



# HTTP ヘッダー ロード バランシングの設定

この章では、HTTP ヘッダー フィールド グループを作成して HTTP ヘッダー フィールドを設定することにより、HTTP ヘッダー ロード バランシングを設定する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。

この章の内容は次のとおりです。

- [HTTP ヘッダー ロード バランシングの概要](#)
- [HTTP ヘッダー ロード バランシングの設定のクイック スタート](#)
- [ヘッダー フィールド グループの作成](#)
- [ヘッダー フィールド グループの説明](#)
- [ヘッダー フィールド エントリの設定](#)
- [ヘッダー フィールド グループのコンテンツ ルールへの関連付け](#)
- [コンテンツ ルールのヘッダー フィールド グループ設定の表示](#)
- [ヘッダー フィールド グループの表示](#)
- [ヘッダー フィールド グループの設定例](#)



(注)

HTTP ヘッダー ロード バランシングを正常に動作させるには、サービスの再マッピングを有効にする必要があります。サービスの再マッピング機能の詳細については、[第 10 章「コンテンツ ルールの設定」](#)を参照してください。

## HTTP ヘッダー ロード バランシングの概要

CSS では、HTTP ヘッダー ロード バランシングを設定することにより、着信コンテンツ要求の HTTP ヘッダー フィールドを検査することができます。HTTP ヘッダー ロード バランシングにより、CSS は、HTTP ヘッダー フィールドの情報に基づいてロード バランシングを行い、要求されている種類のコンテンツを処理するサーバにコンテンツ要求を転送することが可能になります。

CSS は、ブラウザの種類や、同一コンテンツをエンド ユーザ向けにカスタマイズしたさまざまな表示形式に基づいて、コンテンツ要求を特定のサーバに転送することができます。たとえば、携帯情報端末を使用しているも、PC を使用して得るものと同じ内容で、ただグラフィックが少ないコンテンツを取得したり、特定の言語だけでコンテンツを表示したりすることができます。

HTTP ヘッダー ロード バランシングを使用することにより、すべてのサーバ上に同じコンテンツをさまざまな形式で複製する必要がなくなり、サーバの貴重なスペースを節約することができます。HTTP ヘッダー ロード バランシングを使用すると、クライアントの種類ごとにサーバを分類できるだけでなく、キャッシュできないトラフィックをバイパスして、検索エンジン サービスからのクライアントブラウザトラフィックを優先させることも可能になります。

## コンテンツ ルールでの HTTP ヘッダー ロード バランシングの使用

レイヤ 5 コンテンツ ルールで HTTP ヘッダー フィールド グループを使用すると、URL だけを定義するルールよりも明確なルールを作成することができます。HTTP ヘッダー フィールド グループを使用すると、コンテンツをより絞り込んで特定できます。コンテンツ ルールは階層構造になっているため、コンテンツの要求が複数のルールに一致した場合は、固有性の最も高いルールの特性がフローに適用されます。レイヤ 5 ルールの階層構造は、次のように定義されています。CSS では、以下の優先順位を使用してコンテンツへの要求を処理します。最も一致度の高いものが 1、最も一致度が低いものが 4 です。

1. ドメイン ネーム、IP アドレス、プロトコル、ポート、URL、HTTP ヘッダー フィールド グループ
2. IP アドレス、プロトコル、ポート、URL、HTTP ヘッダー フィールド グループ
3. ドメイン名、プロトコル、ポート、URL、HTTP ヘッダー フィールド グループ
4. プロトコル、ポート、URL、HTTP ヘッダー フィールド グループ

# HTTP ヘッダー ロード バランシングの設定のクイック スタート

表 12-1 に、HTTP ヘッダー ロード バランシングの作成および設定に必要な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。各機能および HTTP ヘッダー ロード バランシング設定のすべてのオプションについては、次の

表 12-1.

コンテンツ ルールに使用するサービスと所有者が作成されていることを確認します。表 12-1 のコマンド例では、所有者 arrowpoint、コンテンツ ルール rule1 の HTTP ロード バランシングを作成します。

表 12-1 HTTP ヘッダー ロード バランシングの設定のクイック スタート

---

## 作業とコマンドの例

---

1. `config` と入力して設定モードに入ります。

```
(config)#
```

2. ヘッダー フィールド グループを作成します。この例では、グループ `ppilot` を作成します。

```
(config)# header-field-group ppilot  
(config-header-field-group [ppilot])#
```

3. ヘッダー フィールド グループに説明を記述します (オプション)。

```
(config-header-field-group [ppilot])# description "ppilot content"
```

4. ヘッダー、フィールド、名前、フィールド タイプ、および演算子を定義してヘッダー フィールドのエントリを設定します。

```
(config-header-field-group [ppilot])# header-field palm1  
user-agent contain "MSIE" 20
```

5. ヘッダー フィールド グループをコンテンツ ルールに関連付けます。

```
(config-owner-content [arrowpoint-rule1])# header-field-rule  
ppilot
```

6. (推奨) ヘッダー フィールド グループを表示して設定内容を確認します。

```
(config)# show header-field-group
```

---

表 12-1 に示した各コマンドを実行すると、次のような実行設定が得られます。

```
!***** HEADER FIELD GROUP *****
header-field-group ppilot
  description "ppilot content"
  header-field palml user-agent contain "MSIE" 20

!***** OWNER *****
owner arrowpoint
  address "200 Beaver Brook Road, Boxborough, MA 01719"

content rule1
  vip address 192.1.1.100
  protocol tcp
  port 80
  add service server1
  header-field-rule ppilot
```

## ヘッダー フィールド グループの作成

ヘッダー フィールド グループ設定モードでは、ヘッダー フィールド グループを作成することができます。ヘッダー フィールド グループには、CSS コンテンツ ルールの検索プロセスに使用されるユーザ定義のヘッダー フィールド エントリのリストが含まれます。1つのグループには、複数のヘッダー フィールド エントリを含めることができます。



(注) CSS は最大 1024 のヘッダー フィールド グループと最大 4096 のヘッダー フィールド エントリをサポートします。



(注) 1つのグループ内に複数のヘッダー フィールド エントリがある場合、CSS は、各ヘッダー フィールド エントリが完全に一致してから関連するコンテンツ ルールを使用します。

ヘッダー フィールド グループの作成、または、ヘッダー フィールド グループ設定モードへのアクセスを実行するには、ブート モードおよび RMON モード以外のすべての設定モードから **header-field-group** コマンドを使用します。

プロンプトが (config-header-field-group [group\_name]) に変わります。また、ヘッダー フィールド グループ設定モードでこのコマンドを使用して他のグループにアクセスすることもできます。

このモード変換コマンドのシンタックスは次のとおりです。

**header-field-group** *group\_name*

作成するヘッダー フィールド グループの *group\_name* を入力します。さまざまなコンテンツ ルールでヘッダー フィールド グループを使用できるようにするため、各グループには一意の名前を定義してください。名前には、32 文字以内のテキスト文字列を入力します。既存のヘッダー フィールド グループのリストを表示するには、**header-field-group ?** コマンドを使用します。

## ■ ヘッダー フィールド グループ の説明

たとえば、次のように入力します。

```
(config)# header-field-group ppilot  
(config-header-field-group [ppilot])#
```

ヘッダー フィールド グループ を削除するには、**no header-field-group** コマンドを使用します。次に使用例を示します。

```
(config)# no header-field-group ppilot
```

## ヘッダー フィールド グループ の説明

**description** コマンドを使用して、ヘッダー フィールド グループ に説明を付けます。このコマンドのシンタックスは次のとおりです。

**description** *“text”*

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、次のように入力します。

```
(config-header-field-group [ppilot])# description "ppilot content"
```

ヘッダー フィールド グループ の説明を削除するには、次のコマンドを入力します。

```
(config-header-field-group [ppilot])# no description
```

## ヘッダー フィールド エントリの設定

ヘッダー フィールド エントリをヘッダー フィールド グループに設定して、HTTP ヘッダーのフィールドとそのフィールドで機能を実行する演算子を指定します。CSS が HTTP コンテンツ要求を受信すると、CSS はヘッダー フィールド エントリで指定された HTTP ヘッダー フィールドを調べて、*operator* 変数で指定されているフィールドで機能を実行します。CSS は、ヘッダー フィールドの演算結果を使用して、それ以降のフロー内の全パケットの負荷を分散します。

ヘッダー フィールド エントリには、ヘッダー フィールド名、使用するフィールド タイプ、実行する演算子、検索に使用されるヘッダー文字列、および検索の長さ（オプション）があります。

ヘッダー フィールド グループに複数のヘッダー フィールド エントリを設定した場合、ルールが適用されるためには、コンテンツ要求がこれらのエントリすべてと一致する必要があります。



(注)

CSS は最大 1024 のヘッダー フィールド グループと最大 4096 のヘッダー フィールド エントリをサポートします。

**header-field** コマンドを使用して、ヘッダー フィールド グループにヘッダー フィールド エントリを定義します。このコマンドのシンタックスは次のとおりです。

```
header-field name field_type {custom_string} operator {header_string  
  {search_length}}
```

変数とオプションは、次のとおりです。

- *name* : ヘッダー フィールド エントリを一意に識別する名前。名前は、1 ～ 31 文字で入力します。CSS では、ヘッダー フィールド グループ内で同じフィールド タイプを何度も使用する場合がありますため、ヘッダー フィールド エントリ名は必ず定義してください。

## ■ ヘッダー フィールド エントリの設定

- *field\_type* : フィールドタイプには、次のタイプのいずれかを入力する。
  - **user-agent** : 要求を発信しているユーザ エージェント (ソフトウェア プログラムなど) に関する情報。この情報は統計、プロトコル違反のトレース、およびユーザ エージェントの自動認識 (特定のユーザ エージェントの制約を回避するために応答を変更する) に使用されます。
  - **language** : 文書作成に使用される言語の ISO コード。言語コードは ISO 3316 (同じ言語で複数の地域に分かれている場合は、必要に応じて ISO639 の国番号を指定します)。
  - **host** : 要求されているリソースのインターネット ホスト名とポート番号。ユーザーまたは参照元リソースから最初に与えられた URI から取得されます。Host フィールドの値は、元のサーバまたは元の URL で指定されるゲートウェイの命名機関を表す必要があります。
  - **cache-control** : 要求と応答の過程を通じて、すべてのキャッシュ メカニズムが従うべき指令。この指令は、要求や応答がキャッシュから悪影響を受けないようにするための動作を指定します。
  - **pragma** : Pragma 指令で、関連するサーバで認識される。シンタックスは **accept** フィールドなど、HTTP の他の複数値フィールドの場合と同じです。複数のエントリをカンマ (,) で区切り、必要に応じてオプションのパラメータをセミコロン (;) で区切って指定します。
  - **encoding** : 使用される符号化方式
  - **charset** : 応答に使用できる文字セット。このフィールドによって、クライアントはより多彩な文字を含む文字セットや、特定用途向けの文字セットに対応していることをサーバに通知し、それらの文字セットを文書に使用できるようにします。
  - **connection** : 接続オプション
  - **referer** : リソースのアドレス (URI) (要求内の URI はそのリソースから取得された)
  - **accept** : セミコロンで区切られた表現方式 (コンテンツ タイプ メタ情報値) のリスト (この要求に対する応答として受け入れられる)
  - **request-line** : ブラウザでインターネット上のリソース (<http://www.cisco.com> など) にアクセスを試みると、そのリソースの要求を含む HTTP 要求メッセージがブラウザから送信される。要求行には、HTTP のメソッド (GET、HEAD、または PUSH)、要求先の Uniform Resource Locator (URL)、および HTTP のバージョンが含まれています。URI は、英数字 (および特殊文字) から構成される文字列で、インターネット上のリソースを識別します。要求行は、HTTP 要求メッセージの必須フィールドの 1 つです。



たとえば、HTTP 要求に次の URI が含まれているとします。

```
/cgi-bin/some-app.pl?session=123456789123456789&user=CiscoUser&action=LoadBalanceMe&foo=bar
```

ヘッダー フィールド グループとヘッダー フィールド ルールを作成すれば、URI の文字列に基づいてコンテンツ ルールを選択するように CSS を設定できます。上に示した URI 内の文字列 `LoadBalanceMe` を基にコンテンツ ルールを選択させる場合は、次のように設定します。

```
header-field-group url
header-field urlString request-line contain "LoadBalanceMe"
owner arrowpoint
content rule UrlString
vip address 192.168.128.151
protocol tcp
port 80
url "/"
add service server1
add service server2
header-field-rule url
active
content rule2
vip address 192.168.128.151
protocol tcp
port 80
url "/"
add service server21
add service server22
active
```

- **cookies** : クライアントを特定のサーバに固定するために使用される HTTP ヘッダー内の文字列
- **msisdn** : Wireless Application Protocol (WAP) 用のヘッダー フィールド タイプ。特定の無線ゲートウェイからの HTTP 要求では、HTTP ヘッダーに MSISDN フィールドが存在します。ヘッダー フィールド グループにこの **msisdn** ヘッダー フィールド タイプを設定することで、無線通信の要求の負荷を分散することができます。この章で後述する「[例 3 : HTTP 要求を MSISDN ヘッダー フィールドに基づいて負荷分散する無線設定](#)」を参照してください。

このオプションは単独で使用することも、スティッキ コマンド **advanced-balance wap-msisdn** と共に使用することもできます。第 11 章「[コンテンツ ルールへのスティッキ パラメータの設定](#)」の「[スティッキ コンテンツへの拡張ロード バランシング方式の指定](#)」を参照してください。

## ■ ヘッダー フィールド エントリの設定

- **custom** : ユーザ定義ヘッダー フィールドを示すフィールド タイプ キーワード。 *custom\_string* 変数と共に **custom** ヘッダー フィールドを使用して、HTTP ヘッダー フィールドの Name: value で HTTP ヘッダーを照合します。
- *custom\_string* : **custom** フィールド タイプで使用される、大文字小文字を区別しない英数字文字列。1 ~ 31 文字の英数字文字列を引用符で囲んで入力します。次に示す ASCII 文字は使用できません。制御文字 (10 進数: 0 ~ 31)、DEL (10 進数: 127)、および特殊文字、(、)、<、>、@、“、”、;、\、/、[、]、?、=、{、}、SP、HT。各 CSS で一意のカスタムヘッダー フィールドを最大で 16 個定義できます。



**(注)** すでに定義されている現在のヘッダー フィールド タグのいずれかと同じカスタムヘッダー フィールドは設定できません。

**custom** ヘッダー フィールドでは、現在のヘッダー フィールド照合ルールを使用します。つまり、新しい照合は追加しません。「例 4 : HTTP 要求をユーザ定義ヘッダー フィールドに基づいて負荷分散する設定」を参照してください。

たとえば、次のように入力します。

```
header-field customtag2 custom "Peak" contain "CD"
```

- *operator* : 次の演算子の 1 つを入力する。
  - **exist|not-exist** : **exist** 演算子および **not-exist** 演算子は、指定したヘッダー フィールドがコンテンツ要求のヘッダー内に存在するかどうかを確認する場合に使用する。
  - **equal|not-equal** {"*header\_string*"} : **equal** 演算子および **not-equal** 演算子は、定義した *header\_string* を指定のヘッダー フィールドの内容と照合し、*header\_string* がヘッダー文字列と等しいかどうかを調べる場合に使用する。*header\_string* には、スペースを含めて大文字小文字を区別しない 31 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。
  - **contain|not-contain** {"*header\_string*" {*search\_length*}} : **contain** 演算子および **not-contain** 演算子は、設定した *header\_string* を指定のフィールドタイプのコンテンツ内に含まれるサブ文字列と照合し、そのコンテンツに *header\_string* が含まれるかどうかを判別する場合に使用する。*header\_string* には、スペースを含めて大文字小文字を区別しない 31 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。

オプションとして、演算に使用するヘッダー フィールド部分を定義する *search length* を入力することもできます。検索の長さを定義しない場合は、演算子に (CR および LF で区切られた) ヘッダー フィールド全体が使用されます。検索の長さを定義する場合は、0 ~ 1024 の値を入力します。

たとえば、次のように入力します。

```
(config-header-field-group[ppilot])# header-field palm1 user-agent  
contain "MSIE" 20
```

```
(config-header-field-group[ppilot])# header-field palm2 user-agent  
contain "palm"
```

ヘッダー フィールドのエントリを削除するには、**no header-field** コマンドを使用します。次に使用例を示します。

```
(config-header-field-group[ppilot])# no header-field palm1
```



(注)

---

CSS からカスタム ヘッダー フィールドを完全に削除して再利用できるようにするには、**no header-field** コマンドを使用して CSS 上のすべてのヘッダー フィールドグループから該当のカスタム ヘッダー フィールドのインスタンスをすべて削除します。

---

## ヘッダー フィールド グループのコンテンツ ルールへの関連付け

ヘッダー フィールド グループをコンテンツ ルールに関連付けるには、**header-field-rule** コマンドを使用します。このコマンドでは、必要に応じてヘッダー フィールド グループに重みも設定できます。CSS では、この値を使用して、あるコンテンツ ルールを、類似のコンテンツ ルールよりも優先させることができます。たとえば、特定のサーバに対してフランス語を使用するクライアントのロード バランスを行い、さらに Microsoft Internet Explorer を使用するクライアントと、Netscape Navigator を使用するクライアントを区別する場合を考えます。フランス語を使用するクライアントを特定のサーバに転送することが、Internet Explorer または Netscape の使用に応じてサーバへ転送するよりも重要な場合は、「フランス語」のコンテンツ ルールを「Internet Explorer/Netscape」のコンテンツ ルールより優先させる値を指定します。



(注)

CSS は各コンテンツ ルールに対して、ヘッダー フィールド グループを 1 つだけサポートします。

このコンテンツ モード コマンドのシンタックスは次のとおりです。

```
header-field-rule name {weight number}
```

変数の内容は次のとおりです。

- **name** : コンテンツ ルールに使用するヘッダー フィールド グループの名前。グループのリストを表示するには、**header-field-rule ?** コマンドを入力します。
- **weight number** : ヘッダー フィールド グループに割り当てる重要度の値。0 ~ 1024 の値を入力します。デフォルト値は 0 です。

たとえば、次のように入力します。

```
(config-owner-content [arrowpoint-rule1])# header-field-rule french  
weight 3
```

コンテンツ ルールからヘッダー フィールド グループを削除するには、次のように入力します。

```
(config-owner-content [arrowpoint-rule1])# no header-field-rule
```

## コンテンツ ルールのヘッダー フィールド グループ 設定の表示

**show rule-header-field** コマンドを使用して、コンテンツ ルールに関連付けられたヘッダー フィールド グループの情報を表示します。たとえば、特定のコンテンツ ルールに関連付けられたヘッダー フィールド ルールおよびグループについての情報を表示するには、次のように入力します。

```
(config-owner-content [arrowpoint-rule1])# show rule header-field
```

## ヘッダー フィールド グループ の表示

**show header-field-group** コマンドを使用して、すべてのヘッダー フィールド グループまたは特定のグループの設定を表示します。このコマンドは、すべてのモードで使用可能です。

このコマンドのシンタックスとオプションは次のとおりです。

- **show header-field-group** : 設定されたすべてのヘッダー フィールド グループの要約情報を表示する。
- **show header-field-group all** : 設定されたすべてのヘッダー フィールド グループの詳細情報を表示する。
- **show header-field-group name** : 特定のヘッダー フィールド グループの詳細情報を表示する。

たとえば、設定されたすべてのヘッダー フィールド グループの要約情報を表示するには、次のように入力します。

```
(config)# show header-field-group
```

表 12-2 に、**show header-field-group** コマンドで表示されるフィールドについて説明します。

**表 12-2 show header-field-group コマンド出力のフィールド**

フィールド	説明
Header field group	ヘッダー フィールド グループの名前
Description	ヘッダー フィールド グループに設定されている説明

## ヘッダー フィールド グループの設定例

ヘッダー フィールド グループを設定する場合、一致度が高い明確なルールを設定することが重要です（たとえば、例 2）。ルールの特定が十分にできない場合、CSS はクライアント要求を最初に検出されたルールに一致させることがあるので、次の要求に対しては、最初に一致するルールが変わる可能性があります。

ここでは次の設定例を説明します。

- 例 1 : ルールの一致があいまいな設定
- 例 2 : ルールの一致機能を拡張した設定
- 例 3 : HTTP 要求を MSISDN ヘッダー フィールドに基づいて負荷分散する無線設定
- 例 4 : HTTP 要求をユーザ定義ヘッダー フィールドに基づいて負荷分散する設定

### 例 1 : ルールの一致があいまいな設定

例 1 は、あいまいな設定の例です。クライアント要求が、言語にフランス語を、ユーザ エージェントに Netscape を指定している場合、この要求はルール A2 およびルール A3 の両方に同等に一致する可能性があり、ルール一致の一貫性が維持されません。このあいまいさを解決する 1 つの方法として、ルール A2 とルール A3 との間に重要度が異なる値（設定例には示されていません）を使用します。ヘッダー フィールド グループ B をルール A2 に関連付ける際にこのヘッダー フィールド グループ B に重要度 10 の値を割り当てると、CSS は、このクライアント要求に対する一致ルールを常にルール A2 とみなします。また、例 2 の設定のように、より明確なルールを設定する方法もあります。

```
! ***** HEADER FIELD GROUP *****  
  
header-field-group A  
  header-field ual language equal "en"  
  
header-field-group B  
  header-field ua2 language equal "fr"  
  
header-field-group C  
  header-field-group ua3 user-agent contain "Netscape"  
! ***** OWNER *****
```

## ■ ヘッダー フィールド グループ の設定例

```
owner arrowpoint
  content ruleA
    protocol tcp
    vip address 192.168.128.151
    port 80
    url "/*"
    add service server1
    add service server2

  content ruleA1
    protocol tcp
    vip address 192.168.128.151
    port 80
    url "/*"
    header-field-rule A
    add service server11
    add service server12

  content ruleA2
    protocol tcp
    vip address 192.168.128.151
    port 80
    url "/*"
    header-field-rule B
    add service server21
    add service server22

  content ruleA3
    protocol tcp
    vip address 192.168.128.151
    port 80
    url "/*"
    header-field-rule C
    add service server31
    add service server32
```



## 例 2 : ルールの一致機能を拡張した設定

例 2 は、例 1 と同じ設定で、ルールの一致機能の部分だけを拡張した例です。各コンテンツ ルールは明確に定義されています。言語にフランス語を、ユーザ エージェントに Netscape を指定したクライアント要求は、コンテンツ ルール ruleA2 だけに一致します。

```
! ***** HEADER FIELD GROUP *****

header-field-group A
  header-field ua1 language equal "en"
  header-field ua2 user-agent contain "Netscape"

header-field-group B
  header-field ua3 language equal "fr"
  header-field ua4 user-agent contain "Netscape"

header-field-group C
  header-field ua5 language equal "en"
  header-field ua6 user-agent not-contain "Netscape"

header-field-group D
  header-field ua7 language equal "fr"
  header-field ua8 user-agent not-contain "Netscape"

! ***** OWNER *****

owner arrowpoint
  content ruleA
    protocol tcp
    vip address 192.168.128.151
    port 80
    url "/"
    add service server1
    add service server2

  content ruleA1
    protocol tcp
    vip address 192.168.128.151
    port 80
    url "/"
    header-field-rule A
    add service server11
    add service server12
```

## ■ ヘッダー フィールド グループの設定例

```
content ruleA2
  protocol tcp
  vip address 192.168.128.151
  port 80
  url "/"*
  header-field-rule B
  add service server21
  add service server22

content ruleA3
  protocol tcp
  vip address 192.168.128.151
  port 80
  url "/"*
  header-field-rule C
  add service server31
  add service server32

content ruleA4
  protocol tcp
  vip address 192.168.128.151
  port 80
  url "/"*
  header-field-rule D
  add service server41
  add service server42
```

**例 3 : HTTP 要求を MSISDN ヘッダー フィールドに基づいて負荷分散する無線設定**

例 3 では、クライアントが無線通信を利用しているかどうかでロード バランシングを決定する設定を示します。無線装置では、Wireless Application Protocol (WAP) を使用します。無線を利用するクライアントがコンテンツを要求すると、WAP プロトコル ゲートウェイ (要求を WAP プロトコル スタックから WWW プロトコル スタックに変換する装置) が、MSISDN フィールドを生成し HTTP ヘッダーにこれを追加します。MSISDN ヘッダー フィールドがあるかどうかは、ヘッダー フィールド グループのこのヘッダー フィールド エントリで、**exist** および **not-exist** 演算子を使用して確かめることができます。その後、MSISDN ヘッダー フィールドの有無に基づいて、ロード バランシングの決定を行うことができます。MSISDN ヘッダー フィールド タイプの設定については、この章で前述した「[ヘッダー フィールド エントリの設定](#)」を参照してください。

次の例では、送信先が VIP 192.168.128.151 の TCP ポート 80 のトラフィックで、HTTP ヘッダーに MSISDN フィールドがあるものは、すべてコンテンツ ルール ruleWap に一致します。送信先が VIP 192.168.128.151 の TCP ポート 80 のトラフィックで、HTTP ヘッダーに MSISDN フィールドがないものは、すべてコンテンツ ルール ruleNoWap に一致します。

```
header-field-group wap
  header-field 1 msisdn exist

owner arrowpoint
  content ruleWap
    vip address 192.168.128.151
    protocol tcp
    port 80
    url "/*"
    add service server1
    add service server2
    header-field-rule wap
    active

  content ruleNoWap
    vip address 192.168.128.151
    protocol tcp
    port 80
    url "/*"
    add service server21
    add service server22
    active
```



(注) MSISDN ヘッダー フィールドと、**advanced-balance wap-msisdn** コマンドを使用して、無線のユーザが電子商取引アプリケーションを利用できるように設定することができます。無線ユーザ設定の詳細については、[第 11 章「コンテンツ ルールへのスティッキ パラメータの設定」](#)の「[電子商取引アプリケーションを利用する無線ユーザの設定](#)」の項を参照してください。

#### 例 4 : HTTP 要求をユーザ定義ヘッダー フィールドに基づいて負荷分散する設定

例 4 では、カスタム ヘッダー フィールドに基づいて CSS がロード バランシングを決定できるような設定方法を示します。各 CSS で一意のカスタム ヘッダー フィールドを最大で 16 個定義できます。ただし、各ヘッダー フィールドグループ

## ■ ヘッダー フィールド グループ の設定例

ブで複数のカスタム ヘッダー フィールドを定義できます。複数のヘッダー フィールド グループで同じカスタム ヘッダー フィールドを設定すると、カスタム ヘッダー フィールドでは、該当の CSS で設定できる最大 16 個のカスタム ヘッダー フィールドの 1 つとしてのみカウントします。

次の例では、2 つの一意のカスタム ヘッダー フィールドが設定されます。タグ「Acme」および文字列「CD」を含むタグ「Peak」が設定されていて、送信先が VIP 192.168.128.15 である TCP ポート 80 のトラフィックは、コンテンツ ルール HTTPrule1 に一致します。文字列「CD」を含むタグ「Peak」が設定されていて、送信先が VIP 192.168.128.15 である TCP ポート 80 のトラフィックは、コンテンツ ルール HTTPrule2 に一致します。コンテンツ ルールに設定されているすべてのコマンドに基づいて最も適合するものを検索します。コンテンツ ルールの設定の詳細については、[第 10 章「コンテンツ ルールの設定」](#)を参照してください。

```
header-field-group group1
  header-field customtag1 custom "Acme" exist
  header-field customtag2 custom "Peak" contain "CD"

header-field-group group2
  header-field customtag1 custom "Peak" contain "CD"

owner arrowpoint

content HTTPrule1
  vip address 192.168.128.15
  protocol tcp
  port 80
  url "/*"
  add service server1
  add service server2
  header-field-rule group1
  active

content HTTPrule2
  vip address 192.168.128.15
  protocol tcp
  port 80
  url "/*"
  add service server3
  add service server4
  header-field-rule group2
  active
```