CHAPTER 13

スタンドアロン Content Engine の Rules Template の設定

この章では、スタンドアロン Content Engine の Rules Template（ルール テンプレート）の設定方法について説明します。Rules Template を使用して、Content Engine が HTTP、HTTPS、RTSP のトラフィックをフィルタリングするルールを指定します。これらの設定済みのルールにより、特定のヘッダーが書き換えられたり、要求がリダイレクトされたり、他の方法で要求が処理されることがあります。

この章の内容は、次のとおりです。

（注） *HTTP トラフィック*という用語は、HTTP、FTP-over-HTTP、HTTPS-over-HTTP を含む、HTTP からの要求を表します。Rules Template は、FTP ネイティブ要求に対してはサポートされません。

この章で使用する CLI（コマンドライン インターフェイス）コマンドの構文および使用方法については、『*Cisco ACNS Software Command Reference*』Release 5.5 を参照してください。

Rules Template の概要

Rules Template 機能を使用し、アクションやパターンで識別される一連のルールを指定します。この機能によって、HTTP、HTTPS、RTSP のトラフィックをフィルタリングするときに個々の規則を使用するようにスタンドアロン Content Engine を設定できます。この機能を使用して主に、要求された Web ページが既知のインターネット ワームのパターンに一致するかどうかをチェックし、一致した場合には要求を自動的にブロックし、組織内でのインターネット ワームやウィルスの蔓延を防ぐように Content Engine を設定します。

Content Engine 上でルール処理をイネーブルにしていた（Content Engine で Rules Template 機能をイネーブルにし、Content Engine 用のルールを設定していた）場合には、Content Engine は、着信の各クライアント要求をチェックし、要求されたコンテンツに一致するルール パターンがあるかどうかを調べます。要求に一致するルール パターンがあると、Content Engine は、指定されたアクション（ポリシー）を使用して、この着信トラフィックを処理します。

Content Engine により、着信要求を次のパターンと照合できます。

- IP アドレス、ネットワーク マスク、ポート リストを含む、コンテンツを要求しているクライアントの IP アドレス（送信元 IP アドレス）内のパターン
- IP アドレス、ネットワーク マスク、ポート リストを含む、オリジン Web または メディア サーバの IP アドレス（送信先 IP アドレス）内のパターン
- URL の正規表現
- URL のドメイン部の正規表現
- クライアントが要求している Web オブジェクトの MIME タイプ
- ドメイン名を記号化する正規表現
- 次のものを含む、要求の中で送信されるヘッダー
  - どのクライアント ソフトウェアが要求を発行しているかを示す「要求のユーザ エージェント」
  - ブラウザがどの Web ページからリンクにジャンプしたかを示す「参照元（referer）」
  - 要求ラインを示す「要求ライン（Request Line）」

Rules Template は、HTTP、FTP、HTTPS の各プロトコルに最適な機能です。HTTP、FTP、および HTTPS のほかに、RTSP などのストリーミング メディア オブジェクト プロトコルをサポートするポリシーを表 13-1 に示します。

表 13-1　Rules Template ポリシー

ポリシー（アクション）	HTTP 要求	WMT 要求	RTSP 要求
要求を許可します。	表 13-2 を参照してください。	*	*
ユーザ名を要求ヘッダーに追加します。			
要求をブロックします。		*	*
HTTP 応答ヘッダーを上書きして、オブジェクトをキャッシングします。			
HTTP 応答ヘッダーに応じてオブジェクトをキャッシングします。			
要求に対する認証をバイパスします。		*	*
特定のオブジェクト フレッシュネス計算係数を使用します。			
要求をリセットします。			

表 13-1　Rules Template ポリシー（続き）

ポリシー（アクション）	HTTP 要求	WMT 要求	RTSP 要求
オブジェクトをキャッシングしません。			
要求のアップストリーム プロキシをバイパスします。			
別の URL に要求をリダイレクトします。			*
オリジン サーバでオブジェクトを再検証します。			
URL を書き換えます。		*	*
指定された HTTP および HTTPS 要求について URL フィルタリングは行いません。			
特定の ICAP サーバを使用します。			
特定のアップストリーム プロキシを使用します。			
要求に特定のサーバを使用します。			
クライアントに送信する応答に ToS または DSCP を設定します。			
サーバに送信する応答に ToS または DSCP を設定します。			

プロトコルごとにサポートされるルール アクション

表 13-2 は、ACNS 5.3.1 ソフトウェアおよびそれ以降のリリースを実行しているスタンドアロン Content Engine でプロトコルごとにサポートするルール アクションの一覧です。アスタリスク（*）は、該当のプロトコルで 1 つのルール アクションがサポートすることを示しています。WCCP は透過的サポートを意味します。「*1」は、ACNS 5.1.5 ソフトウェアおよびそれ以降のリリースで、特定プロトコルで 1 つのルール アクションがサポートされることを示します。

RTSP ストリーミング プロトコルについて、**redirect** および **redirect_url_for_cdn rule** のアクションは、RealMedia Player からの RTSP 要求に対してはサポートされますが、WMT に対してはサポートされません。したがって、Windows Media Player からの RTSP 要求に対し、この 2 つのルール アクションはサポートされません。たとえば、Windows Media Services 9（WMS 9）は RTSP 要求に対する **block**、**reset**、**rewrite**、および **allow** の各ルール アクションをサポートします。ただし、RTSP 要求に対する **redirect** および **redirect_url_for cdn** の各ルール アクションはサポートしません。

表 13-2　プロトコルごとにサポートされるルール アクションのマトリックス

ルール アクション	プロトコル							
	HTTP-over-HTTP	FTP-over-HTTP	HTTPS-over-HTTP	HTTP WCCP	FTP-WCCP（ネイティブ FTP）	HTTPS-WCCP	HTTPS-WCCP（トンネル化、ACNS 5.1.5 またはそれ以降のソフトウェアでのみ使用可能）	RTSP
allow	*	*	*	*		*	ここにはいずれのルールも適用されません。	*
append-username-header	*			*		*		
block	*	*	*	*		*		*
cache-cookie	*			*		*		
cache-non-cacheable	*			*		*		
cache-only	*			*		*		
dscp	*	*		*		*		
freshness-factor	*			*		*		
insert-no-cache	*	*		*		*		
no-auth	*	*		*		*		
no-cache	*	*		*		*		
no-persistent-connection	*					*		
no-proxy	*	*		*				
no-url-filtering	*		*	*		*		
redirect	*	*		*		*1		*
redirect-url-for-cdn	*							*
refresh	*	*		*		*		
reset	*	*		*		*		*
rewrite	*			*		*1		*
use-dns-server	*			*				
use-icap-service	*							
use-proxy	*	*		*				
use-server	*			*		*1		
use-xforward-clt-ip	*					*		

アクションとパターンの概要

ルールは、*パターン*と *アクション*で指定します。

- *パターン*は、着信要求の制限を指定します。たとえば、パターンによって、送信元 IP アドレスがサブネット範囲 172.16.*.* に含まれることを指定できます。

  パターンを定義するときには、次の情報を指定します。

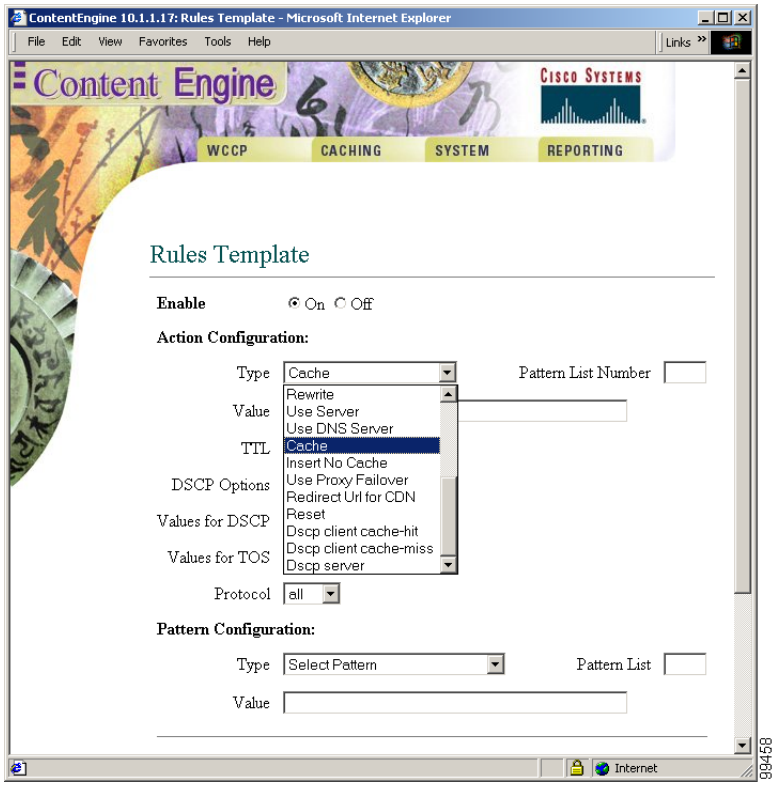
  - パターン タイプ（送信元 IP アドレスを表す「src-ip」）
  - パターン値（たとえば、送信元 IP アドレスがこのサブネット範囲に含まれることを示す「172.16.*.*」）
  - このパターンの追加先となるパターン リストの番号（たとえば、パターン リスト 10）

  サポートされているパターンの詳細リストについては、表 13-3 を参照してください。
- 着信要求が指定のパターンに一致すると、要求に基づいて 1 つの *アクション*が実行されます。アクションは、Content Engine が要求の処理時に実行します。たとえば、要求をブロックするアクションや、代替プロキシを使用するアクションなどがあります。サポートされているアクションの詳細リストについては、表 13-4 を参照してください。

Content Engine のルールは、動的に追加、表示、削除できます。ルールは、該当の Content Engine CLI コマンドや GUI を使用して NVRAM などの不揮発性ストレージに書き込まれるため、再起動後も保持されます。ルールはリソースを消費するため、定義するルールの数が多いほど、Content Engine のパフォーマンスに大きく影響します。Content Engine に設定できるルール数の制限についての詳細は、「Rules Template の設定」（p.13-21）を参照してください。

Content Engine CLI または GUI（図 13-1 を参照）を使用して、パターンとそれに対応するアクションを指定できます。

図 13-1　Rules Template ウィンドウ



（注）　Rules Template ウィンドウを表示するには、Content Engine GUI から **System > Rules Templates** の順に選択します。ウィンドウによる Rules Template の詳しい設定方法を表示するには、**HELP** ボタンをクリックします。

rule pattern-list グローバル コンフィギュレーション コマンドを使用して、1 つのパターン リストを作成し、そのリストに個々のパターンを追加できます。パターン リスト内のパターンは、相互に AND または OR 演算できます。パターン リスト（複数）を定義したら、**rule action** グローバル コンフィギュレーション コマンドを使用して、定義済みパターン リストに個々のアクションを関連付けることができます。次の例は、同じパターン リスト番号でさまざまなパターンを設定し、そのパターン リストを 1 つのアクションに適用することにより、どのようにパターンが AND 演算されるかを示したものです。

```
ContentEngine(config)# rule pattern-list 1 url-regex yahoo
ContentEngine(config)# rule pattern-list 1 dst-port 80
ContentEngine(config)# rule action block pattern-list 1
```

これ以降の説明では CLI を使用して、スタンドアロン Content Engine での Rules Template の設定方法を示します。Content Engine CLI による Rules Template の詳しい設定方法については、「Rules Template の設定」（p.13-21）を参照してください。

サポートされているルール パターン

ACNS 5.1 ソフトウェア リリースでは、新しい 3 つのルール パターン（**groupname**、**username**、および **groupname-regex**）が追加されました。これら 3 つのルール パターンは、認証済みの NTLM ユーザと LDAP ユーザのグループ名とユーザ名に基づいたアクセス制御ポリシーをサポートします。グループ名に基づいたルールは、NTLM と LDAP を通して認証されたユーザに適用されます。ユーザ名に基づいたルールは、LDAP、NTLM、RADIUS、TACACS+（認証用にユーザ名を含んだ要求認証方式）で認証されるユーザに適用されます。

次の例は **rule enable** グローバル コンフィギュレーション コマンドを使用して Content Engine でルール処理をイネーブルにし、「java」という文字列を含む FTP URL（クライアント ブラウザからの FTP 要求）からの、Engineering グループのすべてのエンド ユーザによるダウンロードをブロックする方法を表示しています。

```
ContentEngine(config)# rule enable
ContentEngine(config)# rule pattern-list 1 group-type and
ContentEngine(config)# rule pattern-list 1 groupname Engineering
ContentEngine(config)# rule pattern-list 1 url-regex java
ContentEngine(config)# rule action block pattern-list 1 protocol ftp
```

（注）　グループ名とユーザ名に基づいたルールによる許可は、HTTP 要求認証とグループに基づいたアクセス リスト許可の発生後にのみ発生します。Rules Template 内の設定とアクセス リストが一致した場合は、アクセス リストが優先されます。

表 13-3 は、パターン リストに追加できるルール パターンのタイプについて説明しています。

表 13-3　サポートされているパターンのタイプ

パターン	説明
domain	URL または Host ヘッダー内のドメイン名を正規表現と照合します。たとえば、「.*ibm.*」は、「ibm」サブストリングを含むすべてのドメイン名と一致します。「\.foo\.com$」は、「.foo.com」サブストリングで終わるすべてのドメイン名と一致します。正規表現の構文では、メタ文字のドル記号「$」は、このパターンが行の末尾にある場合だけ一致するように指定します。
dst-ip	要求の送信先 IP アドレスおよびネットマスクと、ルールに指定されている IP アドレスおよびネットマスクを照合します。
dst-port	要求の送信先ポート番号とルールに指定されているポート番号を照合します。
groupname	LDAP または NTLM で認証されたエンド ユーザ（コンテンツを要求している Web クライアント）のグループ名とルールに指定されているグループ名を照合します。たとえば、パターン リスト 1 にグループ名 Engineering を次のように指定します。 `ContentEngine(config)# rule pattern-list 1 groupname Engineering` このパターンは、LDAP または NTLM を通して認証されたユーザに対する要求認証にのみ適用できます。このパターンは、正確な文字列比較をサポートし、大文字と小文字が区別されます。グループ名の最大長は 129 文字です。有効な文字はアンダースコア、英数字、および Active Directory で許可されている特殊文字です。Rules Template 内のグループ名設定と、グループ名に基づいたアクセス リストが一致した場合は、アクセス リストの方が優先されます。 **ヒント**　グループ名パターンを使用する場合には、認証グループ キャッシュ内の最大グループ エントリの正しい数値を必ず設定してください（**http authentication cache max-group-entries** *number* グローバル コンフィギュレーション コマンド）。この数値は、許可クエリー中に返される最大グループ数（たとえば、AAA サーバで定義されている総グループ数）に対応します。数値は、500 ～ 12000 にし、認証グループ キャッシュ内のエントリ数は、Content Engine で使用可能なリソースにより異なります。
groupname-regex	エンド ユーザ（コンテンツを要求している Web クライアント）のグループ名と、ルールに指定されている正規表現を照合します。グループ名の最大長は 255 文字です。
group-type	パターン リストを AND と OR のどちらのタイプにするかを指定します。デフォルトは OR です。
header-field	要求ヘッダー フィールド パターン **referer**、**request-line**、および **user-agent** の要求ヘッダー フィールド パターンは、**block**、**reset**、**redirect**、**rewrite** の各アクションに対してサポートされます。**referer** パターンは要求内の Referer ヘッダーと、**request-line** パターンは要求の先頭行と、**user-agent** パターンは要求内の User-Agent ヘッダーと照合されます。
header-field-sub	要求の要求ヘッダーのパターンを照合します。
icap-service-name	使用する ICAP サービスの名前を指定します。

表 13-3　サポートされているパターンのタイプ（続き）

パターン	説明
mime-type	応答の MIME タイプと、ルールに指定されている MIME タイプ文字列（たとえば、http://www.faqs.org/rfcs/rfc2046.html の RFC 2046 に定義されている image/gif）を比較します。サブストリング（たとえば、java）を指定し、java のサブストリングを持つすべての MIME タイプ（たとえば、application/x-javascript）に、そのサブストリングを適用することができます。
src-ip	要求の送信元 IP アドレスおよびネットマスクと、ルールに指定されている IP アドレスおよびネットマスクを照合します。
username	LDAP、NTLM、RADIUS、TACACS+ を通して認証されたエンド ユーザ（コンテンツを要求している Web クライアント）のユーザ名とルールに指定されているユーザ名（複数）を照合します。LDAP、RADIUS、TACACS+ の認証の場合のユーザ名の最大長は 129 文字です。ユーザ名の有効な文字は英数字（a ～ z、A ～ Z、0 ～ 9）および次の特殊文字です。 ! @ # $ % ^ & ( ) - _ { } . ~ ` ただし、ユーザ名には次の特殊文字を含めることはできません。 \ + = : ; ? < > , このパターンは、正確な文字列比較をサポートし、大文字と小文字が区別されます。 同じパターン リストの同一ラインに複数のユーザ名を指定する場合は、区切り文字を使用します。 次に例を示します。 `ContentEngine(config)# rule pattern-list 1 username jdoe8,dsmith7,jsmith50` デフォルトでは、照合時にはドメイン名は考慮されず、ユーザ名のみが照合されます。照合にドメイン名とユーザ名を含める場合は、domainname\username のように指定します。 次に例を示します。 `ContentEngine(config)# rule pattern-list 1 username cisco\jdoe8` NTLM 認証の場合は、domain\username:password:NTLM 文字列を 50 文字以下にする必要があります。この文字列が 50 文字を超えると、ドメイン名が切り詰められるため、ルールのユーザ名パターンが一致しなくなります。その場合は、システム ログにエラー メッセージが生成されます。 特定ドメイン内のすべてのユーザを照合するには、次のように入力します。 `ContentEngine(config)# rule pattern-list 1 username domain` *`domainname`*`*` *domainname* とは、ドメインの名前（たとえば、cisco）です。

表 13-3　サポートされているパターンのタイプ（続き）

パターン	説明
URL-regex	URL と、ルールに指定されている正規表現を照合します。この照合では大文字と小文字は区別されません。次の Web サイトに示されている構文の正規表現を指定してください。 http://yenta.www.media.mit.edu/projects/Yenta/Releases/Documentation/regex-0.12/
URL-regsub	**rewrite** と **redirect** のアクションに対して、URL と正規表現を照合し、パターンの置換指定に基づいて新しい URL を作成します。この照合では大文字と小文字は区別されません。有効な置換インデックスの範囲は 1 ～ 9 です。

rule グローバル コンフィギュレーション コマンドによるスタンドアロン Content Engine へのパターンの指定方法については、「スタンドアロン Content Engine のためのルールの設定」（p.13-22）を参照してください。

サポートされているルール アクション

サポートされているルール アクションのリストを表示するには、**rule action ?** グローバル コンフィギュレーション コマンドを入力します。

```
ContentEngine(config)# rule action ?
  allow                     Allow the request
  append-username-header    Append the username in the header to the request
                            sent to the server
  block                     Block
  cache-cookie              Cache request containing Cookies
  cache-non-cacheable       Cache this object (Overriding HTTP response headers)
  cache-only                Cache this object only
  dscp                      Set IP TOS/DSCP (Differentiated Services) Field
  freshness-factor          Caching heuristic modifiers
  insert-no-cache           Insert no-cache headers to the response
  no-auth                   Do not authenticate
  no-cache                  Do not cache this object
  no-persistent-connection  Dont use persistent connection to all/client/server
                            connections
  no-proxy                  Do not use any upstream proxy
  no-url-filtering          Do not use URL filtering
  redirect                  Redirect request to rewritten URL
  redirect-url-for-cdn      Redirect alternate url for cdn content
  refresh                   Revalidate the object with the web server
  reset                     Issue a TCP RST
  rewrite                   Rewrite URL and fetch
  use-dns-server            Use a specific DNS server
  use-icap-service          Use a specific ICAP service
  use-proxy                 Use a specific upstream proxy
  use-server                Use a specific server
  use-xforward-clt-ip       Client IP in x-forwarded header will be used for
                            filtering.
```

表 13-4 は、定義済みのパターン リストに関連付けることができるアクション タイプの説明です。

表 13-4　サポートされているアクションのタイプ

アクション	説明
allow	着信要求を許可します。
append-username-header	サーバに送信される要求ヘッダーにユーザ名を追加します。
block	着信要求をブロックします。
cache-cookie	クッキーを含んだ要求をキャッシングします。

表 13-4　サポートされているアクションのタイプ（続き）

アクション	説明
cache-non-cacheable	HTTP 応答ヘッダーを上書きして、オブジェクトをキャッシングします。
cache-only	HTTP 応答ヘッダーに応じてオブジェクトをキャッシングします。オブジェクトが一致し、HTTP によるキャッシングが可能な場合にのみ、このオブジェクトをキャッシングします。ルール（複数）にこのアクションを指定した場合、オブジェクトがキャッシュに保存されるのは、オブジェクトが少なくとも 1 つの cache-only ルールに一致し、オブジェクト サイズ チェックや no-cache-on-authenticated-object チェックなど、他のキャッシング制限をすべて通過した場合のみです。オブジェクトがどの cache-only ルールにも一致しない場合、そのオブジェクトはキャッシングされません。
dscp client	クライアントの IP の Type of Service/Differentiated Services Code Point（ToS; サービス タイプ /DSCP）フィールド応答を設定します。ToS または DSCP の設定は、パケット マーキングと呼ばれます。これを使用すると、ネットワーク データを複数の優先レベルまたは ToS に分割できます。
dscp client cache-hit	キャッシュ ヒットによる DSCP クライアントへの応答
dscp client cache-miss	キャッシュ ミスによる DSCP クライアントへの応答
dscp server	オリジン サーバへの要求用に IP の ToS または DSCP コード ポイント フィールドを設定します。
dscp server match-client	クライアントの ToS または DSCP 値を使用します。
dscp server set-dscp	DSCP 値を指定します。DSCP 値のリストについては、表 11-1 を参照してください。
dscp server set-tos	ToS 値を指定します。ToS 値のリストについては、表 11-2 を参照してください。
freshness- factor	要求 URL が指定の正規表現と一致した場合に、Time to Live（TTL）を決定します。**refresh** 設定は、**freshness-factor** 設定より優先されます。
insert-no-cache	no-cache ヘッダーを応答に挿入します。
no-auth	認証を行いません。
no-cache	このオブジェクトをキャッシングしません。**no-cache** アクションと **selective-cache** アクションの両方が一致した場合には、**no-cache** が優先します。
no-persistent-connection all	どの接続にも固定接続を使用しません。
no-persistent-connection client	クライアント接続にのみ固定接続を使用しません。
no-persistent-connection server	サーバ接続にのみ固定接続を使用しません。
no-proxy	キャッシュ ミスの場合に、設定済みのアップストリーム プロキシを使用せず、ダイレクトにサーバに接続します。このアクションの使用方法の例については、「no-proxy アクションの例」（p.13-14）を参照してください。
no-url-filtering	HTTP および HTTPS 要求の URL フィルタリングをバイパスします。この機能は、ローカル リスト URL フィルタリング（優良サイトおよび悪質サイトのリスト）だけでなく、Websense、SmartFilter、または ACNS 5.2.3 ソフトウェアおよびそれ以降のリリースの N2H2 URL フィルタリングでサポートされています。このアクションの使用方法の例については、「no-url-filtering アクションの例」（p.13-11）を参照してください。

表 13-4　サポートされているアクションのタイプ（続き）

アクション	説明
redirect-url-for-cdn	ACNS ネットワーク コンテンツの場合に、元の要求を代替の URL にリダイレクトします。このルール アクションは、Content Distribution Manager に登録された Content Engine にのみ適用可能です。つまり、スタンドアロン Content Engine には適用されません。
redirect	元の要求を指定の URL にリダイレクトします。リダイレクションは、RADIUS サーバに **redirect** が設定されている場合にのみ、その RADIUS サーバに関連します。
refresh	キャッシュ ヒットの場合に、オブジェクト フレッシュネス チェックをサーバに要求します。
reset	TCP RST を発行します。このリセット要求は、Code Red または Nimda のウィルス要求をリセットする場合に便利です。
rewrite	オリジナルの要求を指定の URL に書き換えます。Content Engine はキャッシュ内で書き換えられた URL を検索し、キャッシュ ミスの場合は、書き換えられた URL を取り出して、そのオブジェクトを透過的にクライアントに送信します。パフォーマンスに影響を与える可能性があるため、**rewrite** ではなく、**redirect** ルールを使用することを推奨します。 URL の書き換えによって、URL のドメイン名が変わる可能性があります。その場合、要求の送信先の新しいサーバの送信先 IP アドレスを見付けるために DNS ルックアップが必要になります。WCCP リダイレクト パケットから得られた元の IP アドレスは使用できません。
selective-cache	HTTP が許可したオブジェクトをキャッシングします。
use-dns-server	指定の DNS サーバを使用します。
use-icap-service	指定のパターン リストで指定の ICAP サービスを使用します。
use-proxy	キャッシュ ミスの場合に、指定のアップストリーム プロキシを使用します。アップストリーム プロキシの IP アドレス（またはドメイン名）とポート番号を指定します。このアクションの使用方法の例については、「use-proxy アクションの例」（p.13-13）を参照してください。
use-server	キャッシュ ミス時に、サーバ形式の HTTP 要求を Content Engine から指定の IP アドレスとポートに送信します。このアクションの使用方法の例については、「use-server アクションの例」（p.13-14）を参照してください。
use-xforward-clt-ip	フィルタリング用として、X-forwarded ヘッダー内の クライアント IP アドレスを使用します。

no-url-filtering アクションの例

ここでは、URL フィルタリング機能を Websense URL フィルタリングでバイパスする **no-url-filtering** アクションの使い方の例を示します。

rule action no-url-filtering コマンドを指定し、それを特定のパターン リスト（パターン リスト 100）と関連付けます。**domain** パターン タイプをパターン リスト 100 に追加して Content Engine を設定し、foo.com のドメインを持つ要求を照合します。この例では Websense URL フィルタリングはすでに設定済みであり、Content Engine で使用可能です。

```
ContentEngine (config)# rule action no-url-filtering pattern-list 100
ContentEngine (config)# rule pattern-list 100 domain .*foo.com
ContentEngine (config)# rule enable
```

（注）　**no-url-filtering** アクションは、src-ip、dst-ip、dst-port、domain、group-name、groupname-regex、header-field、url-regex、および username のルール パターンをサポートしています。パターンは、グループ タイプ パターン（たとえば、**rule pattern-list 1 group-type and**）を使用して AND または OR 演算することが可能です。デフォルトは OR です。

Content Engine が foo.com のドメインを持つ HTTP または HTTPS 要求を受信すると、**rule action no-url-filtering** ルールが照合されます。したがって、**debug http proxy** コマンドの次の出力の一部に示されるように、Content Engine は この特定の要求に対する URL フィルタリングのバイパスを行います。

```
Oct 28 12:25:12 Content Engine 3: Rule action no-url-filtering match
- Bypassing urlfiltering
```

rule action no-url-filtering ルールが照合され、Websense URL フィルタリングの代わりに SmartFilter URL フィルタリングが使用されている場合、**debug http proxy** コマンドの出力は次のようになります。

```
Oct 28 12:25:12 Content Engine 3: Rule action no-url-filtering match
- Bypassing SmartFilter processing
```

Content Engine が foo.com 以外の Web サイトの HTTP または HTTPS 要求（www.abc.com をドメインとして持つ要求）を受信すると、**rule action no-url-filtering** ルールが照合されません。したがって、**debug http proxy** コマンドの次の出力の一部に示されるように、Content Engine は この特定の要求に対する Websense URL フィルタリングを続行します。

```
Oct 28 12:28:06 Content Engine 3: Rule action no-url-filtering not hit - Proceed with
urlfiltering
```

rule action no-url-filtering ルールが照合されず、Websense URL フィルタリングの代わりに SmartFilter URL フィルタリングが使用されている場合、**debug http proxy** コマンドの出力は次のようになります。

```
Oct 28 12:25:12 Content Engine 3: Rule action no-url-filtering not hit- Proceed with
SmartFilter processing
```

no-url-filtering アクションについての統計情報を表示するには、**show statistics rule http action no-url-filtering** EXEC コマンドを入力します。

no-url-filtering アクションについての情報を表示するには、**show run** および **show rule all** の各 EXEC コマンドを入力します。

use-proxy アクションの例

次に示すのは、**use-proxy** アクションの標準的な使用例です。

rule action use-proxy *proxy* **pattern-list** *number* グローバル コンフィギュレーション コマンドは、特定のパターン リストに 1 つのプロキシのみを設定する場合に使用できます。同一のパターン リスト（たとえば、パターン リスト 1）に 2 番めのプロキシを設定しようとすると、ルール エントリがすでに存在することを示すエラー メッセージを受信します。

```
ContentEngine12 (config)# rule action use-proxy 10.16.0.0 8080 pattern-list 1
ContentEngine12 (config)# rule action use-proxy 10.77.157.42 8080 pattern-list 1
Rule entry is duplicate
Rule use-proxy exists with IP:10.16.0.0.Please remove it and reconfigure
When rule action use-proxy is configured with "failover" qualifier, first it would try
to connect to proxy configured in use-proxy, if it fails, it would fall back to
outgoing proxy, configured through the cli "http proxy outgoing host <host> <port>"
```

フェールオーバーを適用せずに use-proxy 機能が設定されている場合（たとえば、**rule action use-proxy 10.16.0.0 8080 pattern-list 1** コマンドを入力した場合）、Content Engine は要求を use-proxy（10.16.0.0 の IP アドレスを持つサーバ）に送信します。use-proxy から応答を取得しない場合、Content Engine は HTTP 発信プロキシにフェールオーバーせずにクライアントにエラー メッセージを送信します。

フェールオーバーを適用して use-proxy 機能が設定されている場合（たとえば、**rule action use-proxy 10.16.0.0 8080 failover pattern-list 1** コマンドを入力した場合）、Content Engine は要求を use-proxy（たとえば、10.16.0.0 の IP アドレスを持つサーバなど）に送信します。use-proxy から応答を取得しない場合、Content Engine は、指定された HTTP 発信プロキシ（たとえば、**http proxy outgoing host 10.77.157.42 8080 primary** グローバル コンフィギュレーション コマンドを使用してプライマリ発信ホストとして指定されているサーバ）にフェールオーバーします。次に、フェールオーバーを適用して use-proxy 機能が設定されるコンフィギュレーション例を示します。

```
ContentEngine 12# show run
Sep  1 06:42:32 ContentEngine 12-admin-shell:%CE-PARSER-6-350232:CLI_LOG
shell_parser_log:sh run
! ACNS version 5.4.0
!
!
hostname ContentEngine 12
!
http client-no-cache-request ignore
http proxy incoming 8080
http proxy outgoing host 10.77.157.42 8080 primary
!
ftp-over-http proxy incoming 8080
!
!
(テキスト出力は省略)
!
rule enable
rule action use-proxy 10.16.0.0 8080 failover pattern-list 1
rule pattern-list 1 domain yahoo.com
!
(テキスト出力は省略)
```

use-server アクションの例

次に示すのは、**use-server** アクションの標準的な使用例です。指定の基準に一致する HTTP 要求に対して、Content Engine は、オリジン サーバにアクセスする必要がある場合（たとえば、キャッシュ ミスが発生した場合）に、要求されたオブジェクトを検索するために要求に示されたサーバにアクセスするのではなく、ルールに指定されている別の送信先サーバを使用します。この機能は主に、オンデマンド要求で使用され、通常はリバース プロキシの配置で使用されます。

この例では、Content Engine が www.abcbigcorp.com のリバース プロキシとなり、インターネットの残りがプロキシになります。この企業の Web サイト（www.abcbigcorp.com）の IP アドレスは、実際には Content Engine の IP アドレスであり、その企業の Web サイト サーバの IP アドレスではありません。Content Engine が要求 http://www.abcbigcorp.com/main.html を受け取ると、通常の処理として www.abcbigcorp.com の IP アドレスが取得され、その IP アドレスに要求が送信されます。ただし、このケースでは、www.abcbigcorp.com の IP アドレス が Content Engine の IP アドレスであるため、管理者は Content Engine が要求を自身に送信するのを防ぐ必要があります。

そのため、CE1 の管理者は、これらの要求（たとえば、キャッシュ ミス）を www.abcbigcorp.com の Web サーバに送信することを CE1 に指示する次のルールを設定できます。

```
CE1(config)# rule use-server 1.2.3.4 80 domain www.abcbigcorp.com
```

1.2.3.4 は、www.abcbigcorp.com の Web サーバの IP アドレスです。

（注）　このルールは HTTP 処理にのみ適用されます。

no-proxy アクションの例

no-proxy アクションは、Content Engine の管理者が Content Engine 用の発信プロキシ サーバをすでに設定しているときに適用できます。**no-proxy** アクションは、基準に一致した要求に対して、オリジン サーバとの接続が必要になった場合（たとえば、キャッシュ ミスが原因で）に、Content Engine 側で指定のプロキシ サーバを使用してオリジン サーバとの接続を設定してはならないことを意味します。このルールは、企業がすべてのインターネット コンテンツをキャッシングする Content Engine（CE1）をインターネット ゲートウェイに設置していて、各支店に Content Engine（CE2、CE3、CE4）を設置している場合に役立ちます。このケースでは、管理者は、発信プロキシ サーバとして CE4 を使用するように支店の CE2、CE3、CE4 を設定できますが、社内のコンテンツに対する要求に **no-proxy** ルールを設定できます。CE2、CE3、CE4 がクライアント要求を受信し、要求されたコンテンツがローカル キャッシュにまだ保存されていないときには、Content Engine は要求を次のように処理します。

- インターネット コンテンツに対するクライアント要求の場合、CE2、CE3、CE4 は、オリジン サーバに直接にアクセスするのではなく、インターネット ゲートウェイにある CE1 を使用する必要があります。
- 社内のコンテンツに対するクライアント要求の場合は、CE2、CE3、CE4 は、CE1 にアクセスするのではなく、オリジン サーバに直接に接続する必要があります。

rule action グローバル コンフィギュレーション コマンドの使用方法についての詳細は、「既存パターン リストへのアクションの関連付け」（p.13-27）を参照してください。サポートされているアクションとパターンの組み合わせのリストについては、表 13-5 と表 13-6 を参照してください。プロトコルごとにサポートされているルール アクションのリストについては、表 13-2 を参照してください。

サポートされているアクションとパラメータの組み合わせ

アクションによってはパターンが意味を持たない場合があるため、要求の照合に関して、すべてのアクションがすべてのパターンをサポートするとは限りません。サポートされているアクションとパターンの組み合わせのリストについては、表 13-5 と表 13-6 を参照してください。

アスタリスク「*」は、ACNS 5.2.1 ソフトウェアおよびそれ以降のリリースでアクションとパラメータの特定の組み合わせがサポートされることを示しています。

表 13-5　スタンドアロン Content Engine でサポートされているアクションとパターンの組み合わせ — パート 1

アクション	パターン													
	domain	dst-ip	dst-port	header-field-referrer	header-field-req-line	header-field-user-agent	header-field-sub-referrer	header-field-sub-req-line	header-field-sub-user-agent	mime-type	src-ip	url-regex	url-regsub	groupname、username、groupname regex
allow	*	*	*	*	*	*					*	*		*
append-username header	*	*	*								*	*		*
block	*	*	*	*	*	*					*	*		*
cache-cookie	*	*								*		*		*
cache-non-cacheable	*	*								*		*		
cache-only	*	*								*		*		
dscp client	*	*	*							*	*	*		
dscp server	*	*	*								*	*		
freshness-factor	*	*	*							*	*	*		
insert-no-cache	*	*										*		
no-auth	*	*	*								*	*		
no-cache	*	*	*							*	*	*		

表 13-6　スタンドアロン Content Engine でサポートされているアクションとパターンの組み合わせ — パート 2

アクション	パターン													
	domain	dst-ip	dst-port	header-field-referrer	header-field-req-line	header-field-user-agent	header-field-sub-referrer	header-field-sub-req-line	header-field-sub-user-agent	mime-type	src-ip	url-regex	url-regsub	groupname、username、groupname regex
no-persistent-connection	*		*	*	*	*					*	*		
no-proxy	*	*	*								*	*		
no-url-filtering	*	*	*	*	*	*					*	*		*
redirect				*	*	*						*		
refresh	*	*	*							*	*	*		
reset	*	*	*	*	*	*					*	*		
rewrite							*	*	*				*	
selective-cache	*	*	*							*	*	*		
use-dns-server	*	*												
use-icap-service	*	*	*	*	*	*					*	*		
use-proxy	*	*	*								*	*		
use-server	*	*	*								*	*		
use-xforward clt-ip	*		*	*	*	*					*	*		

Rules Template 処理の考慮事項

スタンドアロン Content Engine に複数のルールを設定しているときには、ルールがある特定の順序で実行されます。

ACNS 5.x でのルールの処理順序を理解するには、ルール処理の次の部分を理解する必要があります。

- 定義済みのアクションの実行順序は事前に定められています。つまり、同じアクションに関係するルールのグループは、常に別のアクションに関連する別のルール グループの前またはあとに実行されます。この順序は事前に定義されているため、ルールの入力順序の影響を受けません。
- 同じアクションのルールの中で、ルール パターンに事前に定義されている順序があります。つまり、同じアクション内では、同じパターンに関連するルール グループは、常に別のパターンの別のルール グループの前またはあとに実行されます。この場合も、この順序は事前に定義されているため、ルールの入力順序の影響を受けません。
- 同じパターン タイプを持つパターン リストに設定されている、同じアクションのすべてのルールでは、各パターンが設定順序によって照合されます。

たとえば、次のようなルールを指定すると、Content Engine はまず .asf で終了する URL と照合を試みます。

```
ContentEngine(config)# rule action block pattern-list 1
ContentEngine(config)# rule pattern-list 1 url-regex .*\.asf$
ContentEngine(config)# rule pattern-list 1 url-regex .*\.avi$
ContentEngine(config)# rule pattern-list 1 url-regex .*\.wav$
ContentEngine(config)# rule pattern-list 1 url-regex .*\.wmv$
```

次に、ContentEngine .avi で終了する URL と照合を試みます。.avi で終了する URL と照合すると Content Engine は、.wav で終了する URL、.wmv で終了する URL との照合を試みます。

- ACNS　ソフトウェアの最新リリースでルール　アクションとパターンが追加されると（たとえば、ACNS 5.2.1 ソフトウェアでは、**groupname**、**username**、**groupname-regex** のパターンと **cache-cookie** アクションが追加されています）、既存のアクションとパターンの処理順序が変わりません。

ルール パターンの実行順序については、「ルール パターンの実行順序」（p.13-19）を参照してください。この順序は、Content Engine GUI または CLI コマンドでルールを入力する順序の影響を受けることはありません。

ヒント　ACNS 5.x ソフトウェアでは、スタンドアロン Content Engine で **show rule** EXEC コマンドを入力すると、ルールがランダムに表示されます。ただし、**show statistics rule** EXEC コマンドを入力した場合には、ルール アクションの実行順序でルールが表示されます。そのため、このコマンドを使用して、Content Engine がユーザ定義のルールを処理する方法を調べることができます。

ホスト名から IP アドレスへの変換に対するルール コマンド

ACNS 5.3.1 ソフトウェアおよびそれ以前のリリースでは、**use_proxy** ルール アクションと **use-proxy** ルール アクションの **failover** オプションが、CLI コンフィギュレーションの際にホスト名から IP アドレスへの変換を実行します。指定されたホスト名の IP アドレスが変わると、サービス ルールが機能しなくなります。

ACNS 5.3.3 ソフトウェアおよびそれ以降のリリースでは、**rule dns-resolve each-request** グローバル コンフィギュレーション コマンドを使用できます。この CLI コマンドが有効になっている場合、Content Engine のキャッシュ プロセスが要求を処理するたびにホスト名を変換し、**use-proxy** ルール アクションおよび **use-proxy** ルール アクションの **failover** オプションのパターンを照合します。

ACNS 5.3.3 ソフトウェアおよびそれ以降のリリースで **rule dns-resolve each-request** CLI コマンドが無効になっている場合、キャッシュ プロセスは最初に変換された（つまり、CLI コンフィギュレーションの際に実行された）IP を使用して、**use-proxy** ルール アクションおよび **use-proxy** ルール アクションの **failover** オプションを処理します。たとえば、次は ACNS 5.3.3 ソフトウェアおよびそれ以降のリリースの設定における yahoo と abc Web サイトの CLI コマンド構文の例です。

```
ContentEngine(config)# rule action use-proxy www.yahoo.com 8080 failover pattern-list
10
ContentEngine(config)# rule action use-proxy www.abc.com 8090 pattern-list 20
```

以下は ACNS 5.3.1 ソフトウェアおよびそれ以前のリリースの設定における yahoo と abc Web サイトの CLI コマンド構文の例です。

```
ContentEngine(config)# rule action use-proxy 66.94.230.42 8080 failover pattern-list
10
ContentEngine(config)# rule action use-proxy 199.181.132.250 8090 pattern-list 20
```

ルール アクションの実行順序

ACNS 5.2.3 ソフトウェアおよびそれ以降のリリースでは、ルール アクションの実行順序は次のようになります。

1. Redirect-url-for-cdn（このアクションは、Content Distribution Manager に登録された Content Engine にのみ適用可能です。スタンドアロン Content Engine には適用されません）。
2. No-auth（RADIUS、LDAP、または NTLM を使用した認証の前）
3. Reset
4. Block
5. Redirect（キャッシュ ルックアップの前）
6. Rewrite（キャッシュ ルックアップの前）
7. No-url-filtering
8. Refresh（キャッシュ ヒットの場合はキャッシュ ルックアップの後）
9. Freshness-factor（キャッシュ ヒットの場合はキャッシュ ルックアップの後）
10. Use-server
11. No-proxy
12. Use-proxy
13. Use-dns-server
14. ToS/DSCP server（サーバとの接続時の ToS ビット）
15. ToS/DSCP client（サーバがクライアントに応答を送信するときに使用する接続時の ToS ビット）
16. DSCP client cache-miss
17. DSCP client cache-hit
18. Insert-no-cache
19. No-cache
20. Cache（サーバから応答が受信されたとき）
21. Selective-cache（サーバから応答が受信されたとき）
22. Append-username-header
23. Use-icap-service
24. Use-xforward-clt-ip
25. No-persistent-connection
26. Cache-cookie
27. No-selective-cache
28. Allow

reset、**block**、**rewrite**、**redirect** のルール アクションは、次に示す別のパターンをサポートします。つまり、**request-line**、**referer**、**user-agent** の正規表現です。**request-line** 正規表現は、要求の最初のラインを照合します。**user-agent** 正規表現は、要求の User-Agent ヘッダー値を照合します。**referer** 正規表現は、要求 referer ヘッダー値を照合します。

次の例では、要求内の internal.domain.com という文字列をサーバ名 dummy に置き換えるように Content Engine を設定します。

```
ContentEngine(config)# rule rewrite header-field referer internal.domain.com dummy
```

次の例では、置換パターンとして空の文字列を指定した場合、referer ヘッダーが除去されます。このルールは、ABCBigCorp の社内サーバを示す referer ヘッダーを含んだ要求に対して、外部の Web サーバから社内サーバの名前を確認できないように referer フィールドを除去することを意味します。これにより、ネットワーク セキュリティを強化できます。

```
ContentEngine(config)# rule rewrite header-field referer internal.abcbigcorp.com ""
```

referer ヘッダーの除去は、**user-agent** パターンでも発生します。

（注）　**rule action no-proxy**、**rule action use-proxy** *hostname port-number* **failover**、および **rule action use-proxy** のコマンドは、**https proxy outgoing**、**http proxy outgoing**、および **ftp proxy outgoing** のグローバル コンフィギュレーション コマンドより優先されます。

use-server ルール、**no-proxy** ルール、**use-proxy** ルールの中では、**use-server** ルールが最初にチェックされます。いずれのルールも一致しなかった場合には、**no-proxy** ルールと **use-proxy** ルールが続けて実行されます（**no-proxy** ルールが一致した場合には、**use-proxy** ルールはチェックされません）。ルールが Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）を使用して設定され、要求が透過モードでドメイン名の一部として受信された場合には、FQDN が要求 URL 内にないため、このルールは実行されません。透過モードでは、要求が特定のドメイン（ドメイン ルールが設定されているドメイン）を送信先とし、Host ヘッダーを含んでいない場合には、ルール パターンの照合に失敗します。Rules Template の設定は、**ip dscp** コマンドより優先されます。**url-filter** コマンドは、**rule** コマンドより優先されます。したがって、**rule no-block** コマンドが実行されるのは、**url-filter** コマンドが要求をブロックしなかった場合のみです。

ルール パターンの実行順序

実行順序は次のとおりです。

1. **Header-field**
2. **Header-field-sub**
3. その他のパターン：**url-regsub**、**dst-port**、**src-ip**、**url-regex**、**domain**、**dst-ip**、**mime-type**。

これらのパターンで実行されるアクションに対しては実行順序は定められていません。

（注）　MIME タイプは応答内にのみ存在するため、**freshness-factor**、**refresh**、**no-cache**、**selective-cache** の各アクションのみが MIME タイプのルールに適用されます。

たとえば、次の一連のアクションでは、**pattern-list 2 header-field** パターンが最初に実行され、次に **pattern-list 1 domain** パターンが実行されます。この順序が適用されるのは、ヘッダー情報が使用可能になったあとにのみルール アクションが実行されるからです。

```
ContentEngine(config)# rule action block pattern-list 1
ContentEngine(config)# rule action block pattern-list 2
ContentEngine(config)# rule pattern-list 1 domain roti
ContentEngine(config)# rule pattern-list 2 header-field user-agent browser
```

次の例に示すパターンの AND 演算では、パターン エントリに基づいた実行順序は適用されません。

```
ContentEngine(config)# rule pattern-list 1 group-type and
ContentEngine(config)# rule action block pattern 3
ContentEngine(config)# rule pattern-list 3 dst-port 80
ContentEngine(config)# rule pattern-list 3 header-field user-agent browser
```

上記の例では、送信先ポート（**dst-port**）が最初にチェックされ、次に header-field がチェックされます。

ある特定のルールに通常のパターン タイプのヘッダー フィールドが含まれている場合は、特別な実行順序はありません。

残りのパターン リストに一致するルールの検索は、一致していることがすでにわかっている場合は実行されません。たとえば、**rule action block** アクション コマンドの一致が **URL-regex** 要求に見つかった場合、残りのパターンの **domain**、**dst-ip**、**MIME-type** は検索されません。すべてのパターンが **rewrite** と **redirect** のアクションに適用できるわけではありません。

デフォルトでは、ルールは相互に OR 演算されます。複数のルールすべてが 1 つの要求に一致する場合があります。その場合、すべてのアクションが実行されますが、「ルール アクションの実行順序」（p.13-18）に定義されているルール アクションの実行順序に基づいて、競合するアクション間で優先順位が設定されます。**rule action** グローバル コンフィギュレーション コマンドには、**rule pattern-list** グローバル コンフィギュレーション コマンドで設定する複数のパターンを含めることができます。

特定のルールを適用対象外にできます。たとえば、**domain** パターンを含むルールを適用対象外にするには、ブラウザの中にドメイン名ではなく Web サーバの IP アドレスを入力します。ルールが予期せぬ影響を受ける場合があります。たとえば、ibm として指定された **domain** パターンを持つルールは www.ibm.com と一致することを意図したものですが、www.ribman.com のようなドメイン名とも一致する場合があります。

（注）　**src-ip** ルールは、Content Engine が別のプロキシまたは別の Content Engine から受信する要求に対して、意図した通りに適用されない場合があります。これは、元のクライアント IP アドレスが X-Forwarded-For ヘッダー内に含まれているためです。つまり、元の要求の送信元 IP アドレスが、オリジン サーバに向かう途中で、送信 Content Engine の IP アドレスに透過的に置き換えられています。

1 つのルール パターンが一致すると、残りのパターンは検索されません。サーバがすでにオブジェクトにキャッシュ不能（noncacheable）のマークを付けている場合、サーバ側では、このオブジェクトがキャッシングされないことがわかっているため、**no-cache** ルールはチェックされません。**no-cache** ルールのチェックはすべて、キャッシング可能な要求に対してのみ実行されます。

Rules Template の設定

スタンドアロン Content Engine でルールを設定するときには、次の点に留意することが重要です。

- アクションの数は無制限です。
- パターン リストの最大数は 512 です。
- アクションあたりのパターンの最大数は 128 です。
- パターン タイプあたりにパターン リストに設定できるパターンの最大数は 128 です。
- Content Engine CLI からルールの正規表現に疑問符（?）を入力するには、エスケープ文字を使用し、そのあとに疑問符（?）を入力します。これにより、状況依存ヘルプが CLI に表示されなくなります。
- Rules Template の設定は、**ip dscp** コマンドより優先されます。**url-filter** コマンドは、**rule** コマンドより優先されます。そのため、**rule no-block** コマンドが実行されるのは、**url-filter** コマンドが要求をブロックしなかった場合のみです。
- グループ名とユーザ名に基づいたルールによる許可は、HTTP 要求認証とグループに基づいた Access Control List（ACL; アクセス制御リスト）許可の発生後にのみ発生します。Rules Template 内の設定と ACL が一致した場合は、ACL の方が優先されます。
- オブジェクトが認証されている場合には、**rule action cache-non-cacheable** コマンドはそのオブジェクトをキャッシングできません。つまり、認証済みのオブジェクトに関しては、オリジン サーバの中には Last-Modified、または ETag エンティティ ヘッダーを送信しないものがあります。そのため、これらの許可されたオブジェクトは Content Engine 側で再検証されません。認証されたオブジェクトは、オリジン サーバからのみ使用可能となります。認証されたオブジェクトに対して、サーバが Last-Modified および ETag ヘッダーを送信すると、それらのオブジェクトが正しく再検証され、キャッシュから取り出されます。
- 次のような状況では、**no-auth** ルールを適用すると、複数の認証ウィンドウが表示されます。
  - **no-auth** ルールを使用して、メイン ウィンドウ（たとえば、index.htm）がプロキシ認証から除外される場合
  - ユーザ入力情報が Content Engine 認証キャッシュにまだ入っている場合
  - index.htm ウィンドウに別のドメインに属するオブジェクトが含まれている場合

  複数の認証ウィンドウが表示されないようにするには、グローバル コンフィギュレーション モードで **http avoid-multiple-auth-prompts** 非表示コマンドを入力します。次の例に示すように、**show http avoid-multiple-auth-prompts** EXEC 非表示コマンドを使用して設定をチェックしてください。

```
ContentEngine# show http avoid-multiple-auth-prompts
Avoiding multiple authentication prompts due to no-auth rules is enabled
```

  この例で示すコマンドは、この特定の状況にのみ適用されため、非表示になります。

（注）　Rules Template は、CLI または Content Engine GUI から設定できます（図 13-1 を参照）。

Content Engine のルール処理の設定方法については、以下の各項を参照してください。

- ルール処理の有効化（p.13-22）
- スタンドアロン Content Engine のためのルールの設定（p.13-22）

ルール処理の有効化

デフォルトでは、Content Engine 上のルール処理は無効になっています。スタンドアロン Content Engine でのルール処理を有効にするには、**rule** グローバル コンフィギュレーション コマンドを次のように使用します。

```
ContentEngine(config)# rule enable
```

スタンドアロン Content Engine のためのルールの設定

スタンドアロン Content Engine で HTTP、HTTPS、MMS、RTSP のトラフィックをフィルタリングするルールを設定するには、**rule** グローバル コンフィギュレーション コマンドを使用します。

rule {**action** *action-type* **pattern-list** *list_num* [**protocol** {**all** | *protocol-type*}] | **enable** | **pattern-list** *list_num pattern-type*}

表 13-7 は、**rule** コマンドのパラメータの説明です。ルール コマンド パラメータがアクション タイプやパターン タイプの場合は、表 13-7 に明記してあります。

（注）　ほとんどのアクションがパラメータをとりません。例外は、**use-server**、**freshness-factor**、**use-proxy** のアクションです。

表 13-7　rule CLI コマンドのパラメータ

パラメータ	アクションまたはパターンタイプ	説明
action		着信要求が指定のパターンに一致した場合に、このルールで実行されるアクションを設定します。
action-type		着信要求が指定のパターンに一致した場合に実行するアクションのタイプ（たとえば、**allow**）を指定します。
allow	アクションタイプ	要求を許可します。
pattern-list		パターン リストを設定します。
list_num		パターン リスト番号（1 ～ 512）
protocol		規則が照合されるプロトコル
all		ルールとアクションに適用可能なすべてのプロトコルを照合します。
protocol-type		ルールの照合対象のプロトコル タイプ（着信トラフィックのタイプ）を指定します。
enable		ルール処理を有効にします。
http		着信 HTTP トラフィックとこのルールを照合します。
https		ルールと着信の HTTPS トラフィックを照合します。
rtsp		着信 RTSP トラフィックとこのルールを照合します。
append-username-header	アクションタイプ	要求ヘッダーにユーザ名を追加します。
block	アクションタイプ	要求をブロックします。
cache-cookie	アクションタイプ	クッキーを含んだ要求をキャッシングします。
groupname		文字列とグループ名を照合します（たとえば、Engineering）。このグループ名に基づいたルール ポリシーは、LDAP や NTLM を通して認証されたユーザに対する要求認証にのみ適用できます。

表 13-7　rule CLI コマンドのパラメータ（続き）

パラメータ	アクションまたはパターンタイプ	説明
groupname		グループ名の文字列
username		文字列と指定のユーザ名を照合します。ユーザ名に基づいたルール ポリシーは、認証用のユーザ名を含む、サポートされているすべての要求認証方式（たとえば、LDAP、NTLM、RADIUS、TACACS+）に適用できます。
username		ユーザ名の文字列（たとえば、jdoe8）
groupname-regex		正規表現とグループ名を照合します。
groupname-regex		グループ名と照合する正規表現
cache-non-cacheable	アクションタイプ	HTTP 応答ヘッダーを上書きして、オブジェクトをキャッシングします。
ttl		オブジェクトの TTL 値
days		TTL 単位（日数）
days		TTL 値（日数）（1 ～ 1825）
hours		TTL 単位（時間数）
hours		TTL 値（時間数）（1 ～ 43800）
minutes		TTL 単位（分数）
minutes		TTL 値（分数）（1 ～ 2628000）
seconds		TTL 単位（秒数）
seconds		TTL 値（秒数）（1 ～ 157680000）
cache-only	アクションタイプ	オブジェクトのみをキャッシングします。
dscp client	アクションタイプ	クライアントの IP の ToS/DSCP フィールド応答を設定します。
cache-hit		キャッシュ ヒット時に応答をクライアントに送信します。
match-server		サーバの元の ToS または DSCP 値を使用します。
set-dscp		DSCP 値を設定します。DSCP 値のリストについては、表 11-1 を参照してください。
set-tos		ToS 値を設定します。ToS 値のリストについては、表11-2 を参照してください。
cache-miss		キャッシュ ミス発生時に応答をクライアントに送信します。
dscp server	アクションタイプ	発信応答用の ToS または DSCP サービス コード ポイントを設定します。
match-client		クライアントの元の ToS または DSCP 値を使用します。
enable		ルール処理を有効にします。
freshness-factor	アクションタイプ	ヒューリスティック モディファイアをキャッシングします。
exp_time		オブジェクトの経過時間を有効期限のパーセント値（0 ～ 100）で設定します。
insert-no-cache	アクションタイプ	no-cache ヘッダーを応答に挿入します。
no-auth		認証を行いません。
no-cache	アクションタイプ	オブジェクトをキャッシングしません。
no-persistent connection		固定接続を使用できないようにします。
all		クライアントまたはサーバとの固定接続を使用できないようにします。
client-only		クライアントとの固定接続を使用できないようにします。
server-only		サーバとの固定接続を使用できないようにします。
no-proxy	アクションタイプ	アップストリーム プロキシを使用しません。このルール アクションの使用方法の例については、「no-proxy アクションの例」（p.13-14）を参照してください。

表 13-7　rule CLI コマンドのパラメータ（続き）

パラメータ	アクションまたはパターンタイプ	説明
no-url-filtering	アクションタイプ	特定の HTTP および HTTPS 要求の URL フィルタリングをバイパスします。この機能は、ローカル リスト URL フィルタリング（優良サイトおよび悪質サイトのリスト）だけでなく、Websense、SmartFilter、または ACNS 5.2.3 ソフトウェアおよびそれ以降のリリースの N2H2 URL フィルタリングでサポートされています。このルール アクションの使用方法の例については、「no-url-filtering アクションの例」（p.13-11）を参照してください。
redirect	アクションタイプ	書き換えられた URL に要求をリダイレクトします。
url		URL をリダイレクトします。
redirect-url-for-cdn	アクションタイプ	ACNS ネットワーク コンテンツに対する要求を代替の URL にリダイレクトします。Content Distribution Manager に登録された Content Engine にのみ適用可能です。スタンドアロン Content Engine には適用されません。
refresh	アクションタイプ	オブジェクトを Web サーバで再検証します。
reset	アクションタイプ	TCP RST を発行します。
rewrite	アクションタイプ	オリジナルの要求を指定の URL に書き換えて、その URL をキャッシュ ミス時にフェッチします。
selective-cache	アクションタイプ	HTTP が許可したオブジェクトをキャッシングします。
use-dns-server	アクションタイプ	特定の DNS サーバを使用します。
hostname		DNS サーバのホスト名
ip-address		DNS サーバの IP アドレス
use-icap-service	アクションタイプ	特定の ICAP サーバを使用します。
icap-service-name	パターンタイプ	ICAP サービス名を使用します。
service name		ICAP サービスの名前
use-proxy	アクションタイプ	特定のアップストリーム プロキシを使用します。このルール アクションの使用方法の例については、「use-proxy アクションの例」（p.13-13）を参照してください。
hostname		特定のプロキシのホスト名
ip-address		特定のプロキシの IP アドレス
port		特定のプロキシのポート番号（1 ～ 65535）
use-server	アクションタイプ	特定のサーバを使用します。このルール アクションの使用方法の例については、「use-server アクションの例」（p.13-14）を参照してください。
use-xforward-clt-ip		フィルタリング用として、x-forwarded ヘッダー内の クライアント IP アドレスを使用します。
hostname		特定のサーバのホスト名
ip-address		特定のサーバの IP アドレス
port		特定のサーバのポート番号（1 ～ 65535）
domain	パターンタイプ	ドメイン名に一致する正規表現
dn_regexp		ドメイン名と照合する正規表現
dst-ip	パターンタイプ	要求の送信先 IP アドレス
d_ipaddress		要求の送信先 IP アドレス
d_subnet		送信先 IP サブネット マスク
dst-port	パターンタイプ	送信先ポート番号
port		宛先ポート番号（1 ～ 65535）

表 13-7　rule CLI コマンドのパラメータ（続き）

パラメータ	アクションまたはパターンタイプ	説明
group-type	パターンタイプ	パターン リストを AND と OR のどちらのタイプにするかを指定します。デフォルトは OR です。
and		パターン リストに AND パターンを指定します。
あるいは		パターン リストに OR パターンを指定します。
header-field	パターンタイプ	要求ヘッダー フィールド パターン
referer		参照元（referer）要求ヘッダー
ref_regexp		参照元要求ヘッダーと照合する正規表現
request-line		要求メソッド ライン
req_regexp		要求メソッド ラインと照合する正規表現
user-agent		ユーザ エージェント要求ヘッダー
ua_regexp		ユーザ エージェント要求ヘッダーと照合する正規表現
header-field-sub	パターンタイプ	要求ヘッダー フィールド パターンと代替置換パターン
referer		参照元（referer）要求ヘッダー
ref_regexp		参照元要求ヘッダーと照合する正規表現
ref_sub		要求ヘッダーの正規表現の置換文字列
request-line		要求メソッド ライン
req_regexp		要求メソッド ラインと照合する正規表現
req_sub		要求メソッド ラインの正規表現の置換文字列
user-agent		ユーザ エージェント（User Agent）要求ヘッダー
ua_regexp		ユーザ エージェント要求ヘッダーと照合する正規表現
ua_sub		ユーザ エージェント要求ヘッダーの正規表現の置換文字列
mime-type	パターンタイプ	Content-Type（コンテンツ タイプ）HTTP ヘッダーと照合する MIME タイプ
mt_regexp		コンテンツ タイプと照合する正規表現
src-ip	パターンタイプ	要求の送信元 IP アドレスを使用するように設定します。
s_ipaddress		要求の送信元 IP アドレス
s_subnet		送信元 IP サブネット マスク
url-regex	パターンタイプ	URL のサブストリングと照合する正規表現を使用するように設定します。
url_regexp		URL 文字列と照合する正規表現
url-regsub	パターンタイプ	URL および置換パターンと照合する正規表現を設定します。
url_regexp		URL 文字列と照合する正規表現
url_sub		URL 文字列置換パターン

ACNS 5.x ソフトウェアでは、一致した URL、ファイル タイプ、ドメイン、送信先 IP アドレス、送信元 IP アドレス、または送信先ポートに基づいて、IP パケット内に ToS 値または DSCP 値を設定できます。個々の ToS 値または DSCP 値を次の項目に設定できます。

- Content Engine からサーバへの要求
- キャッシュ ヒット時のクライアントへの応答
- キャッシュ ミス時のクライアントへの応答

ToS または DSCP の設定は、**src-ip** *s_ipaddress s_subnet*、**dst-ip** *d_ipaddress d_subnet*、**dst-port** *port*、**domain** *LINE*、**url-regex** *LINE*、または **mime-type** *LINE* のオプションと一致するポリシーのいずれかに基づいて行います。**ip dscp** コマンドを使用して、ToS または DSCP のグローバル設定値を指定することもできます。

パターン リストの設定

スタンドアロン Content Engine でパターン リストを作成するには、**rule pattern-list** グローバル コンフィギュレーション コマンドを次のように使用します。

```
ContentEngine(config)# rule pattern-list list_num
```

ここで各パラメータの意味は、次のとおりです。

list_num はパターン リスト番号（1 ～ 152）です。

たとえば、次のようにしてパターン リスト 10 を作成します。

```
ContentEngine(config)# rule pattern-list 10
```

既存のパターン リストへのパターンの追加

スタンドアロン Content Engine の既存のパターン リストに新しいパターンを追加する手順は、次のとおりです。

ステップ 1　既存のパターン リストへのパターンの追加

```
ContentEngine(config)# rule pattern-list list_num pattern type pattern value
```

次の例は、パターン リスト 10 にパターンを追加する方法を示しています。このパターンでは、**dst-ip**（送信先 IP アドレス）のパターン タイプを使用して、送信先 IP アドレス 172.16.25.25 に定義するアクションが実行されます。

```
ContentEngine(config)# rule pattern-list 10 dst-ip 172.16.25.25 255.255.255.0
```

（注）　サポートされているパターン タイプの詳細リストについては、表 13-3 を参照してください。

ステップ 2　指定のパターン リストに新しいパターンが追加されているか確認してください。

次の例は、ステップ 1 で作成したパターンがパターン リスト 10 に追加されているか検証する方法を示しています。

```
ContentEngine# show rule pattern-list 10 all
Rules Template Configuration
----------------------------
Rule Processing Enabled

Pattern-Lists :

rule pattern-list 10 dst-ip 172.16.25.25 255.255.255.0
```

アクションをパターン リストに関連付ける方法については、次の「既存パターン リストへのアクションの関連付け」を参照してください。

既存パターン リストへのアクションの関連付け

スタンドアロン Content Engine の既存のパターン リストにアクションを関連付ける手順は、次のとおりです。

ステップ 1　既存パターン リストにアクションを関連付けます。

```
ContentEngine(config)# rule action action-type pattern-list list_num
protocol {protocol-type | all}
```

アクションを特定のプロトコルまたは複数のプロトコルに適用できます。プロトコルが設定されていない場合には、Content Engine を通過するすべてのトラフィックに対して指定のアクションが実行されます。次の例では、すべてのプロトコルに対して、パターン リスト 10 に **block** アクションが関連付けられます。

```
ContentEngine(config)# rule action block pattern-list 10 protocol all
```

（注）　**block** アクションと同様に、ほとんどのアクションはパラメータをとりません。**rule action** グローバル コンフィギュレーション コマンドのパラメータの詳細については、表 13-7 を参照してください。

ステップ 2　指定のパターン リストに新しいアクションが関連付けられているかを検証します。

```
ContentEngine# show rule action action-type protocol {protocol-type | all}
```

次の例は、**block** アクションがパターン リスト 10 に関連付けられているかを検証する方法を示しています。

```
ContentEngine# show rule action block
Rules Template Configuration
----------------------------
Rule Processing Enabled

Actions :

rule action block pattern-list 10 protocol all
ContentEngine#
```

パターン リストに基づいて実行されるアクションの検証

指定のパターン リストに基づいて特定のアクションが実行されるかを確認するには、アクションの実行後に、ローカルの Rules Template 設定統計情報を表示します。

```
ContentEngine# show statistics rule action action-type
```

次の例では、**rule action block** コマンドを設定して、既存のパターン リストに関連付けます。ここでは、パターンとして yahoo.com ドメインが表示されます。

```
ContentEngine(config)# rule action block pattern-list 10 protocol all
ContentEngine# show statistics rule action block
Rules Template Statistics
-------------------------
Rule hit count = 3   Rule: rule action block pattern-list 10 protocol all
ContentEngine#
```

この例では、統計情報（ルール ヒット カウント）として、yahoo.com への要求が 3 回拒否されたことが示されています。

スタンドアロン Content Engine でのルール設定の例

ここでは、スタンドアロン Content Engine でルールを設定する例を示します。

（注） 以降の例では、特に注記がないかぎり、すべてのアクションとパターンがすべてのプロトコルに適用されることを前提としています。

- .foo.com を含んだドメインを指定するには、**domain** パターン タイプを使用して、このパターンをパターン リスト 12 に追加します。

  ```
  ContentEngine(config)# rule pattern-list 12 domain \.foo.com
  ```

  block アクションをパターン リスト 12 に関連付けて、.foo.com を含んだドメインに対する URL 要求をすべて拒否するように、Content Engine を設定します。

  ```
  ContentEngine(config)# rule action block pattern-list 12
  ```

  同じパターン リスト（パターン リスト 12）に複数のパターンを設定します。いずれかのパターンが着信要求と一致した場合、該当のアクションがとられます。次の例では、**rule action block** グローバル コンフィギュレーション コマンド（アクション）によって、パターン リスト 12 に指定されているすべてのパターンを拒否します。

  ```
  ContentEngine(config)# rule pattern-list 12 domain \.foo.com
  ContentEngine(config)# rule pattern-list 12 dst-ip 172.16.25.25 255.255.255.0
  ContentEngine(config)# rule action block pattern-list 12
  ```

- *cgi-bin* という文字列を含んだ URL 要求をキャッシュしないように、Content Engine を設定します。

  ```
  ContentEngine(config)# rule pattern-list 13 url-regex \.*cgi-bin.*
  ContentEngine(config)# rule action no-cache pattern-list 13
  ```

- rule **action** グローバル コンフィギュレーション コマンドの前に **no** を使用して、ルールを削除します。

  ```
  ContentEngine(config)# no rule use-proxy foo.com 8080 pattern-list 13
  ContentEngine(config)# no rule action block pattern-list 2
  ```

- MIME-type イメージの Content Engine フレッシュネス係数を設定します。

```
ContentEngine(config)# rule pattern-list 13 mime-type image/.*
ContentEngine(config)# rule action freshness-factor 75 pattern-list 13
```

- 送信先 IP アドレス 10.1.1.1 への発信要求のため、Content Engine の ToS 値を最小遅延に設定します。

```
ContentEngine(config)# rule action dscp server set-tos min-delay protocol all
ContentEngine(config)# rule pattern-list 2 dst-ip 10.1.1.1 255.255.255.255
```

- すべての発信要求用のため、Content Engine の ToS 値を最小遅延に設定します。

```
ContentEngine(config)# ip dscp server set-tos min-delay
```

- 同じオブジェクトへの以降のすべてのキャッシュ ヒット応答に対して、（オブジェクトが最初にフェッチされたときに）サーバが当初送信した ToS、または DSCP 値を使用するように、Content Engine を設定します。

```
ContentEngine(config)# ip dscp client cache-hit match-server
ContentEngine(config)# rule action no-cache pattern-list 3 protocol all
ContentEngine(config)# rule pattern-list 3 url-regex \.*cgi-bin.*
ContentEngine(config)# rule pattern-list 4 dst-ip 172.31.120.0 255.255.192.0
```

- new-domain-name に変更された old-domain-name に対する要求を新しいドメイン名にリダイレクトするように、Content Engine を設定します。

```
ContentEngine(config)# rule action redirect http://old-domain-name/
pattern-list 1 protocol http
ContentEngine(config)# rule pattern-list 1 url-regsub
http://old-domain-name/http://new-domain-name/
```

- IETF　サイトからの要求をローカルにミラーリングされるサイトにリダイレクトするように、Content Engine を設定します。

```
ContentEngine(config)# rule action redirect http://www.ietf.org/rfc/(.*)
pattern-list 2 protocol http
```

- 次の例では、要求 URL が http://www.ietf.org/rfc/rfc1111.txt である場合、Content Engine は、その URL を http://wwwin-eng.cisco.com/RFC/RFC/rfc1111.txt として書き換え、Loation ヘッダー内に書き換えられた URL を持つ 302 Temporary Redirect 応答をクライアントに送ります。ブラウザは、書き換えられた URL に対する要求を自動的に開始します。

```
ContentEngine(config)# rule pattern-list 2 url-regsub
http://www.ietf.org/rfc/(.*) http://wwwin-eng.cisco.com/RFC/RFC/\1
```

- linux.org へのすべての要求を、インドにある Content Engine の場所に近いローカル サーバにリダイレクトするように、Content Engine を設定します。

```
ContentEngine(config)# rule action redirect http://linux.org/(.*) pattern-list 3
protocol http
```

- **rule action no-auth** グローバル コンフィギュレーション コマンドは、LDAP、RADIUS、SSH、TACACS+ などの認証と許可の機能をバイパスする特定のログイン要求とコンテンツ要求を許可します。次の例では、送信元 IP アドレス（src-ip）172.16.53.88 からの要求はどれも認証されません。

```
ContentEngine(config)# rule enable
ContentEngine(config)# rule action no-auth pattern-list 1 protocol all
ContentEngine(config)# rule pattern-list 1 src-ip 172.16.53.88 255.255.255.255
```

- 認証と SmartFilter URL フィルタリング用に ACNS 5.x ソフトウェアを設定している場合には、認証をバイパスすることが許可されている要求は、SmartFilter URL フィルタもバイパスします。172.22.73.34 の送信先 IP アドレス（dst-ip）に対するどの要求も認証しないように、Content Engine を設定します。

```
ContentEngine(config)# rule action no-auth pattern-list 2 protocol all
ContentEngine(config)# rule pattern-list 2 dst-ip 172.22.73.34 255.255.255.255
```

- 9090 の送信先 IP ポート（dst-port）に対するどの要求も認証しないように、Content Engine を設定します。

```
ContentEngine(config)# rule action no-auth pattern-list 3 protocol all
ContentEngine(config)# rule pattern-list 3 dst-port 9090
```

- cgi-bin という文字列を含む URL 要求を認証しないように、Content Engine を設定します。

```
ContentEngine(config)# rule action no-auth pattern-list 4 protocol all
ContentEngine(config)# rule pattern-list 4 url-regex .*cgi-bin.*
```

- ドメインとして cisco.com を含むどの要求も認証しないように、Content Engine を設定します。たとえば、roti.cisco.com や badal.cisco.com に対する要求が Content Engine 認証から除外されます。

```
ContentEngine(config)# rule action no-auth pattern-list 5 protocol all
ContentEngine(config)# rule pattern-list 5 domain cisco.com
```

設定済みルールの統計情報の表示

スタンドアロン Content Engine に設定されているルールに関する統計情報を表示するには、**show statistics rule** EXEC コマンドを使用します。ACNS 5.3.1 ソフトウェア リリースでは、**show statistics rule** コマンドのオプションが変更され、RTSP および WMT RTSP ルールの統計情報を表示することが可能になりました。

ACNS 5.2.x ソフトウェアおよびそれ以前のリリースでは、**show statistics rule** コマンドのコマンド オプションは次のとおりでした。

```
ContentEngine# show statistics rule ?
  all   Display statistics of all the Rules
  http  Display statistics of http/https/all Rules
  wmt   Display statistics of wmt Rules
```

ACNS 5.3.1 ソフトウェアおよびそれ以降のリリースでは、**show statistics rule** コマンドのコマンド オプションは次のとおりです。

```
ContentEngine# show statistics rule ?
  all   Display statistics of all the Rules
  http  Display statistics of http/https/wmt-http Rules
  mms   Display statistics of mms Rules
  rtsp  Display statistics of rtsp/wmt-rtsp Rules
```

ACNS 5.3.1 ソフトウェア リリースでは、**wmt** オプションは **mms** および **rtsp** オプションに置き換えられました。

たとえば、**show statistics rule rtsp** コマンドを入力して、RTSP ルール（RealMedia Player からの RTSP 要求のルール［RTSP ルール］および Windows Media 9 Player からの RTSP 要求のルール［WMT-RTSP ルール］）の統計情報を表示します。

ルール ヒット カウント総数を表示するには、**show statistics rule all** EXEC コマンドを入力します。特定の設定済みルールに関する統計情報を表示するには、**show statistics rule http action** *rule action name* を入力します。たとえば、**no-url-filtering** ルール アクションについての統計情報を表示するには、**show statistics rule http action no-url-filtering** EXEC コマンドを入力します。

設定済みルールの統計情報のクリア

スタンドアロン Content Engine に設定されているルールに関する統計情報をクリアするには、**clear statistics rule** EXEC コマンドを使用します。ACNS 5.3.1 ソフトウェアおよびそれ以降のリリースでは、**clear statistics rule** コマンドのコマンド オプションは次のとおりです。

```
ContentEngine# clear statistics rule ?
  action  Clear statistics of all the rules with same action
  all     Clear statistics of all the rules
  rtsp    Clear statistics of rtsp/wmt-rtsp rules
```

たとえば、**clear statistics rule rtsp** コマンドを入力して、設定済み RTSP ルール（RealMedia Player からの RTSP 要求に対して設定されたルール［RTSP ルール］および Windows Media 9 Player からの RTSP 要求に対して設定されたルール［WMT-RTSP ルール］）の統計情報をクリアします。