オブジェクトを応答として送信します。

HTTP 呼び出しが成功すると、次のような内容を返します。

```
<result>
  <level>status</level>
  <code>NetConnection.Call.Success</code>
  <timestamp>7/23/2007 8:25:52 AM</timestamp>
  <data>
    <_0>Adaptor2</_0>
    <_1>_defaultRoot_</_1>
  </data>
</result>
```

呼び出しが失敗すると、次のような内容を返します。

```
<result>
  <level>error</level>
  <code>NetConnection.Connect.Rejected</code>
  <description>Administrator login failed for user Admin.</description>
  <timestamp>7/23/2007 8:26:36 AM</timestamp>
</result>
```

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

```
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "JLee", "x52z49ab");
nc_admin.call("getAdaptors", new Result());
```

getAdminContext()

RTMP/E

```
getAdminContext([adminName:String] [, adaptorName:String] [, vhostName:String]) : Object
```

HTTP

```
http://www.example.com:1111/admin/getAdminContext?user=username&apswd=password[&adminName=name] [&adaptorName=name] [&vhostName=name]
```

指定したユーザーの管理権限、ユーザーが接続しているアダプタと仮想ホストの名前、ユーザーが現在 Flash Media Server に接続しているかどうかなど、管理者の管理コンテキストを取得します。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

adminName 管理者の名前を示す文字列。

adaptorName 管理者の検索を実行する代替アダプタ (`_defaultRoot_` 以外) を示す文字列。指定しない場合は、`_defaultRoot_` が使用されます。

vhostName 仮想ホスト管理者の検索を実行する代替仮想ホスト (`_defaultVHost_` 以外) を示す文字列。指定しない場合は、サーバーレベルの管理者が使用されます。

user 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、`level` プロパティの `status`、`code` プロパティの `NetConnection.Call.Success` およびオブジェクトである `data` プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
<code>admin_type</code>	管理者のタイプを示す文字列 (server または virtual host)。
<code>adaptor</code>	ユーザーが管理者となっているアダプタの名前を示す文字列。
<code>vhost</code>	ユーザーが管理者となっている仮想ホストの名前を示す文字列。
<code>connected</code>	このプロパティは使用されなくなりました。したがって、常に <code>true</code> が返されます。

HTTP 呼び出しが成功すると、次のような内容を返します。

```
<result>
  <level>status</level>
  <code>NetConnection.Call.Success</code>
  <timestamp>7/23/2007 8:57:26 AM</timestamp>
  <data>
    <admin_type>server</admin_type>
    <adaptor>_defaultRoot_</adaptor>
    <vhost>_defaultVHost_</vhost>
    <connected>true</connected>
  </data>
</result>
```

呼び出しが失敗すると、次のような内容を返します。

```
<result>
  <level>error</level>
  <code>NetConnection.Connect.Rejected</code>
  <description>Administrator login failed for user Admin.</description>
  <timestamp>7/23/2007 9:00:44 AM</timestamp>
</result>
```

注意：HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

getAdmins()

RTMP/E

getAdmins(adaptorName:String) : Object

HTTP

http://www.example.com:1111/admin/getAdmins?user=username&apswd=password&adaptorName=name

Flash Media Server のすべての管理者の配列を返します。返されるデータには、最上位レベルのグループが2つ（サーバーレベルの管理者および仮想ホストレベルの管理者）含まれます。サーバーレベルの管理者のグループは、名前を記した単純なリストです。仮想ホスト管理者は仮想ホスト別に分けられ、各仮想ホストについて管理者名がリストされます。

使用できるバージョン

Flash Media Server 2.0

パラメータ

adaptorName 仮想ホスト管理者の検索を実行する代替アダプタ（_defaultRoot_ 以外）を示す文字列。指定しない場合は、_defaultRoot_ が使用されます。

user 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success および配列である data プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
server_admins	管理者名の配列。
admin	管理者名の文字列。
vhost_admins	仮想ホスト名の配列。
admin	ユーザーが管理者となっている仮想ホストの名前を示す文字列。

HTTP 呼び出しが成功すると、次のような内容を返します。

```
<result>
  <level>status</level>
  <code>NetConnection.Call.Success</code>
  <timestamp>7/23/2007 9:05:00 AM</timestamp>
  <data>
    <server_admins>
      <admin>Admin</admin>
      <admin>John</admin>
      <admin>Mary</admin>
    </server_admins>
    <vhost_admins>
      <_defaultVHost_>
        <_0>Kent</_0>
      </_defaultVHost_>
    </vhost_admins>
  </data>
</result>
```

呼び出しが失敗すると、次のような内容を返します。

```
<result>
  <level>error</level>
  <code>NetConnection.Connect.Rejected</code>
  <description>Administrator login failed for user Admin.</description>
  <timestamp>7/23/2007 9:08:25 AM</timestamp>
</result>
```

注意：HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

getApps()

RTMP/E

```
getApps([verbose:Boolean][, force:Boolean]):Object
```

HTTP

```
http://www.example.com:1111/admin/getApps?auser=username&apswd=password [&verbose=true|false [&force=true|false]]
```

インストールされているすべてのアプリケーションの名前を含む文字列の配列を返します。[addApp\(\)](#) または [removeApp\(\)](#) の呼び出しにより、キャッシュされたアプリケーションリストが更新されます。

注意：ファイルシステムを使用して追加または削除したアプリケーションは、キャッシュされたリストには反映されません。

使用できるバージョン

Flash Communication Server 1.0 verbose パラメータおよび force パラメータは Flash Media Server 3 で使用することができます。

パラメータ

verbose ブール値。true の場合、仮想ホストで実行されているアプリケーションがすべて表示されます。false の場合、アプリケーションの合計数が表示されます。デフォルト値は true です。

force ブール値。true の場合、キャッシュされたアプリケーションリストを強制的に更新して、リストを取得します。false の場合、キャッシュされたアプリケーションリストを取得します。強制的に更新するには、verbose パラメータと force パラメータの両方を指定する必要があります。デフォルト値は false です。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、**level** プロパティの **status**、**code** プロパティの **NetConnection.Call.Success** およびインストールされているすべてのアプリケーションの名前を含む文字列の配列である **data** プロパティを含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、**level** プロパティの **error** および **code** プロパティの **NetConnection.Call.Failed**、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む **description** プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level>status</level>
  <code>NetConnection.Call.Success</code>
  <timestamp>7/23/2007 9:10:37 AM</timestamp>
  <data>
    <total_apps>3</total_apps>
    <_0>live</_0>
    <_1>livetest</_1>
    <_2>vod</_2>
  </data>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。失敗の原因について説明する文字列を含む **description** Element が含まれている XML もあります。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getActiveInstances\(\)](#)、[getAppStats\(\)](#)、[getInstanceStats\(\)](#)

getAppStats()

RTMP/E

`getAppStats (appname:String):Object`

HTTP

`http://www.example.com:1111/admin/getAppStats?auser=username&apswd=password&app|appname=name`

指定したアプリケーションのすべてのインスタンスの包括的なパフォーマンスデータを取得します。

Flash Media Server 3.5 以降では、アプリケーションの送信先となるコアを動的に制御できます。コアを制御することで、パフォーマンスカウンタに基づいて複数のコアに負荷を分散することができます。**getAppStats** では、アプリケーションを実行しているコアの **pid** とコア ID で構成される追加フィールドが、応答で送信されます。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

appname 追加するアプリケーションの名前を示す文字列。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびオブジェクトである data プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
accepted	アプリケーションが許可した接続の合計数を示す数字。
bytes_in	アプリケーションが読み取った合計バイト数を示す数字。
bytes_out	アプリケーションが書き込んだ合計バイト数を示す数字。
connected	現在アクティブな接続の合計数を示す数字。
launch_time	アプリケーションの開始時間を示す ActionScript の Date オブジェクト。
msg_dropped	アプリケーションがドロップしたメッセージの合計数を示す数字。
msg_in	アプリケーションが処理したメッセージの合計数を示す数字。
msg_out	アプリケーションから送信されたメッセージの合計数を示す数字。
normal_connects	通常の接続の合計数を示す数字。
virtual_connects	リモートエッジによる接続の合計数を示す数字。
group_connects	接続されるリモートエッジの合計数を示す数字。
service_connects	サービス接続の合計数を示す数字。
service_requests	要求されたサービスの合計数を示す数字。
admin_connects	管理者による接続の合計数を示す数字。
debug_connects	デバッグ接続の合計数を示す数字。
rejected	アプリケーションが拒否した接続要求の合計数を示す数字。
bw_in	現在の入力帯域幅 (bps) を示す数字。
bw_out	現在の出力帯域幅 (bps) を示す数字。
total_connects	アプリケーション開始時以降のアプリケーションへのソケット接続の合計数を示す数字。
total_disconnects	アプリケーション開始時以降のアプリケーションの接続解除の合計数を示す数字。
total_instances_loaded	アプリケーション開始時以降にロードされたインスタンスの合計数を示す数字。 このプロパティは、ロードされたアクティブなインスタンスの合計数ではありません。ロードされたアクティブなインスタンスの数を取得するには、total_instances_loaded の値から total_instances_unloaded の値を引きます。
total_instances_unloaded	アプリケーション開始時以降にアンロードされたインスタンスの合計数を示す数字。
up_time	アプリケーションの実行時間 (秒) を示す数字。
swf_verification_attempts	SWF 検証を試みた回数を示すカウンタ。確認のためにサーバーに渡された SWF 検証の証明書の合計を表します。接続ごとに複数の証明書が提示される場合があります。
swf_verification_exceptions	SWF 検証で発生した例外の数を示すカウンタ。Application.xml ファイルでの明示的な設定により、特定のユーザーエージェントが SWF 検証要求を省略できるようにする例外が許可されます。例外として許可されるすべての接続がここにカウントされます。

プロパティ	説明
swf_verification_failures	SWF 検証が失敗した回数を示すカウンタ。提示された SWF 検証の証明書がロード済みの証明書と一致しないことが検出されると、失敗になります。失敗はそれぞれ接続の切断に対応します。
swf_verification_unsupported_rejects	拒否された SWF 検証のうち、サポートされない拒否の回数を示すカウンタ。SWF 検証をサポートしていないバージョンの Flash Player が、SWF 検証を必要とするアプリケーションに接続を試みる場合があります。その場合、サポートされていない拒否の数が増加し、接続中のクライアントの接続が切断されます。
swf_verification_matches	一致の合計数を示すカウンタ。信頼できる SWF 検証の証明書が提示されると、この数は増加します。接続ごとに複数の一致がある場合があります。
swf_verification_remote_misses	プロキシ/リモートサーバープロセスにおいて、SWF 検証の試行が成功しなかった回数を示すカウンタ。プロキシサーバーまたはリモートサーバーは、SWF 検証の試行を受け取ると、ローカルキャッシュに有効な SWF 検証がないか探します。一致するものが見つからないと、リモートミスを記録し、検証の試行の応答をオリジンサーバーに任せます。
cores	エレメント「pid」および「core_id」を含む配列。core_id 配列内のエレメント数は、アプリケーションで実行されているコア数によって異なります。

次に getAppStats の出力を示します。

```
<result>
  <level>status</level>
  <code>NetConnect.Call.Success</code>
  <timestamp>7/24/2008 3:50:55 PM</timestamp>
  <data>
    <launch_time>7/24/2008 3:21:02 PM</launch_time>
    <up_time>1792</up_time>
    <bw_in>0</bw_in>
    <bw_out>0</bw_out>
    <bytes_in>3073</bytes_in>
    <bytes_out>3417</bytes_out>
    <msg_in>0</msg_in>
    <msg_out>1</msg_out>
    <msg_dropped>0</msg_dropped>
    <total_connects>1</total_connects>
    <total_disconnects>0</total_disconnects>
    <connected>1</connected>
    <normal_connects>1</normal_connects>
    <virtual_connects>0</virtual_connects>
    <group_connects>0</group_connects>
    <service_connects>0</service_connects>
    <service_requests>0</service_requests>
    <admin_connects>0</admin_connects>
    <accepted>1</accepted>
    <rejected>0</rejected>
```

```
<total_instances_loaded>1</total_instances_loaded>
<total_instances_unloaded>0</total_instances_unloaded>
<swf_verification_attempts>0</swf_verification_attempts>
<swf_verficiation_exceptions>0</swf_verification_exceptions>
<swf_verification_failures>0</swf_verification_failures>
<swf_verification_unsupported_rejects>0</swf_verification_unsupported_rejects>
<swf_verification_matches>0</swf_verification_matches>
<swf_verification_remote_misses>0</swf_verification_remote_misses>
<cores>
  <_0>
    <pid>1234</pid>
    <core_id>0</code_id>
  </_0>
  <_1>
    <pid>5678</pid>
    <core_id>1</core_id>
  </_1>
</cores>
</data>
</result>
```

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Call.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、サーバーは次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意： HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

次の 2 つの例では、オリジンサーバーとエッジサーバーの両方で app_1 というアプリケーションインスタンスを使用します。オリジンサーバーで getAppStats() を呼び出すには、次の URL を使用します。

```
http://origin.example.com:1111/admin/getAppStats?user=username&passwd=password &appName=app_1
```

エッジサーバーで getAppStats() を呼び出すには、次の URL を使用します。

```
http://edge.example.com:1111/admin/getAppStats?user=username&passwd=password&appName=app_1@edge.example.com
```

注意： アプリケーション名を取得するには、Administration Console でエッジサーバーにログインするか、getActiveInstances() を呼び出します。appName=app_1 がエッジサーバーで実行されていると、すべてゼロの情報が表示されます。

関連項目

[getApps\(\)](#)、[getInstanceStats\(\)](#)

getConfig()

RTMP/E

```
getConfig(key:String [,scope:String]) : Object
```

HTTP

```
http://www.example.com:1111/admin/getConfig?user=username&passwd=password&key=configkey[&scope=scope]
```

この API は使用されなくなりました。代わりに、[getConfig2\(\)](#) を使用してください。

設定ファイルの設定パラメータの値を取得します。

仮想ホスト管理者は、自分の仮想ホストの `Vhost.xml` ファイルと `Application.xml` ファイルの設定パラメータを表示できます。 `Server.xml` および `Adaptor.xml` ファイルの設定パラメータの大部分は、サーバー管理者のみが見ることができます。XML 設定ファイルについては、[Adobe Flash Media Server 設定および管理ガイド](#)』を参照してください。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

`key` 情報を取得する設定パラメータを示す文字列。

キーは、スラッシュ (/) で区切られたサブキーのリストとして指定されます。最初のサブキーは、目的の設定パラメータを含む XML 設定ファイルを指定します。2 番目以降のサブキーは、XML 設定ファイルに関連したタグに対応します。したがって、サブキーの階層と名前は、XML ファイルのタグと一致することになります。

権限に応じて、次のファイルの設定パラメータを取得することができます。

- `Server.xml` ファイルでは、最初のサブキーとして、`Admin` または `Server` を指定します。2 番目以降のすべてのキーは、`Server.xml` ファイルの `Admin` または `Server` タグに関連したタグに対応します。

`Server` タグの設定パラメータを表示できるのは、サーバー管理者のみです。

仮想ホスト管理者は、自分の仮想ホストについてのみ `Admin` タグの設定パラメータを表示できます。一部の機密性のある情報については、表示できない場合があります。例えば、自分の仮想ホストの他の管理者名は表示できますが、これらの管理者のパスワードや権限の設定は表示できません。

- `Adaptor.xml` ファイルでは、最初のサブキーとして `Adaptor:adaptor_name` (`adaptor_name` はアダプタ名) を指定します。2 番目以降のすべてのパラメータは、`Adaptor.xml` ファイルの `Adaptor` タグに関連したキーに対応します。
- `Vhost.xml` では、最初のサブキーとして `Adaptor:adaptor_name/VirtualHost:vhost_name` (`adaptor_name` はアダプタ名、`vhost_name` は仮想ホスト名) を指定します。2 番目以降のすべてのキーは、`Vhost.xml` ファイルの `VirtualHost` タグに関連したタグに対応します。
- 管理サーバーにログインしたときに接続した仮想ホスト上で実行中のアプリケーションの `Application.xml` ファイルについては、最初のサブキーとして `Application:app_name` (`app_name` はアプリケーション名) を指定します。

別の仮想ホストで実行中のアプリケーションの `Application.xml` ファイルのパラメータを取得するには、完全なキー (`Adaptor:adaptor_name/VirtualHost:vhost_name/Application:app_name`) を指定します。 `scope` パラメータも指定する必要があります。

デフォルトの `Application.xml` ファイルを取得するには、コロン (:) と属性 `app_name` を付けずに「`Application`」と指定します。

scope 文字列。Server.xml ファイル、Adaptor.xml ファイルまたは Vhost.xml ファイルの設定パラメータを取得するには、スラッシュ (/) を指定します。

Flash Media Server にログオンしたときに接続した仮想ホスト上で実行中のアプリケーションの Application.xml ファイルの設定パラメータを取得するには、このパラメータを省略します。

注意：接続しているアダプタまたは仮想ホストを確認するには、getAdminContext() メソッドを使用します。

user 管理者のユーザー名を示す文字列。

passwd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success および指定したタグの値のプロパティを持つ data オブジェクトを含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると（つまり、指定した設定パラメータが検出されない場合）、サーバーは空の文字列を返します。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意：HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

次の例では、4 つの XML ファイルそれぞれについて、設定キーの取得方法を示します。

```
// Establish connection to server.
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "JLee", "xm1472dy");

// For a virtual host administrator, find key in Server.xml.
key = "Admin/Server/UserList/User:JLee/Password";
nc_admin.call("getConfig", new onGetConfig(), key, "/");

// For a server administrator; find key in Server.xml.
key = "Server/LicenseInfo";
nc_admin.call("getConfig", new onGetConfig(), key, "/");

// Find key in Adaptor.xml.
key = "Adaptor:_defaultRoot_/HostPortList/HostPort";
nc_admin.call("getConfig", new onGetConfig(), key, "/");

// Find key in Vhost.xml.
key = "Adaptor:_defaultRoot_/VirtualHost:_defaultVhost_/RecordAccessLog";
nc_admin.call("getConfig", new onGetConfig(), key, "/");

// Find key in Application.xml for an application on the virtual host you
// connected to when you logged on to the administration server.
// Note that the previous subkeys and the second parameter "/" are not necessary.
key = "Application:FinanceApp/RecordAppLog";
nc_admin.call("getConfig", new onGetConfig(), key);

// Find key in Application.xml for an application on a different virtual host.
key = "Adaptor:_defaultRoot_/VirtualHost:www.redpin.com/Application:ChatApp/ ~ RecordAppLog";
nc_admin.call("getConfig", new onGetConfig(), key, "/");
```

関連項目

[getAdminContext\(\)](#)、[setConfig\(\)](#)

getConfig2()

RTMP/E

`getConfig2(key:String, scope:String) : Object`

HTTP

`http://www.example.com:1111/admin/getConfig2?user=username&apswd=password&key=configkeyname&scope=scope`

指定した設定ファイルの設定パラメータの値を取得します。Flash Media Server には、Users.xml、Logger.xml、Server.xml、Adaptor.xml、Vhost.xml および Application.xml という 6 つのサーバー設定ファイルがあり、そこから情報を取得することができます。XML 設定ファイルについては、『[Adobe Flash Media Server 設定および管理ガイド](#)』を参照してください。

仮想ホスト管理者は、自分の仮想ホストの Vhost.xml ファイルと Application.xml ファイルの設定パラメータを表示できます。Server.xml および Adaptor.xml ファイルの設定パラメータの大部分は、サーバー管理者のみが表示できます。

注意：XML ツリーの同一レベルに同じ名前の XML タグが共存することも可能です。設定ファイル内では、XML タグの name 属性を使用して、このようなタグを区別する必要があります。例えば、複数の VirtualHost タグがある場合は、`<VirtualHost name="www.redpin.com"></VirtualHost>` という形式を使用します。getConfig() メソッドを呼び出して設定のサブキーを指定するときは、タグ名、コロン、正しい name 属性の順に指定することによって必要なタグを示すことができます。例えば、Admin/Adaptor:_defaultRoot_/VirtualHost:www.redpin.com と指定します。

使用できるバージョン

Flash Media Server 2.0

パラメータ

key 情報を取得する設定パラメータを示す文字列。

キーは、スラッシュ (/) で区切られたサブキーのリストとして指定されます。最初のサブキーは、目的の設定キーを含む XML 設定ファイルを指定します。2 番目以降のサブキーは、XML 設定ファイルに関連したタグに対応します。したがって、サブキーの階層と名前は、XML ファイルのタグと一致することになります。同一名で同一の親を持つタグが複数存在する場合は、name 属性を指定することによって、これらのタグを区別できます。key パラメータでは、コロンで区切ったタグ名に name 属性を追加します。指定したタグがリーフノードである場合は、そのタグデータが返されます。指定したタグがリーフノードでない場合は、タグ全体が XML 文字列として返されます。

scope key パラメータで指定した設定タグを検索する設定ファイルを示す文字列。

Flash Media Server には、Server.xml、Users.xml、Logger.xml、Adaptor.xml、Vhost.xml および Application.xml という 6 つのサーバー設定ファイルが含まれています。ユーザーに与えられた権限によっては、以下のリストに示すように、これらのファイルすべての設定キーを取得できます。

- / は、Server.xml を指定します。
- Users は、サーバー管理者用の Users.xml を指定します。
- Logger は、Logger.xml を指定します。
- Adaptor:<adaptor_name> は、Adaptor.xml を指定します。<adaptor_name> プレースホルダーのところにアダプタ名を指定します。Adaptor.xml ファイルにアクセスできるのは、サーバー管理者の特権を持つユーザーのみです。呼び出し側が接続しているアダプタの名前が <adaptor_name> でない場合、呼び出しは失敗します。
- Adaptor:<adaptor_name>/VHost:<vhost_name> は、VHost.xml を指定します。<vhost_name> プレースホルダーのところに仮想ホスト名を指定します。呼び出し側が接続しているアダプタの名前が <adaptor_name> ではない場合や、呼び出し側が接続している仮想ホストの名前が <vhost_name> ではない場合、呼び出しは失敗します。
- Adaptor:<adaptor_name>/VHost:<vhost_name>/Users.xml は、仮想ホスト管理者の Users.xml を指定します。
- Adaptor:<adaptor_name>/VHost:<vhost_name>/App[:<app_name>] は、Application.xml を指定します。<app_name> が指定されていない場合は、デフォルトの Application.xml ファイルが使用されます。指定されている場合は、指定したアプリケーション用の Application.xml が使用されます。指定されたアプリケーションが定義されていない場合や、そのアプリケーション用の Application.xml ファイルがない場合は、呼び出しは失敗します。

注意：接続しているアダプタまたは仮想ホストを確認するには、`getAdminContext()` メソッドを呼び出します。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success および指定したタグに関する情報を格納したプロパティを持つ data オブジェクトを含む情報オブジェクトを応答として送信します。

指定したタグがリーフノードである場合は、タグデータが返されます。指定したタグがリーフノードでない場合は、タグ自体が返されます。例えば、必要なタグが <foo>bar</foo> である場合は、「bar」が返されます。ただし、必要なタグに <foo><bar>foobar</bar></foo> などの子タグが含まれる場合は、「<foo><bar>foobar</bar></foo>」が返されます。

呼び出しが失敗すると（つまり、指定した設定キーが検出されない場合）、サーバーは空の文字列を返します。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

次の例では、さまざまな設定ファイルから XML データを取得します。

```
tSocket = new NetConnection();
tSocket.connect("rtmp://localhost/admin", "user", "password");

// find key in Server.xml
key = "Server/LicenseInfo";
tSocket.call("getConfig2", new onGetConfig(), key, "");

// find key in Adaptor.xml
key = "HostPortList/HostPort";
scope = "Adaptor:_defaultRoot_";
tSocket.call("getConfig2", new onGetConfig(), key, scope);

// find key in Vhost.xml
key = "AppsDir";
scope = "Adaptor:_defaultRoot_/VHost:_defaultVHost_";
tSocket.call("getConfig2", new onGetConfig(), key, scope);

// find key in Application.xml for app "foo"
key = "Process/Scope";
scope = "Adaptor:_defaultRoot_/VHost:_defaultVHost_/App:foo";
tSocket.call("getConfig2", new onGetConfig(), key, scope);

// find key in default Application.xml
key = "Process/Scope";
scope = "Adaptor:_defaultRoot_/VHost:_defaultVHost_/App";
tSocket.call("getConfig2", new onGetConfig(), key, scope);

// return the whole Vhost.xml
key = "";
scope = "Adaptor:_defaultRoot_/VHost:_defaultVHost_";
tSocket.call("getConfig2", new onGetConfig(), key, scope);
```

関連項目

[getAdminContext\(\)](#)、[setConfig2\(\)](#)

getFileCacheStats()

RTMP/E

```
getFileCacheStats([verbose:Boolean] [, file:String]) : Object
```

HTTP

```
http://www.example.com:1111/admin/getFileCacheStats?auser=username&apswd=password[&verbose=value] [&file=string]
```

キャッシュのサイズ、キャッシュに保持されているファイルオブジェクトの数、セグメントのヒット数とミス数など、ファイルキャッシュに関するデータを取得します。verbose パラメータを true に設定すると、キャッシュ内のファイルごとに統計が返されます。

Flash Media Server 3.5 以降では、アプリケーションの送信先となるコアを動的に制御できます。コアを制御することで、パフォーマンスカウンタに基づいて複数のコアに負荷を分散することができます。getFileCacheStats では、結果はコア単位でグループ化されます。

使用できるバージョン

Flash Media Server 3

パラメータ

verbose ブール値。true の場合、キャッシュ内のファイルごとに統計が表示されます。false の場合、キャッシュ内のすべてのオブジェクトのリストが表示されます。デフォルト値は false です。

file ファイルの統計を示す文字列。[object_name] 値の形式で値を指定します。

file プロパティの値は、「flv:[path]」、「mp3:[path]」、「mp4:[path]」などの構文で指定します。この値を確認するには、次のように最初に verbose=true で API を呼び出し、結果からパスを取得します。

```
<object_name>5888_C:¥Program Files¥Adobe¥Flash Media Server  
3¥applications¥app_1¥streams¥_definst_¥Coral_Reef_Adventure_spark_150k_105s</object_name>
```

注意：file プロパティで使用するときは 5888_ を削除し、ファイルのパスだけを使用します。

verbose パラメータおよび file パラメータは同時に使用しないでください。両方を使用すると、最初のパラメータが使用され、2 番目のパラメータは無視されます。キャッシュに含まれるファイルが多い場合、verbose を true に設定して使用すると、時間がかかることがあります。また、Web ブラウザから API を呼び出すと、ブラウザがタイムアウトになる場合があります。返されるデータの量を少なくするには、単一のオブジェクトの名前をパラメータとして渡します。単一のオブジェクト名を渡すと、そのオブジェクトに関する統計だけが返されます。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびオブジェクトである data プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
num_objs	キャッシュ内に保持されるファイルオブジェクトの合計数を示す数字。
hits	サーバーの起動以降のセグメント「ヒット」の合計数を示す数字。
misses	サーバーの起動以降のセグメントミスの合計数を示す数字。

プロパティ	説明
bytes	キャッシュの現在のサイズ (バイト数) を示す数字。
bytes_inuse	現在使用中のキャッシュの量 (バイト数) を示す数字。
released	サーバーの起動以降にリリースされたセグメントの合計数を示す数字。
cores	<data> 以下のすべてのプロパティおよびコア ID。すべてのプロパティはコア固有です。例えば、<data> 以下の <hits> ではサーバー全体から集約されています。反対に、<core><hits> は特定のコアの統計です。実行されているコアがない場合、コアエレメントは空です。
num_segments	キャッシュ内に保持されるセグメントの合計数を示す数字。

verbose パラメータが true に設定されている場合、data プロパティには、キャッシュ内のファイルごとに次の個々の統計値を持つ objects プロパティが含まれます。

プロパティ	説明
object_name	ストリームオブジェクトの名前。
num_segments	現在キャッシュされているこのオブジェクトのセグメントの合計数を示す数字。
hits	このオブジェクトのヒット数。
misses	このオブジェクトのミス数。
useCount	このオブジェクトにアクセスしているクライアント (内部サーバークライアントも含む) の合計数を示す数字。
released	このオブジェクトからリリースされたセグメントの合計数を示す数字。
recording	このファイルが録音されている場合。
bytes	キャッシュ内のこのオブジェクトのサイズを示す数字。
num_kfs	このオブジェクトに対するキーフレームキャッシュ内のフレームの合計数を示す数字。
kfbytes	このオブジェクトに対するキーフレームキャッシュのサイズ (バイト数) を示す数字。
bytes_inuse	このオブジェクトのアクティブに使用されている量 (バイト数) を示す数字。
overhead	このオブジェクトの固定メモリアーヘッド (バイト数) を示す数字。

次に、verbose が false に設定された状態で getFileCacheStats() を呼び出したときの出力を示します。

```
<result>
  <level>status</level>
  <code>NetConnect.Call.Success</code>
  <timestamp>7/24/2008 3:59:22 PM</timestamp>
  <data>
    <num_objs>0</num_objs>
    <hits>0</hits>
    <misses>0</misses>
    <bytes>0</bytes>
    <bytes_inuse>0</bytes_inuse>
    <released>0</released>
    <cores>
      <_26432>
        <num_objs>0</num_objs>
        <hits>0</hits>
        <misses>0</misses>
        <bytes>0</bytes>
        <bytes_inuse>0</bytes_inuse>
        <released>0</released>
        <core_id>0</core_id>
      <_26432>
    </cores>
  </data>
</result>
```

注意：コアが開始されていない（アプリケーションが実行されていない）場合は、集約データだけが返されます。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Call.Failed、またはより具体的な値（存在する場合）を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します（このデータはサンプルデータです）。

```
<result>
  <level>status</level>
  <code>NetConnection.Call.Success</code>
  <timestamp>7/16/2008 10:23:59 AM</timestamp>
  <data>
    <num_objs>1</num_objs>
    <hits>10</hits>
    <misses>17</misses>
    <bytes>2064203</bytes>
    <bytes_inuse>261875</bytes_inuse>
    <released>0</released>
    <cores>
      <_3644>
        <num_objs>1</num_objs>
        <hits>10</hits>
        <misses>17</misses>
        <bytes>2064203</bytes>
        <bytes_inuse>261875</bytes_inuse>
        <released>0</released>
        <objects>
          <_0>
            <_3644>
              <num_objects>1</num_objects>
              <hits>6</hits>
              <misses>28</misses>
              <bytes>2440227</bytes>
              <bytes_inuse>528112</bytes_inuse>
              <released>0</released>
              <core_id>1</core_id>
            </objects>
          </_0>
        </objects>
      </_3644>
    </cores>
  </data>
</result>
```

```

        </_3364>
        <_0>
        </objects>
    <_3644>
</cores>
<objects>
    <_0>
        <object_name>3644_C:¥Program Files¥Adobe¥Flash Media Server
            3.5¥applications¥vod¥media¥sample</object_name>
        <num_segments>9</num_segments>
        <hits>10</hits>
        <misses>9</misses>
        <useCount>3</useCount>
        <released>0</released>
        <recording>>false</recording>
        <bytes>2064203</bytes>
        <num_kfs>0</num_kfs>
        <kfbytes>0</kfbytes>
    </_0>
</objects>
</data>
</result>

```

呼び出しが失敗すると、次の構造の XML を返します。

```

<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>

```

注意： HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getAppStats\(\)](#)、[getGroupStats\(\)](#)

getGroupMembers()

RTMP/E

`getGroupMembers(appInst:String, groupid:String) : Object`

HTTP

`http://www.example.com:1111/admin/getGroupMembers?user=username&pswd=password&appInst=name&groupid=string`

特定グループに属するメンバーのリストを返します。グループ接続は、リモートのエッジサーバーからオリジンサーバーへのマルチプレックス接続です。各グループ接続は、このサーバーのエッジサーバーとして動作している別の Flash Media Server に接続しているユーザーが少なくとも 1 人存在することを示します。

`groupid` パラメータの値を取得するには、[getGroups\(\)](#) を呼び出します。

使用できるバージョン

Flash Media Server 2.0

パラメータ

appInst グループが存在するアプリケーションインスタンスの名前を、`application_name/instance_name` の形式で示す文字列。アプリケーションのデフォルトのインスタンスについてのパフォーマンス統計を取る場合でも、アプリケーション名とインスタンス名の両方をスラッシュ (`/`) で区切って指定する必要があります。例えば、アプリケーション ChatApp のデフォルトのインスタンスを指定する場合は、「ChatApp/_defInst_」と指定します。

groupid グループのクライアント ID を示す文字列。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、`level` プロパティの `status`、`code` プロパティの `NetConnection.Call.Success` およびこのグループから接続するすべてのクライアントのクライアント ID を含む配列である `data` プロパティを含む情報オブジェクトを応答として送信します。

HTTP 呼び出しが成功すると、次のような XML を返します。

```
<result>
  <level>status</level>
  <code>NetConnection.Call.Success</code>
  <timestamp>10/23/2007 12:19:24 PM</timestamp>
  <data>
    <name>_defaultRoot_: _defaultVHost_::__2</name>
    <_0>DDAAQMsI</_0>
  </data>
</result>
```

`data` エレメントには、RTMP/E セクションに記載された `data` オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level>error</level>
  <code>NetConnection.Call.BadValue</code>
  <timestamp>10/23/2007 1:59:00 PM</timestamp>
  <description>Invalid group ID (0).</description>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される `Object` のプロパティと同じ情報が含まれます。

注意 : HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getGroups\(\)](#)、[getGroupStats\(\)](#)

getGroupStats()

RTMP/E

`getGroupStats(appInst:String, groupid:String) : Object`

HTTP

`http://www.example.com:1111/admin/getGroupStats?auser=username&apswd=password&appInst=name&groupid=string`

特定のグループ接続についての統計を取得します。これは、複数の接続についてマルチプレックス接続を実行し、members_count という独自の統計を含んでいる特殊な接続です。グループ接続は、サーバー間でプロキシとして確立されません。

getGroups() を呼び出すと、groupid の値を取得できます。

使用できるバージョン

Flash Media Server 2.0

パラメータ

applInst グループが存在するアプリケーションインスタンスの名前を、application_name/instance_name の形式で示す文字列。アプリケーションのデフォルトのインスタンスについてのパフォーマンス統計を取る場合でも、アプリケーション名とインスタンス名の両方をスラッシュ (/) で区切って指定する必要があります。例えば、アプリケーション ChatApp のデフォルトのインスタンスを指定する場合は、「ChatApp/_defInst_」と指定します。

groupid グループのクライアント ID を示す文字列。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびクライアントのパフォーマンスデータを含むオブジェクトである data プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
connect_time	アプリケーションがサーバーに接続した時間を示す ActionScript の Date オブジェクト。
protocol	クライアントがサーバーに接続するために使用するプロトコルを示す文字列 (rtmp、rtmpe または rtmpt)。
msg_in	アプリケーションが処理したメッセージの合計数を示す数字。
msg_out	アプリケーションから送信されたメッセージの合計数を示す数字。
msg_dropped	アプリケーションがドロップしたメッセージの合計数を示す数字。
bytes_in	アプリケーションが読み取った合計バイト数を示す数字。
bytes_out	アプリケーションが書き込んだ合計バイト数を示す数字。
msg_queue	クライアントメッセージキューの統計を示すオブジェクト。
total_queues	クライアントに対するキューの合計数を示す数字。
audio	すべてのオーディオキューに含まれるオーディオメッセージの合計数を示す数字。
video	すべてのビデオキューに含まれるビデオメッセージの合計数を示す数字。
other	「other」キューに含まれるコマンドおよびデータメッセージの合計数を示す数字。
stream_ids	数字の配列 (ストリーム ID)。
members_count	当該グループ接続でマルチプレックス接続を実行しているクライアント数を示す数字。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level>status</level>
  <code>NetConnection.Call.Success</code>
  <timestamp>10/23/2007 12:43:47 PM</timestamp>
  <data>
    <name>_defaultRoot::_defaultVHost::_:2</name>
    <bytes_in>3283</bytes_in>
    <bytes_out>3878</bytes_out>
    <msg_in>5</msg_in>
    <msg_out>1</msg_out>
    <msg_dropped>0</msg_dropped>
    <connect_time>10/23/2007 12:16:08 PM</connect_time>
    <protocol>rtmp</protocol>
    <msg_queue>
      <total_queues>1</total_queues>
      <audio>0</audio>
      <video>0</video>
      <other>0</other>
    </msg_queue>
    <stream_ids />
    <members_count>1</members_count>
  </data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level>error</level>
  <code>NetConnection.Call.BadValue</code>
  <timestamp>10/23/2007 3:30:12 PM</timestamp>
  <description>Invalid group ID (0).</description>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意：HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getGroups\(\)](#)、[getGroupMembers\(\)](#)

getGroups()

RTMP/E

getGroups(appInst:String) : Object

HTTP

http://www.example.com:1111/admin/getGroups?user=username&passwd=password&appInst=name

特定のアプリケーションインスタンスへのグループ接続のリストを返します。グループ接続は、リモートのエッジサーバーからオリジンサーバーへのマルチプレックス接続です。各グループ接続は、このサーバーのエッジとして動作している別の Flash Media Server に接続しているユーザーが少なくとも 1 人存在することを示します。

使用できるバージョン

Flash Media Server 2.0

パラメータ

appInst パフォーマンス統計の対象となるアプリケーションインスタンスの名前を、`application_name/instance_name` の形式で示す文字列。アプリケーションのデフォルトのインスタンスについてのパフォーマンス統計を取る場合でも、アプリケーション名とインスタンス名の両方をスラッシュ (/) で区切って指定する必要があります。例えば、アプリケーション `ChatApp` のデフォルトのインスタンスを指定する場合は、「`ChatApp/_defInst_`」と指定します。

user 管理者のユーザー名を示す文字列。

passwd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、`level` プロパティの `status`、`code` プロパティの `NetConnection.Call.Success` およびこのアプリケーションに接続するすべてのグループのクライアント ID の配列である `data` プロパティを含む情報オブジェクトを応答として送信します。

HTTP 呼び出しが成功すると、次のような XML を返します。

```
<result>
  <level>status</level>
  <code>NetConnection.Call.Success</code>
  <timestamp>7/24/2007 9:08:59 AM</timestamp>
  <data>
    <name>_defaultRoot_:_defaultVHost_::_2</name>
    <_0>CCAOQVeF</_0>
  </data>
</result>
```

呼び出しが失敗すると、次のような XML を返します。

```
<result>
  <level>error</level>
  <code>NetConnection.Admin.CommandFailed</code>
  <timestamp>7/24/2007 9:10:11 AM</timestamp>
  <description>Application instance is not loaded : vo</description>
</result>
```

注意：HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getGroupMembers\(\)](#)、[getGroupStats\(\)](#)

getInstanceStats()

RTMP/E

`getInstanceStats(appInst:String) : Object`

HTTP

`http://www.example.com:1111/admin/getInstanceStats?user=username&passwd=password&appInst=name`

指定したアプリケーションインスタンスについて、パフォーマンスデータを取得します。

特定のスクリプトのパフォーマンス情報のみが必要な場合には、`getScriptStats()` メソッドを使用します。

Flash Media Server 3.5 以降では、アプリケーションの送信先となるコアを動的に制御できます。コアを制御することで、パフォーマンスカウンタに基づいて複数のコアに負荷を分散することができます。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

applnst パフォーマンス統計が必要なアプリケーションインスタンスの名前を、`application_name/instance_name` の形式で示す文字列。アプリケーション名とインスタンス名をスラッシュ (/) で区切って指定します。アプリケーションのデフォルトインスタンスのパフォーマンス統計を取得する場合にも、2つの名前を指定します。例えば、アプリケーション ChatApp のデフォルトのインスタンスを指定する場合は、「ChatApp/_deflnst_」と指定します。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、`level` プロパティの `status`、`code` プロパティの `NetConnection.Call.Success` およびオブジェクトである `data` プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
<code>launch_time</code>	インスタンスの開始時刻を示す ActionScript の Date オブジェクト。
<code>up_time</code>	インスタンスの実行時間 (秒) を示す数字。
<code>msg_in</code>	アプリケーションインスタンスが処理したメッセージの合計数を示す数字。
<code>msg_out</code>	アプリケーションインスタンスが送信したメッセージの合計数を示す数字。
<code>msg_dropped</code>	アプリケーションインスタンスがドロップしたメッセージの合計数を示す数字。
<code>bytes_in</code>	アプリケーションインスタンスが読み込んだ合計バイト数を示す数字。
<code>bytes_out</code>	アプリケーションインスタンスが書き込んだ合計バイト数を示す数字。
<code>accepted</code>	アプリケーションが許可した接続の合計数を示す数字。
<code>rejected</code>	アプリケーションが拒否した接続要求の合計数を示す数字。
<code>connected</code>	現在アクティブな接続の合計数を示す数字。
<code>total_connects</code>	インスタンス開始時以降のアプリケーションインスタンスへのソケット接続の合計数を示す数字。
<code>total_disconnects</code>	インスタンス開始時以降のアプリケーションインスタンスのソケット接続解除の合計数を示す数字。
<code>script</code>	スクリプトエンジンのパフォーマンスデータを含むオブジェクト。以下は、スクリプトオブジェクトのプロパティです。 <code>time_high_water_mark</code> : スクリプトがイベントの実行に要した最長時間 (秒) を示す数字。 <code>queue_size</code> : スクリプトエンジンのキューに現在入っているイベントの合計数を示す数字。 <code>total_processed</code> : スクリプトエンジンが処理したイベントの合計数を示す数字。 <code>total_process_time</code> : <code>total_processed</code> での数のイベント処理に要した秒数を示す数字。 <code>queue_high_water_mark</code> : キューに入れられたイベントの最大数を示す数字。
<code>normal_connects</code>	通常の接続の合計数を示す数字。
<code>virtual_connects</code>	リモートエッジによる接続の合計数を示す数字。
<code>group_connects</code>	接続されるリモートエッジの合計数を示す数字。

プロパティ	説明
service_connects	サービス接続の合計数を示す数字。
service_requests	要求されたサービスの合計数を示す数字。
admin_connects	管理者による接続の合計数を示す数字。
debug_connects	デバッグ接続の合計数を示す数字。
swf_verification_attempts	SWF 検証を試みた回数を示すカウンタ。確認のためにサーバーに渡された SWF 検証の証明書の合計を表します。接続ごとに複数の証明書が提示される場合があります。
swf_verification_exceptions	SWF 検証で発生した例外の数を示すカウンタ。Application.xml ファイルでの明示的な設定により、特定のユーザーエージェントが SWF 検証要求を省略できるようにする例外が許可されます。例外として許可されるすべての接続がここにカウントされます。
swf_verification_failures	SWF 検証が失敗した回数を示すカウンタ。提示された SWF 検証の証明書がロード済みの証明書と一致しないことが検出されると、失敗になります。失敗はそれぞれ接続の切断に対応します。
swf_verification_unsupported_rejects	拒否された SWF 検証のうち、サポートされない拒否の回数を示すカウンタ。SWF 検証をサポートしないバージョンの Flash Player が、SWF 検証を必要とするアプリケーションに接続すると、サポートされない拒否の数が増えます。また、接続中のクライアントが切断されます。
swf_verification_matches	一致の合計数を示すカウンタ。信頼できる SWF 検証の証明書が提示されると、この数は増加します。接続ごとに複数の一致がある場合があります。
swf_verification_remote_misses	プロキシ/リモートサーバープロセスにおいて、SWF 検証の試行が成功しなかった回数を示すカウンタ。プロキシサーバーまたはリモートサーバーは、SWF 検証の試行を受け取ると、ローカルキャッシュに有効な SWF 検証がないか探します。一致するものが見つからないと、リモートミスを記録し、検証の試行の応答をオリジンサーバーに任せます。
core_id	インスタンスを実行しているコアのコア ID。
pid	インストールを実行しているコアプロセスの pid。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Call.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level>status</level>
  <code>NetConnect.Call.Success</code>
  <timestamp>7/24/2008 7:16:27 PM</timestamp>
  <data>
    <name>_defaultRoot::_defaultVHost::_:0</name>>
    <bw_in>0</bw_in>
    <bw_out>0</bw_out>
    <bytes_in>3073</bytes_in>
    <bytes_out>3453</bytes_out>
    <msg_in>0</msg_in>
    <msg_out>1</msg_out>
    <msg_dropped>0</msg_dropped>
    <total_connects>1</total_connects>
    <total_disconnects>0</total_disconnects>
    <connected>1</connected>
    <normal_connects>1</normal_connects>
    <virtual_connects>0</virtual_connects>
    <group_connects>0</group_connects>
    <service_connects>0</service_connects>
    <service_requests>0</service_requests>
    <admin_connects>0</admin_connects>
```

```

<accepted>1</accepted>
<rejected>0</rejected>
<launch_time>7/24/2008 6:41:10 PM</launch_time>
<pid>26912</pid>
<core_id>0</core_id>
<up_time>2117.079311</up_time>
<mode>local</mode>
<debug>false</debug>
<script>
  <name>_defaultRoot_:_defaultVHost_:::_0</name>
  <time_high_water_mark>22</time_high_water_mark>
  <queue_size>0</queue_size>
  <total_processed>10</total_processed>
  <queue_high_water_mark>1</queue_high_water_mark>
  <total_process_time>44</total_process_time>
</script>
<swf_verification_attempts>0</swf_verification_attempts>
<swf_verification_exceptions>0</swf_verification_exceptions>
<swf_verification_failures>0</swf_verification_failures>
<swf_verification_unsupported_rejects>0</swf_verification_unsupported_rejects>
<swf_verification_matches>0</swf_verification_matches>
<swf_verification_remote_misses>0</swf_verification_remote_misses>
</data>
</result>

```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```

<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>

```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意：HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getActiveInstances\(\)](#)、[getAppStats\(\)](#)、[getScriptStats\(\)](#)

getIOStats()

RTMP/E

getIOStats() : Object

HTTP

http://www.example.com:1111/admin/getIOStats?auser=username&apswd=password

接続しているアダプタのネットワーク出入力の特性についての詳細情報を返します。

この操作を実行できるのは、サーバー管理者のみです。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびオブジェクトである data プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
msg_in	アダプタが処理したメッセージの合計数を示す数字。
msg_out	アダプタから送信されたメッセージの合計数を示す数字。
bytes_in	アダプタが読み取った合計バイト数を示す数字。
bytes_out	アダプタが書き込んだ合計バイト数を示す数字。
reads	システムの読み取り呼び出しの合計数を示す数字。
writes	システムの書き込みの合計数を示す数字。
connected	アダプタへの現在アクティブなソケット接続数を示す数字。
total_connects	アダプタ開始以降のアダプタへのソケット接続の合計数を示す数字。
total_disconnects	アダプタのソケット接続解除の合計数を示す数字。
msg_dropped	ドロップされたメッセージの合計数を示す数字。
tunnel_responses	トンネリング応答の累計数を示す数字。
normal_connects	通常の接続の合計数を示す数字。
virtual_connects	リモートエッジによる接続の合計数を示す数字。
group_connects	接続されるリモートエッジの合計数を示す数字。
service_connects	サービス接続の合計数を示す数字。
service_requests	要求されたサービスの合計数を示す数字。
admin_connects	管理者による接続の合計数を示す数字。
debug_connects	デバッグ接続の合計数を示す数字。
server_bytes_in	Number : SSAS NetConnection によって読み込まれる rtmp バイト数の合計。
server_bytes_out	Number : SSAS NetConnection によって送信される rtmp バイト数の合計。
swf_verification_attempts	SWF 検証を試みた回数を示すカウンタ。確認のためにサーバーに渡された SWF 検証の証明書の合計を表します。接続ごとに複数の証明書が提示される場合があります。
swf_verification_exceptions	SWF 検証で発生した例外の数を示すカウンタ。Application.xml ファイルでの明示的な設定により、特定のユーザーエージェントが SWF 検証要求を省略できるようにする例外が許可されます。例外として許可されるすべての接続がここにカウントされます。
swf_verification_failures	SWF 検証が失敗した回数を示すカウンタ。提示された SWF 検証の証明書がロード済みの証明書と一致しないことが検出されると、失敗になります。失敗はそれぞれ接続の切断に対応します。

プロパティ	説明
swf_verification_unsupported_rejects	拒否された SWF 検証のうち、サポートされない拒否の回数を示すカウンタ。SWF 検証をサポートしないバージョンの Flash Player が、SWF 検証を必要とするアプリケーションに接続すると、サポートされない拒否の数が増えます。また、接続中のクライアントが切断されません。
swf_verification_matches	一致の合計数を示すカウンタ。信頼できる SWF 検証の証明書が提示されると、この数は増加します。接続ごとに複数の一致がある場合があります。
swf_verification_remote_misses	プロキシ/リモートサーバープロセスにおいて、SWF 検証の試行が成功しなかった回数を示すカウンタ。プロキシサーバーまたはリモートサーバーは、SWF 検証の試行を受け取ると、ローカルキャッシュに有効な SWF 検証がないか探します。一致するものが見つからないと、リモートミスを記録し、検証の試行の応答をオリジンサーバーに任せます。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Call.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

getLicenseInfo()

RTMP/E

getLicenseInfo() : Object

HTTP

http://www.example.com:1111/admin/getLicenseInfo?auser=username&apswd=password

ライセンスで許可されている最大帯域幅、接続、アダプタ、仮想ホストおよび CPU の最大数など、ライセンスに関する完全な情報を取得します。すべてのライセンスについてライセンス情報のサマリーが作成され、続いて各ライセンスの情報が作成されます。この呼び出しは、ライセンスキー (「ライセンス番号」とも呼ばれます) に関する情報およびライセンスファイルを返します。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびオブジェクトである data プロパティを含む情報オブジェクトを応答として送信します。data オブジェクトには、HTTP 呼び出しで返される XML の data エレメントと同じプロパティが含まれます。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Call.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します (この XML はライセンスファイル用です)。

注意: 値 -1 は制限がないことを意味します。

```
<result>
  <level>status</level>
  <code>NetConnection.Call.Success</code>
  <timestamp>10/31/2007 1:36:05 PM</timestamp>
  <data>
    <name>Flash Media Interactive Server</name>
    <version>3.0.0</version>
    <build>d1107</build>
    <copyright>Copyright (c) 1993-2007 Adobe Systems Incorporated.
All Rights Reserved.</copyright>
    <max_connections>unlimited</max_connections>
    <max_edge_connections>unlimited</max_edge_connections>
    <max_adaptors>3</max_adaptors>
    <max_vhosts>unlimited</max_vhosts>
    <max_cpu>-1</max_cpu>
    <max_bandwidth>unlimited</max_bandwidth>
    <max_core_process>-1</max_core_process>
    <proxy>enabled</proxy>
    <custom_plugin>enabled</custom_plugin>
    <license_stacking>yes</license_stacking>
    <apps>enabled</apps>
    <web-proxy>enabled</web-proxy>
    <screenshare>enabled</screenshare>
    <app-isolation>enabled</app-isolation>
    <inst-isolation>enabled</inst-isolation>
    <license_files>
      <_0>
        <filename>license0108.lic</filename>
        <description>Beta version</description>
        <type>FlashMediaServer</type>
        <id></id>
        <account-id></account-id>
        <expires>Jan 01, 2008</expires>
        <limits>
          <connections>
            <value>unlimited</value>
          </connections>
          <bandwidth>
            <value>unlimited</value>
          </bandwidth>
        </limits>
      </_0>
    </license_files>
  </data>
</result>
```

```

        <vhosts>
          <value>unlimited</value>
        </vhosts>
      </limits>
    <features>
      <apps>
        <value>enabled</value>
      </apps>
      <proxy>
        <value>enabled</value>
      </proxy>
      <screenshare>
        <value>enabled</value>
      </screenshare>
      <web-proxy>
        <value>enabled</value>
      </web-proxy>
      <app-isolation>
        <value>enabled</value>
      </app-isolation>
      <inst-isolation>
        <value>enabled</value>
      </inst-isolation>
    </features>
  </_0>
</license_files>
<active_profile>N/A</active_profile>
<key_details></key_details>
</data>
</result>

```

呼び出しが成功すると、次の構造の XML を返します（この XML はライセンスキー用です）。

```

<result>
  <level>status</level>
  <code>NetConnection.Call.Success</code>
  <timestamp>10/31/2007 1:34:17 PM</timestamp>
  <data>
    <name>Flash Media Interactive Server</name>
    <version>3.0.0</version>
    <build>d1107</build>
    <copyright>Copyright (c) 1993-2007 Adobe Systems Incorporated.
All Rights Reserved.</copyright>
    <max_connections>unlimited</max_connections>
    <max_edge_connections>unlimited</max_edge_connections>
    <max_adaptors>3</max_adaptors>
    <max_vhosts>unlimited</max_vhosts>
    <max_cpu>4</max_cpu>
    <max_bandwidth>unlimited</max_bandwidth>
    <max_core_process>-1</max_core_process>
    <proxy>enabled</proxy>
    <custom_plugin>enabled</custom_plugin>
    <license_stacking>yes</license_stacking>
    <apps>enabled</apps>
    <web-proxy>enabled</web-proxy>
    <app-isolation>enabled</app-isolation>
    <inst-isolation>enabled</inst-isolation>
    <license_files>
    </license_files>
    <active_profile>N/A</active_profile>
    <key_details>
      <_0>
        <key></key>

```

```

    <product_code>N/A</product_code>
    <type>N/A</type>
    <family>N/A</family>
    <edition>Flash Media Interactive Server</edition>
    <edition_text>N/A</edition_text>
    <max_connections>unlimited</max_connections>
    <max_adaptors>3</max_adaptors>
    <max_vhosts>unlimited</max_vhosts>
    <max_cpu>4</max_cpu>
    <max_bandwidth>unlimited</max_bandwidth>
    <expires></expires>
    <valid>true</valid>
  </_0>
  <_1>
    <key></key>
    <product_code>N/A</product_code>
    <type>N/A</type>
    <family>N/A</family>
    <edition>Flash Media Streaming Server</edition>
    <edition_text>N/A</edition_text>
    <max_connections>unlimited</max_connections>
    <max_adaptors>1</max_adaptors>
    <max_vhosts>unlimited</max_vhosts>
    <max_cpu>4</max_cpu>
    <max_bandwidth>unlimited</max_bandwidth>
    <expires></expires>
    <valid>true</valid>
  </_1>
</key_details>
</data>
</result>

```

呼び出しが失敗すると、次の構造の XML を返します。

```

<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>

```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意： HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

getLiveStreams()

RTMP/E

getLiveStreams (appInst:String) : Object

HTTP

http://www.example.com:1111/admin/getLiveStreams?auser=username&apswd=password&appInst=name

指定したアプリケーションインスタンスにパブリッシュされているすべてのライブストリームの名前を含む文字列の配列を取得します。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

appInst アプリケーションインスタンスの名前を、`application_name/instance_name` の形式で示す文字列。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、`level` プロパティの `status`、`code` プロパティの `NetConnection.Call.Success` および指定したアプリケーションインスタンスにパブリッシュされているすべてのライブストリームの名前を含む文字列の配列である `data` プロパティを含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、`level` プロパティの `error` および `code` プロパティの `NetConnection.Call.Failed`、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む `description` プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

`data` エLEMENTには、RTMP/E セクションに記載された `data` オブジェクトの各プロパティのエLEMENTがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML ELEMENTには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getLiveStreamStats\(\)](#)

getLiveStreamStats()

RTMP/E

```
getLiveStreamStats(appInst:String, stream:String) : Object
```

HTTP

```
http://www.example.com:1111/admin/getLiveStreamStats?auser=username&apswd=password&appInst=name&stream=name
```

ライブストリームの詳細情報を返します。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

applnst アプリケーションインスタンスの名前を、`application_name/instance_name` の形式で示す文字列。

stream ストリームの名前を示す文字列。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、`level` プロパティの `status`、`code` プロパティの `NetConnection.Call.Success` およびオブジェクトである `data` プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
<code>publisher</code>	パブリッシュする側の統計を示すオブジェクト。オブジェクトには、次のプロパティがあります。 <code>name</code> : パブリッシュされたライブストリーム名を示す文字列。 <code>time</code> : ストリームがパブリッシュされた日時を示す <code>Date</code> オブジェクト。このプロパティは <code>publish_time</code> と重複しており、下位互換性のために存在します。 <code>type</code> : パブリッシュ元のストリームのタイプを示す文字列。値は「publishing」となります。 <code>client</code> : パブリッシュ元のクライアント ID を示す数字。 <code>stream_id</code> : パブリッシュ元のストリーム ID を示す数字。 <code>publish_time</code> : ストリームがパブリッシュされた日時を示す <code>Date</code> オブジェクト。 <code>client_type</code> : パブリッシュするクライアントのタイプを示す文字列。
<code>subscribers</code>	加入者の統計を示す配列。配列には、以下のプロパティを含むオブジェクトである <code>subscriber</code> プロパティが含まれます。 <code>client</code> : ユーザー ID を示す数字。 <code>subscribe_time</code> : ユーザーがストリームをサブスクライブした日時を示す <code>Date</code> オブジェクト。

呼び出しが失敗すると、サーバーは、`level` プロパティの `error` および `code` プロパティの `NetConnection.Call.Failed`、またはより具体的な値（存在する場合）を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む `description` プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

`data` エレメントには、RTMP/E セクションに記載された `data` オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される `Object` のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getLiveStreams\(\)](#)

getMsgCacheStats()

RTMP/E

getMsgCacheStats() : Object

HTTP

http://www.example.com:1111/admin/getMsgCacheStats?auser=username&apswd=password

サーバーの TCMessage キャッシュの統計を返します。このコマンドを実行できるのは、サーバー管理者のみです。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびオブジェクトである data プロパティを含む情報オブジェクトを応答として送信します。data オブジェクトには、以下のメッセージキャッシュの統計が含まれます。

プロパティ	説明
allocated	割り当てられているメッセージオブジェクトの合計数を示す数字。
reused	使用されているオブジェクトの合計数を示す数字。
size	現在のサイズを示す数字。

プロパティ	説明
thread_count	使用されているスレッド単位のプールの合計数
units	次の プロパティを持つオブジェクトです。 global_size 使用可能なグローバルプールのサイズ。 thread_size 使用可能なスレッド単位のプールの合計サイズ。 size 使用可能なグローバルプールとスレッド単位のプールの合計サイズ。 reused 再使用されたメッセージの合計数。 allocated ヒープから割り当てられたメッセージの合計数。 released ヒープにリリースバックされたメッセージの合計数。 relocated 再割り当てされたメッセージの合計数。 bulk_allocated グローバルプールから割り当てられたメッセージの合計数。 bulk_released グローバルプールにリリースバックされたメッセージの合計数。 huge_allocated 割り当てられた大きいサイズ (16KB 以上) のメッセージの合計数。 huge_released リリースされた大きいサイズ (16KB 以上) のメッセージの合計数。
bytes	次の プロパティを持つオブジェクトです。 global_size 使用可能なグローバルプールのサイズ。 thread_size 使用可能なスレッド単位のプールの合計サイズ。 size_total 使用可能なグローバルプールとスレッド単位のプールのサイズ。 reused 再使用された合計バイト数。 allocated ヒープから割り当てられた合計バイト数。 released ヒープにリリースバックされた合計バイト数。 relocated 再割り当てされた合計バイト数。 bulk_allocated グローバルプールから割り当てられた合計バイト数。 bulk_released グローバルプールにリリースバックされた合計バイト数。 huge_allocated 割り当てられた大きいサイズ (16KB 以上) のメッセージの合計バイト数。 huge_released リリースされた大きいサイズ (16KB 以上) のメッセージの合計バイト数。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意： HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

getNetStreams()

RTMP/E

`getNetStreams(appInst:String) : Object`

HTTP

`http://www.example.com:1111/admin/getNetStreams?auser=username&apswd=password&appInst=name`

指定したアプリケーションインスタンスに接続中のすべてのネットワークストリームに対してサーバーが割り当てた ID を示す数字の配列を返します。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

`appInst` アプリケーションインスタンスの名前を、`application_name/instance_name` の形式で示す文字列。

`auser` 管理者のユーザー名を示す文字列。

`apswd` 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、`level` プロパティの `status`、`code` プロパティの `NetConnection.Call.Success` および数字の配列である `data` プロパティを含む情報オブジェクトを応答として送信します。数字列は、指定したアプリケーションインスタンスに接続中のすべてのネットワークストリームにサーバーが割り当てた ID を示します。

呼び出しが失敗すると、サーバーは、`level` プロパティの `error` および `code` プロパティの `NetConnection.Call.Failed`、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む `description` プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

`data` エレメントには、RTMP/E セクションに記載された `data` オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される `Object` のプロパティと同じ情報が含まれます。

注意： HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getNetStreamStats\(\)](#)

getNetStreamStats()

RTMP/E

```
getNetStreamStats(appInst:String, streamids:Number) : Object
```

HTTP

```
http://www.example.com:1111/admin/getNetStreamStats?auser=username&apswd=password&appInst=name&streamids=number
```

指定したアプリケーションインスタンスに接続しているネットワークストリームの詳細情報を取得します。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

appInst アプリケーションインスタンスの名前を、**application_name/instance_name** の形式で示す文字列。

streamids ネットワークストリームの ID を示す数字、またはネットワークストリーム ID を示す数字の配列。現在接続しているすべてのネットワークストリームの情報を取得するには、**streamids** パラメータに値 **-1** を指定します。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、**level** プロパティの **status**、**code** プロパティの **NetConnection.Call.Success** およびネットワークストリームの統計の配列である **data** プロパティを含む情報オブジェクトを応答として送信します。配列内の各エレメントは、以下のプロパティを含むオブジェクトです。

プロパティ	名前
streamid	ストリーム ID を示す数字。
name	ストリーム名の文字列。ストリームがアイドル状態の場合は empty となります。
type	ストリームのタイプを示す文字列。可能な値は以下のとおりです。 "idle" "publishing" "playing live" "play recorded"
client	ユーザー ID を示す数字。
time	ActionScript の Date オブジェクト。可能な値は以下のとおりです。 type の値が idle であるとき、値は 0 になります。 type の値が publishing であるとき、値はストリームがパブリッシュされた時刻となります。 type の値が playing live であるとき、値はストリーム再生が開始した時刻となります。 type の値が play recorded であるとき、値はストリーム再生が開始した時刻となります。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Call.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getNetStreams\(\)](#)

getRecordedStreams()

RTMP/E

```
getRecordedStreams(appInst:String) : Object
```

HTTP

```
http://www.example.com:1111/admin/getRecordedStreams?user=username&apswd=password&appInst=name
```

特定のアプリケーションインスタンスから再生中のすべての記録ストリームの名前を含む配列を返します。

使用できるバージョン

Flash Media Server 2.0

パラメータ

appInst アプリケーションまたはアプリケーションインスタンスの名前を、application_name[/instance_name] の形式で示す文字列。

user 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびすべての記録ストリームの名前の配列である data プロパティを含む情報オブジェクトを応答として送信します。記録ス

トリームの拡張名が使用されます。名前は `keyName*type:streamName` の形式でエンコーディングされます。keyName は仮想キー、type は flv、mp3 などのストリームタイプ、streamName はストリームのテキスト名を表します。

次の表は、プロパティについて説明しています。

プロパティ	名前
streamName	記録ストリームの名前を示す文字列。
type	記録ストリームのタイプを示す文字列。
keyName	記録ストリームの仮想キーを示す文字列。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティの要素がネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意： HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getRecordedStreamStats\(\)](#)

getRecordedStreamStats()

RTMP/E

```
getRecordedStreamStats(appInst:String, stream:String) : Object
```

HTTP

```
http://www.example.com:1111/admin/getRecordedStreamStats?user=username&apswd=password&appInst=name&stream=name
```

記録ストリームの詳細情報を返します。

使用できるバージョン

Flash Media Server 2.0

パラメータ

appInst アプリケーションインスタンスの名前を、application_name/instance_name の形式で示す文字列。

stream ストリームの名前を示す文字列。ストリームの仮想キーやタイプがデフォルトではない場合は、ストリーム名の一部としてエンコーディングする必要があります。キーとタイプは、`key?type:name` の形式でエンコーディングします (例、`on2key?flv:myStream`)。

user 管理者のユーザー名を示す文字列。

passwd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、`level` プロパティの `status`、`code` プロパティの `NetConnection.Call.Success` および次のプロパティを含むオブジェクトである `data` プロパティを含む情報オブジェクトを応答として送信します。

プロパティ	名前
<code>cache_bytes</code>	記録ストリームのキャッシュされた合計バイト数を示す数字。
<code>cache_segments</code>	記録ストリームのキャッシュされたセグメント数を示す数字。
<code>cache_hits</code>	キャッシュ内の記録ストリームのヒット数を示す数字。
<code>cache_misses</code>	記録ストリームのキャッシュミスの数を示す数字。
<code>modified_time</code>	ファイルの最終変更日を示す ActionScript の Date オブジェクト。
<code>size</code>	記録ファイルのバイト数を示す数字。
<code>length</code>	ファイルの長さ (秒) を示す数字。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

`data` エLEMENT には、RTMP/E セクションに記載された `data` オブジェクトの各プロパティのエLEMENTがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エLEMENT には、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getRecordedStreams\(\)](#)

getScriptStats()

RTMP/E

getScriptStats(appInst:String) : Object

HTTP

http://www.example.com:1111/admin/getScriptStats?user=username&apswd=password&appInst=name

アプリケーションの指定したインスタンスで実行中のスクリプトのパフォーマンスデータを取得します。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

appInst アプリケーションインスタンスの名前を、application_name/instance_name の形式で示す文字列。

user 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびオブジェクトである data プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
time_high_water_mark	スクリプトがイベントの実行に要した最長時間 (秒) を示す数字。
queue_size	スクリプトエンジンのキューに現在入っているイベントの合計数を示す数字。
total_processed	スクリプトエンジンが処理したイベントの合計数を示す数字。
total_process_time	total_processed での数のイベント処理に要した秒数を示す数字。
queue_high_water_mark	キューに入れられたイベントの最大数を示す数字。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Call.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

getServerStats()

RTMP/E

getServerStats() : Object

HTTP

http://www.example.com:1111/admin/getServerStats?user=username&apswd=password

サーバーステータスと、サーバーの動作に関する統計を取得します。報告される情報としては、サーバーが実行された時間数、I/O、メッセージキャッシュの統計などがあります。

この操作を実行できるのは、サーバー管理者のみです。

サーバーの入出力特性の情報のみが必要な場合には、getIOStats() メソッドを使用します。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

user 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびオブジェクトである data プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
launchTime	サーバーが開始した日時を示す ActionScript の Date オブジェクト。
up_time	サーバーの実行時間 (秒) を示す数字。
io	<p>入出力の統計。以下のプロパティを使用したオブジェクトとして返されます。</p> <p>bw_in：受信帯域幅を示す数字。</p> <p>bw_out：送信帯域幅を示す数字。</p> <p>msg_in：サーバーが処理したメッセージの合計数を示す数字。</p> <p>msg_out：サーバーが送信したメッセージの合計数を示す数字。</p> <p>bytes_in：サーバーが読み取った合計バイト数を示す数字。</p> <p>bytes_out：サーバーが書き込んだ合計バイト数を示す数字。</p> <p>reads：システムの読み取り呼び出しの合計数を示す数字。</p> <p>writes：システムの書き込みの合計数を示す数字。</p> <p>connected：サーバーへのアクティブなソケット接続の合計数を示す数字。</p> <p>total_connects：サーバーへのソケット接続の合計数を示す数字。</p> <p>normal_connects：サーバーへの Flash Player 接続の数を示す数字。</p> <p>virtual_connects：グループでプロキシされている仮想接続の数を示す数字。</p> <p>group_connects：接続されているグループの数を示す数字。</p> <p>service_connects：サービス接続の数を示す数字。</p> <p>admin_connects：管理接続の数を示す数字。</p> <p>debug_connects：デバッグ接続の数を示す数字。</p> <p>total_disconnects：サーバーのソケット接続解除の合計数を示す数字。</p> <p>service_requests：サービスに対して行われた要求の数を示す数字。</p> <p>working_threads：動作しているスレッドの数を示す数字。</p> <p>msg_dropped：サーバーにより削除されたメッセージの合計数を示す数字。</p> <p>swf_verification_attempts：SWF 検証を試みた回数を示すカウンタ。確認のためにサーバーに渡された SWF 検証の証明書の合計を表します。接続ごとに複数の証明書が提示される場合があります。</p> <p>swf_verification_failures：SWF 検証が失敗した回数を示すカウンタ。提示された SWF 検証の証明書がロード済みの証明書と一致しないことが検出されると、失敗になります。失敗はそれぞれ接続の切断に対応します。</p> <p>swf_verification_exceptions：SWF 検証で発生した例外の数を示すカウンタ。Application.xml ファイルでの明示的な設定により、特定のユーザーエージェントが SWF 検証要求を省略できるようにする例外が許可されます。例外として許可されるすべての接続がここにカウントされます。</p> <p>swf_verification_unsupported_rejects：SWF 検証のサポートされていない拒否の数を示すカウンタ。SWF 検証をサポートしないバージョンの Flash Player が、SWF 検証を必要とするアプリケーションに接続すると、サポートされない拒否の数が増えます。また、接続中のクライアントが切断されます。</p> <p>swf_verification_matches：一致の合計数を示すカウンタ。信頼できる SWF 検証の証明書が提示されると、この数は増加します。接続ごとに複数の一致がある場合があります。</p> <p>swf_verification_remote_misses：プロキシ/リモートサーバープロセスにおいて、SWF 検証の試行が成功しなかった回数を示すカウンタ。プロキシサーバーまたはリモートサーバーは、SWF 検証の試行を受け取ると、ローカルキャッシュに有効な SWF 検証がないか探します。一致するものが見つからないと、リモートミスを記録し、検証の試行の応答をオリジンサーバーに任せます。</p>
msg_cache	<p>Flash Media Server メッセージパケットキャッシュの統計。以下のプロパティを使用したオブジェクトとして返されます。</p> <p>allocated：割り当てられてたメッセージオブジェクトの合計数を示す数字。</p> <p>reused：再使用されたオブジェクトの合計数を示す数字。</p> <p>size：キャッシュサイズ (メッセージパケット数) を示す数字。</p>

プロパティ	説明
memory_Usage	Number. Microsoft Windows NT 4.0 では、最近使用した物理メモリ 1,000 ページ分の百分率を概算で示します。 Windows 2000 または Windows XP では、使用中の合計物理メモリの概算の比率を示します。
cpu_Usage	システム全体ではなく、Flash Media Server プロセスによる CPU 使用率の概算の比率を示す数字。
server_bytes_in	Server-Side ActionScript NetConnection により読み取られた RTMP バイトの合計数を示す数字。
server_bytes_out	Server-Side ActionScript NetConnection により送信された RTMP バイトの合計数を示す数字。
num_cores	コアプロセスの合計数を示す数字。
physical_mem	使用されている物理メモリの合計を示す数字。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Call.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getIOStats\(\)](#)

getSharedObjects()

RTMP/E

```
getSharedObjects(appInst:String) : Object
```

HTTP

```
http://www.example.com:1111/admin/getSharedObjects?user=username&passwd=password&appInst=name
```

現在アクティブなすべての共有オブジェクトの名前を取得します。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

applnst アプリケーションインスタンスの名前を、`application_name/instance_name` の形式で示す文字列。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、`level` プロパティの `status`、`code` プロパティの `NetConnection.Call.Success` およびオブジェクトである `data` プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
<code>persistent</code>	永続的な共有オブジェクトの名前を含む文字列の配列。
<code>volatile</code>	永続的でない共有オブジェクトの名前を含む文字列の配列。

呼び出しが失敗すると、サーバーは、`level` プロパティの `error` および `code` プロパティの `NetConnection.Call.Failed`、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む `description` プロパティが含まれているオブジェクトもあります。

存在しないアプリケーションまたはインスタンスの名前を指定すると、呼び出しは失敗します。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

`data` エLEMENT には、RTMP/E セクションに記載された `data` オブジェクトの各プロパティのエLEMENTがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エLEMENT には、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getSharedObjectStats\(\)](#)

getSharedObjectStats()

RTMP/E

```
getSharedObjectStats(appInst:String, sharedObject:String, persistent:Boolean) : Object
```

HTTP

```
http://www.example.com:1111/admin/getSharedObjectStats?auser=username&apswd=password&appInst=name&sharedObject=name&persistent=value
```

共有オブジェクトの詳細情報を取得します。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

appInst アプリケーションインスタンスの名前を、application_name/instance_name の形式で示す文字列。

SharedObject 共有オブジェクトの名前を示す文字列。すべてのアクティブな共有オブジェクトの名前を取得するには、getSharedObjects() メソッドを使用します。

persistent ブール値。永続の場合は true、非永続の場合は false。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびオブジェクトである data プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
version	共有オブジェクトのバージョン番号を示す数字。
num_properties	共有オブジェクトのプロパティの合計数を示す数字。
msg_in	共有オブジェクトが受信したメッセージの合計数を示す数字。
msg_out	共有オブジェクトから送信された合計数を示す数字。
total_connects	共有オブジェクトへの接続の合計数を示す数字。
total_disconnects	共有オブジェクトの接続解除の合計数を示す数字。
connected	アクティブな加入者の数を示す数字。
resync_depth	再同期化まで保持されるバージョン間の相違の最大値を示す数字。サーバーのバージョン番号とクライアントのバージョン番号の間の相違が resync_depth の値よりも大きい場合、Flash Media Server からはバージョン間の変更部分のみが送信されます。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Call.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意：HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getSharedObjects\(\)](#)

getUsers()

RTMP/E

```
getUsers (appInst:String) : Object
```

HTTP

```
http://www.example.com:1111/admin/getUsers?user=username&apswd=password&appInst=name
```

指定したアプリケーションインスタンスに接続しているすべてのユーザーに対してサーバーが割り当てた ID を示す文字列の配列を取得します。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

appInst アプリケーションインスタンスの名前を、`application_name/instance_name` の形式で示す文字列。

user 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの `status`、code プロパティの `NetConnection.Call.Success` および指定したアプリケーションインスタンスに接続しているユーザーの ID を示す文字列の配列である data プロパティを含む情報オブジェクトを応答として送信します。ユーザー ID は、サーバーが割り当てます。

呼び出しが失敗すると、サーバーは、level プロパティの `error` および code プロパティの `NetConnection.Call.Failed`、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む `description` プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data>
    <names></names>
  </data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意： HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getUserStats\(\)](#)

getUserStats()

RTMP/E

```
getUserStats(appInst:String, userid:String) : Object
```

HTTP

```
http://www.example.com:1111/admin/getUserStats?user=username&apswd=password&appInst=name&userid=ID
```

指定したユーザーの接続についてのパフォーマンスデータを取得します。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

appInst アプリケーションインスタンスの名前を、application_name/instance_name の形式で示す文字列。

userid サーバーが割り当てたユーザー ID を示す文字列。getUsers() メソッドを使用すると、ユーザー ID を取得できます。

user 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびオブジェクトである data プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
connect_time	アプリケーションの指定したインスタンスにユーザーが接続している時間 (秒) を示す ActionScript の Date オブジェクト。
msg_in	ユーザーが処理したメッセージの合計数を示す数字。
msg_out	ユーザーから送信されたメッセージの合計数を示す数字。
msg_dropped	ユーザーがドロップしたメッセージの合計数を示す数字。
bytes_in	ユーザーが読み取った合計バイト数を示す数字。
bytes_out	ユーザーが書き込んだ合計バイト数を示す数字。
msg_queue	クライアントのメッセージキューの統計を含むオブジェクト msg_queue には、以下のプロパティが含まれます。 total_queues : このクライアントのキューの合計数。 audio : すべてのオーディオキューに含まれるオーディオメッセージの合計数。 video : すべてのビデオキューに含まれるビデオメッセージの合計数。 other : other キューに含まれるコマンドおよびデータメッセージの合計数。
protocol	クライアントがサーバーに接続するために使用するプロトコルを示す文字列 (rtmp、rtmpe、rtmpt または rtmpe)。
stream_ids	ストリーム ID を示す数字の配列。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Call.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[getUsers\(\)](#)

getVHosts()

RTMP/E

getVHosts([adaptor:String]) : Object

HTTP

http://www.example.com:1111/admin/getVHosts?auser=username&apswd=password [&adaptor=name]

指定したアダプタに定義される仮想ホストの配列を返します。このメソッドを呼び出すことができるのは、サーバー管理者のみです。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

adaptor アダプタのユーザー名を示す文字列。指定しない場合は、「_defaultRoot_」が使用されます。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success およびすべての仮想ホストの名前を含む文字列の配列である data プロパティを含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Call.Failed、またはより具体的な値（存在する場合）を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意：HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

getVHostStats()

RTMP/E

```
getVHostStats([adaptor:String, vhost:String]) : Object
```

HTTP

```
http://www.example.com:1111/admin/getVHostStats?auser=username&apswd=password&adaptor=name&vhost=name
```

指定した仮想ホストのすべてのアプリケーションのすべてのインスタンスについて包括的なパフォーマンスデータを返します。このメソッドを呼び出すことができるのは、サーバー管理者のみです。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

adaptor アダプタのユーザー名を示す文字列。指定しない場合は、_defaultRoot_ が使用されます。

vhost 仮想ホストのユーザー名を示す文字列。指定しない場合は、_defaultVHost_ が使用されます。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success および仮想ホストのパフォーマンスデータを含むオブジェクトである data プロパティを含む情報オブジェクトを応答として送信します。オブジェクトには、次のプロパティがあります。

プロパティ	説明
msg_in	仮想ホストが処理したメッセージの合計数を示す数字。
msg_out	仮想ホストが送信したメッセージの合計数を示す数字。
msg_dropped	仮想ホストによって欠落したメッセージの合計数を示す数字。
bytes_in	仮想ホストが読み取った合計バイト数を示す数字。
bytes_out	仮想ホストが書き込んだ合計バイト数を示す数字。
accepted	仮想ホストが許可した接続の合計数を示す数字。
rejected	仮想ホストが拒否した接続要求の合計数を示す数字。
connected	現在アクティブな接続の合計数を示す数字。
total_apps	インスタンスが作成されたアプリケーションの合計数を示す数字。
total_connects	サーバーへの接続の合計数を示す数字。
total_disconnects	サーバーの接続解除の合計数を示す数字。
total_instances_loaded	ロードされたインスタンスの合計数を示す数字。 このプロパティは、ロードされたアクティブなインスタンスの合計数ではありません。ロードされたアクティブなインスタンスの数を取得するには、total_instances_loaded の値から total_instances_unloaded の値を引きます。
total_instances_unloaded	アンロードされたインスタンスの合計数を示す数字。
server_bytes_in	Number : SSAS NetConnection によって読み込まれる rtmp バイト数の合計

プロパティ	説明
server_bytes_out	Number : SSAS NetConnection によって送信される rtmp バイト数の合計
bw_in	現在の入力帯域幅 (bps) を示す数字。
bw_out	現在の出力帯域幅 (bps) を示す数字。
normal_connects	通常の接続の合計数を示す数字。
virtual_connects	リモートエッジによる接続の合計数を示す数字。
group_connects	接続されるリモートエッジの合計数を示す数字。
service_connects	サービス接続の合計数を示す数字。
service_requests	要求されたサービスの合計数を示す数字。
admin_connects	管理者による接続の合計数を示す数字。
debug_connects	デバッグ接続の合計数を示す数字。
swf_verification_attempts	SWF 検証を試みた回数を示すカウンタ。確認のためにサーバーに渡された SWF 検証の証明書の合計を表します。接続ごとに複数の証明書が提示される場合があります。
swf_verification_exceptions	SWF 検証で発生した例外の数を示すカウンタ。Application.xml ファイルでの明示的な設定により、特定のユーザーエージェントが SWF 検証要求を省略できるようにする例外が許可されます。例外として許可されるすべての接続がここにカウントされます。
swf_verification_failures	SWF 検証が失敗した回数を示すカウンタ。提示された SWF 検証の証明書がロード済みの証明書と一致しないことが検出されると、失敗になります。失敗はそれぞれ接続の切断に対応します。
swf_verification_unsupported_rejects	拒否された SWF 検証のうち、サポートされない拒否の回数を示すカウンタ。SWF 検証をサポートしないバージョンの Flash Player が、SWF 検証を必要とするアプリケーションに接続すると、サポートされない拒否の数が増えます。また、接続中のクライアントが切断されます。
swf_verification_matches	一致の合計数を示すカウンタ。信頼できる SWF 検証の証明書が提示されると、この数は増加します。接続ごとに複数の一致がある場合があります。
swf_verification_remote_misses	プロキシ/リモートサーバープロセスにおいて、SWF 検証の試行が成功しなかった回数を示すカウンタ。プロキシサーバーまたはリモートサーバーは、SWF 検証の試行を受け取ると、ローカルキャッシュに有効な SWF 検証がないか探します。一致するものが見つからないと、リモートミスを記録し、検証の試行の応答をオリジンサーバーに任せます。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Call.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

ping()

RTMP/E

ping() : Object

HTTP

http://www.example.com:1111/admin/ping?user=username&passwd=password

サーバーが実行していることを確認します。サーバーは状態メッセージを使用して応答します。このメソッドを使用すると、サーバーの状態を確認できます。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

user 管理者のユーザー名を示す文字列。

passwd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status、code プロパティの NetConnection.Call.Success および Date オブジェクトである timestamp プロパティを含む情報オブジェクトを応答として送信します。Date オブジェクトは、メソッドが実行された時刻を示します。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Call.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意： HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

reloadApp()

RTMP/E

```
reloadApp(appInst:String) : Object
```

HTTP

```
http://www.example.com:1111/admin/reloadApp?user=username&passwd=password&appInst=name
```

アプリケーションインスタンスが実行中の場合は終了し、再ロードします。すべてのユーザーとの接続は解除されます。

注意： アプリケーション名を指定すると、_definst_ インスタンスが再ロードされます。

メソッド実行後、ユーザーはアプリケーションまたはアプリケーションインスタンスに再接続する必要があります。

このメソッドを呼び出すと、アプリケーションインスタンスを事前にロードすることができます。また、アプリケーションの設定が変更されたり、アプリケーションに関連付けられているスクリプトが変更されたときにこのメソッドを呼び出すと、アプリケーションインスタンスを再ロードできます。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

appInst アプリケーションまたはアプリケーションインスタンスの名前を、application_name[/instance_name] の形式で示す文字列。デフォルトのアプリケーションインスタンスを再ロードするには、アプリケーション名のみを指定します。

user 管理者のユーザー名を示す文字列。

passwd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは level プロパティの status および code プロパティの NetConnection.Call.Success を含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Admin.Command.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

data エレメントには、RTMP/E セクションに記載された data オブジェクトの各プロパティのエレメントがネストされます。呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[unloadApp\(\)](#)

removeAdmin()

RTMP/E

```
removeAdmin(user:String [,scope:String]) : Object
```

HTTP

```
http://www.example.com:1111/admin/removeAdmin?auser=username&apswd=password&user=name [&scope=scope]
```

システムから管理者を削除します。指定したパラメータに応じて、サーバー管理者または仮想ホスト管理者を削除できます。システムから管理者を削除できるのは、サーバー管理者のみです。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

user 削除する管理者のユーザー名を示す文字列。

scope 削除する管理者を示す文字列。

接続している仮想ホストから仮想ホスト管理者を削除するには、このパラメータを省略します。別の仮想ホストから仮想ホスト管理者を削除するには、仮想ホストを `adaptor_name/virtual_hostname` の形式で指定します。

サーバー管理者を削除するには、「server」と指定します。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは `level` プロパティの `status` および `code` プロパティの `NetConnection.Call.Success` を含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、`level` プロパティの `error` および `code` プロパティの `NetConnection.Admin.Command.Failed`、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む `description` プロパティが含まれているオブジェクトもあります。

指定した管理者が存在しない場合は、呼び出しは失敗します。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意： HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

次の例では、removeAdmin() メソッドの使用方法を示します。

```
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "JLee", "x52z49ab");

/* Remove a server administrator named "DYoung" */
nc_admin.call("removeAdmin", new onRemoveAdmin(), "DYoung", "server");

/* Remove a virtual host administrator named "LPark" */
nc_admin.call("removeAdmin", new onRemoveAdmin(), "LPark");

/* Remove a virtual host administrator "JGarcia" from vhost tree.oak.com */
nc_admin.call("removeAdmin", new onRemoveAdmin(), "JGarcia", "_defaultRoot_/tree.oak.com");
```

関連項目

[addAdmin\(\)](#)

removeApp()

RTMP/E

removeApp(appName:String) : Object

HTTP

http://www.example.com:1111/admin/removeApp?user=username&apswd=password&appName=name

仮想ホストから指定したアプリケーションまたはアプリケーションインスタンスを削除します。最初に、指定したアプリケーションのすべてのインスタンスがアンロードされ、続いて、アプリケーションのディレクトリが仮想ホストから削除されます。アプリケーションインスタンスを指定する場合は、対象となるインスタンスのみがアンロードされて削除され、続いて、このインスタンスのすべてのストリームと共有オブジェクトが削除されます。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

appName 削除するアプリケーションまたはアプリケーションインスタンスの名前を、`application_name[/instance_name]` の形式で示す文字列。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは `level` プロパティの `status` および `code` プロパティの `NetConnection.Call.Success` を含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、`level` プロパティの `error` および `code` プロパティの `NetConnection.Admin.Command.Failed`、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む `description` プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される `Object` のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

次の例では、アプリケーション `ChatApp` 全体を削除する方法およびアプリケーション `ChatApp` のインスタンス `Instance1` のみを指定して削除する方法を示します。

```
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "JGarcia", "ezcabby1");

/* Removes the application along with all instances of the application */

nc_admin.call("removeApp", new onRemoveApp(), "ChatApp");

/* Removes only the specified instance */

nc_admin.call("removeApp", new onRemoveApp(), "ChatApp/Instance1");
```

関連項目

[addApp\(\)](#)

removeVHostAlias()

RTMP/E

removeVHostAlias (vhost:String, alias:String) : Object

HTTP

http://www.example.com:1111/admin/removeVHostAlias?auser=username&apswd=password&vhost=name&alias=name

仮想ホストからエイリアスを削除します。エイリアスは、Flash Media Server 接続が確立されるときにターゲットとして使用される仮想ホストの代替名です。エイリアスを削除すると、接続が確立されるときにエイリアスを使用できなくなります。

使用できるバージョン

Flash Media Server 2.0

パラメータ

vhost エイリアスを削除する仮想ホストを示す文字列。

alias 指定した仮想ホストから削除するエイリアスを示す文字列。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは、level プロパティの status および code プロパティの NetConnection.Call.Success を含む情報オブジェクトを応答として送信します。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

```
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "JLee", "x52z49ab");
admin_nc.call("removeVHostAlias", new Result(), "myvhost", "myalias");
```

関連項目

[addVHostAlias\(\)](#)

restartVHost()

RTMP/E

`restartVHost([vhost:String])` : Object

HTTP

`http://www.example.com:1111/admin/restartVHost?auser=username&apswd=password[&vhost=vhost]`

現在実行している仮想ホストを再起動します。仮想ホストを再起動すると、ユーザーとの接続がすべて切断され、現在ロードされているアプリケーションがすべてアンロードされ、この仮想ホストの設定ファイルが再ロードされます。

`restartVHost()` が適切に機能するには、仮想ホスト上に 1 つ以上のアプリケーションインスタンスが存在する必要があります。

仮想ホストの設定ファイルを変更する場合、このメソッドを呼び出すと、サーバーを再起動せずに仮想ホストを再起動できます。

仮想ホストを再起動するときは、そのつどユーザーは再接続する必要があります。このメソッドを呼び出す前に、接続しているユーザーに再起動することを通知してください。

仮想ホスト管理者が再起動できるのは、自分が接続している仮想ホストのみです。接続している仮想ホスト以外の仮想ホストを起動するには、サーバー管理者のアクセス権限が必要となります。

注意：停止している仮想ホストを起動するには、`startVHost()` メソッドを呼び出します。このメソッドを呼び出すことができるのは、サーバー管理者のみです。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

vhost 再起動する仮想ホストを示す文字列。

接続している仮想ホストを再起動するには、このパラメータを省略します。

別の仮想ホストを再起動するには、仮想ホストを `adaptor_name/virtual_hostname` の形式で指定します。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは `level` プロパティの `status` および `code` プロパティの `NetConnection.Call.Success` を含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、`level` プロパティの `error` および `code` プロパティの `NetConnection.Admin.Command.Failed`、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む `description` プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意： HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

次の例では、接続している仮想ホストを再起動する呼び出しおよびデフォルトのアダプタ上の仮想ホスト tree.oak.com を指定して再起動する呼び出しを示します。

```
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "JLee", "x52z49ab");

/* Restart the connected virtual host */
nc_admin.call("restartVHost", new onRestartVHost());

/* Restart the virtual host tree.oak.com on the default adaptor. */
nc_admin.call("restartVHost", new onRestartVHost(), "_defaultRoot_/tree.oak.com");
```

関連項目

[reloadApp\(\)](#)、[startVHost\(\)](#)、[stopVHost\(\)](#)

setConfig()

RTMP/E

```
setConfig(key:String, value:String, [,scope:String]) : Object
```

HTTP

```
http://www.example.com:1111/admin/setConfig?user=username&passwd=password&key=string&value=string
[&scope=scope]
```

この API は使用されなくなりました。代わりに、[setConfig2\(\)](#) を使用してください。指定した設定ファイルの指定した設定キーの値を変更します。XML 設定ファイルについては、『[Adobe Flash Media Server 設定および管理ガイド](#)』を参照してください。

仮想ホスト管理者は、自分の仮想ホストの Vhost.xml ファイルと Application.xml ファイルの設定キーの値を変更できます。Server.xml および Adaptor.xml ファイルの大部分の設定キーの値は、サーバー管理者のみが変更できます。

注意： XML ツリーの同一レベルに同じ名前の XML タグが共存することも可能です。設定ファイル内では、XML タグの name 属性を使用して、このようなタグを区別する必要があります。例えば、複数の仮想ホストについては、<VirtualHost name="www.redpin.com"></VirtualHost> という形式を使用します。setConfig() メソッドを呼び出して設定のサブキーを指定するときは、タグ名、コロン、正しい name 属性の順に指定することによって必要なタグを示すことができます。例えば、Adaptor:_defaultRoot_/VirtualHost:www.redpin.com と指定します。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

key 値を変更する設定キーを示す文字列。

キーは、スラッシュ (/) で区切られたサブキーのリストとして指定されます。最初のサブキーは、目的の設定キーを含む XML 設定ファイルを指定します。2 番目以降のサブキーは、XML 設定ファイルに関連したタグに対応しています。したがって、サブキーの階層と名前は、XML ファイルのタグと一致することになります。

権限に応じて、次のファイルの設定キーの値を変更することができます。

- Server.xml ファイルでは、最初のサブキーとして、Admin または Server を指定します。2 番目以降のすべてのキーは、Server.xml ファイルの <Admin> タグまたは <Server> タグに関連したタグに対応します。

<Server> タグの設定キーを設定できるのは、サーバー管理者のみです。

仮想ホスト管理者は、自分の仮想ホストについてのみ、設定キーを設定できます。一部の機密性のある情報については、設定できない場合があります。例えば、自分のパスワードは設定できますが、他の仮想ホスト管理者のパスワードや権限は設定できません。

- Adaptor.xml ファイルでは、最初のサブキーとして Adaptor:adaptor_name (adaptor_name はアダプタ名) を指定します。2 番目以降のすべてのキーは、Adaptor.xml ファイルの <Adaptor> タグに関連したタグに対応します。
- Vhost.xml では、最初のサブキーとして Adaptor:adaptor_name/VirtualHost:vhost_name (vhost_name は仮想ホスト名) を指定します。2 番目以降のすべてのキーは、Vhost.xml ファイルの <VirtualHost> タグに関連したタグに対応します。
- 管理サーバーにログオンしたときに接続した仮想ホスト上で実行中のアプリケーションの Application.xml ファイルについては、最初のサブキーとして Application:app_name (app_name はアプリケーション名) を指定します。

別の仮想ホストで実行中のアプリケーションの Application.xml ファイルのキーを取得するには、完全なキー (Adaptor:adaptor_name/VirtualHost:vhost_name/Application:app_name) を指定します。scope パラメータも指定する必要があります (この項目のコード例を参照)。

デフォルトの Application.xml ファイルを取得するには、コロン (:) と属性 app_name を付けずに「Application」と指定します。

注意: サブキーを指定しても、対応するタグが XML ファイルに存在しない場合は、XML ファイルに新しいタグが作成されます。

value 指定した設定キーに設定される値を示す文字列。

scope 文字列。Server.xml ファイル、Adaptor.xml ファイルまたは Vhost.xml ファイルの設定キーを変更するには、このパラメータにスラッシュ (/) を指定します。

Flash Media Server にログオンしたときに接続した仮想ホスト上で実行中のアプリケーションの Application.xml ファイルの設定パラメータを変更するには、このキーを省略します。

注意: 接続しているアダプタまたは仮想ホストを確認するには、getAdminContext() メソッドを使用します。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは level プロパティの status および code プロパティの NetConnection.Call.Success を含む情報オブジェクトを応答として送信します。設定キーの値は変更されます。

呼び出しが失敗すると、サーバーは、level プロパティの error、code プロパティの NetConnection.Call.BadValue および失敗の原因について説明する文字列を含む description プロパティを含む情報オブジェクトを応答として送信します。

指定した設定キーが検出されない場合、または設定キーの値を変更する権限をユーザーが持っていない場合は、呼び出しは失敗します。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意：HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

次の例では、4つの XML ファイルそれぞれについて、設定キーに新しい値を設定する方法を示します。

```
// Establish connection to server
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "LGreen", "123jn098");

// For a virtual host administrator, change key in Server.xml
// Set Password for user LGreen to "strawman28"
key = "Admin/Server/UserList/User:LGreen/Password";
val = "strawman28"
nc_admin.call("setConfig", new onSetConfig(), key, val, "/");

// For a server administrator, change key in Server.xml
// Set LicenseInfo to "Helloworld"
key = "Server/LicenseInfo";
val = "Helloworld";
nc_admin.call("setConfig", new onSetConfig(), key, val, "/");

// Change key in Adaptor.xml
// Set HostPort to 128.0.0.1:1938
key = "Adaptor:_defaultRoot_/HostPortList/HostPort";
val = "128.0.0.1:1938";
nc_admin.call("setConfig", new onSetConfig(), key, val, "/");
```

```
// Change key in Vhost.xml
// Set RecordAccessLog to true
key = "Adaptor:_defaultRoot_/VirtualHost:_defaultVHost_/RecordAccessLog";
val = "true";
nc_admin.call("setConfig", new onSetConfig(), key, val, "/");

// Change key in Application.xml for an application on the virtual host
// you connected to when you logged on to the administration server.
// Note that the previous subkeys and the final parameter "/" are not
// necessary.
key = "Application:FinanceApp/RecordAppLog";
val = "true";
nc_admin.call("setConfig", new onSetConfig(), key, val);

// Change key in Application.xml for a different virtual host.
// The adaptor and virtual host names and the second parameter are necessary.
key = "Adaptor:_defaultRoot_/VirtualHost:www.redpin.com/Application:ChatApp/ ~ RecordAppLog";
val = "true";
nc_admin.call("setConfig", new onSetConfig(), key, val, "/");
```

関連項目

[getAdminContext\(\)](#)、[getConfig\(\)](#)

setConfig2()

RTMP/E

```
setConfig2(key:String, value:String, scope:String) : Object
```

HTTP

```
http://www.example.com:1111/admin/setConfig2?auser=username&apswd=password&key=string&value=string&scope=scope
```

指定した設定ファイルの指定した設定キーの値を変更します。Flash Media Server には、Server.xml、Users.xml、Logger.xml、Adaptor.xml、Vhost.xml および Application.xml という 6 つのサーバー設定ファイルが含まれています。XML 設定ファイルについては、『[Adobe Flash Media Server 設定および管理ガイド](#)』を参照してください。

仮想ホスト管理者は、自分の仮想ホストの Vhost.xml ファイルと Application.xml ファイルの設定キーの値を変更できます。Server.xml および Adaptor.xml ファイルの大部分の設定キーの値は、サーバー管理者のみが変更できます。

注意：XML ツリーの同一レベルに同じ名前の XML タグが共存することも可能です。設定ファイル内では、XML タグの name 属性を使用して、このようなタグを区別する必要があります。例えば、複数の仮想ホストについては、<VirtualHost name="www.redpin.com"></VirtualHost> という形式を使用します。setConfig() メソッドを呼び出して設定のサブキーを指定するときは、タグ名、コロン、正しい name 属性の順に指定することによって必要なタグを示すことができます。例えば、Adaptor:_defaultRoot_/VirtualHost:www.redpin.com と指定します。

使用できるバージョン

Flash Media Server 2.0

パラメータ

key 値を変更する設定キーを示す文字列。

キーは、スラッシュ (/) で区切られたサブキーのリストとして指定されます。最初のサブキーは、目的の設定キーを含む XML 設定ファイルを指定します。2 番目以降のサブキーは、XML 設定ファイルに関連したタグに対応しています。した

がって、サブキーの階層と名前は、XML ファイルのタグと一致することになります。例えば、「/Server/LicenseInfo」となります。

value 指定した設定キーに設定される値を示す文字列。指定した値が単純な文字列の場合、指定したタグのタグデータとして設定されます。指定した値が、`<foo>bar</foo>` のように有効な XML の場合、その XML は指定したタグの子タグとして追加されます。指定した値が空の文字列の場合、指定したタグは削除されます。

scope key パラメータによって指定されたタグに設定する値を示す文字列。指定した値が単純な文字列の場合、指定したタグのタグデータとして設定されます。指定した値が、`<foo>bar</foo>` のように有効な XML の場合、その XML は指定したタグの子タグとして追加されます。指定した値が空の文字列の場合、指定したタグは削除されます。

- / は、Server.xml を指定します。このファイルの大部分は、サーバー管理者の特権を持つユーザーのみがアクセスできます。サーバー管理者の権限を必要としないセクションは、呼び出し側が所属する仮想ホストの管理者を含む `<VirtualHost>` セクションのみです。
- Users は、サーバー管理者用の Users.xml を指定します。
- Logger は、Logger.xml を指定します。
- Adaptor:<adaptor_name> は、Adaptor.xml を指定します。<adaptor_name> は該当するアダプタの名前です。このファイルにアクセスできるのは、サーバー管理者の特権を持つユーザーのみです。呼び出し側が接続しているアダプタの名前が <adaptor_name> ではない場合、呼び出しは失敗します。
- Adaptor:<adaptor_name>/VHost:<vhost_name> は、VHost.xml を指定します。<vhost_name> は該当する仮想ホストの名前です。呼び出し側が接続しているアダプタの名前が <adaptor_name> でない場合や、呼び出し側が接続している仮想ホストの名前が <vhost_name> でない場合は、呼び出しが失敗します。
- Adaptor:<adaptor_name>/VHost:<vhost_name>/Users は、仮想ホスト管理者の Users.xml を指定します。
- Adaptor:<adaptor_name>/VHost:<vhost_name>/App[:<app_name>] は、Application.xml を指定します。<app_name> が指定されていない場合は、デフォルトの Application.xml が使用されます。指定されている場合は、指定したアプリケーション用の Application.xml ファイルが使用されます。指定されたアプリケーションが定義されていない場合や、そのアプリケーション用の Application.xml ファイルがない場合は、呼び出しは失敗します。

注意：接続しているアダプタまたは仮想ホストを確認するには、getAdminContext() メソッドを使用します。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは level プロパティの status および code プロパティの NetConnection.Call.Success を含む情報オブジェクトを応答として送信します。設定キーの値は変更されます。

呼び出しが失敗すると、サーバーは、level プロパティの error、code プロパティの NetConnection.Call.BadValue および失敗の原因について説明する文字列を含む description プロパティを含む情報オブジェクトを応答として送信します。

指定した設定キーが検出されない場合、または設定キーの値を変更する権限をユーザーが持っていない場合は、呼び出しは失敗します。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

次の例では、設定キーに新しい値を設定する方法を示します。

```
tSocket = new NetConnection();
tSocket.connect("rtmp://localhost/admin", "user", "password");

// set tag data in Server.xml
key = "Server/LicenseInfo";
val = "SFD150-XXXXX-XXXXX-XXXXX";
tSocket.call("setConfig2", new onSetConfig(), key, val, "/");

// set tag data in Adaptor.xml
key = "HostPortList/HostPort";
val = ":1935, 80, 443";
scope = "Adaptor:_defaultRoot_";
tSocket.call("setConfig2", new onSetConfig(), key, val, scope);

// set tag data in Vhost.xml
key = "AppsDir";
val = "c:\\¥¥applications";
scope = "Adaptor:_defaultRoot_/VHost:_defaultVHost_";
tSocket.call("setConfig2", new onSetConfig(), key, val, scope);

// set tag data in Application.xml for app "foo"
key = "Process/Scope";
val = "inst";
scope = "Adaptor:_defaultRoot_/VHost:_defaultVHost_/App:foo";
tSocket.call("setConfig2", new onSetConfig(), key, val, scope);

// add child tag to a tag in default Application.xml
key = "Process";
val = "<Scope>inst</Scope>";
scope = "Adaptor:_defaultRoot_/VHost:_defaultVHost_/App";
tSocket.call("setConfig2", new onSetConfig(), key, val, scope);

// delete tag in default Application.xml
key = "Process";
val = "";
scope = "Adaptor:_defaultRoot_/VHost:_defaultVHost_/App";
tSocket.call("setConfig2", new onSetConfig(), key, val, scope);
```

関連項目

[getAdminContext\(\)](#)、[getConfig2\(\)](#)

startServer()

RTMP/E

startServer([mode:String]) : Object

HTTP

http://www.example.com:1111/admin/startServer?auser=username&apswd=password[&mode=string]

Flash Media Server サービスを開始するか、サービスを停止して再開します。

この操作を実行できるのは、サーバー管理者のみです。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

mode サーバーを再起動するかどうかを示す文字列。

実行していないサーバーを起動する場合は、このパラメータを削除します。実行中のサーバーを終了して再起動する場合は、このパラメータに「restart」と指定します。サーバーが実行中の場合にこのパラメータを省略すると、このメソッドは何もしません。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは level プロパティの status および code プロパティの NetConnection.Call.Success を含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Admin.Command.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
  <data></data>
</result>
```

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[stopServer\(\)](#)

startVHost()

RTMP/E

```
startVHost(vhost:String) : Object
```

HTTP

```
http://www.example.com:1111/admin/startVHost?auser=username&apswd=password&vhost=name
```

停止している仮想ホストを起動、または新しい仮想ホストを有効にします。

このメソッドにより、サーバーを再起動させずに、ランタイムに新しい仮想ホストを有効にすることができます。
startVHost() メソッドは、ディレクトリや設定ファイルを作成またはコピーしません。したがって、サーバーの `conf` ディレクトリには、新しい仮想ホストのディレクトリと設定ファイルを事前に作成しておく必要があります。仮想ホストの追加については、『[Adobe Flash Media Server 設定および管理ガイド](#)』を参照してください。

startVHost() メソッドを使用できるのは、サーバー管理者のみです。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

vhost 起動する仮想ホスト名または新規に有効にする仮想ホスト名を、`[adaptor/]vhost` の形式で示す文字列。

接続しているアダプタ上で新規に仮想ホストを有効にする場合は、`adaptor/` の部分を省略します。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは `level` プロパティの `status` および `code` プロパティの `NetConnection.Call.Success` を含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、`level` プロパティの `error` および `code` プロパティの `NetConnection.Admin.Command.Failed`、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む `description` プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

次の例では、現在接続しているアダプタ上で仮想ホスト `diamond.world.com` を起動する方法およびアダプタ `gem` 上で仮想ホスト `diamond.world.com` を起動する方法を示します。

```
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "JLee", "x52z49ab");

/* Starts a virtual host named diamond.world.com */
nc_admin.call("startVHost", new onStartVHost(), "diamond.world.com");

/* Starts a virtual host named diamond.world.com on the adaptor, gem */
nc_admin.call("startVHost", new onStartVHost(), "gem/diamond.world.com");
```

関連項目

[restartVHost\(\)](#)

stopServer()

RTMP/E

```
stopServer(mode:String) : Object
```

HTTP

```
http://www.example.com:1111/admin/stopServer?auser=username&apswd=password&mode=string
```

Flash Media Server を終了します。ユーザーが接続しているときにこのメソッドを呼び出す場合は、アプリケーションが即座に終了する旨をユーザーに通知し、ユーザーの作業が失われないように配慮してください。

この操作を実行できるのは、サーバー管理者のみです。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

mode サーバをシャットダウンする方法を示す文字列。可能な値は `normal` または `abort` です。値 `normal` を使用する場合は、サーバーは終了し、実行中のアプリケーションは正常に終了します。値 `abort` を使用する場合は、サーバーは即座に終了し、実行中のアプリケーションは正常に終了しません。`abort` は、緊急時または `normal` を指定しても正常に動作しない場合に以外には使用しないでください。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは `level` プロパティの `status` および `code` プロパティの `NetConnection.Call.Success` を含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Admin.Command.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意： HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[startServer\(\)](#)

stopVHost()

RTMP/E

stopVHost([vhost:String]) : Object

HTTP

http://www.example.com:1111/admin/stopVHost?auser=username&apswd=password[&vhost=name]

実行中の仮想ホストを停止します。デフォルトの仮想ホストに対して stopVHost() コマンドを使用することはできません。仮想ホストの停止後、すべてのアプリケーションはアンロードされ、すべてのユーザーとの接続が切断されます。startVHost() メソッドを使用して仮想ホストを再起動するか、サーバーが再起動するまでは、仮想ホストは要求を一切受け付けません。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

vhost 停止する仮想ホストを示す文字列。接続している仮想ホストを停止するには、このパラメータを省略します。

別の仮想ホストを停止するには vhost_name を adaptor_name/vhost_name の形式で指定します。接続している仮想ホスト以外の仮想ホストを停止するには、サーバー管理者のアクセス権限が必要となります。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは level プロパティの status および code プロパティの NetConnection.Call.Success を含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、level プロパティの error および code プロパティの NetConnection.Admin.Command.Failed、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む description プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エlement には、RTMP/E 呼び出しで返される Object のプロパティと同じ情報が含まれます。

注意: HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

例

次の例では、仮想ホスト tree.oak.com を停止する方法およびデフォルトのアダプタ上の仮想ホスト tree.oak.com を停止する方法を示します。

```
nc_admin = new NetConnection();
nc_admin.connect("rtmp://localhost:1111/admin", "JLee", "x52z49ab");

/* Stop the vhost named tree.oak.com */
nc_admin.call("stopVHost", new onStopVHost(), "tree.oak.com");

/* Stop the vhost named tree.oak.com on the default adaptor */
nc_admin.call("stopVHost", new onStopVHost(), "_defaultRoot_/tree.oak.com");
```

関連項目

[restartVHost\(\)](#)、[startVHost\(\)](#)

unloadApp()

RTMP/E

unloadApp(appInst:String) : Object

HTTP

http://www.example.com:1111/admin/unloadApp?user=username&passwd=password&appInst=name

指定したアプリケーションのすべてのインスタンス、または指定したアプリケーションインスタンスを終了します。アプリケーション名を指定すると、このアプリケーションのすべてのインスタンスは終了し、アプリケーションインスタンスへのユーザー接続はすべて即座に切断されます。アプリケーションインスタンスを指定する場合は、指定したインスタンスのみが終了し、このインスタンスへのすべてのユーザー接続が即座に切断されます。

注意：このメソッドを呼び出す前に、アプリケーションが即座に終了する旨をユーザーに通知してください。

使用できるバージョン

Flash Communication Server 1.0

パラメータ

applnst アプリケーションまたはアプリケーションインスタンスの名前を、`application_name[/instance_name]` の形式で示す文字列。

auser 管理者のユーザー名を示す文字列。

apswd 管理者のパスワードを示す文字列。

戻り値

RTMP/E 呼び出しが成功すると、サーバーは `level` プロパティの `status` および `code` プロパティの `NetConnection.Call.Success` を含む情報オブジェクトを応答として送信します。

呼び出しが失敗すると、サーバーは、`level` プロパティの `error` および `code` プロパティの `NetConnection.Admin.Command.Failed`、またはより具体的な値 (存在する場合) を含む情報オブジェクトを応答として送信します。失敗の原因について説明する文字列を含む `description` プロパティが含まれているオブジェクトもあります。

HTTP 呼び出しが成功すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <timestamp></timestamp>
</result>
```

呼び出しが失敗すると、次の構造の XML を返します。

```
<result>
  <level></level>
  <code></code>
  <description></description>
  <timestamp></timestamp>
</result>
```

この XML エレメントには、RTMP/E 呼び出しで返される `Object` のプロパティと同じ情報が含まれます。

注意：HTTP 経由のタイムスタンプ応答の形式は、Windows (9/23/2007 6:16:40 PM) および Linux (Sun 23 Sep 2007 06:16:40 PM IST) で異なります。

関連項目

[reloadApp\(\)](#)

