CHAPTER 11

スタンドアロン Content Engine 上での事前コンテンツ ロードおよび URL フィルタリングの設定

この章では、ACNS 5.4.x 以降のソフトウェア リリースで稼働するスタンドアロン Content Engine でサポートされている事前コンテンツ ロード、およびさまざまなタイプの URL フィルタリングの概要を示します。また、スタンドアロン Content Engine に事前コンテンツ ロードおよび URL フィルタリングを設定する方法を示します。

この章の内容は、次のとおりです。

ACNS 5.2.3 以降のソフトウェア リリースでは、特定の URL のパフォーマンスをモニタするように Content Engine を設定することができます。Content Engine には、各モニタ対象 URL のさまざまな応答特性に関する統計情報が保持されています。詳細は、「スタンドアロン Content Engine でのクリティカル ディスク ドライブのモニタリング」（p.21-18）を参照してください。

（注） 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法については、『*Cisco ACNS Software Command Reference*』Release 5.5 を参照してください。

（スタンドアロン Content Engine ではなく）Content Distribution Manager に登録されている Content Engine に URL フィルタリングを設定する方法については、『*Cisco ACNS Software Configuration Guide for Centrally Managed Deployments*』Release 5.5 を参照してください。

スタンドアロン Content Engine の事前コンテンツ ロードの設定

ここでは、ACNS 5.4.1 以降のソフトウェア リリースを実行しているスタンドアロン Content Engine のコンテンツ事前ロードの概要について説明します。また、スタンドアロン Content Engine の事前ロード機能の設定方法も説明します。

事前ロードされたコンテンツとは、Content Engine の管理者が特定のコンテンツに関するユーザ要求を予測して、そのコンテンツの取得をスケジュールすることにより、取得され、スタンドアロン Content Engine に保存されるコンテンツです。コンテンツの事前ロードを行うには、プライマリ コンテンツを保存するオリジン Web サーバにあるすべてのコンテンツに対するキャッシュ要求を作成するように、スタンドアロン Content Engine を設定します。

事前ロード プロセスの帯域幅制限を指定して、事前ロード プロセス時に指定された帯域幅制限を超えて帯域幅が消費されないようにします。事前ロード プロセス時に、スタンドアロン Content Engine は、コンテンツを求めて Web サイトを数リンク レベル下方までスキャンし、特定のコンテンツを取得し、将来の要求のためにローカルに保存します。Content Engine は、指定された時刻に、Web サイトのいくつかのレベルをスキャンして、そのコンテンツが最新の状態であることを確認し、変更されているコンテンツを更新します。

ACNS 5.x ソフトウェアにより、URL のファイルを読み取り、指定された URL のコンテンツをスタンドアロン Content Engine に事前ロードできます。スタンドアロン Content Engine に事前ロードできるコンテンツは、HTTP URL、FTP-over-HTTP URL、および MMS-over-HTTP URL（WMT ストリーミング メディア ファイル）です。ACNS 5.3.1 以降のソフトウェア リリースでは、Windows Media 9 クライアントおよびサーバの RTSP URL（rtsp://）もサポートされています。この URL リストは、*事前ロード URL リスト ファイル*といいます。

（注）　設定済みのすべての HTTP、FTP-over-HTTP、MMS-over-HTTP、および RTSP パラメータとルールが、事前ロードされるオブジェクトに適用されます。

NTLM で認証されたオブジェクトの事前ロードのサポート

ACNS 5.1.1 以降のソフトウェア リリースでは、NTLM で認証されたオブジェクトの事前ロードがサポートされています。この機能を使用すると、NTLM 認証オブジェクト（NTLM 認証だけを行うサーバに置かれる認証オブジェクト）を Content Engine に事前ロードできます。

（注）　ACNS 5.1.1 以降のソフトウェア リリースでは、スタンドアロン Content Engine に NTLM で認証されたオブジェクトを事前ロードするための NTLMv1 サポートを使用できます。ACNS 5.4.1 ソフトウェア リリースでは、NTLM で認証されたオブジェクトの事前ロードに関する NTLMv2 サポートが追加されています。ACNS 5.4.1 ソフトウェア リリースに追加された **ntlm version** グローバル コンフィギュレーション コマンドには、コマンド オプションが 2 つあります。NTLMv1 を再イネーブルにするための **version 1** オプションと、スタンドアロン Content Engine で NTLMv2 をイネーブルにするための **version 2** オプションです。**version 1** コマンド オプションがデフォルト オプションです。

URL リスト ファイル内のエントリの形式を次に示します。

URL [*depth*] [*domain-name*:*host-name*:*host-domain-name*]

hostname および *host-domain-name* は、null にも設定できます。ただし、NTLM 証明書が設定されている場合には、*domain name* を指定する必要があります（区切り文字が必要です）。

```
http//www.cisco.com 3 apac::
```

事前ロード URL リスト ファイル エントリに NTLM 関連情報が存在しない場合、認証方式は基本認証方式に機能が低下してしまいます。

デフォルトでは、Content Engine は基本認証オブジェクトと NTLM 認証オブジェクトをキャッシュしません。スタンドアロン Content Engine をイネーブルにして特定のオブジェクトを取得し、何らかの認証方式（基本認証または NTLM 認証）によって認証されているこれらのオブジェクトをキャッシュするには、**http cache-authenticated all** 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)# http cache-authenticated all
```

NTLM 認証オブジェクトだけをキャッシュするように Content Engine を設定するには、**http cache-authenticated ntlm** 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。キャッシュ オブジェクトは、Content Engine がクライアントにコンテンツを提供する前に、同じオブジェクトに対する次の要求を認証できるよう、NTLM 保護としてタグ化されます。

WMT ストリーミング メディア ファイルを Content Engine に事前ロードする場合は、事前に Content Engine の WMT 機能をイネーブルにしておく必要があります。Setup ユーティリティを使用して、Content Engine の WMT キャッシング（「Setup ユーティリティによるスタンドアロン Content Engine の基本設定」のステップ 15 を参照）を設定した場合、WMT は Content Engine ですでにイネーブルになっています。それ以外の場合は、Content Engine CLI（Setup ユーティリティではなく）を使用してスタンドアロン Content Engine 上で Windows Media サーバをイネーブルにする方法について、「スタンドアロン Content Engine での WMT RTSP ストリーミングおよびキャッシング サービスの設定」（p.9-16）を参照してから、この Content Engine に関する Windows Media ストリーミング ファイルの事前ロードをイネーブルにしてください。

事前ロード URL リスト ファイルの作成

事前ロード URL リスト ファイルには、Content Engine で事前ロードされる URL（HTTP、FTP-over-HTTP または RTSP URL）がリストされています。このファイルのメンテナンスは管理者が行い、リモート システム上に作成する必要があります。このファイルは事前ロード アクセスのためにスタンドアロン Content Engine に転送されるか、またはリモート サーバからアクセスされます。このファイルのパスは、**pre-load url-list-file** *path* グローバル コンフィギュレーション コマンドで指定します。

（注）　**pre-load url-list-file** *path* グローバル コンフィギュレーション コマンドで、*path* の値は、URL またはローカルのファイル パスになります。

URL のリストをローカル ディスク上のファイルに置くことができます。また、**mkdir** EXEC コマンドを使用して、事前ロード URL リスト ファイルに含まれているサブディレクトリを作成することもできます。たとえば、**mkdir/local1/preload-directory** コマンドを使用すると、preload-directory と呼ばれるサブディレクトリをローカル ディスク上に作成できます。

事前ロード URL リスト ファイル内の各 URL には、オプションの depth パラメータがあります。depth パラメータは、事前ロードが実行されるレベル数を指定します。たとえば、http://www.espn.com 3 は、http://www.espn.com と、そこから下の 3 レベル内にあるすべてのコンテンツをダウンロードします。depth レベルを指定しない場合、事前ロードのデフォルトの depth レベル 3 が使用されます。URL は、次のように改行で区切られます。

```
<cr>
. . .
http://www.cnn.com  3 <cr>
ftp://ftp.lehigh.edu/  2 <cr>
mms://www.aol.com/<dir>/<streaming-file>
http://www.yahoo.com <cr>
. . .
<cr>
```

認証済みのコンテンツを Content Engine に事前ロードする場合、URL リスト ファイルのエントリには次のように書き込む必要があります。

```
http://username:password@www.authenticatedsite.com/ depth level
```

（注）　ACNS 5.4.1 以降のソフトウェア リリースでは、事前ロード済みコンテンツのプロキシ認証のサポートが追加されています。詳細は、「事前コンテンツ ロードのプロキシ認証サポート」（p.11-9）を参照してください。

ACNS 5.1.5 より前の ACNS 5.1.x ソフトウェア リリースでは、Content Engine CLI を使用して事前ロード URL リスト ファイルを設定する場合、**pre-load url-list-file** グローバル コンフィギュレーション コマンドには HTTP または FTP オプションしかなく、事前ロード URL リスト ファイルを安全に取得する方法はありませんでした。

ACNS 5.1.5 ソフトウェア リリースでは、HTTPS を介して事前ロード URL リスト ファイルを取得する機能が追加されました。事前ロード URL リスト ファイルにユーザ名とパスワードが含まれている場合、企業や団体は、HTTPS を介して事前ロード URL リスト ファイルを取得できます。実際には、HTTPS リンクの事前ロードはサポートされていません。HTTPS プロトコルを使用して、事前ロード URL リスト ファイルをダウンロードする機能だけがサポートされています。

スタンドアロン Content Engine 上でのコンテンツ事前ロードのイネーブル化と設定

Content Engine GUI（グラフィカル ユーザ インターフェイス）または CLI のいずれかを使用して、スタンドアロン Content Engine でコンテンツ事前ロードをイネーブル化し、設定することができます。

（注）　Content Engine GUI から、**Caching > Content Preload** の順に選択します。表示された Content Preload ウィンドウで、スタンドアロン Content Engine 上のコンテンツ事前ロード機能をイネーブルにして設定します。Content Preload ウィンドウでこのタスクを実行する方法については、ウィンドウの **HELP** ボタンをクリックします。

Content Engine CLI を使用して、スタンドアロン Content Engine 上でコンテンツ事前ロードをイネーブルにし、設定する手順は、次のとおりです。

ステップ 1　Content Engine 上でコンテンツ事前ロードをイネーブルにします。

```
ContentEngine(config)# pre-load enable
```

ステップ 2　事前ロード URL リスト ファイルを作成します。作成方法は、「事前ロード URL リスト ファイルの作成」（p.11-3）を参照してください。

ステップ 3　URL を取得する場合の最大同時要求数を指定します。1 ～ 30（たとえば、24 など）までの値を指定できます。デフォルトは 10 です。事前ロード URL リスト ファイル内の URL 数が、指定された同時要求数より少ない場合、少ない方の値がアクティブになります。

```
ContentEngine(config)# pre-load concurrent-requests 24
```

ステップ 4　URL が取得するデフォルト depth レベルを指定します（4 レベルなど）。0 ～ 20 の値を指定できます。デフォルトは、3 です。URL を preload.txt ファイルに指定し、Content Engine が他の URL を事前ロードしないように設定する場合は、depth レベルをデフォルトから 0 に設定すると便利です。

```
ContentEngine(config)# pre-load depth-level-default 4
```

ステップ 5　URL リストまたは特定の URL を含むファイルのパスを指定します。

```
ContentEngine(config)# pre-load url-list-file path
```

ここで、*path* は、URL リストまたは特定の URL を含むファイルのパスです。次に例を示します。

```
pre-load url-list-file /local1/myurllist
pre-load url-list-file ftp://ftpserver/ftpdirectory/urllist.txt
pre-load url-list-file http://server/directory/urllist.txt
pre-load url-list-file https://httpsserver/directory/urllist.txt
pre-load url-list-file rtsp://server/directory/urllist.txt
```

実際には、HTTPS リンクの事前ロードはサポートされていません。HTTPS プロトコルを使用して事前ロード URL リスト ファイルをダウンロードする機能だけがサポートされています（前述の例を参照）。

ACNS 5.3.1 以降のソフトウェア リリースでは、事前ロード URL リスト ファイル内で RTSP URL を指定できます。

ステップ 6　事前ロード プロセス中に取得するドメインを指定します（cisco.com など）。

```
ContentEngine(config)# pre-load fetch domain cisco.com
```

ステップ 7　HTML ページ内のその他のドメインを所得するように指定します。デフォルトでは、HTML ページにある他のドメインは、コンテンツ事前ロード時には検索されません。

```
ContentEngine(config)# pre-load traverse-other-domains
```

ステップ 8　事前ロード処理から除外するサフィックスを指定します。次の例では、除外されるオブジェクト用のフィルタを作成します。

```
ContentEngine(config)# pre-load no-fetch suffix .mil .su .ca
```

ステップ 9　事前ロード プロセスの最大帯域幅を設定します（50,000 kbps など）。

```
ContentEngine(config)# pre-load max-bandwidth 50000
```

（注） ACNS 5.x ソフトウェアでは、コンテンツ事前ロード用に指定された URL で、ビットレートが異なる WMT ストリーミング メディア ファイルを事前にロードできます。また、**bandwidth wmt outgoing** および **bandwidth incoming** 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、WMT 帯域幅を制御することもできます。詳細は、「着信および発信 WMT 帯域幅とビット レートの設定」（p.9-26）を参照してください。

ステップ 10　コンテンツ事前ロードをただちに実行するには、**pre-load force** EXEC コマンドを入力します。

ステップ 11　特定のコンテンツを今後に備えて事前ロードするように Content Engine を設定するには、**pre-load schedule** 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。Content Engine は指定された事前ロード スケジュール（**pre-load schedule** グローバル コンフィギュレーション コマンドまたは Content Engine GUI [**Caching > Content Preloading** の順に選択] を使用して設定される）の頻度で、指定されている事前ロード URL リスト ファイルにアクセスします。

事前ロード操作のデフォルト開始時刻は、00:00（深夜 0 時）です。終了時刻が指定されない場合、すべてのオブジェクトがダウンロードされると、事前ロード操作が完了します。このデフォルトを変更する手順は、次のとおりです。

a.　事前ロードを毎日または毎週行う場合、開始時刻と終了時刻を指定するには、*hh*:*mm*（*hh* は時、*mm* は分。たとえば、01:00 など）を使用します。事前ロードを時間ごとに行う場合は、*mm* を使用して開始時刻と終了時間を指定します。次の例では、コンテンツ事前ロードをスケジュールする場合の、毎日の時間間隔を指定する方法を示しています。この例では、事前ロード処理は毎日午前 1:00 に開始し、午前 2:00 に終了します。

```
ContentEngine(config)# pre-load schedule every-day start-time 01:00 end-time 02:00
```

b.　事前ロードを時間ごとに行う場合、開始時刻と終了時刻を指定するには、開始時刻は 0、終了時刻は 59 になります。事前ロードを毎日または毎週行う場合、開始時刻は 0 ～ 23、終了時刻は 0 ～ 59 になります。終了時刻が指定されない場合、事前ロード処理は完了するまで行われます。

事前ロードを毎週 2 日以上設定するには、**pre-load schedule every-week** 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。次の例では、事前ロード処理を毎週日曜日と水曜日の午前 1:00 ～ 6:00 にスケジュールする方法を示しています。

```
ContentEngine#(config)# pre-load schedule every-week Sun Wed
start-time 01:00 end-time 06:00
```

ステップ 12　**pre-load dscp** 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、事前ロードされるすべてのトラフィックの Type of Service（ToS; サービス タイプ）値および Differentiated Services Code Point（DSCP）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。URL の照合、ファイル タイプ、ドメイン、宛先 IP アドレス、送信元 IP アドレス、または宛先ポートに基づいて、IP パケット内に ToS または DSCP の値を設定できます。

ACNS 5.x ソフトウェアには、HTTP、FTP、および WMT の事前ロード トラフィックに対する ToS、または DSCP のサポートが含まれます。コンテンツ事前ロードは、オリジン サーバに接続されたとき、要求クライアントによってではなく、Content Engine によって開始されるため、オリジン サーバと通信する前に、サーバへ向かうトラフィックに ToS、または DSCP のコードポイントを設定する必要があります。

次の例では、ToS を Normal にセットする方法を示します。

```
ContentEngine(config)# pre-load set-tos normal
```

（注）　**pre-load dscp** グローバル コンフィギュレーション コマンドの使用は、DSCP サーバ設定に関係する Rules Template コンフィギュレーション コマンドの使用より優先します。

表 11-1 では、DSCP 値について説明しています。

表 11-1　DSCP 値

DSCP 値	説明
<0-63>	有効な DSCP 値の範囲
af11	AF11 dscp（001010）を指定したパケット
af12	AF12 dscp（001110）を指定したパケット
af13	AF13 dscp（001110）を指定したパケット
af21	AF21 dscp（011010）を指定したパケット
af22	AF22 dscp（010110）を指定したパケット
af23	AF23 dscp（010110）を指定したパケット
af31	AF31 dscp（011010）を指定したパケット
af32	AF32 dscp（011110）を指定したパケット
af33	AF33 dscp（011110）を指定したパケット
af41	AF41 dscp（110010）を指定したパケット
af42	AF42 dscp（110110）を指定したパケット
af43	AF43 dscp（110110）を指定したパケット
cs1	CS1（優先順位 1）dscp（001100）を指定したパケット
cs2	CS2（優先順位 2）dscp（011000）を指定したパケット
cs3	CS3（優先順位 3）dscp（011100）を指定したパケット
cs4	CS4（優先順位 4）dscp（110000）を指定したパケット
cs5	CS5（優先順位 5）dscp（101100）を指定したパケット
cs6	CS6（優先順位 6）dscp（111000）を指定したパケット
cs7	CS7（優先順位 7）dscp（111100）を指定したパケット
デフォルト	デフォルト dscp（000000）を指定したパケット
ef	EF dscp（101110）を指定したパケット

表 11-2 では、ToS 値について説明しています。

表 11-2　ToS 値

ToS 値	説明
<0-127>	有効な ToS 値の範囲
critical	クリティカルの優先順位（110）を指定したパケット
flash	フラッシュの優先順位（48）を指定したパケット
flash-override	フラッシュ上書きの優先順位（64）を指定したパケット
immediate	即時の優先順位（32）を指定したパケット
internet	インターネットワーク制御の優先順位（96）を指定したパケット
max-reliability	最高の信頼性 ToS（2）を指定したパケット
max-throughput	最大スループット ToS（4）を指定したパケット
min-delay	最小遅延 ToS（8）を指定したパケット
min-monetary-cost	最小金銭コスト ToS（1）を指定したパケット
network	ネットワーク制御の優先順位（112）を指定したパケット
normal	通常の ToS（0）を指定したパケット
priority	高い優先順位（16）を指定したパケット

ステップ 13　現在の事前ロード処理のステータスを表示します。

次の例では、**pre-load set-tos** および **pre-load max-bandwidth** コマンドを使用したあとの、現在の事前ロード処理のステータスを示します。

```
ContentEngine# show pre-load
Preloading is enabled
Number of concurrent sessions: 10
Depth level: 4
URL List File: /local1/url.txt
DSCP: set-tos normal
Max Bandwidth: 50000 Kbps
Previous preloading operation will be continued.
Preload will not traverse other domains.
Fetch Domains:
Fetch Suffix:
Fetch Directory:
No-fetch Domain:
No-Fetch Suffix:
No-Fetch Directory:
Scheduling on all days
    Start Time: 00:00
    End Time  : Till completion
```

ステップ 14　事前ロード開始後の現在の事前ロードに関連する統計情報を表示します。

```
ContentEngine# show statistics pre-load
Statistics of last Preloading operation
---------------------------------------

Preloading is in progress.
List of preloaded URLs are in /local1/preload_dir/downloaded_urls.

83 objects downloaded, 2842292 bytes transferred.
```

ステップ 15 エンド ユーザに事前ロードされたファイルの URL の状況を知らせることにより、エンド ユーザはブラウザまたはメディア プレーヤーを使用して、事前ロードされたコンテンツにアクセスできます。

事前ロードされた VOD ファイルがキャッシュされていて、クライアントに正常に配信されているかどうかを確認する方法については、「事前にロードされた VOD ファイルがキャッシュされ、Windows Media クライアントに適切に配信されたことの確認」（p.9-44）を参照してください。

事前コンテンツ ロードのプロキシ認証サポート

事前ロードを機能させるには、通常、URL リスト ファイル（url.txt ファイル）で指定されたコンテンツを事前ロードするように スタンドアロン Content Engine を設定します。URL リスト ファイルでは、URL を取得する depth レベルも指定します（4 レベルなど）。保護されたオブジェクトを取得する必要がある場合は、サーバ認証を実行するときに、URL 内でユーザ名およびパスワードを使用することができます。中間プロキシが存在する場合は、Content Engine がプロキシ認証を実行できないため、コンテンツの事前ロードが機能しません。

ACNS 5.4.1 より前のソフトウェア リリースでは、通常、Content Engine（CE1）のアップストリーム プロキシ サーバとして機能する別の Content Engine（CE2）を導入しました。CE2 で **http authentication header 401** グローバル コンフィギュレーション コマンドを指定した場合、CE1 は NTLM 認証をサポートできました。NTLM 認証の場合、要求は認証され（WWW-Auth として処理され）、証明書が一致すれば、CE1 はコンテンツを取得し、CE1 に事前ロードします。

ACNS 5.4.1 以降のソフトウェア リリースでは、事前コンテンツ ロードのプロキシ認証サポートが追加されています。事前コンテンツ ロードのプロキシ認証がサポートされるのは、発信プロキシ サーバで NTLM、LDAP、RADIUS、および TACACS+ のいずれかの認証方式を使用するように Content Engine が設定されている場合です。この機能は、アップストリーム プロキシ サーバを 1 レベルだけサポートします（事前ロード ファイルから取得したプロキシ情報は、直近のアップロード プロキシ サーバにのみ適用されます）。

ACNS 5.4.1 以降のソフトウェア リリースでは、プロキシ認証方式として NTLM 認証または基本認証を使用するように CE1 が設定されている場合、CE1 は初期プロキシ認証を実行し、その後、要求されたオブジェクトをオリジン サーバから取得できます。認証されていないオブジェクト、または認証されたオブジェクトを取得できます。

事前ロードでのプロキシ認証をサポートするには、NTLM 認証または基本認証でプロキシ認証を実行できるように、Content Engine の事前コンテンツ ロードでプロキシ ユーザおよびプロキシ ドメイン名を許可する必要があります。現在サポートされている事前ロード プロセスの URL フォーマットは、次のとおりです。

http://*user1*:*user1*@10.77.157.131/kerberos/kerberos.htm 1 *acns*:*acns*:*acns*

user1 はユーザ名、*user1* はパスワード、1 は depth レベル、*acns* はホストが配置されたドメイン名、ホスト名、およびドメインです。

プロキシ認証のユーザ名、パスワード、およびドメイン名を使用できるフォーマットは、次のとおりです。

proxy_user:*username*:proxy_pwd:*pwd*:proxy_domain_name:*domain*

各ユーザ名、パスワード、ドメイン名の文字長は最大で 50 です。この情報は、事前ロード リスト ファイルの先頭行で指定する必要があります。事前ロード リスト ファイルから情報を取得するための区切り文字には、コロン（:）を使用します。したがって、ユーザのパスワードにはコロンを使用しないでください。

事前ロード ファイル内でプロキシ認証のユーザ名、パスワード、およびドメイン名を使用できない場合、Content Engine は URL フォーマットを使用して事前ロード処理を実行します。

Content Engine がプロキシ認証方式に NTLM を使用するように設定されている場合、事前ロード ファイルには次のタイプの情報が含まれます。

proxy_user:user1:proxy_pwd:user1:proxy_domain_name:acns

Content Engine がプロキシ認証方式に基本認証を使用するように設定されている場合、事前ロード ファイルには次のタイプの情報が含まれます。

proxy_user:tuser1:proxy_pwd:tpass1:proxy_domain_name:null

スタンドアロン Content Engine にコンテンツ事前ロードのプロキシ認証を設定する例については、次の「事前コンテンツ ロードのプロキシ認証の設定例」を参照してください。

事前コンテンツ ロードのプロキシ認証の設定例

次の例では、Content Engine（CE1）は、発信プロキシ サーバに Content Engine CE2 を使用し、事前コンテンツ ロードに NTLM 認証を使用するように設定されています。

ステップ 1　プロキシ認証方式に NTLM 認証を使用するように Content Engine（CE1）を設定します。

```
CE1(config)# ip name-server name-server
CE1(config)# ntlm server host ip-address
CE1(config)# ntlm server enable
```

ステップ 2　CE1 のキャッシュおよび統計情報をクリアします。

```
CE1# clear cache
CE1# clear statistics all
```

ステップ 3　CE1 の事前ロード機能をイネーブルにします。

```
CE1(config)# pre-load enable
```

ステップ 4　CE1 の URL リスト ファイルのパスを設定します。

```
CE1(config)# pre-load url-list-file /local1/preload.txt
```

ステップ 5　CE1 の発信プロキシ サーバ（CE2）を設定します。

```
CE1(config)# http proxy outgoing host ip-address port-number
```

ステップ 6　CE1 で次のコマンドを入力して、407 HTTP 認証ヘッダーを保護します。

```
CE1(config)# http proxy outgoing preserve-407
```

ステップ 7　CE2 で認証をイネーブルにします。

```
CE2(config)# http proxy incoming port-number
```

ステップ 8　CE1 の URL リスト ファイルの先頭行で、NTLM オリジン サーバに対して証明書を照合するように指定します。

```
CE1# type preload.txt
proxy_user:preload:proxy_pwd:preload1:proxy_domain_name:acns
http://www.yahoo.com/
http://10.77.157.60/
```

上記の事前ロード ファイルでは、ユーザ名が preload、パスワードが preload1、ドメイン名が acns です。CE1 が発信プロキシ サーバ（CE2）から 407 メッセージを受信すると、これらの証明書が NTLM オリジン サーバに送信されます。

ステップ 9　CE1 に事前ロード処理を設定します。

```
CE1# preload force
```

ステップ 10　CE1 に設定された事前ロード処理が完了したら、すべてのオブジェクトが CE1 に適切に事前ロードされたかを確認します。

a.　CE1 で、/local/local1/preload_dir ディレクトリ内の latest_preloaded_objects ファイルを調べます。

b.　CE1 で、latest_preload_error ファイルを調べて、プロキシ認証障害または 407 メッセージによってエラーが発生したことを示すエントリがないことを確認します。

c.　CE1 で、次のコマンドを入力して、CE1 にオブジェクトが適切に事前ロードされたことを確認します。

```
CE1# show statistics preload
         Preloading was initiated by force.
         Preloading started at Thu Mar 13 06:42:40 2003
         Preloading ended   at Thu Mar 13 06:42:53 2003
         List of preloaded URLs are in
/local1/preload_dir/latest_preloaded_objects.
         Preload errlog is /local1/preload_dir/latest_preload_error.

         Number of invalid entries in URL list file  =          0
         Total number of preloaded objects           =          1
         Total number of preloaded bytes             =        570
```

d.　CE2 で、次のコマンドを入力して、CE2 の認証キャッシュ内にユーザ（preload という名前のユーザ）があるか確認します。

```
CE2# show http-authcache
```

ステップ 11　プロキシ サーバとして CE1 が設定された Web クライアントから、ブラウザを使用して、http://10.77.157.60 に関する要求を発行します。

ステップ 12　このクライアント要求が CE1 のキャッシュから提供されたことを確認します。

a.　CE1 で、次のコマンドを入力します。

```
CE1# show statistics http savings
```

b.　コマンド出力を調べて、CE1 のヒット カウントが増加したことを確認します。

c. CE1 で、次のコマンドを入力して、CE1 にオブジェクトが適切に事前ロードされたことを確認します。

```
CE1# show statistics preload
         Preloading was initiated by force.
         Preloading started at Thu Mar 13 06:42:40 2003
         Preloading ended   at Thu Mar 13 06:42:53 2003
         List of preloaded URLs are in
/local1/preload_dir/latest_preloaded_objects.
         Preload errlog is /local1/preload_dir/latest_preload_error.

         Number of invalid entries in URL list file  =          0
         Total number of preloaded objects           =          1
         Total number of preloaded bytes             =        570
```

d. CE2 で、次のコマンドを入力して、CE2 の認証キャッシュ内にユーザ（preload という名前のユーザ）があるか確認します。

```
CE2# show http-authcache
```

ステップ 13 プロキシとして CE1 が設定された Web クライアントから、ブラウザを使用して、http://10.77.157.60 に関する要求を発行します。

ステップ 14 このクライアント要求が CE1 のキャッシュから提供されたことを確認します。

a. CE1 で、次のコマンドを入力します。

```
CE1# show statistics http savings
```

b. コマンド出力を調べて、CE1 のヒット カウントが増加したことを確認します。

スタンドアロン Content Engine 上でのコンテンツ事前ロードの停止と再開

スタンドアロン Content Engine 上で現在進行中の事前ロード プロセスを停止するには、**no pre-load enable** グローバル コンフィギュレーション コマンドを使用します。

コンテンツ事前ロードがスケジュールされた終了時刻までに完了しなかった場合、**pre-load resume** 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、コンテンツを取得する事前ロード プロセスを再開できます。このコマンドを使用すると、事前ロード URL リスト ファイルの最初からでなく、事前ロードを前回中断した部分からダウンロードを再開できます。

（注）　**pre-load resume** グローバル コンフィギュレーション コマンドが Content Engine 上で設定されてなく、コンテンツ事前ロードがスケジュールされた終了時刻以前に中断された場合、次のスケジュールされたコンテンツ事前ロードは、URL リスト ファイルの最初から始まります。

スタンドアロン Content Engine 上での URL フィルタリングの設定

一部の企業や団体では、インターネット上のビジネス以外のコンテンツ、および好ましくないコンテンツへの社員のアクセスを監視、管理、制限する必要性を認識しています。社員や学生に対して、Web サイトへのアクセスを許可または拒否したり、インターネットの正しい使用方法を指導することができます。Content Engine に URL フィルタリング方式を設定すると、生産性を向上させ、ネットワーク帯域幅を本来の業務のみに使用することによって、企業はすぐに投資を回収できるうえ、ネットワークの不正使用による法的責任問題が軽減されます。

表 11-3 は、クライアントの Web サイトへのアクセスを制御するため、スタンドアロン Content Engine に設定するさまざまな URL フィルタリング方式を表示しています。

表 11-3　スタンドアロン Content Engine での URL フィルタリング方式

URL フィルタリング方式	詳細
ローカル リスト ファイル	リストで指定する URL へのアクセスを拒否する。リストで指定する URL へのアクセスだけを許可する。「スタンドアロン Content Engine 上での ローカル リスト URL フィルタリングの設定」（p.11-14）を参照してください。
N2H2 外部サーバ	URL フィルタリングのため、コンテンツに対するクライアント要求を外部の N2H2 サーバへ転送する。「N2H2 URL フィルタリングのためのスタンドアロン Content Engine の設定」（p.11-20）を参照してください。
Websense サーバ	URL フィルタリングのため、コンテンツに対するクライアント要求をローカルの Websense プラグイン、または外部の Websense サーバへ転送する。「Websense URL フィルタリングのためのスタンドアロン Content Engine の設定」（p.11-22）を参照してください。
SmartFilter プラグイン	URL フィルタリングのため、コンテンツに対するクライアント要求を SmartFilter プラグインに転送する。「SmartFilter ソフトウェアを使用した URL フィルタリングの設定」（p.11-42）を参照してください。

各プロトコルでサポートされている URL フィルタリング方式（たとえば、SmartFilter や Websense）のリストについては、表 B-6 を参照してください。

一度にプロトコルごとにアクティブにできる URL フィルタリング方式は 1 つだけですが、複数の URL フィルタリング方式を同時にサポートできます。たとえば、N2H2 フィルタが HTTP 要求に適用されている場合は、他の URL フィルタリング方式（たとえば、Websense や SmartFilter）をこのプロトコルに適用できません。ただし、ローカル リスト URL フィルタリング方式（good リストと bad リスト）は、ストリーミング メディア プロトコル（WMT クライアント要求と RTSP を介したクライアント要求）へ適用できます。特定のプロトコルに対してイネーブルにされている方式は、ほかのプロトコルから影響を受けません。

（注）　**url-filter** グローバル コンフィギュレーション コマンドは、**rule** 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。したがって、**url-filter** コマンドが要求をブロックしなかった場合だけ、**rule no-block** コマンドが実行されます。

URL フィルタリングが、Content Engine を通過するすべての URL に適用されるようにするには、すべてのバイパス機能をディセーブルにします。デフォルトでは、ロード バイパスはイネーブルです。

- Content Engine GUI を使用して、ロード バイパスを手動でディセーブルにするには、**Caching > Bypass** の順に選択し、次に Bypass ウィンドウの **Load Bypass Off** オプション ボタンをクリックします。

- Content Engine CLI を通してロード バイパスを手動でディセーブルにするには、**bypass load** 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。

```
ContentEngine(config)# no bypass load enable
```

- Content Engine CLI を通してエラー処理を手動でディセーブルにするには、**error-handling send-cache-error** または **error-handling reset-connection** 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。デフォルトでは、エラー処理は Content Engine 上でイネーブルです。

```
ContentEngine(config)# no error-handling send-cache-error
ContentEngine(config)# no error-handling reset-connection
```

RADIUS 認証および URL フィルタリングが Content Engine 上でイネーブルのときは、RADIUS サーバ データベース内のユーザ Filter-Id 属性を設定し、URL フィルタリングをバイパスできます。

次の例では、RADIUS サーバ データベースのユーザ Filter-Id 属性エントリを示しています。

```
test        Password = "test"
            Service-Type = Framed-User,
            Filter-Id = "No-Web-Blocking"
```

Filter-Id 属性は、No-Web-Blocking または Yes-Web-Blocking として定義できます。Yes-Web-Blocking とは、要求が URL フィルタリングされることを意味し、No-Web-Blocking は、要求が URL フィルタリングされないことを意味します。ブロッキングが未指定の場合は、Yes-Web-Blocking が RADIUS フィルタリングのデフォルトとなります。

（注） RADIUS サーバの認証情報および URL フィルタリングの情報は、「RADIUS 認証および許可の概要」（p.17-6）を参照してください。

スタンドアロン Content Engine 上での ローカル リスト URL フィルタリングの設定

badurl.lst ファイルにリストされている URL へのクライアント要求を拒否するように、スタンドアロン Content Engine を設定できます。また、goodurl.lst ファイルにリストされている URL への要求だけを許可するようにも設定できます。ローカル リスト ファイル（URL リスト）は、RTSP ストリーミング メディア プロトコルだけでなく HTTP（HTTP、HTTPS-over-HTTP、FTP-over-HTTP）にも適用されます。このタイプの URL フィルタリングは*ローカル リスト URL フィルタリング*といいます。

ヒント プロトコルごとに、優良サイト ファイルまたは悪質サイト ファイルを 1 つだけ同時にアクティブにできます。

各プロトコルのローカル リスト ファイルには、他のプロトコルに属している URL を含めることはできません。たとえば、HTTP ローカル リスト ファイルには、HTTP、HTTPS、または FTP URL のタイプの URL のみを含める必要があります。ACNS 5.3.1 以降のソフトウェア リリースでは、WMT ローカル ファイルに RTSP URL を含めることができます。

注意 ローカル リスト ファイルが大き過ぎると、プロキシのパフォーマンスに影響します。これは、ローカル リスト フィルタリングがイネーブルになると、そのローカル リスト ファイルがメモリにロードされるからです。ファイル サイズが 5 MB を超えると警告メッセージが表示されますが、ACNS ソフトウェアによってローカル リスト ファイルのサイズが制限されることはありません。ユーザの責任において、ローカル リスト ファイルのサイズを管理し、パフォーマンスに影響するほど大きくならないようにしてください。

ローカル リスト URL フィルタリングを使用して、コンテンツに関する次の種類のクライアント要求をフィルタリングするようにスタンドアロン Content Engine を設定できます。

- HTTP を介した要求（HTTP、FTP-over-HTTP、および HTTPS-over-HTTP 要求）
- RealMedia 要求（RealNetworks 独自の拡張を含む IETF スタンドアロン RTSP プロトコル）
- WMT 要求（MMS-over-HTTP および Windows Media 9 クライアントや Windows Media 9 サーバ対応の RTSP-over-RTP）

ネイティブ FTP 要求 とネイティブ HTTPS 要求のフィルタリングはサポートしていません。

ACNS 5.3.1 以降のソフトウェア リリースでは、Windows Media 9 Player の RTSP-over-RTP サポート（別名 *WMT RTSP 要求*）を使用できます。WMT RTSP 要求には、rtsp、rtspu、および rtspt の 3 つの有効なプロトコル プレフィックスがあります。

ユーザが URL のプロトコル プレフィックスとして rtsp: を入力した場合、Windows Media 9 Player は RTSPT または RTSPU を使用するように選択できます。rtsp bad ファイルに URL rtsp://*hostname*/*pathname* があり、ユーザの URL 要求が rtspt://*hostname*/*pathname* である場合、Windows Media 9 Player からの RTSP 要求は URL フィルタリングを通過することがあります。ACNS 5.3.1 以降のソフトウェア リリースでは、Windows Media 9 Players からの RTSP 要求に専用の URL フィルタリングを使用できます。

WMT URL フィルタリングの場合、RTSP URL（rtsp://）のフィルタリングのみがサポートされます。RTSPT および RTSPU URL のフィルタリング サポートされていません。ただし、badurl.lst ファイルに RTSP URL を設定した場合は、RTSPT と RTSPU の両方の URL がブロックされます。

Content Engine CLI を使用して、スタンドアロン Content Engine 上でローカル リスト URL フィルタリングを設定するには、**url-filter** 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。ACNS 5.3.1 ソフトウェア リリースでは、Windows Media 9 Players からの RTSP 要求に対してローカル リスト URL フィルタリングをサポートするように、**url-filter** コマンドが変更されました。ACNS 5.2.x 以前のソフトウェア リリースでは、**url-filter** コマンド オプションは次のとおりでした。

```
ContentEngine(config)# url-filter ?
  http  For requests over HTTP
  rtsp  For requests over RTSP
  wmt   For WMT requests
```

ACNS 5.3.1 以降のソフトウェア リリースでは、**url-filter** コマンド オプションは次のとおりです。

```
ContentEngine(config)# url-filter ?
  http  For requests over HTTP and MMS over HTTP
  rtsp  For requests over RTSP - applies to real proxy, real server and cisco
        streaming engine
  wmt   For WMT requests - applies to MMS and RTSP
```

ローカル リスト URL フィルタリングは、WMT 要求（WMT クライアントからの MMS-over-HTTP および RTSP 要求）や、RTSP 要求（RealMedia プレーヤーからの要求）に対してサポートされている唯一のフィルタリング方式です。サードパーティ製 URL フィルタリング方式（N2H2、SmartFilter、および Websense ソフトウェア）では、WMT および RTSP 要求がサポートされていません。HTTP 要求については、N2H2、SmartFilter、Websense URL フィルタリングだけでなく、ローカル リスト URL フィルタリング方式もサポートされています。各プロトコルでサポートされている URL フィルタリングのリストについては、表 B-4 を参照してください。

表 11-4 は、HTTP 要求（HTTP、FTP-over-HTTP、MMS-over-HTTP、および HTTPS-over-HTTP 要求）に対してローカル リスト URL フィルタリングを使用するようにスタンドアロン Content Engine を設定する、Content Engine CLI グローバル コンフィギュレーション コマンドを説明しています。

表 11-4　HTTP を介した要求にローカル リスト URL フィルタリングを使用するためのスタンドアロン Content Engine の設定

CLI コマンド	説明
url-filter http bad-sites-deny enable	HTTP 悪質サイト リストにある URL のクライアント要求を拒否するように Content Engine を設定する。
url-filter http bad-sites-deny file *filename*	HTTP 悪質サイト リストのファイル名を指定する。
url-filter http good-sites-allow enable	HTTP 優良サイト リストにある URL のクライアント要求を許可するように Content Engine を設定する。
url-filter http good-sites-allow file *filename*	HTTP 優良サイト リストのファイル名を指定する。

表 11-5 は、RTSP を介した要求に対してローカル リスト URL フィルタリングを使用するようにスタンドアロン Content Engine を設定する Content Engine CLI グローバル コンフィギュレーション コマンドを説明しています。このタイプの URL フィルタリングは、RealProxy（スタンドアロン Content Engine が稼働しているバックエンド RTSP サーバ）と組み合わせて使用します。Content Engine が登録されている場合、登録済み Content Engine で稼働している RealProxy、RealSubscriber、および Cisco Streaming Engine は、このタイプの URL フィルタリングを使用します。

表 11-5　RTSP を介した要求に対してローカル リスト URL フィルタリングを使用するためのスタンドアロン Content Engine の設定

CLI コマンド	説明
url-filter rtsp bad-sites-deny enable	RTSP 悪質サイト リストにある URL のクライアント要求を拒否するように Content Engine を設定する。
url-filter rtsp bad-sites-deny file *filename*	RTSP 悪質サイトリストのファイル名を指定する。
url-filter rtsp good-sites-allow enable	RTSP 優良サイト リストにある URL のクライアント要求を許可するように Content Engine を設定する。
url-filter rtsp good-sites-allow file *filename*	RTSP 優良サイト リストのファイル名を指定する。

表 11-6 は、WMT 要求（RTSP を介した WMT 要求）に対してローカル リスト URL フィルタリングを使用するようにスタンドアロン Content Engine を設定する Content Engine CLI グローバル コンフィギュレーション コマンドを説明しています。WMT 悪質サイト リストに RTSP URL を設定すると、悪質サイト リストで指定された RTSP URL だけでなく、RTSPT と RTSPU URL もブロックされます。

表 11-6　WMT 要求に対してローカル リスト URL フィルタリングを使用するためのスタンドアロン Content Engine の設定

CLI コマンド	説明
url-filter wmt bad-sites-deny enable	WMT 悪質サイト リストにある URL のクライアント要求を拒否するように Content Engine を設定する。
url-filter wmt bad-sites-deny file *filename*	WMT 悪質サイト リストのファイル名を指定する。
url-filter wmt good-sites-allow enable	WMT 優良サイト リストにある URL のクライアント要求を許可するように Content Engine を設定する。
url-filter wmt good-sites-allow file *filename*	WMT 優良サイト リストのファイル名を指定する。

ACNS 5.3.1 以降のソフトウェア リリースでは、**url-filter wmt** グローバル コンフィギュレーション コマンドは RTSP に適用されます。RTSP 要求の URL フィルタリングは、クライアントが Windows Media 9 Player、サーバが Windows Media 9 Server の場合に使用されます。初期バージョンの Windows Media Player を使用している場合（Windows Media 7 Players など）、RTSP プロトコルでなく MMS-over-HTTP プロトコルを使用して、Windows Media Player からのコンテンツ要求を処理します。

ローカル URL リストを使用した URL フィルタリングの設定例

ローカル リスト ファイルを使用して特定の HTTP URL に対するクライアント要求を拒否するようにスタンドアロン Content Engine を設定する手順は、次のとおりです。

ステップ 1　badurl.lst という名前のプレーン テキスト ファイルを作成します。

このファイルに、ブロックする URL を入力します。badurl.lst ファイル内の URL のリストは、http://www.domain.com/ 形式で入力し、改行キーで区切ります。

ステップ 2　スタンドアロン Content Engine の /local1 システム ファイル システム（sysfs）ディレクトリに、badurl.lst ファイルをコピーします。

ヒント　bad リストを保持するために、local1 の下に別のディレクトリ（たとえば、**/local1/filtered_urls**）を作成することを推奨します。

ステップ 3　Content Engine が悪質 URL リストをポイントするように設定します。

```
ContentEngine(config)# url-filter http bad-sites-deny file local/local1/badurl.lst
```

ステップ 4　この URL をアクティブに拒否するように、Content Engine を設定します。

```
ContentEngine(config)# url-filter http bad-sites-deny enable
```

ステップ 5　スタンドアロン Content Engine に新しい悪質サイト リストをリロードします。

```
ContentEngine# url-filter local-list-reload http
```

ローカル リスト ファイルを使用して特定の HTTP URL を許可し、その他のすべての URL を拒否するようにスタンドアロン Content Engine を設定するには、次の手順に従ってください。

ステップ 1　goodurl.lst という名前のプレーン テキスト ファイルを作成します。

このファイルに、排他的に許可する URL を入力します。goodurl.lst ファイル内の URL のリストは、http://www.domain.com/ 形式で入力し、改行キーで区切ります。

ステップ 2　goodurl.lst ファイルを、Content Engine の /local1 sysfs ディレクトリにコピーします。

ヒント　good リストを保持するために、local1 の下に別個のディレクトリ（たとえば、/local1/filtered_urls）を作成することを推奨します。

ステップ 3　Content Engine が goodurl.lst ファイルをポイントするように設定します。

```
ContentEngine(config)# url-filter http good-sites-allow file local/local1/goodurl.lst
```

ステップ 4　優良 URL のみをアクティブに許可するように、Content Engine を設定します。

```
ContentEngine(config)# url-filter http good-sites-allow enable
```

ステップ 5　スタンドアロン Content Engine に新しい優良サイト リストをリロードします。

```
ContentEngine# url-filter local-list-reload http
```

スタンドアロン Content Engine へのローカル リスト ファイルのリロード

badurl.lst または goodurl.lst ファイルを更新する場合は、**url-filter local-list-reload** EXEC コマンドを使用して優良サイトまたは悪質サイトのリストをスタンドアロン Content Engine 上にリロードします（URL リスト機能がイネーブルの場合）。

url-filter local-list-reload {**http** | **rtsp** | **wmt**}

構文は次のとおりです。

- **http** は HTTP 要求（HTTP、FTP-over-HTTP、MMS-over-HTTP、および HTTPS-over-HTTP 要求）のための新しいローカル リストをリロードします。
- **rtsp** は RTSP を介した要求（RealMedia クライアントからの要求）用のローカル リストをリロードします。
- **wmt** は、WMT 要求（Windows Media 9 Players からの RTSP-over-RTP [Microsoft 独自の拡張を含む標準 IETF RTSP プロトコル] 要求）用のローカル リストをリロードします。

次に、スタンドアロン Content Engine 上に新しい優良サイトまたは悪質サイトのリストをリロードする方法を示します。

```
ContentEngine# url-filter local-list-reload http
ContentEngine# url-filter local-list-reload rtsp
ContentEngine# url-filter local-list-reload wmt
```

カスタム ブロック メッセージの作成

ローカル リスト URL フィルタリングの場合は、カスタマイズしたブロック メッセージを、Content Engine から提供されるコンテンツを要求したクライアントに返すように、スタンドアロン Content Engine を設定できます。カスタム メッセージは、block.html と名付けられた管理者作成の HTML ページである必要があります。カスタム メッセージの HTML ページに関連付けられたすべての内蔵グラフィックスを、block.html ファイルを含む同じディレクトリに必ずコピーしてください。次の例は、block.html ファイルの中身です。

```
<TITLE>Cisco Content Engine example customized message for url-filtering</TITLE>
<p>
<H1>
<CENTER><B><I><BLINK>
<FONT COLOR="#800000">P</FONT>
<FONT COLOR="#FF00FF">R</FONT>
<FONT COLOR="#00FFFF">A</FONT>
<FONT COLOR="#FFFF00">D</FONT>
<FONT COLOR="#800000">E</FONT>
<FONT COLOR="#FF00FF">E</FONT>
<FONT COLOR="#00FFFF">P</FONT>
<FONT COLOR="#FF8040">'</FONT>
<FONT COLOR="#FFFF00">S</FONT>
</BLINK>
<FONT COLOR ="#0080FF">Blocked Page</FONT>
</I></B></CENTER>
</H1>
<p>
<p>
<IMG src="/content/engine/blocking/url/my.gif">
<p>
This page is blocked by the Content Engine.
<p>
```

block.html ファイルが更新されると、**url-filter http custom-message** コマンドを再入力しなくても、新しいメッセージが自動的に表示されます。

次の例では、スタンドアロン Content Engine がブロックされたサイトへの要求を代行受信した場合、block.html ファイルによって次のカスタム メッセージが表示されます。

```
This page is blocked by the Content Engine
```

block.html ファイルでは、オブジェクト（.gif、.jpeg など）は、上記の例に示すように、カスタム メッセージ ディレクトリ ストリング /content/engine/blocking/url 内で参照する必要があります。

カスタマイズされたブロック メッセージをイネーブルにするには、**url-filter http custom-message** グローバル コンフィギュレーション コマンドを使用し、ディレクトリ名を指定します。カスタム メッセージをディセーブルにするには、**no url-filter http custom-message** コマンドを使用します。

url-filter http custom-message コマンドは、**good-sites-allow** および **bad-sites-deny** の設定に影響を与えることなく、イネーブルやディセーブルにすることができます。

（注） local1 または local2 を、カスタム ブロック メッセージ用のディレクトリとして使用しないでください。カスタム メッセージ ファイルを保持するには、local1 または local2 の下に別のディレクトリを作成してください。

ブロックされたサイトへのアクセス要求について質問がある場合は、システム管理者に問い合わせてください。

N2H2 URL フィルタリングのためのスタンドアロン Content Engine の設定

N2H2 は、グローバルに展開される URL フィルタリング ソリューションです。このソフトウェアは、宛先ホスト名、宛先 IP アドレス、およびユーザ名とパスワードに基づいて、HTTP、FTP、または HTTPS の要求をフィルタリングすることができます。N2H2 は、1,500 万サイトを超える高度な URL データベースに基づいており、インターネット テクノロジーと人間によるレビューにより、40 を超えるカテゴリに分類されています。N2H2 フィルタリング製品については、http://www.n2h2.com を参照してください。

（注）　N2H2 サーバを使用した各プロトコルをサポートしている URL フィルタリングのリストについては、表 B-4 を参照してください。

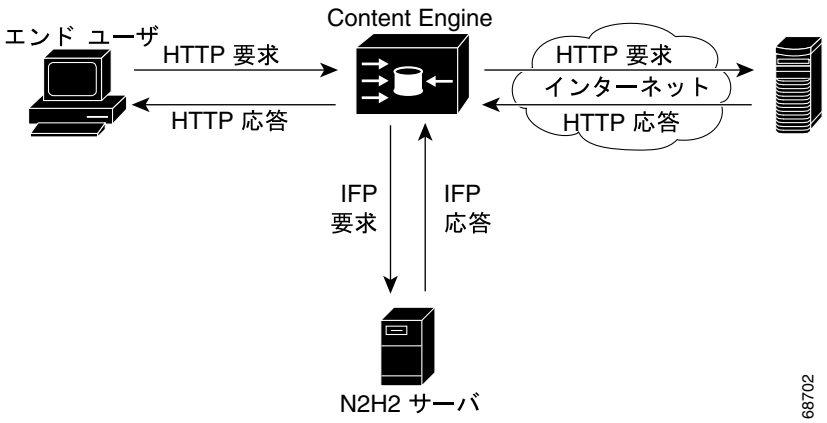
N2H2 は、3 種類のフィルタリング方式をサポートします。表 11-7 では、Content Engine がサポートする N2H2 機能を示しています。1 台の N2H2 サーバで、同時に複数の Content Engine をサポートできます。

表 11-7　サポートされている N2H2 機能

N2H2 機能	説明
グローバル フィルタリング	すべての HTTP 要求（HTTP、FTP-over-HTTP、HTTPS-over HTTP 要求）にフィルタリングを適用する。
ユーザ ベースのフィルタリング	特定のユーザまたはグループにフィルタリングを適用する。
クライアント IP ベースのフィルタリング	特定のクライアント IP アドレスにフィルタリングを適用する。
トランスペアレント認証	トランスペアレント認証は、IFP 応答内の HTML ページを使用して、初期応答ヘッダーをクライアントへ返して行われる。

スタンドアロン Content Engine では、N2H2 Enterprise サーバをフィルタリング エンジンとして使用し、N2H2 サーバ上で設定されたフィルタリング ポリシーを実行できます（図 11-1 を参照）。スタンドアロン Content Engine と N2H2 サーバは、Internet Filtering Protocol（IFP）Version 2 を使用して相互に通信します。Content Engine は URL 要求を受信すると、要求されたその URL を含めた IFP 要求を N2H2 サーバに送信します。N2H2 サーバは、必要な URL 検索を実行し、IFP 応答を返します。N2H2 サーバの IFP 応答に基づいて、Content Engine は、ブロック メッセージが表示されるページにブラウザをリダイレクトして HTTP 要求をブロックするか、URL 要求をオリジン サーバに送信して、通常の HTTP 処理を続行します。

図 11-1　N2H2 フィルタリング



（注）　要求されたオブジェクトがキャッシュ内にあるかどうかにかかわらず、N2H2　サーバを使用する URL フィルタリングが HTTP トラフィック（HTTP、FTP-over-HTTP、HTTPS-over-HTTP 要求）に適用されたあと、Rules Template が適用されます。N2H2 を使用した他のプロトコルをサポートしている URL フィルタリングのリストについては、表 B-4 を参照してください。

URL フィルタリングに外部 N2H2 サーバを使用するようにスタンドアロン Content Engine を設定する手順は、次のとおりです。

ステップ 1　HTTP を介した要求用にこの Content Engine で現在イネーブルになっている URL フィルタリング方式を表示します。

```
ContentEngine# show url-filter http
```

ステップ 2　HTTP を介した要求用にイネーブルになっている URL フィルタリング方式がほかにないことを確認してください（たとえば、Websense や SmartFilter ソフトウェア）。URL フィルタリング方式は、プロトコルごとに一度に 1 つのみアクティブにできます。

ステップ 3　**url-filter http N2H2 server** 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、URL フィルタリングに外部 N2H2 サーバを使用するように Content Engine を設定します。

a. 外部 N2H2 サーバについて必要な情報を指定します（たとえば、IP アドレス）。

url-filter http N2H2 server {[*hostname* | *ip-address*]} [**port** *portnum* [**timeout** *seconds*]]

- *hostname* は 外部 N2H2 サーバのホスト名です。
- *IP address* は 外部 N2H2 サーバの IP アドレスです。
- *portnum* はポート番号（1 ～ 65535）で、Content Engine は指定されている N2H2 サーバのこのポートへ IFP 要求を送信します。デフォルト ポート番号は 4005 です。
- *seconds* は秒数（1 ～ 120）で、接続がタイムアウトするまで Content Engine が N2H2 サーバからの IFP 応答を待つ時間です。デフォルトのタイムアウトは 5 秒です。

次の例では、Content Engine は IP アドレスが 172.16.22.10 の N2H2 サーバを使用するように設定されています。Content Engine は、IFP 要求をこの N2H2 サーバのポート 4008 へ送信し、接続がタイムアウトするまで、サーバからの IFP 応答を最大で 100 秒間待ちます。

```
ContentEngine(config)# url-filter http N2H2 server 172.16.22.10 port 4008 timeout
100
```

スタンドアロン Content Engine に設定されているサーバの IP アドレスとポート番号は、N2H2 サーバの IP アドレス、および N2H2 サーバが IFP 要求を待ち受けるポートと一致する必要があります。Content Engine 上の設定が、N2H2 サーバ上の設定と一致しない場合、Content Engine は、すべての HTTPS 要求（HTTP、FTP-over-HTTP、HTTPS-over-HTTP 要求）をタイムアウトにし、**allowmode** オプション の設定に基づいて、すべての HTTP トラフィックをブロック、または許可します。

（注）　**url-filter http N2H2 server** グローバル コンフィギュレーション コマンドは、指定された IP アドレスで N2H2 サーバに現在アクセスできるかどうか確認しません。N2H2 がイネーブルになっている間に、設定が変わる場合があります。Content Engine は実行時に新しい設定を受け入れます。

ステップ 4　この Content Engine で N2H2 URL フィルタリング方式をイネーブルにします。

```
ContentEngine(config)# url-filter http N2H2 enable
```

ステップ 5　**url-filter http N2H2 allowmode 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、N2H2 サーバがイネーブルになっているにもかかわらず、Content Engine が N2H2 サーバと正常に通信できない場合に、HTTP 要求（HTTP、FTP-over-HTTP、HTTPS-over-HTTP 要求）が通過できるようにします。

- **allowmode** がイネーブルの場合、N2H2 サーバからの応答を受信できなくても、Content Engine はすべての HTTP トラフィックの通過を許可（通常のトラフィック処理を継続）します。
- **allowmode** がディセーブルの場合、N2H2 サーバからの応答を受信できなければ、Content Engine は通過する HTTP トラフィックをすべてブロックします。

デフォルトでは、**allowmode** はイネーブルです。N2H2 がイネーブルでもディセーブルでも **allowmode** オプションは設定可能で、N2H2 サーバの設定とは独立しています。N2H2 URL フィルタリングをすでに行っていても、Content Engine は **allowmode** の新しい設定を受け入れます。

ステップ 6　Content Engine と N2H2 サーバ間の通信に関する要求 / 応答統計情報を表示します。

```
ContentEngine# show statistics url-filter http N2H2
```

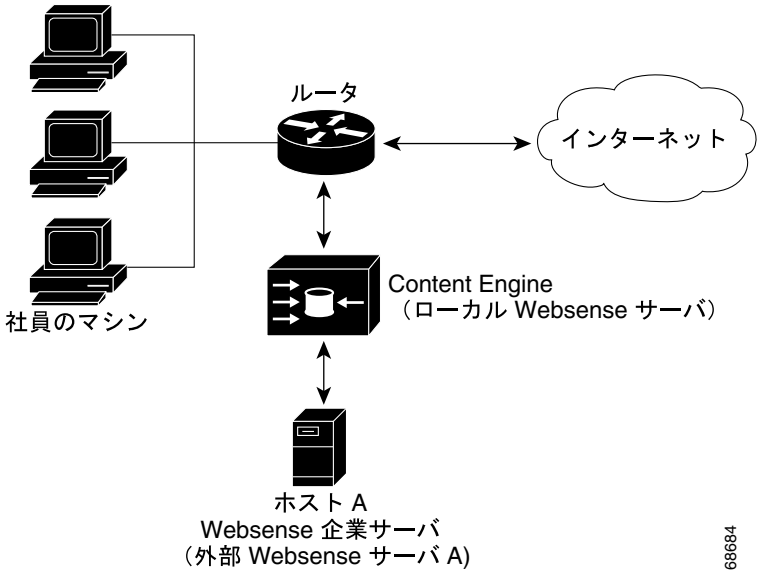
これらの統計情報は、送信された要求、受信された応答、ブロックされたページ、許可されたページ、および障害の数を表示します。さらに詳しい URL フィルタリング統計情報は、N2H2 サーバ上で入手可能です。表示される統計情報をクリアするには、**clear statistics url-filter http N2H2** コマンド、および **clear statistics all** EXEC コマンドを使用します。**clear statistics url-filter http N2H2** EXEC コマンドは、N2H2 サーバの統計カウンタをリセットします。すべての統計カウンタが 0 にリセットされます。

（注）　N2H2 フィルタリング コンフィギュレーションおよびポリシーの詳細は、次の Web サイトを参照してください。http://www.n2h2.com

Websense URL フィルタリングのためのスタンドアロン Content Engine の設定

スタンドアロン Content Engine は、リモート Websense Enterprise サーバをフィルタリング エンジンとして使用し、Websense サーバ上に設定されたフィルタリング ポリシーを実行できます。図 11-2 が示すとおり、リモート Websense サーバはローカル Websense と別のシステム（ホスト A）上で実行され、ネットワーク経由でスタンドアロン Content Engine と通信します。

図 11-2　Websense サーバを使用した URL フィルタリング



統合 Websense サーバを使用するようにスタンドアロン Content Engine を設定することもできます。統合 Websense サーバは Content Engine で稼働する内部サーバです。*ローカル Websense サーバ*ともいいます。

（注）　ACNS 5.1 以前のソフトウェア リリースでは、サポートされている Websense サーバは 1 つのみです。ACNS 5.2.1 以降のソフトウェア リリースでは、最大 2 つの Websense サーバがサポートされています。このトピックに関する詳細は、次の「Websense サーバのフェールオーバーについて」を参照してください。

ACNS 5.4.1 以降のソフトウェア リリースは Websense 5.2.2 ソフトウェアをサポートしています。ACNS 5.3.x ソフトウェアは Websense 5.2.0 ソフトウェアをサポートしています。

Websense サーバのフェールオーバーについて

ACNS 5.2.1 以降のソフトウェア リリースでは、Websense サーバのフェールオーバー機能がサポートされています。この機能を使用すると、フェールオーバーのために最大 2 つの Websense サーバ（プライマリ サーバとセカンダリ サーバが 1 つずつ）を使用するように Content Engine を設定できます。表 11-8 に、サポートされている Websense サーバのフェールオーバー設定を示します。

表 11-8　サポートされている Websense サーバのフェールオーバー設定

サポートされている設定	ローカル（内部）Websense サーバ	リモート Websense サーバ
オプション A	Content Engine でローカル Websense サーバがディセーブルです。	プライマリ Websense サーバは外部ホスト（ホスト A など）で稼働します。 セカンダリ Websense サーバは別の外部ホスト（ホスト B など）で稼働します。

表 11-8　サポートされている Websense サーバのフェールオーバー設定（続き）

サポートされている設定	ローカル（内部）Websense サーバ	リモート Websense サーバ
オプション B	ローカル Websense サーバはプライマリ Websense サーバとして機能します。	セカンダリ Websense サーバは外部ホストで稼働しています。
オプション C	ローカル Websense サーバはセカンダリ Websense サーバとして機能します。	プライマリ Websense サーバは外部ホストで稼働しています。

Websense サーバを設定する順番によって、プライマリ Websense サーバに指定されるサーバが決まります。最初に設定された Websense サーバが、プライマリ サーバに指定されます。セカンダリ Websense サーバの設定は任意です。スタンドアロン Content Engine に Websense サーバのフェールオーバーを設定する方法については、「Websense サーバ フェールオーバーおよび URL フィルタリングの設定例」（p.11-32）を参照してください。

Websense サービス

ACNS 5.4.x 以降のソフトウェア リリースは、Websense 5.2.2 ソフトウェアをサポートしています。Websense 5.2.2 ソフトウェアでは次のサービスがサポートされています。

- ポリシー サーバ
- Employee Internet Management（EIM）サーバ
- ローカル ネットワーク エージェント
- ローカル RADIUS エージェント
- ローカル ディレクトリ エージェント
- ローカル ログオン エージェント（このサポートは、ACNS 5.4.1 ソフトウェア リリースで追加）
- ローカル ユーザ サービス

（注）　*ローカル Websense サーバ*という用語は、Content Engine で内部実行される一連の Websense プロセスを表す場合にも使用されます。これらの Websense プロセスは、*サービス*ともいいます。

Websense 5.2.2 ソフトウェアでサポートされている Websense GUI Manager Version 5.5 および 6.1 では、GUI および CLI コマンドを通して RADIUS および eDirectory エージェントを設定できます。したがって、ACNS 5.4.1 ソフトウェア リリースでは、RADIUS および eDirectory エージェントの設定に関連したすべてのグローバル コンフィギュレーション CLI コマンドが ACNS CLI コマンド セットから削除されています。ただし、RADIUS および eDirectory エージェントをアクティブにする場合に使用する CLI コマンドは維持されています（**websense-server service radius-agent activate** および **websense-server service edirectory-agent activate** グローバル コンフィギュレーション コマンド）。

- RADIUS エージェントの設定に使用する次のグローバル コンフィギュレーション CLI コマンドは、ACNS 5.4.1 ソフトウェア リリースから削除されました。
  - **websense-server service radius-agent incoming** [**auth-***number*] [**accnt-port** *port number*]
  - **websense-server service radius-agent outgoing** [**host** *remote-RADIUS-server IP-address*] [**auth-port** *port number*] [**accnt-port** *port number*]
- eDirectory エージェントの設定に使用する次のグローバル コンフィギュレーション CLI コマンドは、ACNS 5.4.1 ソフトウェア リリースから削除されました。
  - **websense-server service edir-agent edir-server** [**administrative-dn** *administrative-distinguished-name*] [**host** *remote-eDirectory-server IP-address*] [**root-context** *root-context*]

Content Engine で ACNS 5.4.1 以降のソフトウェア リリースが稼働している場合に、上記のグローバル コンフィギュレーション コマンドを入力して RADIUS エージェントまたは eDirectory エージェントを CLI から設定しようとしても、コマンドは無効です。上記の無効コマンドを入力した場合、エラー メッセージは表示されません。

eDirectory 管理パスワードの設定に使用するグローバル コンフィギュレーション コマンドは、ACNS 5.4.1 ソフトウェア リリースで維持されています。

```
ContentEngine(config)# websense-server service edirectory-agent edir-server
administrative-passwd password
```

Websense 5.2.2 ソフトウェアでは、ローカルまたはリモートの Websense Policy Server を使用して、ローカル EIM サーバ、ローカル RADIUS エージェント、ローカル eDirectory エージェント、ローカル ネットワーク エージェント、ローカル ログオン エージェント、およびローカル ユーザ サービスを Content Engine で個別にアクティブにすることができます（表 11-9 を参照）。

表 11-9　ACNS 5.4.1 以降のソフトウェアでサポートされているローカル Websense 5.2.2 サーバのサービス

名前	説明
ポリシー サーバ	外部 Websense Manager の GUI を通して設定したポリシー情報すべてをホストします。ポリシー情報を、ローカル Websense サーバの他のサービス（ローカル EIM サーバ、ローカル ネットワーク エージェント、ローカル ログオン エージェント、およびローカル ユーザ サービス）に送ります。 Content Engine でローカル EIM サーバ、ローカル ネットワーク エージェント、ローカル ログオン エージェント、またはローカル ユーザ サービスをアクティブにするには、ローカル（内部）ポリシー サーバまたは指定されたリモート ポリシー サーバが稼働している必要があります。
ローカル EIM サーバ	プロキシ サーバ、ファイアウォール、キャッシング アプリケーションを使うときに、URL フィルタリング機能を提供します。
ローカル ネットワーク エージェント	HTTP、HTTPS-over-HTTP、FTP-over HTTP 以外のプロトコルを使う要求の URL フィルタリングをイネーブルにします。ローカル ネットワーク エージェントを Content Engine 上でアクティブにすると、そのネットワーク エージェントは次のプロトコルやアプリケーションからの着信要求をフィルタリングできます。 • SQL Net などのデータベース アプリケーション • FTP や Gopher などのファイル転送アプリケーション • Yahoo Messenger や MSN Messenger などのインスタント メッセージングおよびチャット アプリケーション • POP3、SMTP、NetMeeting などのメールと支援ツール • Daytime、finger、NTP、SSH、Telnet などの ネットワーク オペレーティング システム アプリケーション • VNC や pcANYWHERE などのリモート アクセス アプリケーション • RTSP、Windows Media、Liquid Audio などのストリーミング メディア アプリケーション • その他（Network News Transfer Protocol [NNTP] など）

表 11-9　ACNS 5.4.1 以降のソフトウェアでサポートされているローカル Websense 5.2.2 サーバのサービス（続き）

名前	説明
ローカル RADIUS エージェント	外部 RADIUS サーバを通して認証されるユーザに対して、ユーザベースまたはグループベース ポリシーに基づく URL フィルタリングをイネーブルにします。このエージェントは、RADIUS 認証方式を通してネットワークにアクセスしたり、認証を受けるユーザを、トランスペアレントに識別します。Content Engine にこの情報を提供すると、Content Engine はネットワークにリモート アクセスするユーザおよびユーザ グループにポリシーを適用します。 このエージェントは、RADIUS クライアントと外部 RADIUS サーバ間で RADIUS メッセージを転送するプロキシとして機能します。ローカル RADIUS エージェントを適切に機能させるには、Content Engine に RADIUS 設定（外部 RADIUS サーバの IP アドレスなど）を設定する必要があります（「スタンドアロン Content Engine のための RADIUS 認証設定の指定」[p.17-9] を参照）。
ローカル ディレクトリ エージェント	LDAP を通して認証されるユーザに対して、ユーザベースまたはグループベース ポリシーに基づいて URL フィルタリングをイネーブルにします。このエージェントは Novell eDirectory と連携して、LDAP 認証方式を通してネットワークにアクセスしたり、認証を受けるユーザを、トランスペアレントに識別します。Content Engine にこの情報が提供されると、Websense フィルタリング サービスはユーザまたはグループに適用されるポリシーに基づいて要求をフィルタリングすることができます。 このエージェントは LDAP を使用して、Novell eDirectory からユーザ ログイン セッション情報を収集し、この情報に基づいてネットワークにログインしているユーザを認証します。このエージェントは認証された各ユーザと IP アドレスを対応付けます。ローカル eDirectory エージェントは Websense ローカル ユーザ サービスを利用して、この情報を Websense フィルタリング サービスに提供します。このローカル eDirectory エージェントを適切に機能させるには、Websense Manager GUI（バージョン 5.5、または 6.1 以降）を使用して、管理識別名などを設定しておく必要があります。
ローカル ユーザ サービス	ユーザ ベースまたはグループ ベース ポリシーに基づいて URL フィルタリングをイネーブルにします。ユーザ サービスを使用している場合に、Windows NT ディレクトリを使用してユーザ ベースやグループ ベース URL フィルタリングを設定するには、Windows マシン上で外部ユーザ サービスを使用する必要があります。

表 11-9　ACNS 5.4.1 以降のソフトウェアでサポートされているローカル Websense 5.2.2 サーバのサービス（続き）

名前	説明
ローカル ログオン エージェント	Windows クライアント マシンを通してネットワークにログオンしているユーザに対して、ユーザベースまたはグループベース ポリシーに基づく URL フィルタリングをイネーブルにします。このエージェントに対応するログオン アプリケーションは、ユーザがネットワーク内の Windows ドメインにログオンするときに、ログオン セッションをキャプチャします。ログオン エージェントは Websense ユーザ サービスと通信して、Websense にフィルタリング用の最新のユーザ ログオン セッション情報を提供します。 ログオン エージェントは、ドメインにログオンしたユーザをリアルタイムで識別します。DC（ドメイン コントローラ）エージェントは、ドメイン コントローラおよびワークステーションに定期的に問い合わせて、ユーザを識別します。ログオン エージェントはユーザをリアルタイムで識別するため、Websense フィルタリング サービスは特定のユーザ、グループ、ワークステーション、またはネットワークに割り当てられたポリシーに基づいて、インターネット アクセスを正確にフィルタリングできます。 ログオン エージェント ユーザ識別プロセスは、ログオン エージェントが Windows クライアント マシンの自己識別に利用する識別プロセスです。通常、共有ネットワーク ロケーション内のログオン スクリプトはクライアント マシン上で、LogonApp.exe と呼ばれるプロセスを呼び出します。LogonApp.exe には固定と非固定の 2 つのモードがあります。 • 固定モード:LogonApp.exe はクライアント マシンでバックグラウンド タスクとして稼働し、ログオン エージェントにユーザ名とパスワードのペアを定期的に送信します。インターバルは Websense Enterprise Manager の Query Interval（固定モード）設定によって決まります。ユーザがログアウトするときにレジスタにログアウト スクリプトが設定されている場合、LogonApp.exe はログオン エージェントにその時点でログアウト情報を送信します。 • 非固定モード : LogonApp.exe はユーザがログオンしたときに一度ログオン エージェントに問い合わせます。ユーザ ログオン セッションはログオン エージェントのローカル メモリに格納されます。ログオン エージェントのユーザ マップは、Websense Enterprise Manager の Entry Lifetime（非固定モード）設定に従います。 ログオン エージェントは対応するログオン アプリケーション（LogonApp.exe）からログオン セッション情報を取得し、ローカル メモリ内のユーザ マップおよび AuthServer.journal ファイルにユーザ名と IP アドレスのペアを格納します。ログオン セッションを追跡するための重要な要素は、ユーザ名でなく IP アドレスです。同じユーザが複数のワークステーションからネットワーク ドメインにログオンすることがあるためです。AuthServer.journal ファイルのサイズが 1 MB に達するたびに、Logon Agent は内容をハード ディスクの AuthServer.bak ファイルにバックアップします。 eDirectory エージェントなどのログオン エージェントは、ユーザをトランスペアレントに識別します。ログオン エージェントのサポートは、ACNS 5.4.1 ソフトウェア リリースで追加されました。

ACNS 5.4.1 以降のソフトウェア リリースでサポートされている Websense 5.5.2 ソフトウェアには、ログオン エージェントいう追加エージェントが組み込まれています。

表 11-10 に、Content Engine での Websense 5.2.2 ソフトウェア設定に関連する CLI コマンドを示します。

表 11-10　Websense サーバ関連の CLI コマンド

CLI コマンド構文	説明
websense-server service policy local activate	Content Engine 上のローカル ポリシー サーバをアクティブにします。
websense-server service policy remote [**host** *remote-policy-server IP-address*] [**port** *remote-policy-server port-number*]]	Content Engine 上のローカル EIM サーバ、ローカル ネットワーク エージェント、およびローカル ユーザ サービスをアクティブにするために使用されるリモート ポリシー サーバを指定します。デフォルト ポート番号は 55806 です。
websense-server service eim activate	Content Engine 上のローカル EIM サーバをアクティブにします。無効にするには、このコマンドの **no** 形式を使用します。
websense-server service network-agent activate	Content Engine 上のローカル ネットワーク エージェントをアクティブにします。無効にするには、このコマンドの **no** 形式を使用します。
websense-server service user activate	Content Engine 上のローカル ユーザ サービスをアクティブにします。無効にするには、このコマンドの **no** 形式を使用します。
websense-server service radius-agent activate	Content Engine 上のローカル RADIUS エージェントをアクティブにします。無効にするには、このコマンドの **no** 形式を使用します。
websense-server service edir-agent activate	Content Engine 上のローカル eDirectory エージェントをアクティブにします。無効にするには、このコマンドの **no** 形式を使用します。
websense-server service edir-agent edir-server administrative-passwd password	Content Engine が外部 eDirectory サーバと通信して、データベース検索を要求する際に使用する管理パスワードを指定します。 次に、**administrative-passwd** コマンド オプションを使用して、管理パスワードとして default244 を指定する例を示します。 `ContentEngine(config)# websense-server service edir-agent edir-server administrative-passwd default244`
websense-server service logon-agent activate	Content Engine 上のローカル ログオン エージェントをアクティブにします。無効にするには、このコマンドの **no** 形式を使用します。このコマンドは、ACNS 5.4.1 ソフトウェア リリースで追加されました。

（注）　ACNS 5.2 ソフトウェアでは、**websense-server ip-address** および **websense-server user-server external** グローバル コンフィギュレーション コマンドは推奨しません。

ACNS 5.2.1 以降のソフトウェア リリースでは、URL フィルタリングに Websense サーバを 2 つまで使用するように Content Engine を設定できます。

Content Engine に Websense URL フィルタリングを設定するには、URL フィルタリングに使用する Websense サーバ設定のタイプを決定します。

- 2 つの Websense サーバ（ローカル Websense サーバと外部 Websense サーバ、または 2 つの外部 Websense サーバ）を使用する場合は、「Websense サーバ フェールオーバーおよび URL フィルタリングの設定例」（p.11-32）を参照してください。
- ローカル Websense サーバのみを使用する場合は、「ローカル Websense サーバへの URL フィルタリングの設定」（p.11-36）を参照してください。
- 外部 Websense サーバのみを使用する場合は、「外部 Websense サーバを使用した Websense URL フィルタリングの設定」（p.11-39）を参照してください。

URL フィルタリング方式は、プロトコルごとに一度に 1 つのみアクティブにできます。HTTP を介した要求に対して Websense URL フィルタリングをイネーブルにするには、プロトコルごとに他の URL フィルタリング方式が設定されていないことを確認してください。HTTP 要求（HTTP、FTP-over-HTTP、HTTPS-over-HTTP）のために Content Engine 上で現在イネーブル化されている URL フィルタリング方式を表示するには、the **show url-filter http** EXEC コマンドを使用します。Websense サーバを使用した各プロトコルでサポートされている URL フィルタリングのリストについては、表 B-4 を参照してください。

（注）　ACNS 5.4.x 以降のソフトウェア リリースでは、Websense サーバ バージョン 5.2.2 がすべての Cisco Content Engine プラットフォームでサポートされています。Websense ソフトウェア設定についての詳細は、Web サイト（http://www.websense.com）を参照してください。

Websense サーバは、Content Engine 上の /local/local1/WebsenseEnterprise/EIM ディレクトリに常駐する Websense サーバのイメージを提供します。コンフィギュレーション ファイルおよびロギング ファイルとともに、すべての実行ファイルがこのディレクトリに保存されています。

ACNS 5.4.1 以降のソフトウェア リリースでは、Websense インストレーション出力はファイル ws_history.log に記録されます。この特定のログ ファイルは、Content Engine の /local1/logs/urlfilter/websense ディレクトリ内にあります。

ANCS 5.4.1 以降のソフトウェア リリースでは、**clear websense** EXEC コマンドを入力して、既存の Websense 設定を削除できます。このコマンドは、スタンドアロン Content Engine から破損した Websense 設定を削除する場合に便利です。

Websense サーバがイネーブルであり、Websense URL データベースが Content Engine に初めてダウンロードされるときは、CPU の使用率が上昇します。したがって、Websense サーバはオフピーク時、またはネットワーク トラフィックが少ないときにイネーブルにしてください。それ以外の場合に Websense サーバをイネーブルにすると、Content Engine 上のその他の処理に影響を及ぼすことがあります。Websense プロセスの 1 つが停止すると、ローカル Websense サーバが Content Engine 上で自動的に再起動します。

Explorer、Manager、および Reporter などの Websense コンポーネントをダウンロードするか、またはスタンドアロン Content Engine 上で実行される ローカル Websense サーバと一緒に使用するための評価キーを入手するには、次の URL にアクセスし、そこで示される手順を順に実行します。

http://www.websense.com/downloads

Websense サーバ用のポート設定

Websense のプロセスでは、Content Engine の内部プロセスから、または Websense Manager などの外部プロセスからの接続に対して、次の 4 つのポートをオープンにしておく必要があります（表 11-11 を参照）。

表 11-11　Websense サーバ用のポート設定

ポート	説明	デフォルト
Websense サーバ ポート	これは、Websense プロトコルに従って、コンテンツのフィルタリング要求を受信する TCP ポートです。	15868
ブロック メッセージ サーバ ポート	Websense プロセスによってある URL がブロックされる場合、このポートはリダイレクト URL をユーザに送信します。リダイレクト URL は、ブロックされたページおよびポリシーをユーザに出力します。Websense プロセスは、このポート上で待ち受けし、Websense サーバ内のスレッドによってブロックされサービスされるページを受信します。このスレッドは、リダイレクトされた要求に応答して、ブロックされたページを送信します。	15871
診断サーバ ポート	Websense サーバには、ユーザが Websense プロセス内での問題の診断をリモートで実行できる完全な診断セットがあります。このポートは、このような診断ユーティリティが接続するポートです。	15869
Websense 設定サーバ ポート	Websense GUI Manager の接続先となる Websense ポリシー サーバのポートです。このポート用の websense.ini ファイルにはデフォルト エントリはありません。このデフォルト状態を変更しないことを推奨します。	55806
ログオン エージェント ポート	Content Engine で稼働するログオン エージェントが、Windows クライアント マシンで稼働する LogonApp.exe からの通信を待ち受けるポートです。ログオン エージェントはこのポートでユーザ名、ハッシュ化パスワード、およびクライアント マシンの IP アドレスを受信します。ログオン エージェントのサポートは、ACNS 5.4.1 ソフトウェア リリースで追加されました。	15880

表 11-11 の最初の 3 つのポートは、スタンドアロン Content Engine 上の /local1/WebsenseEnterprise/EIM ディレクトリにある eimserver.ini ファイルを修正することによって、設定できます。Websense サーバが新たに設定されたポートを認識できるように、Websense サーバを再起動する必要があります。

これらのポートを修正するには、Content Engine 上の /local1/WebsenseEnterprise/EIM ディレクトリから、FTP を使用して eimserver.ini ファイルのコピーをエクスポートし、このファイルを修正してから、eimserver.ini ファイルを Content Engine 上で削除し、次に修正されたファイルを FTP を使用して Content Engine に送り返します。

（注）　新しく設定されたポートを有効にするには、Websense サーバをディセーブルにし、再度イネーブルにする必要があります。ローカル Websense サーバをディセーブルにするには、**no websense-server enable** グローバル コンフィギュレーション コマンドを使用します。Websense クライアントが正しい Websense サーバ ポートをポイントとするように設定するには、必ず **url-filter http websense server** 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ください。**url-filter http websense server** コマンドの詳細は、『*Cisco ACNS Software Command Reference*』Release 5.5 を参照してください。

ACNS 5.4.x ソフトウェアにアップグレードする場合の Websense の問題点

ACNS 5.3.x ソフトウェアから ACNS 5.4.x ソフトウェアにアップロードすると、Websense バイナリは RADIUS および eDirectory 設定をサポートしなくなります。したがって、RADIUS および eDirectory エージェントの設定に使用するグローバル コンフィギュレーション コマンドは、ACNS 5.4.1 ソフトウェア リリースから削除されています。

CLI コンフィギュレーション コマンドを使用して、ACNS 5.3.x ソフトウェアが稼働する Content Engine に RADIUS および eDirectory エージェントを設定した場合に、ACNS 5.3.x ソフトウェアを ACNS 5.4.x ソフトウェアにアップグレードしても、wsradius.ini および wsedire.init ファイルに格納されている RADIUS および eDirectory エージェントの設定は維持されます。ただし、Websense GUI Manager を通して RADIUS および eDirectory エージェントに行った設定変更は、これらの 2 つの .ini ファイルに反映されません。

（注）　RADIUS および eDirectory エージェントをアクティブにするために使用される CLI コマンド（**websense-server service radius-agent activate** および **websense-server service edirectory-agent activate** グローバル コンフィギュレーション コマンド）および eDirectory 管理パスワードを指定するために使用される CLI コマンド（**websense-server service edirectory-agent edir-server administrative-passwd** *password* グローバル コンフィギュレーション コマンド）は、ACNS 5.4.x ソフトウェアで維持されています。

ACNS 5.2.x ソフトウェアを ACNS 5.4.x ソフトウェアにアップグレードする場合は、以前の Websense 設定を変更する必要がありません。RADIUS および eDirectory エージェントは ACNS 5.2.x ソフトウェア リリースでサポートされていないためです。

古いバージョンの ACNS ソフトウェアにダウングレードする場合の Websense の問題点

ACNS 5.4.x ソフトウェアを ACNS 5.3. ソフトウェアにダウングレードすると、ローカル WebsenseEnterprise ディレクトリおよびその他の関連 Websense ファイルが削除されます。Websense で使用されていた既存の内部コンフィギュレーション ファイルはすべて削除され、ダウングレードの前に行ったすべての変更は失われます。

次のようなエラー メッセージが表示されて、Websense のダウングレードに関する問題点が示されます。

```
WARNING:
Websense does not support downgrade
Hence removing /local/local1/WebsenseEnterprise
Websense will stop working after copy ftp install
```

（注）　以前のリリースでは、ソフトウェアは WebsenseEnterprise ディレクトリが削除されたことを示すエラー メッセージを生成しませんでした。

ダウングレード後に Content Engine をリロードすると、Websense が再インストールされ、内部コンフィギュレーション ファイルが再作成されます。Content Engine Websense サーバの設定（**websense-server service policy local activate** および **websense-server service eim activate** など）が再開され、ダウングレード前のスタートアップ コンフィギュレーションに格納されます。

ACNS 5.4.x ソフトウェアを ACNS 5.3. ソフトウェアにダウングレードした場合、スタートアップ コンフィギュレーションはそのまま残ります。既存のコンフィギュレーション ファイルはすべて削除され、フレッシュ インストールが実行されます。以前に行った変更はすべて失われます。Content Engine の再起動後に、スタートアップ コンフィギュレーションから CLI 設定が取得されます。

ACNS 5.4.x ソフトウェアを ACNS 5.2.x または 5.1.x ソフトウェアにダウングレードした場合、Content Engine に設定されていた CLI コマンドおよび Websense コンフィギュレーション ファイルはそのまま残ります。

ローカル（内部）Websense サーバが Content Engine でイネーブル化されている場合に、ACNS 5.2.x ソフトウェアを ACNS 5.0 ソフトウェアまたは ACNS 5.1 ソフトウェアにダウングレードすると、Content Engine から Websense Enterprise ディレクトリが削除され、ローカル Websense サーバが動作を停止します。ACNS 5.2.x 以降のソフトウェア リリースを ACNS 5.1 ソフトウェアまたは ACNS 5.0 ソフトウェアにダウングレードする場合に、この問題を回避する手順は、次のとおりです。

ステップ 1　Content Engine 上のローカル（内部）Websense サーバをディセーブルにします。

ステップ 2　Content Engine 上で Websense サービスをディセーブルにします。

ステップ 3　ACNS 5.1 ソフトウェアまたは ACNS 5.0 ソフトウェア ダウングレード イメージを Content Engine にインストールします。

Websense サーバ フェールオーバーおよび URL フィルタリングの設定例

次の例では、Content Engine は URL フィルタリング用の HTTP プロキシとして機能します。まず、ローカルまたはリモート ポリシー サーバを使用するように Content Engine が設定され、その後、ローカル Websense サーバ サービス（ローカル EIM サーバ、ローカル ユーザ サービス、およびローカル ネットワーク エージェント）が Content Engine でアクティブになります。

次に、ローカル（内部）Websense サーバをプライマリ Websense サーバに、外部 Websense サーバをセカンダリ Websense サーバに使用するように、Content Engine が設定されます。プライマリ Websense サーバを使用できない場合、Content Engine はこのセカンダリ サーバにフィルタリング要求を送信します。

Content Engine で許可モードが再イネーブルになると、URL フィルタリングが Content Engine 上でイネーブルになります。Websense Manager GUI を使用して、ローカルおよびリモート Websense サーバにデフォルト ポリシーが設定されて、その後、HTTP プロキシが Content Engine 上でイネーブルになります。

Websense サーバのフェールオーバーおよび URL フィルタリングを設定する手順は、次のとおりです。

ステップ 1　Content Engine 上にある個々の Websense サービスをアクティブにするために、ローカル Websense ポリシー サーバを使用するか、またはリモート Websense ポリシー サーバを使用するかを指定します。

- ローカル ポリシー サーバを使用するには、次のように Content Engine でローカル ポリシー サーバをアクティブにします。

```
ContentEngine(config)# websense-server service policy local activate
```

- リモート ポリシー サーバを使用するには、次のように Content Engine でリモート ポリシー サーバに関する必須情報を設定します（ホスト名、IP アドレス、ポート番号など）。

  ```
  ContentEngine(config)# websense-server service policy remote host {hostname|
  IP address} [port policy-server-port]
  ```

  ここで各パラメータの意味は、次のとおりです。

  - *hostname* または *IP address* はリモート ポリシー サーバのホスト名または IP アドレスです。
  - ポート番号は任意です。デフォルト ポート番号は 55806 です。

Content Engine 上のローカル Websense サーバ のサービス（ローカル EIM サーバ、ローカル ネットワーク エージェント、ローカル eDirectory エージェント、ローカル RADIUS エージェント、およびローカル ユーザ サービス）を起動する前に、ローカルまたはリモート ポリシー サーバのどちらかがアクティブになっている必要があります。ローカルとリモート ポリシー サーバの設定は、相互に排他的です。

ステップ 2　Content Engine 上のローカル EIM サーバをアクティブにします。

```
ContentEngine(config)# websense-server service eim activate
```

ステップ 3　Content Engine 上のローカル ユーザ サービスをアクティブにします。

```
ContentEngine(config)# websense-server service user activate
```

ステップ 4　Content Engine 上のローカル ネットワーク エージェントをアクティブにします。

```
ContentEngine(config)# websense-server service network-agent activate
```

ステップ 5　Content Engine 上のローカル eDirectory エージェントをアクティブにします。

```
ContentEngine(config)# websense-server service edir-agent activate
```

ステップ 6　Content Engine 上のローカル RADIUS エージェントをアクティブにします。

```
ContentEngine(config)# websense-server service radius-agent activate
```

ステップ 7　Content Engine 上のローカル ログオン エージェントをアクティブにします。

```
ContentEngine(config)# websense-server service logon-agent activate
```

ステップ 8　Content Engine でアクティブにしたローカル Websense サーバのサービス（ローカル EIM サーバ、ローカル ネットワーク エージェント、ローカル ログオン エージェント、およびローカル ユーザ サービス）をすべてイネーブルにします。

```
ContentEngine(config)# websense-server enable
```

（注） デフォルトでは、ローカル EIM サーバ、ローカル ネットワーク エージェント、ローカル ログオン エージェント、およびローカル ユーザ サービスから構成されるローカル Websense サーバは、Content Engine 上でディセーブルです。スタンドアロン Content Engine のクラスタでローカル Websense サーバを使う場合は、各スタンドアロン Content Engine 上のローカル Websense サーバをイネーブルにしてください（Content Engine クラスタ内の各 Content Engine で **websense-server enable** 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）。

ステップ 9 **url-filter http websense server local** 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、ローカル Websense サーバをプライマリ Websense サーバに使用するように Content Engine を設定します。

（注） **url-filter http websense server** 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、プライマリとセカンダリの Websense サーバで異なる設定値（タイムアウト、ポート番号、接続数など）を設定することもできます。デフォルトでは、Content Engine（HTTP プロキシとして機能）はポート 15868 で Websense サーバにフィルタリング要求を送信し、Websense サーバからの応答を 20 秒間待機してから接続をタイムアウトし、CPU ごとに 40 の固定接続を確立します。

この例では、Content Engine（HTTP プロキシとして機能）はポート 4005 でローカル Websense サーバにフィルタリング要求を送信し、Websense サーバからの応答を 60 秒間待機してから接続をタイムアウトし、このローカル Websense サーバとの 90 の固定接続を確立します。ローカル Websense サーバが先に設定されるため、このサーバが Content Engine のプライマリ Websense サーバに指定されます。

```
ContentEngine(config)# url-filter http websense server local port 4005 timeout 60
connections 90
```

（注） ローカル Websense サーバの IP アドレスは変更できず、127.0.0.1 に設定されています。

ステップ 10 **url-filter http websense server** 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、既存の Websense サーバをセカンダリ Websense サーバに使用するように Content Engine を設定します。

ローカル Websense サーバがすでにプライマリ Websense サーバになっているため、外部 Websense サーバはセカンダリ Websense サーバとして指定する必要があります。

この例では、IP アドレスが 172.18.22.10 の外部 Websense サーバがセカンダリ Websense サーバとして設定されています。ローカル Websense サーバを使用できない場合、Content Engine は要求をポート 4006 からこのセカンダリ Websense サーバに送信し、このサーバからの応答を 90 秒間待機してから接続をタイムアウトし、CPU ごとに 90 の固定接続を確立します。

```
ContentEngine(config)# url-filter http websense server 172.18.22.10 port 4006
timeout 90
```

ステップ 11 デフォルトでは、許可モードがイネーブルです。許可モードを再イネーブルにするには、次のコマンドを入力します。

```
ContentEngine(config)# no url-filter http websense allowmode enable
```

プライマリ Websense サーバを使用できない場合、Content Engine は指定されたセカンダリ サーバに要求を送信します。プライマリとセカンダリの両方の Websense サーバを使用できない場合、要求は許可モードに送信されます。

- 許可モードがイネーブルの場合、Content Engine は Websense サーバからの応答を受信できないときでも、すべての HTTP トラフィックの通過を許可します（通常のトラフィック処理を継続します）。
- 許可モードがディセーブルの場合、Websense サーバからの応答を受信できなければ、Content Engine は通過する HTTP トラフィックをすべてブロックします。

Websense サーバがイネーブルでもディセーブルであっても、**allowmode** オプションは設定可能で、Websense サーバの設定とは独立しています。Websense URL フィルタリングがすでに使用中であっても、Content Engine は **allowmode** の新しい設定を受け入れます。

ステップ 12 Content Engine 上の URL フィルタリングをイネーブルにします。

```
ContentEngine(config)# url-filter http websense enable
```

ステップ 13 Websense Manager GUI を使用してデフォルト ポリシーを設定します。このステップは、ローカルとリモートの両方の Websense サーバに実行する必要があります。

a. Websense Manager GUI を使用してポリシー サーバを追加します。

- Websense Manager メイン ウィンドウの左側ペインを右クリックします。
- **Add Policy Server** を選択します。
- 表示されたダイアログ　ボックスに、ローカル（内部）Websense　サーバが稼働している Content Engine の IP アドレスを入力します。

b. Content Engine で稼働している Websense Policy ポリシー サーバに接続します。

- 左側ペインで、ポリシー サーバ（Content Engine の IP アドレスなど）をダブルクリックします。
- ユーザ名とパスワードを入力し、**OK** をクリックします。

c. Websense Manager GUI を使用して、Websense ポリシーを設定します。

- Websense Manager GUI を使用して、Websense ポリシー サーバを接続します。
- 左側ペインで、**Filter Definition** をダブルクリックしてから、**Policies** をクリックします。
- Choose **Global** を選択します。
- 右側ペインで、**Edit** ボタンをクリックします。
- 表示されたダイアログボックスで、デフォルト設定（default　settings）、基本設定（basic settings）、常にブロック（always block）、ブロックしない（never block）などのカテゴリ設定を適用します。デフォルト ポリシーはグローバル、デフォルト カテゴリはデフォルト設定です。

（注） Websense Enterprise Manager ウィンドウで **Save Changes** ボタンをクリックしても、デバイス再起動後に行った Websense 設定変更は保存されません。再起動後の Websense 設定変更を保存するには、**write memory** コマンドを使用する必要があります。Websense Manager GUI の使用方法については、次の Web サイトを参照してください。
http://www.websense.com

ステップ 14 Content Engine に HTTP プロキシを設定します。

```
ContentEngine(config)# http proxy incoming 8080
```

ステップ 15 プライマリおよびセカンダリ Websense サーバの統計情報を表示します。

```
ContentEngine# show statistics url-filter http websense
```

ローカル Websense サーバへの URL フィルタリングの設定

URL フィルタリングでローカル（内部）Websense サーバを使用するように Content Engine を設定するには、次のタスクを実行する必要があります。

1. ローカルまたはリモート ポリシー サーバを使用して、Content Engine 上のローカル Websense サーバ サービス（ローカル EIM サーバ、ローカル ネットワーク エージェント、ローカル ログオン エージェント、ローカル ユーザ サービス）をアクティブにします。
2. Content Engine 上のローカル Websense サーバをイネーブルにします。デフォルトでは、ディセーブルです。
3. HTTP 要求（HTTP、FTP-over-HTTP、HTTPS-over-HTTP 要求）の URL フィルタリングにローカル Websense サーバを使用するように Content Engine を設定します。ネットワーク エージェントが設定されている場合は、その他のプロトコルもフィルタリングできます。

ACNS 5.2.1 以降のソフトウェア のリリースでは、フェールオーバー用に Websense サーバを 2 つまで設定できます。これらの Websense サーバの 1 つをローカル Websense サーバにすることができます。Websense サーバを設定する順番によって、プライマリ サーバになるサーバが決まります。最初に設定した Websense サーバが自動的にプライマリ Websense サーバになり、2 番めに設定したサーバがセカンダリ Websense サーバになります。サポートされている設定のリストについては、表 11-8 を参照してください。

ACNS 5.3.1 以降のソフトウェア リリースでは、ローカル Websense サーバ サービスを自由な組み合わせ（表 11-9 を参照）でアクティブにできます。ローカル ポリシー サーバが Content Engine 上でアクティブになっていない場合、他のローカル Websense サーバ サービス（ローカル EIM サーバ、ローカル ネットワーク エージェント、ローカル RADIUS エージェント、ローカル eDirectory エージェント、およびローカル ユーザ サービス）をアクティブにするときは、有効な外部ポリシー サーバを指定する必要があります。詳細は、「外部 Websense サーバを使用した Websense URL フィルタリングの設定」（p.11-39）を参照してください。

ACNS 5.0.3 ～ 5.1.x ソフトウェアには、ローカル Websense サーバが含まれています。これらのソフトウェア リリースでは Websense サーバのサービスを個別にアクティブにすることはできません。Content Engine の Websense サーバ サービスは、デフォルトで次のようにアクティブになります。

- ACNS 5.3.x ソフトウェア リリースから ACNS 5.4.x にアップグレードした場合は、ローカル ポリシー サーバ、ローカル EIM サーバ、ローカル eDirectory エージェント、ローカル RADIUS エージェント、およびローカル ユーザ サービスのみがアクティブになります（ACNS 5.4.1 ソフトウェア リリースへのアップグレードの場合、ローカル ログオン エージェントはアクティブになりません）。
- ACNS 5.0.3、5.1.x、または 5.2.x ソフトウェア リリースから ACNS 5.3.x にアップグレードした場合は、ローカル ポリシー サーバ、ローカル EIM サーバ、およびローカル ユーザ サービスのみがアクティブになります（ACNS 5.3.1 ソフトウェア リリースへのアップグレードの場合、ローカル eDirectory エージェント、ローカル RADIUS エージェントはアクティブになりません）。
- ACNS 5.0.3 または 5.1.x ソフトウェア リリースから ACNS 5.2.1 以降のソフトウェア リリースにアップグレードした場合は、Content Engine 上でローカル ポリシー サーバ、ローカル EMI サーバ、およびローカル ユーザ サービスの 3 つのローカル Websense サーバ サービスがアクティブになります。

URL フィルタリングにローカル（内部）Websense サーバを使用するように Content Engine を設定する手順は、次のとおりです。

ステップ 1　Content Engine 上にあるローカル Websense サーバの個々のサービスをアクティブにするために、ローカル ポリシー サーバとリモート ポリシー サーバのどちらを使うか指定します。

- ローカル ポリシー サーバを使用するには、Content Engine でローカル ポリシー サーバをアクティブにします。

```
ContentEngine(config)# websense-server service policy local activate
```

- リモート ポリシー サーバを使用するには、次のように Content Engine でリモート ポリシー サーバに関する必須情報を設定します（ホスト名、IP アドレス、ポート番号など）。

```
ContentEngine(config)# websense-server service policy remote host {hostname|
IP address} [port policy-server-port]
```

ここで各パラメータの意味は、次のとおりです。

- *hostname* または *IP address* はリモート ポリシー サーバのホスト名または IP アドレスです。
- ポート番号は任意です。デフォルト ポート番号は 55806 です。

（注） Content Engine 上のローカル Websense サーバ のサービス（ローカル EIM サーバ、ローカル ネットワーク エージェント、ローカル RADIUS エージェント、ローカル eDirectory エージェント、ローカル ログオン エージェント、およびローカル ユーザ サービス）を起動する前に、ローカルまたはリモート ポリシー サーバのどちらかがアクティブになっている必要があります。ローカルとリモート ポリシー サーバの設定は、相互に排他的です。

ステップ 2　Content Engine 上のローカル EIM サーバをアクティブにします。

```
ContentEngine(config)# websense-server service eim activate
```

ステップ 3　Content Engine 上のローカル ユーザ サービスをアクティブにします。

```
ContentEngine(config)# websense-server service user activate
```

ステップ 4　Content Engine 上のローカル ネットワーク エージェントをアクティブにします。

```
ContentEngine(config)# websense-server service network-agent activate
```

ステップ 5　Websense Manager GUI（バージョン 5.5、またはバージョン 6.1 以降）を使用して、Content Engine で稼働するローカル eDirectory エージェントの設定値を設定します。

ステップ 6　Websense Manager GUI（バージョン 5.5、またはバージョン 6.1 以降）を使用して、Content Engine で稼働するローカル RADIUS エージェントの設定値を設定します。

ステップ 7　Content Engine 上のローカル eDirectory エージェントをアクティブにします。

```
ContentEngine(config)# websense-server service edir-agent activate
```

ステップ 8　Content Engine 上のローカル RADIUS エージェントをアクティブにします。

```
ContentEngine(config)# websense-server service radius-agent activate
```

ステップ 9　Content Engine でアクティブにしたローカル Websense サーバのサービス（ローカル EIM サーバ、ローカル ネットワーク エージェント、ローカル ログオン エージェント、およびローカル ユーザ サービス）をすべてイネーブルにします。

```
ContentEngine(config)# websense-server enable
```

（注）　デフォルトでは、ローカル EIM サーバ、ローカル ネットワーク エージェント、およびローカル ユーザ サービスから構成されるローカル Websense サーバは、Content Engine 上でディセーブルです。ローカル Websense サーバの IP アドレスは変更できず、127.0.0.1 に設定されています。スタンドアロン Content Engine のクラスタでローカル Websense サーバを使用する場合は、各スタンドアロン Content Engine 上のローカル Websense サーバをイネーブルにしてください（たとえば、Content Engine クラスタ内の各 Content Engine で **websense-server enable** 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）。

ステップ 10　HTTP 要求（HTTP、FTP-over-HTTP、HTTPS-over-HTTP 要求）の URL フィルタリングにローカル Websense サーバを使用するように Content Engine を設定します。Websense サーバを使用した各プロトコルでサポートされている URL フィルタリングのリストについては、表 B-4 を参照してください。

```
ContentEngine(config)# url-filter http websense server local [port portnumber]
[timeout seconds] [connections connections]
```

ここで各パラメータの意味は、次のとおりです。

- **local** は、Content Engine が URL フィルタリングに内部 Websense サーバを使用するように指定します。
- *port number* は、フィルタリングする HTTP 要求を待ち受けるローカル Websense サーバのポート（1 ～ 65535）を指定します。デフォルトでは、ローカル Websense サーバはポート 15868 で待ち受けます。
- *seconds* は、接続がタイムアウトするまで Content Engine が内部 Websense サーバからの HTTP 応答を待つ秒数（0 ～ 240）です。デフォルトは 20 秒です。

- *connections* は、CPU あたりの固定接続数（1 ～ 250）です（デフォルトは CPU ごとに 40）。内部 Websense サーバとの固定接続数を設定するには、このオプションを使用します。異なる値が要求されているという確信がないかぎり、デフォルト数を変更しないでください。

ステップ 11　HTTP 要求の Websense URL フィルタリングをイネーブルにします。

```
ContentEngine(config)# url-filter http websense enable
```

ステップ 12　現在の Websense サーバ設定を表示します。

```
ContentEngine# show websense-server
```

Content Engine 上にあるローカル Websense サーバの 1 つまたは複数のサービスを無効にする方法については、「スタンドアロン Content Engine のローカル Websense サーバ サービスの無効化（p.11-41）」を参照してください。

外部 Websense サーバを使用した Websense URL フィルタリングの設定

外部の Websense サーバを使用するように Content Engine を設定する場合は、そのサーバの IP アドレスおよびポート番号を指定する必要があります。指定した IP アドレスおよびポート番号と、外部 Websense サーバの IP アドレス、および外部 Websense サーバがフィルタリング要求を待ち受けるポートが一致する必要があります。一致しない場合、Content Engine はすべての HTTP 要求（HTTP、FTP-over-HTTP、または HTTPS-over-HTTP 要求）をタイムアウトし、**allowmode** オプション設定に基づいて、すべての HTTP トラフィックをブロックまたは許可します。デフォルトでは、Content Engine 上で許可モードがイネーブルです。許可モードがイネーブルの場合、外部 Websense サーバが応答しない場合、Content Engine はクライアントからの HTTP 要求を許可します。許可モードがディセーブルの場合、再度イネーブルにするには、**url-filter http websense allowmode enable** コマンドを使用します。

ACNS 5.2.1 以降のソフトウェア のリリースでは、フェールオーバー用に Websense サーバを 2 つまで設定できます。Websense サーバを設定する順番によって、プライマリ サーバになるサーバが決まります。最初に設定した Websense サーバが自動的にプライマリ Websense サーバになり、2 番めに設定したサーバがセカンダリ Websense サーバになります。サポートされている Websense サーバ設定については、表 11-8 を参照してください。

URL フィルタリングに外部の Websense サーバを使用するようにスタンドアロン Content Engine を設定する手順は、次のとおりです。

ステップ 1　**url-filter http websense server** 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、外部 Websense サーバに関する必須情報を指定します。

url-filter http websense server {[*hostname* | *ip-address*]} [**port** *portnum* [**timeout** *seconds* [**connections** *connection*]]

ここで各パラメータの意味は、次のとおりです。

- *hostname* は 外部 Websense サーバのホスト名です。
- *IP address* は 外部 Websense サーバの IP アドレスです。
- *portnum* は、Content Engine が HTTP 要求を送信する外部 Websense サーバのポート番号（1 ～ 65535）です。デフォルトのポートは 15868 です。

- *seconds* は、接続がタイムアウトするまで Content Engine が外部 Websense サーバからの HTTP 応答を待つ秒数（0 ～ 240）です。デフォルトは 20 秒です。
- *connections* は、CPU あたりの固定接続数（1 ～ 250）です（デフォルトは CPU ごとに 40）。外部 Websense サーバとの固定接続数を設定するには、このオプションを使用します。異なる値が要求されているという確信がないかぎり、デフォルト数を変更しないでください。

次に、ホスト A で稼働している、IP アドレスが 172.18.22.10 の外部 Websense サーバを指定するように、スタンドアロン Content Engine を設定する方法を示します。Content Engine はポート 4006 から外部 Websense サーバに要求を送信し、接続がタイムアウトするまでサーバからの応答を 90 秒間待つように設定されています。

```
ContentEngine(config)# url-filter http websense server 172.18.22.10 port 4006 timeout 90
```

（注）　スタンドアロン Content Engine のクラスタで、URL フィルタリングに外部 Websense サーバを使用するには、Content Engine クラスタ内の各 Content Engine 上で **url-filter http websense server** グローバル コンフィギュレーション コマンドを使用し、すべてのトラフィックが確実にフィルタリングされるようにしてください。

ステップ 2　フェールオーバーのためにセカンダリ Websense サーバを設定する場合は、次のいずれかの操作を実行します。

- ローカル（内部）Websense サーバをセカンダリ Websense サーバとして設定するには、**url-filter http websense server local** 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。ローカル Websense サーバの設定方法については、「ローカル Websense サーバへの URL フィルタリングの設定」（p.11-36）を参照してください。
- プライマリ Websense サーバ（ホスト A）と異なるホスト（ホスト B）で稼働する外部 Websense サーバを設定するには、**url-filter http websense server** コマンドを使用します。今回は、このコマンドで、セカンダリ Websense サーバのパラメータ（ホスト B で稼働する Websense サーバの IP アドレス、ポート番号、タイムアウト、接続数など）を指定する必要があります。

ステップ 3　この Content Engine で現在の HTTP URL フィルタリング方式として Websense をイネーブルにします。

```
ContentEngine(config)# url-filter http websense enable
```

（注）　外部 Websense サーバ設定についての詳細は、次の Web サイトを参照してください。
http://www.websense.com

ステップ 4　現在の Websense サーバ設定を表示します。

```
ContentEngine# show websense-server
```

スタンドアロン Content Engine のローカル Websense サーバ サービスの無効化

スタンドアロン Content Engine 上にあるローカル Websense サーバのサービス（ローカル EIM サーバ、ローカル ネットワーク エージェント、ローカル RADIUS エージェント、ローカル eDirectory エージェント、ローカル ユーザ サービス、またはローカル ポリシー サーバ）を 1 つ以上無効にする手順は、次のとおりです。

ステップ 1　Content Engine 上でローカル Websense サーバが現在イネーブルになっているかどうかを確認します。

```
ContentEngine# show websense-server
```

ステップ 2　Content Engine 上でローカル Websense サーバが現在イネーブルになっている場合は、ディセーブルにします。

```
ContentEngine(config)# no websense-server enable
```

ステップ 3　**websense-server service** グローバル コンフィギュレーション コマンドの **no** 形式を使用して、ローカル Websense サーバの特定のサービスを非アクティブにします。

たとえば、**no websense-server service network-agent activate** を使用して、Content Engine 上のローカル ネットワーク エージェントを停止します。

```
ContentEngine(config)# no websense-server service eim activate
ContentEngine(config)# no websense-server service user activate
ContentEngine(config)# no websense-server service network-agent activate
ContentEngine(config)# no websense-server service edir-agent activate
ContentEngine(config)# no websense-server service radius-agent activate
ContentEngine(config)# no websense-server service logon-agent activate
```

Content Engine 上でローカル EMI サーバ、ローカル ネットワーク エージェント、ローカル eDirectory エージェント、ローカル RADIUS エージェント、ローカル ログオン エージェント、またはローカル ユーザ サービスを非アクティブにする順番は関係ありません。ただし、ポリシー サーバが Content Engine 上で動作している（リモート ポリシー サーバではなくローカル ポリシー サーバが使用されている）場合は、ポリシー サーバが最後に停止するローカル Websense サービスになります（**no websense-server service policy activate** コマンドを使用します）。

逆に、リモート ポリシー サーバを使用している場合は、その動作を確認してから、Content Engine 上でローカル EIM サーバ、ローカル ネットワーク エージェント、ローカル eDirectory エージェント、ローカル RADIUS エージェント、またはローカル ユーザ サービスを停止してください（**no websense-server service** *service-name* **activate** コマンドを使用します）。

ステップ 4　Content Engine 上で使用されているローカル ポリシー サーバを非アクティブにします。

```
ContentEngine(config)# no websense-server service policy
```

ステップ 5　Content Engine 上で使用されているリモート ポリシー サーバの設定を解除します。

```
ContentEngine(config)# no websense-server service policy remote host
```

Websense コンフィギュレーション ファイルの保存

ACNS 5.2.1 以降のソフトウェア リリースでは、**write memory** コマンドは、ディスクの再設定や ACNS ソフトウェア リリースのアップグレード後に変更された Websense コンフィギュレーション ファイル（eimserver.ini、config.xml、および websense.ini ファイルと Blockpages ディレクトリ）を保存します。

websense.ini ファイルの変更、および Websense URL フィルタリングの設定変更を含む、最新の設定変更を保存するには、**write memory** コマンドを実行する必要があります。**write memory** コマンドを使用すると、外部の Websense Manager GUI から加えた変更を、ディスクの再設定およびアップグレード後に（アップグレードにより、ディスクのコンテンツが消去されることがあります）保存できます。

次の場合は、**write memory** コマンドを最後に使用した時点の Websense 設定が維持されています。

- ディスク再設定の再起動を行う前、またはディスク内容を消去する ACNS ソフトウェアのアップグレードの前に、**write memory** コマンドが使用されていない場合。
- **reload** コマンドを使用していて、リロード プロンプトで設定を保存するかどうか確認されたときに、**yes** と応答しなかった場合

ただし、**write memory** コマンドが以前に実行されたことがなく、Content Engine 上の /local1/WebsenseEnterprise/EIM ディレクトリ内の内容が消去されている場合は、デフォルトの設定が適用されます。

Websense URL フィルタリング統計情報の閲覧

スタンドアロン Content Engine 上で現在設定されている HTTP URL フィルタリング方式のすべてのステータスを表示するには、**show url-filter http** EXEC コマンドを使用します。Content Engine と Websense サーバ間の通信の要求および応答に関する統計情報を表示するには、**show statistics url-filter http websense** EXEC マンドを使用します。これらの統計情報は、送信された要求、受信された応答、ブロックされたページ、許可されたページ、および障害の数を表示します。さらに詳しい URL フィルタリング統計情報は、Websense サーバ上で入手可能です。

表示される統計情報をクリアするには、**clear statistics url-filter http websense** コマンド、および **clear statistics all** EXEC コマンドを使用します。すべての統計カウンタが 0 にリセットされます。

SmartFilter ソフトウェアを使用した URL フィルタリングの設定

スタンドアロン Content Engine 上で実行される SmartFilter ソフトウェアは、プロキシ サーバ、ファイアウォール、およびキャッシング機器と一緒に使用した場合、Employee Internet Management（EIM）機能を提供します。SmartFilter フィルタリング機能は、ACNS 5.x ソフトウェアを実行する Content Engine の追加サービスとして使用できます。SmartFilter の追加サービスは、シスコから直接ライセンスが与えられます。

SmartFilter の追加サービスは、サーバ機能を 1 つにまとめたソリューションを提供します。Content Engine は、一連のプラグイン API を使用して、SmartFilter ソフトウェアが、HTTP トランザクション時に戦略ポイントにフックを設け、URL フィルタリングを実行できるようにします。

SmartFilter 追加サービスを設定するには、sfadmin console と呼ばれるエンド ユーザ管理ツールと sfadmin server と呼ばれる管理サーバ ツールを使用します。sfadmin console を使用して SmartFilter 製品を設定し、その設定を sfadmin サーバに保存します。sfadmin server がこの設定をエンド クライアント Content Engine に伝播すると、Content Engine 上で動作する SmartFilter ソフトウェアが使用できるようになります。スタンドアロン Content Engine で SmartFilter URL フィルタリングをイネーブルにするには、**url-filter http smartfilter enable** グローバル コンフィギュレーション コマンドを使用

します。スタンドアロン Content Engine のクラスタで SmartFilter URL フィルタリングを使用するには、クラスタ内の各 Content Engine 上で **url-filter http smartfilter enable** コマンドを入力して、すべてのトラフィックがフィルタリングされるようにしてください。

（注）　ACNS 5.4.1 以降のソフトウェア リリースは、SmartFilter ソフトウェア バージョン 4.1 をサポートしています。ACNS 5.2.1 および 5.3.x ソフトウェア リリースは、SmartFilter ソフトウェア バージョン 4.0 をサポートしています。SmartFilter ソフトウェア バージョン 4.1 がサポートされている場合、エージェントはブロック ページ要求をポート 9014 で直接待ち受けます。SmartFilter でサポートされているプロトコルの完全なリストについては、表 B-4 を参照してください。

Smartfilter ソフトウェアを使用して URL フィルタリングを設定する場合は、次の重要事項に注意してください。

- Content Engine を別の ACNS ソフトウェア リリースにアップグレードまたはダウングレードした場合に、SmartFilter プラグイン バージョンが異なっていれば、SmartFilter データベースおよびコンフィギュレーション ファイルが削除され、デフォルト設定がロードされます。この変更が発生するのは、新しいバージョンの SmartFilter ソフトウェアごとに設定の詳細が変更される可能性があるためです。SmartFilter プラグインをアップグレードまたはダウングレードしたら、SmartFilter の管理コンソールから Content Engine に新しいデータベースをダウンロードする必要があります。
- Content Engine が NAT 環境に配置されている場合は、**external-ip** *external-ip-address* 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 の外部 IP アドレスを設定する必要があります。そのように設定しないと、Advanced ブロック ページ機能が適切に動作しないことがあります。
- Content Engine の Smartfilter ソフトウェアおよびキャッシュ プロセスは、次の場合に再起動します。
  - Content Engine で Smartfilter ソフトウェアが稼働しているときに、Content Engine の外部 IP アドレスを設定した場合
  - Content Engine で Smartfilter ソフトウェアが稼働しているときに、Content Engine の外部 IP アドレスが設定されていない状態で、Content Engine のインターフェイス IP アドレスを変更した場合
- ポート 9014 は Smartfilter Advanced ブロック ページ専用です。

SmartFilter コントロール リストについて

SmartFilter コントロール リストは、200 万の Web サイトをコンテンツ グループに分類しています。SmartFilter コントロール リストには、30 の カテゴリがあらかじめ設定されていて、広範なマテリアルをカバーしています。企業の法的責任を軽減することが目的のカテゴリもあります。これら 30 のカテゴリは、デフォルトの SmartFilter ソフトウェア ポリシーで拒否に設定されています。カテゴリの中には、MP3 サイトのような（大量の帯域幅を消費するコンテンツがある）サイトを含むカテゴリもあります。これら 30 のカテゴリ以外は、ビジネスや教育に有効ではないまたは不適切と考えられます。

また、SmartFilter ソフトウェアは、10 のユーザ定義のカテゴリを提供し、SmartFilter コントロール リストに含まれていないサイトを定義したり、フィルタリングすることが可能です。また、特定のグループや個人がすばやく簡単にアクセスできるサイトを除外することもできます。SmartFilter 管理コンソールを使用して、SmartFilter コントロール リストのダウンロード スケジュールを決めることができます。Download Setup ウィンドウは、ダウンロード サイト、ユーザ名、ユーザ パスワードを追跡します。SmartFilter コントロール リストを少なくとも 1 カ月間更新しないと、SmartFilter ソフトウェアはコントロール リストが失効したとみなし、SmartFilter License ウィンドウで指定したアクションを実行します。

一時ユーザ上書き機能について

ACNS 5.4.1 ソフトウェア リリースの一時ユーザ上書き機能は、SmartFilter ソフトウェア バージョン 4.1 で使用可能な新機能です。この機能を使用すると、特定のユーザは自分たちのユーザ グループに現在適用されているフィルタリング プロセスを上書きできます。この機能は SmartFilter 管理コンソールを通して設定する必要があります。

一時ユーザ上書き機能を使用する手順は、次のとおりです。

ステップ 1　SmartFilter 管理コンソールが稼働しているマシン、または別のマシンに、SmartFilter 認証サーバ ソフトウェアをインストールします。

（注）　Smartfilter 認証サーバ ソフトウェアのコピーを取得するには、次の Web サイト を参照してください。
http://www.securecomputing.com

ステップ 2　SmartFilter 管理コンソールで認証サーバを追加します。

ステップ 3　SmartFilter 管理コンソールで認証サーバにユーザを追加します。

ステップ 4　認証サーバに変更を送信します。

ステップ 5　SmartFilter 管理コンソールで Content Engine を選択し、設定された認証サーバを Content Engine の認証サーバ リストに追加します。

ステップ 6　SmartFilter 管理コンソールで、Content Engine の上書きリストに、フィルタリング プロセスの上書きを許可するユーザを追加します。

ステップ 7　Content Engine に変更を送信します。

（注）　SmartFilter ソフトウェア設定についての詳細は、次の Web サイトを参照してください。
http://www.securecomputing.com

特定の HTTP および HTTPS 要求に関する URL フィルタリングをバイパスする Content Engine の設定

ACNS 5.2.3 以降のソフトウェア リリースでは、特定の HTTP および HTTPS 要求に関する URL フィルタリングをバイパスするように Content Engine を設定することができます。この機能は、ローカル リスト URL フィルタリング（優良および悪質サイト リスト）および Websense、SmartFilter、または N2H2 URL フィルタリングでサポートされています。

たとえば、Content Engine でローカル URL フィルタリングをイネーブルにし、悪質サイト拒否機能をイネーブルにした場合（badfile.txt ファイルにブロックする URL を追加した場合など）、**rule no-url-filtering** のアクションが hit（一致）であれば、Content Engine は該当する特定の要求の URL フィルタリングをバイパスします。それ以外の場合は、URL フィルタリングが継続され、該当する URL 要求がブロックされます。

スタンドアロン Content Engine にこの機能を設定するには、**rule action no-url-filtering** 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。詳細は、「no-url-filtering アクションの例」（p.13-11）を参照してください。

現在の URL フィルタリング設定の表示

スタンドアロン Content Engine の URL フィルタリング設定を表示するには、**show url-filter** EXEC コマンドを使用します。

```
ContentEngine# show url-filter http
ContentEngine# show url-filter rtsp
ContentEngine# show url-filter wmt
```

注意　ローカル リスト ファイルが大き過ぎると、プロキシのパフォーマンスに影響します。これは、ローカル リスト フィルタリングがイネーブルになると、そのローカル リスト ファイルがメモリにロードされるからです。ファイル サイズが 5 MB を超えると警告メッセージが表示されますが、ACNS ソフトウェアによってローカル リスト ファイルのサイズが制限されることはありません。ユーザの責任において、ローカル リスト ファイルのサイズを管理し、パフォーマンスに影響するほど大きくならないようにしてください。

URL フィルタリング統計情報の表示

スタンドアロン Content Engine に設定された各 URL フィルタリング方式の統計情報を表示するには、**show statistics url-filter** EXEC コマンドを使用します。

```
ContentEngine# show statistics url-filter ?
  http  Display URL-filter for http and mms over http statistics
  rtsp  Display URL-filter for rtsp statistics for real proxy, real server and
        cisco streaming engine
  wmt   Display URL-filter for wmt statistics for rtsp requests
```

WMT 要求（Windows Media 9 Players からの RTSP 要求）に関するローカル リスト URL フィルタリング統計情報を表示するには、**show statistics url-filter wmt local-list** EXEC コマンドを使用します。

（注） WMT 要求の場合、URL フィルタリング方式としてサポートされるのはローカル リスト ファイルのみです。

コマンド出力例として、許可された WMT 要求、ブロックされた WMT 要求、および ローカル リスト URL フィルタリングでフィルタリングされなかった WMT 要求の数が表示されます。

```
ContentEngine# show statistics url-filter wmt local-list
Local List URL filtering statistics:
                        Requests Allowed = 25
                        Requests Blocked = 30
                    Requests not filtered = 5
```

スタンドアロン Content Engine の RealMedia 要求の統計情報を表示するには、**show statistics url-filter rtsp local-list** EXEC コマンドを使用します。RealMedia 要求は、スタンドアロン Content Engine で稼働中の RealProxy サーバで処理されます。

```
ContentEngine# show statistics url-filter rtsp?

  local-list  Display local-list URL-filter statistics
```

（注） RealMedia 要求の場合、URL フィルタリング方式としてサポートされるのはローカル リスト ファイルのみです。登録済みの Content Engine（つまり、ACNS ネットワーク内に配置されている Content Distribution Manager に最初に登録されていないスタンドアロン Content Engine ではなく、Content Distribution Manager に登録されている Content Engine）の **show statistics url-filter rtsp local-list** EXEC コマンドの出力には、登録済み Content Engine で 2 つのバックエンド RTSP サーバがイネーブル化されている場合、RealSubscriber および Cisco Streaming Engine からクライアントに送信された統計情報も含まれます。

コマンド出力例として、許可された要求、ブロックされた要求、および ローカル リスト URL フィルタリングでフィルタリングされなかった要求の個数が表示されます。

```
ContentEngine# show statistics url-filter rtsp local-list
Local List URL filtering statistics:
                        Requests Allowed = 15
                        Requests Blocked = 10
                    Requests not filtered = 2
```

HTTP 要求の URL フィルタリング統計情報を表示するには、**show statistics url-filter http** EXEC コマンドを入力します。HTTP 要求に関する URL フィルタリングでは、サードパーティ製ソフトウェア（N2H2 や Websense ソフトウェアなど）によるローカル リスト ファイルおよび URL フィルタリングがサポートされています。

```
ContentEngine# show statistics url-filter http ?
  local-list  Display local-list URL-filter statistics
  N2H2        Display N2H2 URL-filter statistics
  websense    Display websense URL-filter statistics
```

URL フィルタリング情報のクリア

スタンドアロン Content Engine の URL フィルタリング統計情報をクリアするには、**clear statistics url-filter** EXEC コマンドを使用します。

```
ContentEngine# clear statistics url-filter ?
  http  Clear URL-filter for http statistics
  rtsp  Clear URL-filter for rtsp statistics
  wmt   Clear URL-filter for wmt statistics
```

たとえば、スタンドアロン Content Engine の WMT URL フィルタリング統計情報は、次のようにしてクリアします。

```
ContentEngine# clear statistics url-filter wmt local-list
```