サービス キープアライブ、グローバル キープアライブ、スクリプト キープアライブの設定

CSS でサービスを設定すると、その CSS はキープアライブ メッセージを送信してサービスの状態を調べます。デフォルトでは、CSS は各サービスに ICMP タイプのキープアライブを割り当てます。キープアライブ間隔と再試行間隔はどちらも 5 秒、最大失敗回数は 3 回です。サービスのデフォルト キープアライブ設定の変更が必要になった場合は、サービスのキープアライブ アトリビュートを個別に変更できるほか、キープアライブ モードでキープアライブを作成し、対象のサービスに適用することもできます。

この章では、サービス キープアライブ、グローバル キープアライブ、スクリプト キープアライブの設定方法について説明します。この章の内容は、特に指定のない限り、すべての CSS モデルに共通です。

この章で説明する CSS キープアライブの設定に関する主な内容は次のとおりです。

CSS のキープアライブの概要

CSS では合計で 2048 のキープアライブをサポートします。次のキープアライブがあります。

- **(config-service) keepalive type** コマンドにより設定され、サービスに割り当てられる ICMP、HTTP、TCP、FTP、SSL、およびスクリプト キープアライブ。デフォルトでは、ICMP タイプのキープアライブが割り当てられます。このコマンドを使用してこれらのキープアライブのタイプの 1 つをサービスに割り当てるたびに、そのキープアライブは別のキープアライブとしてカウントされます。サービスのキープアライブの設定については、「サービス キープアライブの設定」を参照してください。
- キープアライブ設定モードで設定されるグローバル キープアライブ。1 つのグローバル キープアライブには複数のサービスを割り当てることが可能で、それによって各サービスに必要な設定量も少なくなります。CSS は、グローバル キープアライブを、割り当てられているサービスの数に関係なく、1 つのキープアライブとしてカウントします。

  グローバル キープアライブは、サービス モードで設定する各キープアライブ パラメータより優先されます。グローバル キープアライブの設定については、この章で後述する「グローバル キープアライブの設定」を参照してください。

CSS では、キープアライブのタイプがクラス A キープアライブとクラス B キープアライブという 2 つのカテゴリに分けられます。CSS でサポートされるキープアライブの最大数は、クラス A キープアライブが 2048、クラス B キープアライブが 512 です。

注意　設定するキープアライブの数が 2048 を超えないようにしてください。この数にはクラス B キープアライブの最大数（512）も含まれます。サポートされる合計数を超えるキープアライブに割り当てられたサービスは、コンテンツ ルールの選択の対象にはなりません。

表 4-1 は、それぞれのクラスに含まれるキープアライブのタイプ、各タイプの最大数、および並行して実行できる最大数を示しています。

表 4-1　キープアライブのクラス、タイプ、および制限

クラス	タイプ	CSS がサポートする最大数	並行実行の最大数
A （CSS がクラス A ごとにサポートするキープアライブの最大数：2048）	ICMP	2048	2048
	HTTP-HEAD 非固定	2048	2048
	暗号化 HTTP-HEAD 非固定	256[1]	256
	SSL （Hello）	2048	2048
	TCP	2048	2048
B （CSS がクラス B ごとにサポートするキープアライブの最大数：512）	FTP	256	32
	HTTP-GET 固定 / 非固定	256	32
	HTTP-HEAD 固定	256	32
	暗号化 HTTP-GET 固定 / 非固定	256[1]	32
	暗号化 HTTP-HEAD 固定	256[1]	32
	スクリプト	256	16

1. 1 つの SSL プロキシ リストには、SSL バックエンド サーバのエントリを 256 個まで記述できます。したがって、クラス A とクラス B の暗号化キープアライブの合計数も、最大で 256 となります。

サービスにキープアライブを設定すると（またはサービスにグローバル キープアライブを関連付けると）、サービスの状態を調べるために、サービスにキープアライブの周期に基づいてメッセージが送信されます。「キープアライブの間隔の設定」を参照してください。サービスがキープアライブ メッセージに応答すると、そのサービスは「Alive」状態であると認識されます。

サービスがキープアライブ メッセージへの応答に失敗すると、そのサービスは「Dying」状態に移行されます。再試行間隔に基づいた時間間隔でキープアライブ メッセージを送信して、失敗したサービスが有効かどうかをテストします。「キープアライブの再試行間隔の設定」を参照してください。

キープアライブ メッセージの再試行最大回数に達してもサービスが応答しない場合、そのサービスはダウン状態に移行されます。「キープアライブの最大失敗回数の設定」を参照してください。その後、サービスはロード バランシング アルゴリズムから削除されます。CSS は、再試行間隔に基づく時間間隔で、失敗したサービスが有効かどうかのテストを継続します。

したがって、5 秒のキープアライブ時間間隔、5 秒の再試行間隔、および最大失敗回数 3 回のデフォルト値を使用した場合は、サービスが Alive 状態から 15 秒後に Dead 状態に移行することがあります。キープアライブ時間間隔に基づき、キープアライブ応答とキープアライブの最初の失敗の間隔は 5 秒です。その後、再試行間隔に基づいて 5 秒間隔で 2 回の失敗が発生すると、合計は 15 秒になります。

ただし、クラス B タイプのキープアライブの場合は、Alive から Dead に移行するまでの時間が、これよりも長くなることがあります。この遅れは、CSS で一度に実行できるクラス B キープアライブ数が少ないために発生します。たとえば、256 個の HTTP-GET キープアライブをデフォルトの実行間隔、再試行間隔、および最大失敗回数で設定している場合に、すべてのサービスが失敗すると、これらのすべてのサービスが Alive から Dead に移行するのに 120 秒間かかります。これは、最初の 32 個のサービスが 15 秒で移行してから 256 個のサービスがすべて移行し終えるまで、サービスが 32 個ずつ、それぞれ 15 秒で移行するためです。

サービス キープアライブの設定

サービス キープアライブとは、特定のサービス用に設定されたキープアライブのことです。サービスの設定時に、そのキープアライブ アトリビュートを設定できます。サービスのキープアライブ アトリビュートを設定するには、サービス設定モードで **keepalive** コマンドを使用します。詳細については、「サービスとグローバル キープアライブのアトリビュートの設定」を参照してください。

CSS サービスをグローバル キープアライブに適用する方法については、「グローバル キープアライブの設定」を参照してください。

キープアライブ アトリビュートを含むサービスの設定が完了すると、そのサービスを有効化できます。サービスを有効にすると、そのサービスは、コンテンツ要求のロード バランシングのためのリソース プールに格納され、キープアライブ機能が起動します。サービス *serv1* を有効するには、次のように入力します。

```
(config-service[serv1])# active
```

グローバル キープアライブの設定

グローバル キープアライブを設定すると、キープアライブ アトリビュートの設定後、そのキープアライブに複数のサービスを適用できます。適用したサービスのいずれかが有効な限り、グローバル キープアライブ サービスも有効です。1 つのキープアライブの設定を複数のサービスに利用できるため、サービスごとに設定する手間が省けます。また、グローバル キープアライブは適用するサービスの数に関係なく、1 つのキープアライブとしてカウントされます。

表 4-2 に、グローバル キープアライブの設定に必要な基本手順の概要を示します。それぞれの手順に、作業を実行するために必要な CLI コマンドも示します。CLI コマンドの各機能とすべてのオプションの詳細については、表 4-2 以降の項を参照してください。

表 4-2　グローバル キープアライブの設定のクイック スタート

作業とコマンドの例

1. **config** と入力してグローバル設定モードに入ります。
`# config`
`(config)#`

2. グローバル キープアライブを作成し、キープアライブ設定モードに入ります。「グローバル キープアライブの作成と命名」を参照してください。
`(config)# keepalive keepimages`
`(config-keepalive[keepimages])#`

3. キープアライブ メッセージの送信先 IP アドレスを指定します。「グローバル キープアライブの IP アドレスの設定」を参照してください。 `(config-keepalive[keepimages])# ip address 192.168.7.6`
4. キープアライブに割り当てるキープアライブ　メッセージのタイプを指定します。「キープアライブ タイプの設定」を参照してください。 `(config-keepalive[keepimages])# type http`
5. グローバル キープアライブに割り当てる HTTP キープアライブ方式を指定します。「HTTP キープアライブ方式の設定」を参照してください。 `(config-keepalive[keepimages])# method get`
6. HTTP グローバル キープアライブのコンテンツ情報を指定します。「キープアライブ URI の設定」を参照してください。 `(config-keepalive[keepimages])# uri "/index.html"`

表 4-2　グローバル キープアライブの設定のクイック スタート（続き）

作業とコマンドの例
7. グローバル キープアライブを有効化します。 `(config-keepalive[keepimages])# active`
8. サービスをグローバル キープアライブに関連付けます。 `(config-service[imageserver1])# keepalive type named keepimages`
9. （推奨）**show keepalive** コマンドを使用して、グローバル キープアライブの設定を確認します。「キープアライブ設定の表示」を参照してください。 `(config-keepalive[keepimages])# show keepalive`
10. （オプション）**show service** コマンドを使用して、サービスの基本的なキープアライブ設定を確認します。 `(config-service[imageserver1])# show service`

次の実行設定例は、表 4-2 に示した各コマンド（ボールド体部分）および関連する各種コマンドを実行した結果を示しています。

```
!************************* SERVICE ************************
server2
  ip address 10.3.6.1
  keepalive type named keepimages
  active
!************************ KEEPALIVE ***********************
keepalive keepimages
  ip address 192.168.7.6
  type http
  method get
  uri "/index.html"
  active
```

ここでは、次の内容について説明します。

- グローバル キープアライブの作成と命名
- グローバル キープアライブの IP アドレスの設定
- グローバル キープアライブの説明の設定
- グローバル キープアライブの有効化
- グローバル キープアライブの一時停止
- サービスとグローバル キープアライブの関連付け

キープアライブ アトリビュートの設定については、「サービスとグローバル キープアライブのアトリビュートの設定」を参照してください。

グローバル キープアライブの作成と命名

グローバル キープアライブを作成し、名前を付けるには、**keepalive** コマンドを使用してキープアライブ モードに入ります。キープアライブ モードには、回線、グローバル、インターフェイス、および IP 設定の各モードから移行できます。プロンプトは (config-keepalive [name]) に変わります。キープアライブ モードでこのコマンドを使用して、別のキープアライブにアクセスすることもできます。

作成する新しいキープアライブの名前または既存のキープアライブの名前を入力します。スペースを含まない 1 ～ 31 文字のテキスト文字列を引用符で囲まずに入力します。既存のキープアライブ名のリストを表示するには、**keepalive ?** コマンドを使用します。

たとえば、グローバル キープアライブ keepimages を作成するには、次のように入力します。

```
(config)# keepalive keepimages
```

このモードに入ると、プロンプトが (config-keepalive [keepimages]) に変わります。

```
(config-keepalive[keepimages])#
```

既存のキープアライブを削除するには、次のように入力します。

```
(config)# no keepalive keepimages
```

グローバル キープアライブの IP アドレスの設定

グローバル キープアライブは CSS によってサービスに送信され、割り当てられた各サービスの状態が監視されます。キープアライブ メッセージの送信先の IP アドレスを指定するには、**ip address** コマンドを使用します。IP アドレスはドット付き 10 進表記で入力します。

たとえば、グローバル キープアライブ keepimages の IP アドレスを入力するには、次のように入力します。

```
(config-keepalive[keepimages])# ip address 192.168.7.6
```

グローバル キープアライブの説明の設定

グローバル キープアライブの説明も、必要に応じて指定できます。説明を指定するには、**description** コマンドを使用します。説明は、スペースを含む 64 文字以内のテキスト文字列を引用符で囲んで入力します。

たとえば、グローバル キープアライブ keepimages の説明を指定するには、次のように入力します。

```
(config-keepalive[keepimages])# description "This keepalive is for the
image servers"
```

説明を削除するには、次のように入力します。

```
(config-keepalive[keepimages])# no description
```

グローバル キープアライブの有効化

キープアライブを有効にすると、CSS は IP アドレスへのキープアライブ メッセージの送信を開始します。グローバル キープアライブを有効にするには、**active** コマンドを使用します。

たとえば、グローバル キープアライブ keepimages を有効化するには、次のように入力します。

```
(config-keepalive[keepimages])# active
```

グローバル キープアライブの一時停止

キープアライブを無効するには、**suspend** コマンドを使用します。

次にコマンドの使用例を示します。

```
(config-keepalive[keepimages])# suspend
```

サービスとグローバル キープアライブの関連付け

サービスにグローバル キープアライブを関連付けるには、**keepalive type named** コマンドを使用します。サービスをコンテンツ ルールに追加したときに、そのサービスでグローバル キープアライブのアトリビュートが保持されます。

たとえば、imageserver1 にグローバル キープアライブ keepimages を関連付けるには、次のように入力します。

```
(config-service[imageserver1])# keepalive type named keepimages
```

サービスとグローバル キープアライブのアトリビュートの設定

ここでは、キープアライブに設定できるアトリビュートについて説明します。

キープアライブの間隔の設定

キープアライブ間隔とは、キープアライブ メッセージをサービスに送信する時間間隔（秒）です。2 ～ 255 秒の間隔を指定します。デフォルト値は 5 秒です。

（注） CSS で FTP キープアライブを設定する際には、キープアライブ間隔、キープアライブ再試行間隔のどちらも、必ず 15 秒以上になるように設定してください。15 秒未満の値も設定可能で、デフォルトではキープアライブ間隔と再試行間隔のどちらも 5 秒に設定されていますが、キープアライブ間隔と再試行間隔を指定するコマンドで明示的に値を設定して、デフォルト値を無効にする必要があります。

（注） キープアライブのタイムアウトは、設定したキープアライブの間隔に関連しています。
7.20.1.04 以降のバージョンのタイムアウト値は、キープアライブ間隔より 2 秒短い値（最小値：1 秒）です。バージョン 5.20 ～ 7.20.1.04 のタイムアウト値は、キープアライブ間隔より 1 秒短い値になります。

- 特定のサービスにキープアライブ間隔を設定するには、サービス モードで **keepalive frequency** コマンドを使用します。たとえば、間隔を 15 秒に設定するには、次のように入力します。

  ```
  (config-service[serv1])# keepalive frequency 15
  ```

  間隔をデフォルトの 5 秒にリセットするには、次のように入力します。

  ```
  (config-service[serv1])# no keepalive frequency
  ```

- グローバル キープアライブの間隔を設定するには、キープアライブ モードで **frequency** コマンドを使用し、IP アドレスへのキープアライブ メッセージの送信間隔を指定します。

  たとえば、間隔を 10 秒に設定するには、次のように入力します。

  ```
  (config-keepalive[keepimages])# frequency 10
  ```

  間隔をデフォルトの 5 秒にリセットするには、次のように入力します。

  ```
  (config-keepalive[keepimages])# no frequency
  ```

キープアライブの再試行間隔の設定

サービスが、指定されたキープアライブ メッセージの応答に失敗した場合に（サービスの状態は「dying」状態に遷移）、サービスが機能しているかどうかを確認するためのテストを実行する間隔を retryperiod で指定します。再試行間隔（秒単位）として、2～255 の範囲内の整数を入力します。デフォルト値は 5 秒です。

（注） CSS で FTP キープアライブを設定する際には、キープアライブ間隔、キープアライブ再試行間隔のどちらも、必ず 15 秒以上になるように設定してください。15 秒未満の値も設定可能で、デフォルトではキープアライブ間隔と再試行間隔のどちらも 5 秒に設定されていますが、キープアライブ間隔と再試行間隔を指定するコマンドで明示的に値を設定して、デフォルト値を無効にする必要があります。

- サービスにキープアライブ再試行間隔を設定するには、サービス モードで **keepalive retryperiod** コマンドを使用します。たとえば、再試行間隔を 60 秒に設定するには、次のように入力します。

  ```
  (config-service[serv1])# keepalive retryperiod 60
  ```

再試行間隔をデフォルト値の 5 秒にリセットするには、次のように入力します。

```
(config-service[serv1])# no keepalive retryperiod
```

- グローバル　キープアライブの再試行間隔を設定するには、キープアライブ モードで **retryperiod** コマンドを使用します。たとえば、再試行間隔を 60 秒に設定するには、次のように入力します。

```
(config-keepalive[keepimages])# retryperiod 60
```

再試行間隔をデフォルト値の 5 秒にリセットするには、次のように入力します。

```
(config-keepalive[keepimages])# no retryperiod
```

キープアライブの最大失敗回数の設定

最大失敗回数とは、サービスからの応答が得られなくても許容されるキープアライブ メッセージの送信回数のことです。この回数を超えると、サービスが CSS によってオフラインと認識されます。最大失敗回数は 1 ～ 10 の数値で指定します。デフォルトは 3 です。

- 特定のサービスにキープアライブの最大失敗回数を設定するには、サービス モードで **keepalive maxfailure** コマンドを使用します。たとえば、最大失敗回数を 5 に設定するには、次のように入力します。

```
(config-service[serv1])# keepalive maxfailure 5
```

最大失敗回数をデフォルト値の 3 にリセットするには、次のように入力します。

```
(config-service[serv1])# no keepalive maxfailure
```

- グローバル　キープアライブの最大失敗回数を設定するには、キープアライブ モードで **maxfailure** コマンドを使用します。たとえば、最大失敗回数を 7 に設定するには、次のように入力します。

```
(config-keepalive[keepimages])# maxfailure 7
```

最大失敗回数をデフォルト値の 3 にリセットするには、次のように入力します。

```
(config-keepalive[keepimages])# no maxfailure
```

キープアライブ タイプの設定

キープアライブ タイプとは、キープアライブに割り当てられるキープアライブ メッセージの種類のことであり、ICMP、HTTP-GET、HTTP-HEAD、TCP、FTP、SSL、スクリプト キープアライブの 7 種類があります。また、SSL モジュールを使用すれば、暗号化 HTTP タイプのキープアライブを設定し、HTTPS アプリケーションの検証も可能になります。

サービス キープアライブの場合、named キープアライブ タイプを指定すると、設定済みのグローバル キープアライブにサービスを適用できます。

- サービスに送信するキープアライブ　メッセージのタイプを指定するには、サービス モードで **keepalive type** コマンドを使用して適切なタイプを指定するか、サービスをグローバル キープアライブに関連付けます。たとえば、serv1 のキープアライブ タイプを **ftp** に設定するには、次のように入力します。

  ```
  (config-service[serv1])# keepalive type ftp
  ```

- グローバル キープアライブのキープアライブ タイプを指定するには、キープアライブ モードで **type** コマンドを使用します。たとえば、グローバル キープアライブ keepimages を **type tcp** に設定するには、次のように入力します。

  ```
  (config-keepalive[keepimages])# type tcp
  ```

サービスやグローバル キープアライブにキープアライブ タイプを割り当てるたびに、CSS のキープアライブ カウントが 1 だけ増加します。

注意　設定するキープアライブの数が 2048 を超えないようにしてください。この数にはクラス B キープアライブの最大数（512）も含まれます。サポートされる合計数を超えるキープアライブに割り当てられたサービスは、コンテンツ ルールの選択の対象にはなりません。

keepalive type コマンドと **type** コマンドのオプションは次のとおりです。

- **ftp** *ftp_record* : FTP レコード ファイル内での定義に従って、CSS が FTP サーバにログインする。FTP サーバの既存 FTP レコードの名前として、スペースを含まないテキスト文字列を引用符で囲まずに入力します。FTP レコードを作成するには、**(config) ftp-record** コマンドを使用します。

FTP キープアライブは、クラス B タイプのキープアライブです。CSS は、このタイプのキープアライブを最大 256 サポートしており、最大 32 の FTP キープアライブを並行して実行できます。

CSS で FTP キープアライブを設定する際には、キープアライブ間隔、キープアライブ再試行間隔のどちらも、必ず 15 秒以上になるように設定してください。15 秒未満の値も設定可能で、デフォルトではキープアライブ間隔と再試行間隔のどちらも 5 秒に設定されていますが、デフォルト値を無効にする場合は、**keepalive frequency** コマンドと **keepalive retryperiod** コマンドで明示的に値を設定する必要があります。

- **http** ：固定 HTTP インデックス ページ要求。デフォルトでは、HTTP キープアライブは固定接続を使用します。

  このタイプの HTTP キープアライブの方式を設定する方法については、「HTTP キープアライブ方式の設定」を参照してください。HTTP-HEAD 固定と HTTP-GET 固定は、クラス B タイプのキープアライブです。どちらのタイプについても CSS は最大 256 のキープアライブをサポートし、最大 32 のキープアライブを並行して実行できます。

  HTTP 固定キープアライブで固定接続の確立に失敗すると、非固定接続の確立が試みられます。その結果、非固定接続が確立されると、キープアライブは成功です。その後、次のキープアライブ間隔が来た時点で、同じキープアライブで固定接続の確立が試みられます。

- **http non-persistent** ：非固定 HTTP インデックス ページ要求。このコマンドは、固定接続を確立するデフォルトの動作を無効にします。

  このタイプの HTTP キープアライブの方式を設定する方法については、「HTTP キープアライブ方式の設定」を参照してください。HTTP-GET 非固定キープアライブは、クラス B タイプのキープアライブです。CSS はこのタイプのキープアライブを最大 256 サポートしており、最大 32 の HTTP-GET 非固定キープアライブを並行して実行できます。

  HTTP-HEAD 非固定キープアライブは、クラス A タイプのキープアライブです。CSS は、このタイプのキープアライブを最大 2048 サポートしており、最大 2048 の HTTP-HEAD 非固定キープアライブを並行して実行できます。

- **http** {**non-persistent**} **encrypt** : SSL バックエンド サーバまたは SSL 開始サーバに送信する、暗号化された固定 / 非固定キープアライブの種類（HTTP HEAD または GET）。暗号化 HTTP キープアライブによって、SSL ハンドシェイク全体と、サーバから返されたデータの検証が可能になります。バックエンド SSL サーバの場合、キープアライブによって HTTP GET または HTTP HEAD が実行されます。このキープアライブで SSL モジュールが 1 つ選択され、そのモジュールが設定済みのサーバに接続されます。キープアライブ

メッセージは、キープアライブ エラーが発生するまで、同じモジュールに継続的に送信されます。その後、キープアライブによって CSS 内の他の SSL モジュールが選択されます。

SSL 開始サーバの場合は、キープアライブによって HTTP GET または HTTP HEAD が実行され、そのサーバ用に設定されている SSL モジュールに送信されます。次にこのモジュールは、設定済みのサーバに接続されます。

1 つの SSL プロキシ リストには、SSL バックエンド サーバまたは SSL 開始サーバのエントリを最大 256 個記述できます。したがって、CSS でサポートされる暗号化キープアライブの数も、最大 256 個になります。

（注） 暗号化キープアライブの数は、CSS でサポートされるクラス A およびクラス B の該当する HTTP キープアライブの最大数および現在数に含まれます。

SSL バックエンド サーバや SSL 開始サーバの設定方法の詳細については、『*Cisco Content Services Switch SSL Configuration Guide*』を参照してください。

- **icmp** ：ICMP エコー メッセージ（ping）。これがデフォルトのキープアライブ タイプです。

  ICMP キープアライブは、クラス A タイプのキープアライブです。CSS は、このタイプのキープアライブを最大 2048 までサポートしており、最大 2048 の ICMP キープアライブを並行して実行できます。

- **named** *name*：（サービス モードのみ）サービスを定義済みのグローバル キープアライブに関連付ける。

  このコマンドを使用する場合は、事前に **(config-keepalive) active** コマンドでグローバル キープアライブをアクティブにしておく必要があります。サービスをグローバル キープアライブに割り当てると、サービス モードで割り当てたキープアライブ プロパティが無効になります。グローバル キープアライブの作成については、「グローバル キープアライブの設定」の項を参照してください。

- **none** ：キープアライブ メッセージはサービスに送信されない。

- **script** *script_name* {**“***arguments***”**} {**use-output**}：サービスで使用するスクリプト キープアライブを定義する。スクリプトは、キープアライブが実行されるたびに実行されます。既存のスクリプト キープアライブの名前を入力します。

オプションの *arguments* 変数はキープアライブ スクリプトに引数を渡します。スペースを含む 128 文字以内のテキスト文字列を引用符で囲んで入力します。

use-output オプションを使用すると、実行した各コマンドの出力を解析できます。このオプションのキーワードによって、スクリプト内で **grep** とファイル指定を使用できます。デフォルトでは、スクリプトは出力を解析しません。スクリプト キープアライブの使用方法の詳細については、この章で後述する「サービスでのスクリプト キープアライブの使用」を参照してください。

スクリプト キープアライブは、クラス B タイプのキープアライブです。CSS は、このタイプのキープアライブを最大 256 サポートしており、最大 16 のスクリプト キープアライブを並行して実行できます。

（注） CSS のシステム リソースを節約するために、スクリプト キープアライブは必要な場合だけに使用してください。サービスを検証するのに ICMP または HTTP キープアライブ メッセージで十分な場合は、スクリプト キープアライブではなく、これらのメッセージのタイプを使用してください。

- **ssl** ：このサービスの SSL HELLO キープアライブ。CSS は、クライアント HELLO を送信して SSL サーバに接続します。CSS は、サーバから HELLO を受信すると、TCP RST を送信して接続を閉じます。

  SSL キープアライブは、クラス A タイプのキープアライブです。CSS は、このタイプのキープアライブを最大 2048 サポートしており、最大 2048 の SSL キープアライブを並行して実行できます。

  Cisco CSS 11500 シリーズで SSL モジュールを使用する場合は、キープアライブ タイプ **none** を使用します。SSL モジュールは CSS に内蔵された装置であり、サービスへのキープアライブ メッセージは必要ありません。

- **tcp** :3 ウェイ ハンドシェイクとリセット（SYN、SYN-ACK、ACK、RST-ACK）を通じてサービスの利用可能状況を判定する。デフォルトでは、CSS は RST を送信して TCP キープアライブのサーバ ポートのソケットを閉じます。FIN を使用してサーバのソケットを正常に閉じる必要がある場合は、**tcp-close fin** コマンドを使用することによって、キープアライブで FIN を送信し、ソケットを閉じることができます。**tcp-close** コマンドの詳細については、「正常ソケット クローズ（FIN）を使用する TCP キープアライブの設定」を参照してください。

TCP キープアライブは、クラス A タイプのキープアライブです。CSS は、このタイプのキープアライブを最大 2048 サポートしており、最大 2048 の TCP キープアライブを並行して実行できます。

正常ソケット クローズ（FIN）を使用する TCP キープアライブの設定

デフォルトでは CSS は RFC 1122 に準拠し、リセット（RST）を送信して TCP キープアライブのサーバ ポートのソケットを閉じます。RST に必要なパケットが 1 つであるのに対し、FIN に必要なパケットは 最大 4 つなので、RST は FIN より早く処理されます。FIN を使用してサーバのソケットを正常に閉じる必要がある場合は、FIN を送信してソケットを閉じるようにキープアライブを設定できます。

- FIN　を送信してソケットを閉じるようにキープアライブを設定するには、サービス モードで **keepalive tcp-close fin** コマンドを使用します。次に例を示します。

  ```
  (config-service[serv1])# keepalive tcp-close fin
  ```

  キープアライブをリセットして RST を送信するには、次のように入力します。

  ```
  (config-service[serv1])# keepalive tcp-close rst
  ```

- FIN を送信してソケットを閉じるようにグローバル キープアライブを設定するには、キープアライブ モードで **tcp-close fin** コマンドを使用します。次に例を示します。

  ```
  (config-keepalive[keepimages])# tcp-close fin
  ```

  キープアライブをリセットして RST を送信するには、次のように入力します。

  ```
  (config-keepalive[keepimages])# tcp-close rst
  ```

キープアライブ ポート番号の設定

デフォルトでは、キープアライブに使用されるポートの番号は、キープアライブのタイプで決まります。キープアライブのタイプによって、デフォルトのポート番号は次のようになります。

- HTTP または TCP ：80
- FTP ：21 （変更不可）

- SSL ：443
- ICMP ：サービスの番号

ポート番号は 0 ～ 65535 の範囲内で設定できます。

- 特定のサービスを対象にキープアライブのポート番号を設定するには、サービス モードで **keepalive port** コマンドを使用します。たとえば、キープアライブのポートを 8080 に設定するには、次のように入力します。

  ```
  (config-service[serv1])# keepalive port 8080
  ```

  キープアライブのポート番号をデフォルトの 0 にリセットするには、次のように入力します。

  ```
  (config-service[serv1])# no keepalive port
  ```

- グローバル キープアライブのポートを指定するには、キープアライブ モードで **port** コマンドを使用します。たとえば、ポート 8080 を指定するには、次のコマンドを入力します。

  ```
  (config-keepalive[keepimages])# port 8080
  ```

  キープアライブのポート番号をデフォルトの 0 にリセットするには、次のように入力します。

  ```
  (config-keepalive[keepimages])# no port
  ```

HTTP キープアライブ方式の設定

CSS で HTTP キープアライブ タイプを設定すると、デフォルトでは HTTP-HEAD 方式が使用されます。CSS はサービスに HTTP-HEAD 方式を送信し、200 OK ステータスを要求します。このタイプのキープアライブの参照ハッシュ値は計算されません。200 OK ステータスが返されない場合、CSS はサービスが停止したとみなします。

HTTP GET 方式が使用されるように設定することもできます。CSS はサービスに HTTP GET 方式を送信し、Message Digest Algorithm Version 5 （MD5）ハッシュ値を計算し、その値を参照ハッシュとして格納します。後続の GET では、200 OK ステータス（HTTP コマンドによって OK 応答が完了したことを示す）とハッシュ値が、参照ハッシュ値と同じ値になることが必要です。200 OK ステータスが返されない場合、または 200 OK ステータスが返されてもハッシュ値が参照ハッシュ値と異なる場合、サービスが停止しているとみなされます。

HTTP キープアライブの HTTP URI コンテンツ情報を指定すると、そのコンテンツのハッシュ値が計算されます。コンテンツ情報が変わると、ハッシュ値は元のハッシュ値と一致しなくなり、CSS はサービスが停止したとみなします。ハッシュ値のミスマッチが原因でサービスが停止したとみなされないようにするには、キープアライブ方式を HTTP HEAD に指定します。

HTTP 応答コードの設定については、「キープアライブ HTTP 応答コードの設定」を参照してください。HTTP URI の設定については、「キープアライブ URI の設定」を参照してください。

- 特定のサービスを対象に HTTP キープアライブ方式を指定するには、サービス モードで **keepalive method** コマンドを使用します。たとえば、HTTP GET キープアライブ方式を指定するには、次のように入力します。

  ```
  (config-service[serv1])# keepalive method get
  ```

  HTTP キープアライブ方式を HTTP HEAD にリセットするには、次のように入力します。

  ```
  (config-service[serv1])# keepalive method head
  ```

- グローバル キープアライブの HTTP キープアライブ方式を指定するには、キープアライブ モードで **method** コマンドを使用します。たとえば、HTTP GET キープアライブ方式を指定するには、次のように入力します。

  ```
  (config-keepalive[keepimages])# method get
  ```

  HTTP キープアライブ方式を HTTP HEAD にリセットするには、次のように入力します。

  ```
  (config-keepalive[keepimages])# method head
  ```

アクティブなサービスのキープアライブ方式を変更する場合は、サービスをいったん中断してもう一度有効化するまで、変更は反映されません。

（注） デフォルトでは、HTTP キープアライブは固定接続を使用します。HTTP 固定キープアライブで固定接続の確立に失敗すると、非固定接続の確立が試みられます。その結果、非固定接続が確立されると、キープアライブは成功です。その後、次のキープアライブ間隔が来た時点で、同じキープアライブで固定接続の確立が試みられます。

キープアライブ HTTP 応答コードの設定

デフォルトでは、CSS は HTTP-HEAD キープアライブを送信すると、問い合せ対象のサーバからの応答パケットで応答コード 200 が返されることを予期しています。HTTP-HEAD 非固定キープアライブの場合、200 以外の応答コード（たとえば、302 リダイレクト応答コード）を予期するように CSS を設定できます。応答コードとして、100 ～ 999 の範囲内の整数を入力します。

- 特定のサービスにキープアライブの応答コードを指定するには、サービス モードで **keepalive http-rspcode** コマンドを使用します。たとえば、応答コード 302 を指定するには、次のように入力します。

  ```
  (config-service[serv1])# keepalive http-rspcode 302
  ```

  応答コードをデフォルト値の 200 にリセットするには、次のように入力します。

  ```
  (config-service[serv1])# no keepalive http-rspcode
  ```

- グローバル キープアライブの応答コードを指定するには、**http-rspcode** コマンドを使用します。たとえば、応答コード 302 を指定するには、次のように入力します。

  ```
  (config-keepalive[keepimages])# http-rspcode 302
  ```

  応答コードをデフォルト値の 200 にリセットするには、次のように入力します。

  ```
  (config-keepalive[keepimages])# no http-rspcode
  ```

キープアライブ URI の設定

HTTP キープアライブ タイプを設定すると、CSS は URI 文字列を使用して、対象のサービスが有効かどうかを判定します。デフォルトでは、CSS はルート ディレクトリ（/）を示す URI 文字列を使用します。HTTP Get の場合、CSS は URI 情報を使用してハッシュ値を計算します。HTTP キープアライブには、URI コンテンツ情報を指定できます。

（注） HTTP キープアライブに URI のコンテンツ情報を指定すると、そのコンテンツのハッシュ値が計算されます。コンテンツ情報が変わると、ハッシュ値は元のハッシュ値と一致しなくなり、CSS はサービスが停止したとみなします。ハッシュ値のミスマッチが原因でサービスが停止したとみなされないようにするには、**keepalive method** を **head** として指定します。このタイプのキープアライブのハッシュ値は計算されません。コンテンツが変わる Web ページを指定する場合に head キープアライブ方式を指定しないときは、コンテンツが変わるたびにサービスを一時停止してもう一度アクティブにする必要があります。

URI のコンテンツ情報は、64 文字以内のテキスト文字列を引用符で囲んで入力します。文字列には、ホスト情報は含めないでください。CSS はサービスの IP アドレスとキープアライブのポート番号からホスト情報を取得します。

- 特定のサービスに HTTP キープアライブのコンテンツ情報を指定するには、サービス モードで **keepalive uri** コマンドを使用します。次に例を示します。

  ```
  (config-service[serv1])# keepalive uri "/index.html"
  ```

  キープアライブのコンテンツ情報をクリアするには、次のように入力します。

  ```
  (config-service[serv1])# no keepalive uri
  ```

- グローバル キープアライブの HTTP キープアライブのコンテンツ情報を指定するには、**uri** コマンドを使用します。たとえば、グローバル キープアライブのコンテンツ情報を入力するには、次のように入力します。

  ```
  (config-keepalive[keepimages])# uri "/index.html"
  ```

  このキープアライブに指定されているコンテンツ情報を消去するには、次のように入力します。

  ```
  (config-keepalive[keepimages])# no uri
  ```

キープアライブのハッシュ値の設定

デフォルトでは、CSS は HTTP GET キープアライブに MD5 ハッシュを使用します。このハッシュ値は、すべての HTTP GET 応答の計算されたハッシュ値と比較されます。比較が成功すると、キープアライブが「Alive」状態に維持されます。

サービス キープアライブの場合、デフォルト（MD5 ハッシュ）以外のハッシュ値が使用されるように指定するには、サービス モードで **keepalive hash** コマンドを使用します。サービス キープアライブのハッシュ値を設定する手順は次のとおりです。

1. キープアライブを設定します。次の例では、テスト　ページに対してキープアライブ GET を作成します。

```
(config)# service serv1
(config-service[serv1])# ip address 10.0.3.21
(config-service[serv1])# keepalive type http
(config-service[serv1])# keepalive method get
(config-service[serv1])# keepalive uri "/testpage.html"
(config-service[serv1])# keepalive hash
"1024b91e516637aaf9ffca21b4b05b8c"
(config-service[serv1])# active
```

2. **show keepalive** コマンドを使用してハッシュ値を表示します。次に例を示します。

```
(config-service[serv1])# show keepalive
Keepalives:

Name: serv1
Index: 0          State: ALIVE
Description: Auto generated for service serv1
Address: 10.0.3.21  Port: 80
Type:             HTTP:GET:/testpage.html
Hash:             1024b91e516637aaf9ffca21b4b05b8c
Frequency:        5
Max Failures:     3
Retry Frequency:  5
Dependent Services:
```

3. 表示されているハッシュ値を使用して、キープアライブのハッシュ値を設定します。MD5 ハッシュには、32 文字以内の 16 進文字列を引用符で囲んで入力します。次に例を示します。

```
(config-service[serv1])# keepalive hash
"1024b91e516637aaf9ffca21b4b05b8c"
```

実行設定（running-config）のサービス設定の一部を次に示します。

```
service serv1
    ip address 10.0.3.21
    keepalive type http
    keepalive method get
    keepalive uri "/testpage.html"
    keepalive hash "1024b91e516637aaf9ffca21b4b05b8c"
    active
```

ハッシュ値を消去し、デフォルト値に戻すには、次のように入力します。

```
(config-service[serv1])# no keepalive hash
```

グローバル キープアライブの場合、HTTP GET キープアライブでデフォルト（MD5 ハッシュ）以外のハッシュ値が使用されるように指定するには、**hash** コマンドを使用します。グローバル キープアライブのハッシュ値を設定する手順は次のとおりです。

1. グローバル キープアライブを設定します。次に例を示します。

```
(config-keepalive[keepimages])# method get
(config-keepalive[keepimages])# uri "/testpage.html"
(config-keepalive[keepimages])# hash
"1024b91e516637aaf9ffca21b4b05b8c"
```

2. サービスを設定します。次に例を示します。

```
(config)# service imageserver1
(config-service[imageserver1])# ip address 10.0.3.21
(config-service[imageserver1])# keepalive type named keepimages
(config-service[imageserver1])# active
```

3. **show keepalive** コマンドを使用してハッシュ値を表示します。次に例を示します。

```
(config-keepalive[keepimages])# show keepalive

Keepalives:

Name: imageserver1
Index:           0           State:      ALIVE
Description:     Auto generated for service serv1
Address:         10.0.3.21    Port:80
Type:            HTTP GET:/testpage.html
Hash:            1024b91e516637aaf9ffca21b4b05b8c
Frequency:       5
Max Failures:    3
Retry Frequency: 5
Dependent Services:
```

4. 表示されているハッシュ値を使用して、キープアライブのハッシュ値を設定します。32 文字以内の 16 進文字列の MD5 ハッシュ値を引用符で囲んで入力します。次に例を示します。

```
(config-keepalive[keepimages])# hash
"1024b91e516637aaf9ffca21b4b05b8c"
```

実行設定（running-config）のサービス設定の一部を次に示します。

```
service imageserver1
    ip address 10.0.3.21
    keepalive type http
    keepalive method get
    keepalive uri "/testpage.html"
    keepalive hash "1024b91e516637aaf9ffca21b4b05b8c"
    active
```

ハッシュ値を消去し、デフォルト値に戻すには、次のように入力します。

```
(config-keepalive[keepimages])# no hash
```

キープアライブ設定の表示

サービスのキープアライブ情報を表示するには、**show service** コマンドを使用します。このコマンドと、コマンド出力内容については、第 3 章「サービスの設定」の「サービス設定の表示」を参照してください。

グローバル キープアライブ設定を表示するには、**show keepalive** コマンドを使用します。既存のキープアライブ名のリストを表示するには、**show keepalive ?** を使用します。

（注） キープアライブ データには、2 つのセッション（SSH、コンソール、Telnet など）から同時にアクセスできます。一方のセッションがサービスまたはキープアライブをクリアして、キープアライブ データを再設定しているときに、もう一方のセッションが **show keepalive** コマンドでキープアライブ データの表示を試みると、**show** コマンドが中断され、次のメッセージが表示されることがあります。

```
Command Aborted!!! Configuration changed. Please reissue command.
```

このコマンドには、次のオプションがあります。

- **show keepalive** ：すべてのキープアライブに関する情報を表示する。
- **show keepalive** *keepalive_name* ：特定のキープアライブに関する情報を表示する。
- **show keepalive-summary** ：すべてのキープアライブに関するサマリー情報を表示する。

次に、例を示します。

```
(config)# show keepalive

Keepalives:

Name:          keepimages  Index: 1   State: ALIVE ( ICP Check )
Description:   This keepalive is for image servers
Address:       172.16.1.7 Port: 80
Type: HTTP:HEAD-302:/index.html
Frequency: 5
Max Failures: 3
Retry Frequency: 5
Dependent Services: imageserver1

Name: rualive   Index: 2           State: ALIVE
Description: Auto generated for service serv2
Address:        172.16.1.8         Port: 80
Type: HTTP:HEAD:/index.html
Frequency: 5
Max Failures: 3
Retry Frequency: 5
Dependent Services: serv2

(config)# show keepalive-summary

Keepalives:
Alive1         DOWN     192.25.1.7
Alive2         ALIVE    192.25.1.8
```

表 4-3 に、**show keepalive** コマンド出力で表示されるフィールドについて説明します。

表 4-3　show keepalive コマンド出力のフィールド

フィールド	説明
Name	キープアライブの名前
Index	CSS により各キープアライブに割り当てられた一意のインデックス値
State	キープアライブの状態。Down、Alive、Dying、および Suspended の状態があります。
Description	キープアライブの説明

表 4-3　show keepalive コマンド出力のフィールド（続き）

フィールド	説明
Address	キープアライブ メッセージの送信先 IP アドレス
Port	キープアライブのポート番号
Type	キープアライブに割り当てられているキープアライブ メッセージのタイプ。タイプには、FTP、HTTP、ICMP、script、SSL、TCP、または named があります。HTTP Head キープアライブでは、応答コードも表示されます。
Encryption	HTTP キープアライブの SSL 暗号化状態。値は Enabled（有効）または Disabled （無効）です。
Frequency	キープアライブ メッセージを IP アドレスに送信する間隔（秒数）。範囲は 2 ～ 255 です。デフォルトは 5 です。
Max Failures	IP アドレスがキープアライブ メッセージの応答に失敗して「down」とみなされるまでの回数。範囲は 1 ～ 10 です。デフォルトは 3 です。
Retry Frequency	メッセージをキープアライブ IP アドレスに送信するときの再試行間隔（秒数）。範囲は 2 ～ 255 です。デフォルトは 5 です。
Dependent Services	キープアライブを使用するように現在設定されているサービス。これは主に named タイプのキープアライブで使用されます。

サービスでのスクリプト キープアライブの使用

スクリプト キープアライブは、特定のサービス要件に合せたカスタムのキープアライブを作成するときに使用するスクリプトです。スクリプトを作成するには、CSS ソフトウェアに付属している CSS スクリプト言語を使用します。**socket** コマンドの使用法やキープアライブ スクリプトの例などの、CSS スクリプト言語の使用法の詳細については、『*Cisco Content Services Switch Administration Guide*』を参照してください。

現在、CSS には FTP、HTTP、ICMP、SSL、および TCP に対応するキープアライブが用意されています。キープアライブ メッセージ設定の詳細については、この章で前述した「CSS のキープアライブの概要」を参照してください。

スクリプトを使用すると、デフォルトのキープアライブよりも高度な CSS キープアライブ機能を使用できます。たとえば、CSS を Post Office Protocol 3（POP3）メール サーバに接続するための特別なスクリプトを作成できます。

スクリプトをオフラインで作成し、これを CSS にアップロードして、サービスにスクリプト キープアライブ オプションを設定することができます。

CSS では最大 256 のスクリプト キープアライブをサポートします。実行された各コマンドに対して、出力を解析するスクリプトを指定する場合、スクリプト出力を使用するキープアライブは、16 まで設定できます。

（注） 対応するスクリプトが CSS に存在しない場合でも、スクリプト キープアライブを設定することができます。この場合、適切なスクリプトを CSS にアップロードするまでサービスの状態は「Down」のままになっています。したがってこの間に設定を決めて実装し、すべてのスクリプトを CSS にアップロードすることができます。

スクリプト キープアライブの考慮事項

スクリプト キープアライブを設定する際は、ここに示す例外事項を除き、キープアライブ タイプの一般的なガイドラインに従ってください。キープアライブの詳細については、この章で前述した「CSS のキープアライブの概要」を参照してください。

- スクリプト　キープアライブ以外では処理できないキープアライブ操作をサポートするために、CSS はスクリプト キープアライブを提供しています。スクリプト キープアライブでの I/O 操作は、サーバへのネットワーク接続のプローブに使用するソケット操作と、サーバ上のアプリケーションの状態チェックだけにとどめてください。スクリプト言語は CSS のハード ドライブやフラッシュ ディスク上のファイル I/O をサポートしていますが、スクリプト キープアライブ内ではファイル I/O 操作は行わないでください。スクリプト キープアライブで過度にファイル I/O 操作を行うと、サービスが遷移する場合があります。ファイル システムには、CLI またはコマンド スケジューラから実行するスクリプトでアクセスできます。
- CSS スクリプト言語では、128 文字の引数を引用符で囲んで渡すことができます。1 つの引数につき平均 7 文字（およびスペース区切り）とすると、1 つのスクリプトで最大 16 の引数を使用できます。
- CSS は、スクリプト キープアライブの各行を実行します。アプリケーションに多数のスクリプト キープアライブが必要な場合（60 を超える場合など）、各スクリプトはなるべく短く簡潔にしてください。スクリプトを短くすると、スクリプトが長い場合よりも実行時間がかなり短くなります。CSS システムのパフォーマンスを最大にするために、実行時間が長くなるような複雑なプロトコルや細かいスクリプトは避けてください（データベースへの問い合せ、検証を伴う完全なログインを避けるなど）。
- Tab キーまたは ? キーを押すだけで、使用可能なキープアライブ スクリプトを簡単に参照できるように、スクリプトには命名規則 *ap-kal-type* を使用します。たとえば、SMTP スクリプトの名前は ap-kal-smtp にします。スクリプト名は、32 文字以内で入力します。引数には、128 文字以内のテキスト文字列を引用符で囲んで入力します。
- 設定したスクリプト　キープアライブで、該当のスクリプトを検索できるように、スクリプトを /<*current running version*>/script ディレクトリに置く必要があります。スクリプト キープアライブを設定する場合は、スクリプト名だけを使用します（CSS はパス名を受け入れません）。スクリプトが CSS の他のディレクトリに存在する場合、スクリプト キープアライブではスクリプトが存在しないものとみなされます。

/<*current running version*>/script ディレクトリ内の、使用可能なすべてのスクリプトのリストを表示するには、**Tab** キーまたは **?** キーを押します。リストに存在しないスクリプト名を入力して、そのスクリプトを後からアップロードすることもできます。スクリプトの操作には、**archive**、**clear**、および **copy** コマンドを使用します。ローカルのハード ドライブから CSS の /script ディレクトリへスクリプトをアップロードしたり、CSS の /script ディレクトリからローカルのハード ディスクへスクリプトをダウンロードすることもできます。

- スクリプトの多くは、接続、要求の送信、および特定タイプの応答待ちなどの複数のプロセスを実行するため、スクリプト キープアライブの **frequency** の値は、標準のキープアライブの値よりも長く設定してください。間隔を 10 秒以上に設定すると、キープアライブは確実に完了します。10 秒未満の場合は、通常よりも状態遷移が多く発生します。
- CSS　は、サービス範囲の最初のアドレスだけにキープアライブを送信します。サービスに IP アドレス範囲を設定し、そのサービスへの IP アドレスをスクリプト キープアライブに設定する場合、後者にはサービス範囲の先頭アドレスを使用する必要があります。
- CSS はスクリプト全体をメモリに読み込むため、スクリプト キープアライブの最大容量は 200 KB （約 6,000 行）に制限されます。この制限を超えると、スクリプトは読み込まれません。これは、すべてのアプリケーションにあてはまります。たとえば、CSS ソフトウェアに付属しているスクリプト キープアライブは、約 1KB です。できるだけ CSS メモリを節約するために、複数のサービスで共通のスクリプト キープアライブを共有して、スクリプトの 1 つのインスタンスだけをメモリに常駐させることができます。ただし、スクリプトを実行するサービスごとにスクリプト キープアライブを設定する必要があります。

（注） スクリプト　キープアライブを使用するサービスが多数ある場合は、スクリプト キープアライブの処理に使用するグローバル キープアライブのサブセットを小さくします。グローバル キープアライブの詳細については、この章で前述した「グローバル キープアライブの設定」を参照してください。

スクリプト キープアライブの設定

スクリプト キープアライブは、特定のサービス要件に合せたカスタムのキープアライブを作成するときに使用するスクリプトです。スクリプト キープアライブを設定するには、**keepalive type script** コマンドを使用します。このサービス設定モードのコマンドのシンタックスは次のとおりです。

keepalive type script *script_name* {**“***arguments***”**} {**use-output**}

既存のスクリプト キープアライブの名前を入力します。オプションの *arguments* 変数はキープアライブ スクリプトに引数を渡します。スペースを含む 128 文字以内のテキスト文字列を引用符で囲んで入力します。

オプションの **use-output** キーワードを使用すると、実行した各コマンドの出力を解析できます。このオプションのキーワードによって、スクリプト内で **grep** とファイル指定を使用できます。最大 255 のスクリプト キープアライブのうち、最大 16 のスクリプト キープアライブをスクリプト出力に使用するように設定できます。デフォルトでは、スクリプトは出力を解析しません。

たとえば、ap-kal-httplist というスクリプト キープアライブを設定するには、次のように入力します。

```
(config-service[serv1)# keepalive type script ap-kal-httplist
“10.10.102.105 /default.htm”
```

上記の例では、**keepalive** コマンドを使用して、serv1 のサービス キープアライブを、スクリプト名が ap-kal-httplist で引数が "10.10.102.105 /default.htm" のスクリプト タイプに設定しています。このスクリプトは出力を解析しません。

サービスのスクリプト キープアライブを無効にするには、次のように入力します。

```
(config-service[serv1])# keepalive type none
```

サービスのスクリプト キープアライブの表示

スクリプト キープアライブをサービスに追加すると、このスクリプトは **show service** コマンドの出力に表示されます。スクリプト名は **Keepalive** フィールドに表示され、指定可能な引数は **Script Arguments** フィールドのすぐ下に表示されます。スクリプト引数が指定されていない場合、**Script Arguments** フィールドは表示されません。

次に、例を示します。

```
(config-service[serv1])# show service

Name: serv1                     Index: 1
    Type: Local                   State: Alive
    Rule (10.10.102.105 ANY ANY)
   Session Redundancy: Disabled
   Redirect Domain:
   Redirect String:
    Keepalive: (SCRIPT ap-kal-httplist 10   3   5)
    Script Arguments: "10.10.102.105 /default.htm"
    Script Error: None
    Script Run Time: 1 second
    Script Using Output Parsing:  No
   Last Clearing of Stats Counters  03/15/2002  13:45:01
    Mtu:               1500       State Transitions:   0
    Connections:       0          Max Connections:     0
    Total Connections: 0          Total Reused Conns:  0
    Weight:            1          Load:                2
```

（注） スクリプト キープアライブがエラーで終了した場合は、**Script Error** および **Script Run Time** フィールドを確認して、問題を追求できます。

show running-config コマンドを使用して、スクリプト キープアライブとその引数を表示することもできます。

次に、例を示します。

```
(config-service[serv1])# show running-config

service serv1
    ip address 10.10.102.105
    keepalive frequency 10
    keepalive type script ap-kal-httplist "10.10.102.105
    /default.htm"
    active
```

上記の例では、サービスに設定されているスクリプト キープアライブと引数が示されています。引数がスクリプトで指定されていない場合、スクリプト名の後ろの引用符で囲まれたテキストは表示されません。

スクリプト キープアライブのステータス コード

スクリプトは、0 または 0 以外のステータス コードを返します。0 以外のコードが返されると、CSS はサービス状態に「Dying」または「Down」のフラグを立てます。0 のコードが返されると、サービス状態に「Alive」のフラグを立てます。次に例を示します。

```
! Connect to the remote host
socket connect host einstein port 25 tcp
! Purposely fail
exit script 1
```

上記のスクリプトは **exit** コマンドを実行したときに失敗したため、0 以外の値を返しています。デフォルトでは、**connect** コマンドが失敗すると、スクリプトはシンタックス エラーで失敗します。スクリプトのロジックをチェックして、CSS が正しい値を返しているか確認してください。

WebNS ソフトウェアのアップグレードとスクリプト キープアライブ

CSS の WebNS ソフトウェアをアップグレードすると、アップグレード プロセスにより新しい /<*current running version*>/script ディレクトリが作成されます。そこで、カスタム スクリプト（カスタム スクリプト キープアライブを含む）を新しい /<*current running version*>/script ディレクトリにコピーして、CSS が認識できるようにする必要があります。

次の手順を行うことで、ソフトウェアのアップグレード後、カスタム スクリプト キープアライブは正常に動作します。

1. CSS の WebNS ソフトウェアをアップグレードします。『*Cisco Content Services Switch Administration Guide*』を参照してください。
2. 古い /<*current running version*>/script ディレクトリのスクリプトを新しい /<*current running version*>/script ディレクトリにコピーします。
3. CSS を再度ブートします。