CHAPTER 11

スタンドアロン Content Engine 上でのコンテンツ事前ローディングと URL フィルタリングの設定

この章では、コンテンツ事前ローディングの概要と、ACNS 5.x ソフトウェアが動作するスタンドアロン Content Engine でサポートされている URL フィルタリングの種類を説明し、スタンドアロン Content Engine 上でコンテンツ事前ローディングと URL フィルタリングを設定する方法を説明します。

この章の構成は、次のとおりです。

- スタンドアロン Content Engine のコンテンツ事前ローディングの設定（P. 11-2）
- スタンドアロン Content Engine 上での URL フィルタリングの設定（P. 11-9）
- 特定の HTTP および HTTPS 要求に対して URL フィルタリングをバイパスする Content Engine の設定（P. 11-39）
- 現在の URL フィルタリング設定の表示（P. 11-40）
- URL フィルタリング統計情報の表示（P. 11-41）
- URL フィルタリング統計情報のクリア（P. 11-42）

ACNS 5.2.3 ソフトウェアおよびそれ以降では、特定の URL のパフォーマンスをモニタするよう Content Engine を設定できます。Content Engine は、モニタ対象の各 URL について、さまざまな応答特性の統計情報を管理します。このトピックに関する詳細は、「スタンドアロン Content Engine 上でのクリティカル 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） この章で使用される CLI コマンドの構文と使用方法については、『Cisco ACNS Software Command Reference, Release 5.3』を参照してください。
Content Distribution Manager に登録されている Content Engine（スタンドアロン Content Engine の逆）上で URL フィルタリングを設定する方法については、『中央管理配置に関する Cisco ACNS ソフトウェア コンフィギュレーション ガイド Release 5.3』を参照してください。

スタンドアロン Content Engine のコンテンツ事前ローディングの設定

ここでは、ACNS ソフトウェア 5.3.1 以降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ディングの概要について説明します。また、スタンドアロン Content Engine の事前ローディング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、ユーザのコンテンツ要求を予測して特定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ことを保証します。事前ロード プロセス時に、スタンドアロン Content Engine は、コンテンツを取得するために、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 サイトの複数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 ソフトウェアは、URL のファイルを読み取り、指定された URL をスタンドアロン Content Engine 上に事前ロードできます。スタンドアロン Content Engine 上に事前ロードされるコンテンツは、HTTP URL、FTP-over-HTTP URL、および MMS URL（WMT ストリーミング メディア ファイル）。ACNS ソフトウェア 5.3.1 では、Windows Media 9 のクライント / サーバー用 RTSP URL リストのサポートが追加されました。この URL リストは、「事前ロードされた URL リスト ファイル」として参照されます。

（注）　設定済みのすべての HTTP、FTP-over-HTTP、MMS、および RTSP の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。

ACNS ソフトウェア 5.1.1 以降では、NTLM 認証済み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。

URL リスト ファイル内のエントリの形式は次のとおりです。

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 クレデンシャル（証明書）が設定されている場合には、*domain name* は指定する必要があります（区切り記号が必要です）。

```
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には、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 を有効にして、特定のオブジェクトをフェッチし、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** グローバル設定コマンドを実行します。

```
ContentEngine(config)# http cache-authenticated all
```

Content Engine を設定して、NTLM 認証オブジェクトだけをキャッシュするには、**http cache-authenticated ntlm** グローバル設定コマンドを実行します。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、同じオブジェクトを次の要求が認証できるように、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 機能を有効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 ユーティリティを使用したスタンドアロン Content Engine の基本コンフィギュレーション設定」」の ステップ 15 を参照。）を設定した場合、WMT は Content Engine ですでに有効になっています。それ以外の場合、スタンドアロン Content Engine 上で Windows Media Services を有効にする Content Engine CLI（Setup ユーティリティではなく）の使用方法の説明について、「スタンドアロン Content Engine 上の WMT RTSP ストリーミングおよびキャッシング サービスを設定するためのチェックリスト」（P. 9-23）を参照してから、この Content Engine に対してストリーミング メディア ファイルの事前ロードを有効にしてください。

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 に事前ロードされる URL（HTTP、FTP-over-HTTP、MMS、または RTSP プロトコルの URL）がリストされています。このファイルの保守は管理者が行い、リモート システム上に作成される必要があります。このファイルは事前ロード アクセスを制御するスタンドアロン 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。**pre-load url-list-file** *path* グローバル設定コマンドを使用してこのファイルのパスを指定します。

（注）　**pre-load url-list-file** *path* グローバル設定コマンドで、*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 パラメータには、事前ローディング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 の 3 階層のコンテンツをすべてダウンロードすることを意味します。depth レベルが指定されない場合、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
```

Content Engine CLI を使用して事前ロード URL リスト ファイルを作成する場合、リリース 5.1.5 よりも前の ACNS 5.1.x ソフトウェアでは、**pre-load url-list-file** グローバル設定コマンドだけに、HTTP または FTP オプションがありました。事前ロード URL リスト ファイルをセキュアにフェッチする方法を代替するもの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 経由で事前ロード 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、組織は、HTTPS 経由で事前ロード 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ないことに注意してください。HTTPS プロトコルを使用して、事前ロード URL リスト ファイルのダウンロード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有効化と設定

Content Engine GUI または CLI のいずれかを使用して、スタンドアロン Content Engine 上でコンテンツ事前ロード機能を有効にし、設定できます。

（注）　Content Engine GUI から、**Caching > Content Preload** の順に選択します。表示された Content Preload ウィンドウを使用して、このスタンドアロン Content Engine 上のコンテンツ事前ロード機能を有効にして設定します。このタスクを実行する Content Preload ウィンドウの使用方法の詳細は、このウィンドウの **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機能を有効にし、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機能を有効にします。

```
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。作成方法については、「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 取得の同時実行要求の最大数を指定します。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、少ない方がアクティブになります。

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

ステップ 4　URL 検索のデフォルト depth レベルを指定します（たとえば深度レベル 4）。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 の事前ロードを試行することを望まない場合、depth レベルをデフォルトから 0 に設定することをお勧めします。

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

ステップ 5　URL リストまたは単一の URL が記述された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 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 ファイルのダウンロードだけがサポートされています（前述の例を参照）。

ACNS 5.3.1 ソフトウェア リリースでは、事前ロード URL リスト ファイルに RTSP URL を指定する機能が追加されました。

ステップ 6　事前ロード プロセス時に取得されるドメインを指定します（たとえば cisco.com）。

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

ステップ 7　HTML　ページ内の他のドメインもすべて検索されることを指定します。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 オペレーション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（たとえば 50,000 kbps）。

```
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ディング用の URL に、異なるビットレートを指定する、WMT ストリーミング メディア ファイルも事前ロードできるようになりました。**bandwidth wmt outgoing** および **bandwidth incoming** グローバル設定コマンドを使用して、WMT 帯域幅もコントロールすることができます。詳細は、「着信および発信 WMT 帯域幅、およびビット レートの設定」（P. 9-32）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　Content Engine を設定して、特定のコンテンツを今後に備えて事前ロードするには、**pre-load schedule** グローバル設定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**　グローバル設定コマンドまたは　Content Engine　GUI［**Caching > Content Preloading** ］を使用して設定される）による頻度で、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 オペレーションのデフォルト開始時刻は、00:00（つまり、深夜 0 時）です。終了時刻を指定しないと、すべてのオブジェクトがダウンロードされた後、事前ロード オペレーションが完了します。このデフォルトを変更する場合は、次の手順を実行します。

a. 毎日または毎週事前ロードを行う開始時刻と終了時刻を指定するには、*hh*:*mm*（ここで、*hh* は時、*mm* は分です。たとえば、01:00 など）を使用します。時間ごとに事前ロードを行う場合には、*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に、毎日インターバルを指定する方法を示しています。この例では、事前ロード オペレーションは毎日午前 1:00 に開始し、毎日午前 2:00 に終了します。

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

b. 時間ごとに事前ロードを行う場合、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。毎日および毎週事前ロードを行う場合、開始時刻は 0 ～ 23、終了時刻は 0 ～ 59 になります。終了時刻を指定しないと、事前ロード オペレーション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** グローバル設定コマンドを使用します。次の例では、事前ロード オペレーションを毎週日曜日と水曜日の午前 1:00 ～ 午前 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** グローバル設定コマンドを使用して、すべての事前ロード　トラフィックに対する DSCP（Differentiated Services Code Point）コード ポイントと同様に、ToS（Type of Service）値を設定します。

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サービス タイプに分割できます。URL の照合、ファイル タイプ、ドメイン、宛先 IP アドレス、発信元 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 MMS 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ディングは、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上の 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、Type of Service 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** グローバル設定コマンドの使用は、DSCP　サーバ設定に関係する　Rules Template 設定コマンドの使用より優先されます。

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　現在の事前ローディング オペレーション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後の、現在の事前ローディング オペレーションの状況を示します。

```
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　事前ロードが開始したら、現在の事前ローディングに関連付けられてい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　事前ロードされたファイルの URL の状況をエンド ユーザに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、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. 9-48）を参照してください。

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** グローバル設定コマンドを使用します。

コンテンツ事前ローディング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** グローバル設定コマンドを使用して、コンテンツを取得する事前ローディング プロセスを再開できます。このコマンドを使用すると、再度 URL リスト ファイルの最初から事前ローディングをはじめるのではなく、事前ローディング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** グローバル設定コマンドが Content Engine 上でセットアップされていないで、コンテンツ事前ローディングがスケジュールされた終了時刻以前に中断された場合、次にスケジュールされたコンテンツ事前ローディングは、URL リスト ファイルの最初からはじまります。

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

企業や団体の中には、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し、管理し、制限する必要性を認識している企業や団体があります。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、あるいは、インターネット情報を正しく使用する方法を指導できます。Content Engine に URL フィルタリング方式を備えると、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、企業は投資利益が直ちに得られると同時に、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 では、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 URL へのアクセスを拒否する。リストで指定した 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. 11-10）を参照してください。
N2H2 外部サーバ	URL フィルタリングを行うため、コンテンツに対するクライアント要求を外部の N2H2 サーバへ誘導する。「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. 11-16）を参照してください。
Websense サーバ	URL フィルタリングを行うため、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 サーバへ誘導する。「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. 11-19）を参照してください。
SmartFilter プラグイン	URL フィルタリングを行うため、コンテンツに対するクライアント要求を SmartFilter プラグインに誘導する。「SmartFilter ソフトウェアを使用した URL フィルタリングの設定」（P. 11-37）を参照してください。

異なる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 形式だけですが、多数の 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 フィルターが 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。しかし、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 を経由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有効になっているスキームは、ほかのプロトコルの影響を受けません。

（注）　**url-filter** グローバル設定コマンドは、**rule** グローバル設定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無効にします。デフォルトでは、ロード バイパスは有効になっています。

- Content Engine GUI を使用して、ロード バイパスを手作業で無効にするには、**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 バイパスを手作業で無効にするには、**bypass load** グローバル設定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無効にするには、**error-handling send-cache-error** または **error-handling reset-connection** グローバル設定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 上で有効になっています。

```
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 上で有効になっているときは、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。ブロッキングが指定されていない場合は、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. 17-5）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

スタンドアロン Content Engine を設定して、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、MMS や RTSP などのストリーミング 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。この種類の URL フィルタリングは、「ローカル リスト URL フィルタリング」と呼ばれます。

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 ファイルを同時にアクティブにできるのは、1 つだけです。

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、FTP などのタイプの URL だけが含まれるようにする必要があります。ACNS ソフトウェア 5.3.1 以降では、WMT ローカル リスト ファイルには MMS や RTSP などの URL を含めることができます。

注意　ローカル リスト ファイルが大きくなると、プロキシのパフォーマンスが悪くなります。これは、ローカル リスト ファイル フィルタリングが有効になっていると、そのローカル リスト ファイルは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 ソフトウェアはローカル リスト ファイルにサイズ制限を強制しません。ローカル リスト ファイルのサイズを管理し、パフォーマンスに悪影響を及ぼす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、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 を経由した要求（HTTP、FTP-over-HTTP、および HTTPS-over-HTTP 要求）
- RealMedia 要求（RealNetworks 独自の拡張機能を組み込んだ IETF 標準 RTSP プロトコル）
- WMT 要求（MMS [MMST および MMSU] と、Windows Media 9 クライント / サーバ用 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、サポートされていません。

Windows Media 9 Player では、RTSP-over-RTP（「WMT RTSP 要求」と呼ばれている）が、ACNS ソフトウェア 5.3.1 以降でサポートされています。WMT RTSP 要求には、rtsp、rtspu、および rtspt という 3 つの プロトコル プレフィックスが使用できます。

ユーザが URL のプロトコル プレフィックスとして rtsp: を入力すると、Windows Media 9 Player は RTSPT または RTSPU を選択できます。rtsp bad ファイルに rtsp://*hostname*/*pathname* という URL があって、ユーザの URL 要求が rtspt://*hostname*/*pathname* の場合、Windows Media 9 Player からの RTSP 要求は URL フィルタリングを通る可能性がありました。そのため、ACNS ソフトウェア リリース 5.3.1 では、Windows Media 9 Player からの RTSP 要求のための特別な URL フィルタリングが追加されました。

WMT URL フィルタリングについては、RTSP URL（rtsp://）のフィルタリングのみがサポートされています。RTSPT URL と RTSPU URL に対する別々のフィルタリングはサポートされていません。ただし、badurl.lst ファイルに RTSP URL を設定すると、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** グローバル設定コマンドを使用します。ACNS ソフトウェア リリース 5.3.1 では **url-filter** コマンドが変更され、Windows Media 9 Player からの RTSP 要求用のローカル リスト URL フィルタリングがサポートされるようになりました。ACNS ソフトウェア 5.2.x 以前の **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
  rtsp  For requests over RTSP
  wmt   For WMT requests
```

ACNS ソフトウェア 5.3.1 以降の **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 フィルタリングでは、WMT 要求（WMT クライアントからの MMS および RTSP 要求）と RTSP 要求（RealMedia Player からの要求）用のフィルタリング方式のみサポートされます。WMT および RTSP 要求については、サードパーティの URL フィルタリング方式（N2H2、SmartFilter、および Websense ソフトウェア）をサポートしていません。HTTP 要求については、N2H2、SmartFilter、Websense だけでなく、ローカル リスト URL フィルタリング方式をサポートしています。他の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 では、HTTP 要求（HTTP、FTP-over-HTTP、および HTTPS-over-HTTP 要求）用ローカル リスト URL フィルタリングを使うために、スタンドアロン Content Engine を設定する Content Engine CLI グローバル設定コマンドを説明しています。

表 11-4　HTTP を経由する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スタンドアロン Content Engines 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 では、RTSP を経由する要求用ローカル リスト URL フィルタリングを使用するために、スタンドアロン Content Engine を設定する Content Engine CLI グローバル設定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 で使用されます。RealProxy はスタンドアロン Content Engine 上で実行されているバックエンド RTSP サーバです。登録されている Content Engine とともに、その Content Engine 上で実行されている RealProxy、RealSubscriber、および Cisco Streaming Engine がこのタイプの URL フィルタリングを使用します。

表 11-5　RTSP を経由する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スタンドアロン Content Engines 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 では、WMT 要求（UDP 経由の MMS 要求 [MMSU]、TCP 経由の MMS 要求 [MMS]、および RTSP 経由の WMT 要求）用ローカル リスト URL フィルタリングを使用するために、スタンドアロン Content Engine を設定する Content Engine CLI グローバル設定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 リストに指定された RTSP URL のほか、RTSPT および RTSPU の両方の 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スタンドアロン Content Engines 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 ソフトウェア 5.3.1 以降では、グローバル設定コマンドは MMS（MMSU および MMST）と RTSP に適用します。RTSP 要求用の 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 サーバの場合に使用されます。Windows Media Player の旧バージョン（たとえば、Windows Media 7 Player）を使用している場合は、Windows Media Player からのコンテンツ要求に応じるため、RTSP プロトコルの代わりに MMS プロトコルが使用されます。

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

スタンドアロン Content Engine を設定して、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手順は、次のとおりです。

ステップ 1　badurl.lst という名前の平文テキスト 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/ の形式で入力し、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 の下に別個のディレクトリ（たとえば、**/local1/filtered_urls**）を作成することをお勧めします。

ステップ 3　Content Engine に bad URL リストを参照させます。

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

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

```
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 bad 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 を設定して、ローカル リスト ファイルを使用して特定の HTTP URL を許可し他の拒否する手順は、次のとおりです。

ステップ 1　goodurl.lst という名前の平文テキスト 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）を作成することをお勧めします。

ステップ 3　Content Engine に goodurl.lst ファイルを指し示します。

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

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

```
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 good 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 Engines 上での ローカル リスト フィルタリングのリロード

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 リスト機能が有効になっている場合）。

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）用の新しいローカル リストをリロードします。
- **rtsp** は、RTSP 経由の要求（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（MMS [MMSU および MMST] 要求と、Windows Media 9 Player からの RTSP-over-RTP [Microsoft 独自拡張機能を組み込んだ IETF 標準 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。

次の例では、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 を介して配信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 Engines を設定できます。カスタム メッセージは、block.html という名前の一種の HTML ページで、システム管理者が作成します。埋め込みグラフィックスがそのカスタム メッセージの HTML ページに関連付けられている場合は、すべてのグラフィックスを block.html ファイルと同じディレクトリに必ずコピーしてください。次の例では、block.html ファイルの内容を示しています。

```
<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 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 ファイルを使用して、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
```

前の例に示すように、block.html ファイル内のオブジェクト（.gif、.jpeg など）が、カスタム メッセージのディレクトリ ストリング /content/engine/blocking/url 内で参照されています。

カスタマイズされたブロッキング メッセージを有効にするには、**url-filter http custom-message** グローバル設定コマンドを使用し、ディレクトリ名を指定します。カスタム メッセージを無効にするには、**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、有効または無効にすることができます。

（注）　local1 または local2 を、カスタム ブロッキング 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 の下に別個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の両方を使用して、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。

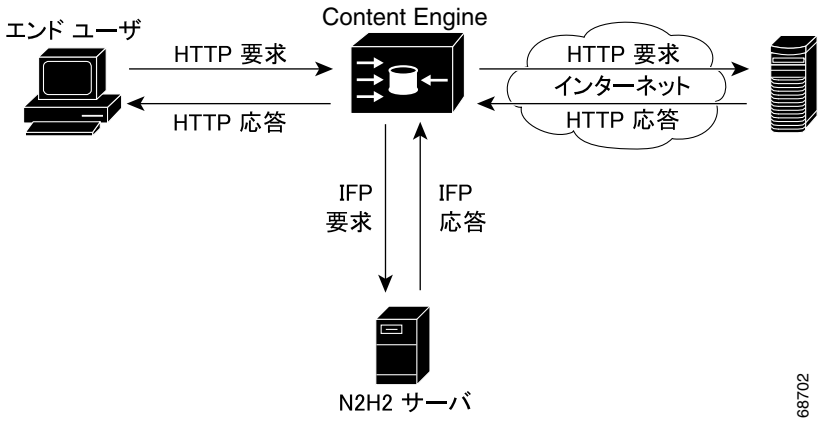
N2H2 は、3 種類のフィルタリング方式をサポートします。表 11-7 では、Content Engine がサポートする N2H2 機能を示しています。1 台の N2H2 サーバで、同時に複数の Content Engine をサポートできます。

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。
透過認証	透過認証は、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戻して行われる。

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 応答を戻します。N2H2 サーバの IFP 応答に基づいて、Content Engine は、ブロッキング 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、または、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）　N2H2 サーバを使用する URL フィルタリングは、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関係なく、HTTP トラフィック（HTTP、FTP-over-HTTP、または HTTPS-over-HTTP 要求）に適用された後、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 フィルタリングを行う外部の N2H2 サーバを使う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 経由の要求に対して、Content Engine 上で現在有効になっている URL フィルタリング方式を表示します。

```
ContentEngine# show url-filterhttp
```

ステップ 2　HTTP 経由の要求に対して有効になっている URL フィルタリング方式が他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** グローバル設定コマンドを使用して、URL フィルタリングに外部の N2H2 サーバを使うように 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（IP アドレスなど）。

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 サーバを使うように設定しています。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 要求を受信す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、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** グローバル設定コマンドは、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 が有効になっている間に、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　Content Engine の N2H2 URL フィルタリング方式を有効にします。

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

ステップ 5　**url-filter http N2H2 allowmode enable** グローバル設定コマンドを使用することにより、N2H2 サーバが有効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** が有効になっている場合、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** が無効になっている場合、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** は有効になっています。N2H2 が有効でも無効であっても、**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 6　Content Engine と N2H2 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
```

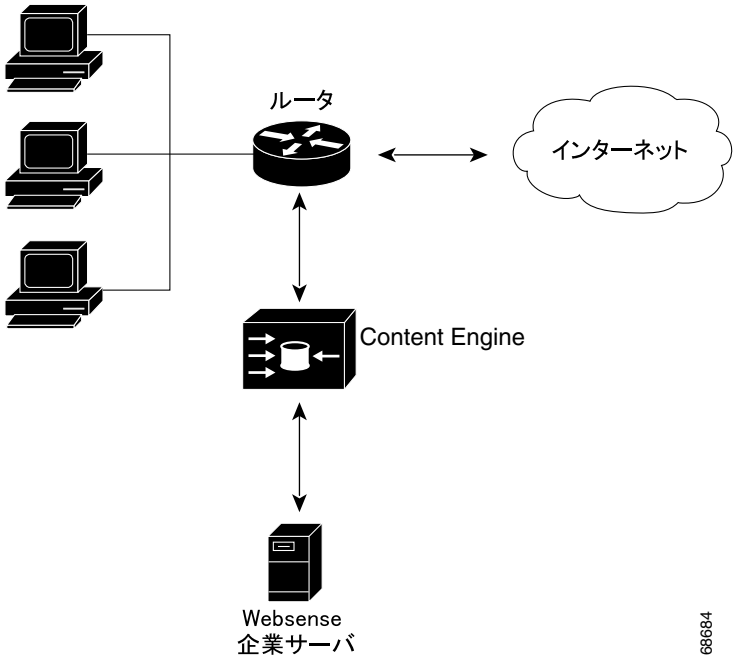
これらの統計情報は、送信された要求、受信された応答、ブロックされたページ、許可されたページ、および障害の数を表示します。さらに詳しい URL フィルタリング統計情報は、N2H2 サーバ上で入手可能です。表示される統計情報をクリアするには、**clear statistics url-filter http N2H2** コマンド、および **clear statistics all** EXEC コマンドを使用します。**clear statistics url-filter http N2H2** EXEC コマンドは、N2H2 サーバの統計カウンタをリセットします。すべての統計カウンタが 0 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、http://www.n2h2.com を参照してください。

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 サーバとは別のシステム（Host 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 サーバを使う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 Web サーバは内部サーバで、Content Engine 上で実行し、「ローカル Websense サーバ」と呼ばれます。

（注）　ACNS ソフトウェア 5.1 以前では、サポートされる Websense サーバは 1 つのみです。ACNS ソフトウェア 5.2.1 以降は、2 つまでの Websense サーバがサポートされます。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。

Websense サーバのフェールオーバーについて

ACNS ソフトウェア 5.2.1 では、Websense サーバのフェールオーバー機能が追加されました。この機能により、URL フィルタリングのフェールオーバーを目的として最大 2 つの Websense サーバ（1 つはプライマリ、もう 1 つはセカンダリ サーバ）を使用するように Content Engine を設定できます。表 11-8 では、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リストしています。

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	ローカル Websense サーバは、Content Engine 上で無効になっています。	プライマリ Websense サーバは、外部ホスト（たとえば、Host A）で実行中です。 セカンダリ Websense サーバは、2 台目の外部ホスト（たとえば、Host B）で実行中です。
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ています。	セカンダリ Websense サーバは、外部ホストで実行中です。
オプション C	ローカル Websense サーバは、セカンダリ Websense サーバとして動作しています。	プライマリ Websense サーバは、外部ホストで実行中です。

Websense サーバの設定順により、どのサーバがプライマリ Websense サーバになるかが決まります。最初に設定された Websense サーバが、プライマリ サーバになります。セカンダリ Websense サーバの設定は、オプションです。たとえば、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は、「Websense サーバ フェールオーバーと URL フィルタリングの設定例」（P. 11-25）を参照してください。

Websense サービスについて

Websense ソフトウェア 5.2.0 では、Websense 5.0.1 でサポートされていた単一の Websense サービスが、次の 5 つのサービスに置き換わりました。

- Employee Internet Management（EIM; 従業員インターネット管理）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

ACNS ソフトウェア 5.3.1 では、上記 5 つの Websense サービスがすべてサポートされます（ACNS ソフトウェア 5.2.x では、ローカル RADIUS エージェント サービスおよびローカル eDirectory エージェントサービスはサポートされていませんでした）。

（注）　「ローカル Websense サーバ」という用語は、Content Engine 内部で実行されている 5 つの Websense プロセスの総称として使用されます。また、Websense プロセスは「サービス」とも呼ばれます。

Websense 5.2.0 ソフトウェアでは、ローカルまたはリモート Websense ポリシー サーバを使用して、Content Engine 上でローカル 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、およびローカル ユーザ サービスを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 ソフトウェア 5.3.1 以降でサポートされているローカル Websense 5.2 サーバのサービス

名前	説明
ポリシー サーバ	外部 Websense Manager GUI を介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 ユーザ サービス）に伝えます。 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、ローカル ユーザ サービスをアクティブにする前に、ローカル（内部）ポリシー サーバーまたは指定したリモート ポリシー サーバを実行しておく必要があります。
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、および FTP-over HTTP 以外のプロトコルを使用する要求の URL フィルタリングを有効にします。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 および Gopher などのフィルタ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 • VNC および 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（たとえば、NNTP [Network News Transfer Protocol]）
ローカル RADIUS エージェント	外部 RADIUS サーバを介して認証されたユーザ用のユーザ ベースまたはグループ ベース ポリシーに基づいて、URL フィルタリングを有効にします。このエージェントは、ネットワークにアクセスして RADIUS 認証方式を通して認証されたユーザを、透過的に識別します。Content Engine は、この情報を提供されると、リモートからネットワークにアクセスするユーザやユーザ グループにポリシーを適用できます。 このエージェントは、RADIUS クライアントと外部 RADIUS サーバの間で RADIUS メッセージを転送するプロキシとして動作します。ローカル RADIUS エージェントが適切に機能するには、Content Engine 上で RADIUS の設定（外部 RADIUS サーバの IP アドレスなど）を行っておく必要があります。設定方法は、「スタンドアロン Content Engine の RADIUS 認証設定」（P. 17-9）を参照してください。

表 11-9　ACNS ソフトウェア 5.3.1 以降でサポートされているローカル Websense 5.2 サーバのサービス（続き）

名前	説明
ローカル eDirectory エージェント	LDAP を介して認証されたユーザ用のユーザ ベース ポリシーまたはグループ ベース ポリシーに基づいて、URL フィルタリングを有効にします。このエージェントは Novell eDirectory と連携して動作し、ネットワークにアクセスして LDAP 認証方式を介して認証されたユーザを、透過的に識別します。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、ユーザやグループに適用されているポリシーに基づいて、要求をフィルタリングできます。 このエージェントは LDAP を使用して、ユーザのネットワークへのログインを認証する Novell eDirectory からユーザのログイン セッション情報を収集します。このエージェントは、認証された各ユーザを IP アドレスと関連付けます。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 エージェントが適切に機能するには、管理識別名などの設定を行っておく必要があります。設定方法は、「ローカル eDirectory エージェントの設定」（P. 11-32）を参照してください。
ローカル ユーザ サービス	ユーザ ベース ポリシーまたはグループ ベース ポリシーに基づき URL フィルタリングを有効にします。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 ベースの 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 11-10 では、スタンドアロン Content Engine 上での Websense ソフトウェア 5.2.0 の設定に関連する CLI コマンドをリストしています。

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 incoming [**auth-port** *port number*] [**accnt-port** *port number*]	ローカル RADIUS エージェントの要求の着信ポート番号を設定するために使用します。詳細は、「ローカル RADIUS エージェントの設定」（P. 11-32）を参照してください。
websense-server service radius-agent outgoing [**host** *remote-RADIUS-server IP-address*] [**auth-port** *port number*] [**accnt-port** *port number*]	Content Engine が使用する外部 RADIUS サーバに必要な設定を指定するために使用します。詳細は、「ローカル RADIUS エージェントの設定」（P. 11-32）を参照してください。
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。

表 11-10　Websense サーバ関連の CLI コマンド（続き）

CLI コマンド構文	説明
websense-server service edir-agent edir-server [**administrative-dn** *administrative-distinguished-name*] [**administrative-passwd** *password*] [**host** *remote-eDirectory-server IP-address*] [**root-context** *root-context*]	Content Engine が使用する外部 eDirectory サーバに必要な設定を指定するために使用します。詳細は、「ローカル eDirectory エージェントの設定」（P. 11-32）を参照してください。
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。

（注）　ACNS 5.2 ソフトウェアでは、**websense-server ip-address** および **websense-server user-server external** グローバル設定コマンドは推奨されません。

ACNS ソフトウェア リリース 5.2.1 以降では、URL フィルタリング用に最大 2 つまで Websense サーバを使用するよう Content Engine を設定できます。

Websense URL フィルタリング用の Content Engine を設定するには、URL フィルタリングに使用する Websense サーバ設定の種類を決定します。

- 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）を使用するには、「Websense サーバ フェールオーバーと URL フィルタリングの設定例」（P. 11-25）を参照してください。
- ローカル Websense サーバのみを使用するには、「ローカル Websense サーバを使用した URL フィルタリングの構成」（P. 11-29）を参照してください。
- 外部 Websense サーバのみを使用するには、「外部 Websense サービスを使った Websense URL フィルタリングの設定」（P. 11-34）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 経由の要求に対して Websense URL フィルタリングを有効にするには、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）用の Content Engine 上で現在有効な URL フィルタリング方式を表示するには、**show url-filter http** EXEC コマンドを使用します。Websense サーバ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。

（注）　ACNS ソフトウェア 5.2.x および 5.3.x では、Websense サーバ Version 5.2 は、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 ソフトウェアを設定する方法の詳細は、Web サイト（ http://www.websense.com）を参照してください。

Websense ソフトウェアは、Content Engine 上の /local/local1/WebsenseEnterprise/EIM ディレクトリ内に常駐する Websense サーバのイメージを提供します。設定ファイルおよびロギング ファイルと共に、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。

Websense サーバが有効になっていない、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 サーバがオフピーク時、またはネットワーク トラフィックが低いときに有効にしてください。それ以外の場合に Websense サーバを有効にすると、Content Engine 上で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 サーバは自動的に Content Engine 上で開始します。

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、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プリント アウトします。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	これは、Websense GUI Manager が接続する 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お勧めします。	55806

表 11-11 にある始めの 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 で Content Engine に送り返して、修正できます。

（注）　新しく設定されたポートを有効にするには、Websense サーバを無効にし、再度有効にする必要があります。ローカル Websense サーバを無効にするには、**no websense-server enable** グローバル設定コマンドを使用します。Websense クライアントを正しい Websense サーバ ポートに誘導するには、必ず **url-filter http websense server** グローバル設定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 5.3』を参照してください。

ACNS ソフトウェア 5.0 または 5.1 へのダウングレード時の Websense の問題

ローカル（内部）Websense サーバが Content Engine 上で有効になっている状態で、ACNS ソフトウェア 5.2.x を 5.0 または 5.1 にダウングレードすると、WebsenseEnterprise ディレクトリが Content Engine から削除され、ローカル Websense サーバが停止します。ACNS ソフトウェア 5.2.x 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。ただし、ACNS ソフトウェア 5.3.1 以降では、この Websense のダウングレード問題について知らせる、次のエラー メッセージが表示されます。

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

ACNS ソフトウェア 5.2.x 以降から 5.1 または 5.0 へのダウングレード時の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 サーバを無効にします。

ステップ 2　Content Engine 上の Websense サービスを停止します。

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

Websense サーバ フェールオーバーと 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ています。最初に Content Engine がローカルまたはリモートのプロキシ サーバを使用するように設定され、次にローカル 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 上でアクティブになります。

次に、Content Engine が、ローカル（内部）Websense サーバをプライマリ Websense サーバとして、外部 Websense サーバをセカンダリ Websense サーバとして使用するように設定されます。プライマリ Websense サーバが使用できない場合、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ります。

Content Engine 上で、allow mode が再び有効になってから、URL フィルタリングが有効になります。ローカルおよびリモートの Websense サーバのデフォルト ポリシーを設定するために Websense Manager GUI が使用され、続いて HTTP プロキシが Content Engine 上で有効になります。

Websense サーバのフェールオーバーと 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 上の個々の Websense サービスをアクティブにするために、ローカルとリモートの Websense ポリシー サーバのどちら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 上のローカル 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 上のリモート ポリシー サーバについて、必要な情報（たとえば、ホスト名または IP アドレスと、そのポート番号）を設定します。

  ```
  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。
  - ポート番号はオプションです。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）を起動する前には、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のポリシー 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 上でアクティブになっているローカル Websense サーバのすべてのサービス（ローカル EIM サーバ、ローカル ネットワーク エージェント、およびローカル ユーザ サービス）を有効にします。

```
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 上で無効になっています。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 サーバを有効にしてください（つまり、Content Engine クラスタ内の各 Content Engine で **websense-server enable** グローバル設定コマンドを入力します）。

ステップ 8　**url-filter http websense server local** グローバル設定コマンドを使用して、Content Engine がローカル Websense サーバをプライマリ Websense サーバとして使用するように設定します。

（注）　**url-filter http websense server** グローバル設定コマンドを使用して、プライマリおよびセカンダリ Websense サーバを異なる設定（たとえば、タイムアウト、ポート番号、および接続数）にすることができます。デフォルトでは、HTTP プロキシとして動作している Content Engine は、ポート 15868 の Websense サーバにフィルタリング要求を送り、接続がタウムアウトする前に Websense サーバからの応答を 20 秒間待機します。そして、CPU ごとに 40 個の永続接続を確立します。

この例では、HTTP プロキシとして動作している Content Engine は、ポート 4005 のローカル Websense サーバにフィルタリング要求を送り、接続がタウムアウトする前に Websense サーバからの応答を 60 秒待機します。そして、このローカル Websense サーバに対して 90 個の永続接続を確立します。ローカル Websense サーバが最初に設定されるので、これが Content Engine のプライマリ Websense サーバになります。

```
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ません。127.0.0.1 に設定されています。

ステップ 9　**url-filter http websense server** グローバル設定コマンドを使用して、Content Engine が外部 Websense サーバをセカンダリ Websense サーバとして使用するように設定します。

ローカル Websense サーバはすでにプライマリ Websense サーバなので、外部 Websense サーバをセカンダリ 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 である外部 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 サーバが使用できない場合、Content Engine はポート 4006 のセカンダリ Websense サーバに要求を送り、接続がタウムアウトする前にこのサーバからの応答を 90 秒待機します。そして、CPU ごとに 90 の永続接続を確立します。

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

ステップ 10　デフォルトでは、allow mode が有効になっています。allow mode を再度有効にするには、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 サーバが使用できない場合、Content Engine は指定したセカンダリ Websense サーバに要求を送ります。プライマリとセカンダリの両方の Websense サーバが使用できない場合、要求は allow mode に送られます。

- allow mode が有効になっている場合、Websense 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。
- allow mode が無効になっている場合、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 サーバが有効無効に関係なく、**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 11　Content Engine 上の URL フィルタリングを有効にします。

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

ステップ 12　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、ローカルおよびリモートの両方の Websense サーバで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 のメイン ウィンドウの左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で、ローカル（内部）Websense　サーバが実行されている Content Engine の IP アドレスを入力します。

b. Content Engine 上で実行されている Websense ポリシー サーバに接続します。

- 左ペインで、ポリシー サーバをダブルクリックします（たとえば Content Engine の IP アドレス）。
- 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 サーバへ接続します。
- 左ペインで、**Filter Definition** をダブルクリックし、続いて **Policies** をダブルクリックします。
- **Global** を選択します。
- 右ペインで、**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、基本設定、常にブロック、ブロックなしなどのカテゴリ セットを適用します。デフォルトポリシーはグローバルで、デフォルト カテゴリ セットがデフォルト設定になります。

（注）　Websense Enterprise Manager ウィンドウで **Save Changes** ボタンをクリックしても、Websense 設定の変更は保存されず、リブートすると無効になります。リブート後まで Websense 設定の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 の使用方法の詳細は、次の Web サイト（http://www.websense.com）を参照してください。

ステップ 13　Content Engine 上に HTTP プロキシを設定します。

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

ステップ 14　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 サーバを使用した URL フィルタリングの構成

URL フィルタリングでローカル（内部）Websense サーバを使うように 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 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 サーバを有効にします。デフォルトでは、無効になっています。
3. HTTP 要求（HTTP、FTP-over-HTTP、HTTPS-over-HTTP 要求）の URL フィルタリングを行うためにローカル Websense サーバを使うように 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、他のプロトコルも同様にフィルタリングできます。

ACNS ソフトウェア 5.2.1 以降では、フェールオーバーの目的で、Websense サーバを 2 つまで設定できます。ローカル Websense サーバが、いずれかの Websense サーバになることができます。Websense サーバの設定順により、どのサーバがプライマリ サーバになるかが決まります。最初に設定された Websense サーバが自動的にプライマリ Websense サーバになり、2 番目に設定され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 ソフトウェア 5.3.1 以降では、Websense サーバ サービスを自由に組み合わせ（表 11-9 を参照）て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、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。このトピックに関する詳細は、「外部 Websense サービスを使った Websense URL フィルタリングの設定」（P. 11-34）を参照してください。

ACNS ソフトウェア 5.0.3 ～ 5.1.x 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 リリースにおける Websense サーバの個々のサービスをアクティブにするオプションは用意されていないため、デフォルトでは、Websense サーバ サービスは Content Engine 上で次のようにアクティブになります。

- 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 5.3.1 以降にアップグレードすると、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 サービスがアクティブになります（ローカル eDirectory エージェントとローカル RADIUS エージェントは、ACNS ソフトウェア 5.3.1 へのアップグレード時にはアクティブになりません）。
- ACNS ソフトウェア 5.0.3 または 5.1.x からリリース 5.2.1 にアップグレードすると、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 サービスが Content Engine 上でアクティブになります。

URL フィルタリングを行うためにローカル（内部）Websense サーバを使うように 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 サーバかリモート 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 上のローカル 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、Content Engine 上のリモート ポリシー サーバの必要な情報（たとえば、ホスト名または IP アドレスと、そのポート番号）を設定します。

```
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、オプションです。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 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）を起動する前には、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のポリシー 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-server service radius-agent outgoing** および **websense-server service radius-agent incoming** グローバル設定コマンドを使用して、Content Engine 上のローカル RADIUS エージェントを設定します。詳細は、「ローカル RADIUS エージェントの設定」（P. 11-32）を参照してください。

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　**websense-server service edir-agent edir-agent edir-server** グローバル設定コマンドを使用して、Content Engine 上のローカル eDirectory エージェントを設定します。詳細は、「ローカル eDirectory エージェントの設定」（P. 11-32）を参照してください。

ステップ 8　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 9　Content Engine 上でアクティブになっているローカル Websense サーバのすべてのサービス（ローカル EIM サーバ、ローカル ネットワーク エージェント、およびローカル ユーザ サービス）を有効にします。

```
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 上で無効になっています。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 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 サーバを有効にしてください（つまり、Content Engine クラスタ内の各 Content Engine で **websense-server enable** グローバル設定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 フィルタリングを行うためにローカル Websense サーバを使うように Content Engine を設定します。Websense サーバ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-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 フィルタリング用に内部 Websense サーバを使うように指定します。
- *port number* は、フィルタリングする HTTP 要求を受信するローカル 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 上で受信します。
- *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 あたりの持続的接続数（1 ～ 250）です（デフォルトは CPU ごとに 40）。内部 Websense サーバへ持続的接続数を設定するには、このオプションを使用します。別の値が要求されていることが明らかでない限り、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 フィルタリングを有効にします。

```
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. 11-35）」を参照してください。

ローカル eDirectory エージェントの設定

ローカル eDirectory エージェントが適切に機能するには、Content Engine で次の設定を行う必要があります。

- 管理識別名（dn）
- 管理パスワード
- 外部 LDAP データベースの検索を要求するために Content Engine が接続する外部 eDirectory サーバの IP アドレスまたはホスト名
- ディレクトリ サービスのルート ロケーション（ルート コンテキスト）

ACNS ソフトウェア 5.3.1 では、**websense-server service edir-agent edir-server** グローバル設定コマンドが追加されました。

```
ContentEngine(config)# websense-server service edir-agent edir-server ?
  administrative-dn      Specify Administrative Distinguished name
  administrative-passwd  Specify Administrative Password
  host                   eDirectory Server IP address.
  root-context           Specify the directory service root location
```

コマンドには複数のオプションがあり、eDirectory サーバを設定できます。次の例では、**host** コマンド オプションを使用して、Content Engine が IP アドレス 172.18.24.10 をもつ外部 eDirectory サーバを参照するように設定する方法を示しています。

```
ContentEngine(config)# websense-server service edir-agent edir-server host
172.18.24.10
```

次の例では、**administrative-passwd** コマンド オプションを使用して、管理用パスワードとして default244 を指定する方法を示しています。この管理パスワードは、データベース検索を要求するため外部 eDirectory サーバに接続する Content Engine によって使用されます。

```
ContentEngine(config)# websense-server service edir-agent edir-server
administrative-passwd default244
```

ローカル eDirectory エージェントが適切に機能するには、発信ホストの IP アドレスおよび認証ポートが、Content Engine が使用するように設定された LDAP サーバのいずれかの IP アドレスおよび認証ポートと一致する必要があります。Content Engine が認証に使用するように設定された外部 LDAP サーバの IP アドレスおよび発信認証ポートを指定する方法は、次のとおりです。

```
ContentEngine(config)# ldap-server host ip_addr [auth-port port-number]
```

ローカル eDirectory エージェントの設定後、Content Engine のローカル eDirectory エージェントをアクティブにするには、**websense-server service edir-agent activate** グローバル設定コマンドを入力します。

ローカル RADIUS エージェントの設定

RADIUS 認証のクライアントは、ACNS 5.x ソフトウェアを実行する Content Engine 上に存在します。これらのクライアントは、可能な場合、中央（リモート）の RADIUS サーバに認証要求を送信します。このサーバには、ログイン認証情報とネットワーク サービス アクセス情報が含まれています。Content Engine で実行されているローカル RADIUS エージェントは、RADIUS クライアントと外部 RADIUS サーバの間で RADIUS メッセージを転送するプロキシとして動作します。

ローカル RADIUS エージェントが適切に機能するには、Content Engine でエージェントについて次の設定を行う必要があります。

- 外部 RADIUS サーバの IP アドレスまたはホスト名

- 次の目的で使用する受信ポート番号
  - RADIUS クライアントからの認証要求の受信
  - RADIUS クライアントからのアカウンティング要求の受信
- 次の目的で使用する発信ポート番号
  - 認証要求の外部 RADIUS サーバへの転送
  - アカウンティング要求の外部 RADIUS サーバへの転送

ACNS ソフトウェア 5.3.1 では、**websense-server service radius-agent** グローバル設定コマンドが追加されました。

```
ContentEngine(config)# websense-server service radius-agent ?
  activate  Install local Radius Agent
  incoming  Configuration for incoming radius-agent requests
  outgoing  Configuration for outgoing radius-agent requests
```

このコマンドには複数のオプションがあり、Content Engine 上でエージェントをアクティブにするほか、ローカル RADIUS エージェントに必要な設定を行うことができます。たとえば、**websense-server service radius-agent incoming** グローバル設定コマンドを使用すると、ローカル RADIUS エージェントの要求に対して受信ポート番号を設定できます。

```
ContentEngine(config)# websense-server service radius-agent incoming auth-port
[port-number]
ContentEngine(config)# websense-server service radius-agent incoming accnt-port
[port-number]
```

認証要求の転送先となる外部 RADIUS サーバのポート番号を指定するには、**websense-server service radius-agent outgoing auth-port** グローバル設定コマンドを使用します。ポート番号の範囲は 1 ～ 65535 で、デフォルトはポート 1645 です。

```
ContentEngine(config)# websense-server service radius-agent outgoing auth-port
[port-number]
```

アカウンティング要求の転送先となる外部 RADIUS サーバのポート番号を指定するには、**websense-server service radius-agent outgoing accnt-port** グローバル設定コマンドを使用します。ポート番号の範囲は 1 ～ 65535 で、デフォルトはポート 1646 です。

```
ContentEngine(config)# websense-server service radius-agent outgoing accnt-port
[port-number]
```

認証要求の転送先となる外部 RADIUS サーバの IP アドレスを指定するには、**websense-server service radius-agent outgoing host** グローバル設定コマンドを使用します。

```
ContentEngine(config)# websense-server service radius-agent outgoing host
[RADIUS server IP address]
```

ローカル RADIUS エージェントが適切に機能するには、発信ホストの IP アドレスおよび認証ポート番号が、Content Engine が使用するように設定された RADIUS サーバのいずれかの IP アドレスおよび認証ポートと一致する必要があります。**radius-server** グローバル設定コマンドを使用して、Content Engine が認証に使用するように設定された外部 RADIUS サーバの IP アドレスと発信認証ポート番号を指定する方法は、次のとおりです。

```
ContentEngine(config)# radius-server host ip_addr [auth-port port-number]
```

RADIUS サーバ ホストのデフォルト ポートは 1645 です。スタンドアロン Content Engine での RADIUS ホスト設定（たとえば、IP アドレス、ポート番号、および RADIUS キー）の詳細に付いては、「スタンドアロン Content Engine の RADIUS 認証設定」（P. 17-9）を参照してください。

ローカル RADIUS エージェントの設定後、Content Engine のローカル RADIUS エージェントをアクティブにするには、**websense-server service radius-agent activate** グローバル設定コマンドを入力します。

外部 Websense サービスを使った Websense URL フィルタリングの設定

Content Engine が外部の Websense サーバを使用するように設定する場合は、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。ここで指定される IP アドレスとポート番号は、外部 Websense サーバの IP アドレスと、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と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、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てしまいます。デフォルトでは、allow mode は Content Engine 上で有効になっています。allow mode が有効になっている場合、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。allow mode が無効になっている場合、再度有効にするには **url-filter http websense allowmode enable** コマンドを使用します。

ACNS ソフトウェア 5.2.1 以降では、フェールオーバーの目的で、Websense サーバを 2 つまで設定できます。Websense サーバの設定順により、どのサーバがプライマリ サーバになるかが決まります。最初に設定された Websense サーバが自動的にプライマリ Websense サーバになり、2 番目に設定されたサーバがセカンダリ Websense サーバになります。サポートされる Websense サーバ設定のリストについては、表 11-8 を参照してください。

URL フィルタリングのために外部の Websense サーバを使う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** グローバル設定コマンドを使用して、外部 Websense サーバについて必要な情報を指定します。

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 あたりの持続的接続数（1 ～ 250）です（デフォルトは CPU ごとに 40）。外部 Websense サーバへ持続的接続数を設定するには、このオプションを使用します。別の値が要求されていることが明らかでない場合、デフォルト数を変更しないでください。

次の例では、IP アドレスが 172.18.22.10 で、Host A で実行されている外部 Websense サーバを指定するようスタンドアロン Content Engine を設定します。また、Content Engine は、この外部 Websense サーバのポート 4006 に要求を送信するように、さらに、接続がタイムアウトするまでこのサーバからの応答を 90 秒間待つように設定されます。

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

（注）　Content Engine のクラスタが存在するときに、外部の Websense URL フィルタリングを使用するには、Content Engine クラスタ内の各 Content Engine 上で **url-filter http websense server** グローバル設定コマンドを使用し、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目的で、セカンダリ Websense サーバを設定する場合は、次のいずれかの作業を行います。

- セカンダリ Websense サーバとしてローカル（内部）Websense サーバを設定するには、**url-filter http websense server local** グローバル設定コマンドを使用します。ローカル Websense サーバの設定に関する詳細は、「ローカル Websense サーバを使用した URL フィルタリングの構成」（P. 11-29）を参照してください。
- プライマリ Websense サーバ（Host A）とは異なるホスト（Host B）で実行されている外部 Websense サーバを設定するには、**url-filter http websense server** コマンドを入力します。ここでは、コマンドに、セカンダリ Websense サーバのパラメータ（たとえば、IP アドレス、ポート番号、タイムアウト、Host B で実行している Websense サーバの接続数）を指定する必要があります。

ステップ 3　Websense を、この Content Engine 上での現在の URL フィルタリング方式として有効にします。

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

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

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（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、ローカル ポリシー サーバ）の 1 つまたは複数のサービスを無効にする手順は、次のとおりです。

ステップ 1　ローカル Websense サーバが、Content Engine 上で無効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　ローカル Websense サーバが Content Engine 上で有効になっている場合は、無効にします。

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

ステップ 3　**no websense-server service** グローバル設定コマンドの **no** 形式を使用して、ローカル Websense サーバの特定サービスを停止します。

たとえば、**no websense-server service network-agent activate** コマンドを使用して、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
```

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、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 サービスを停止する順番は自由です。しかし、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 上でローカル ポリシー サーバを使用している場合は、これを停止します。

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

ステップ 5　Content Engine 上でリモート ポリシー サーバを使用している場合は、これの設定を解除します。

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

Websense 設定ファイルの保存

ACNS ソフトウェア 5.2.1 以降では、**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を通して変更した Websense 設定ファイル（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** コマンドにより、外部の Websense Manager GUI から加えた変更を有効にして、ディスクの再設定およびアップグレード（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場合があります）をして保存できます。

write memory コマンドを最後に使用した Websense 設定は、次の状況で保持されます。

- ディスクの再設定後の再起動か、ディスク内容を消去する ACNS ソフトウェアの更新の前に **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の確認に対して「yes」と応答していない場合。

write memory コマンドが以前に実行されたことがない場合で、Content Engine 上の /local1/WebsenseEnterprise/EIM ディレクトリ内のコンテンツが消去されたときは、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すべての状況を表示するには、**show url-filter http** EXEC コマンドを入力します。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 コ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 サーバから入手可能です。

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されるときに、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、Cisco 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 がこの設定を伝播してエンド クライアントである Content Engines に渡すと、Content Engines 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。Content Engine 上の SmartFilter URL フィルタリングを有効にするには、**url-filter http smartfilter enable** グローバル設定コマンドを使用します。スタンドアロン Content Engine のクラスタが存在するときに、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 ソフトウェア 5.2.1 以降では、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 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。

Content Engine を ACNS ソフトウェアの異なるリリースへアップグレードまたはダウングレードする場合、SmartFilter プラグインのバージョンが異なっていると、SmartFilter データベースと設定ファイルが削除され、デフォルト設定がロードされます。このように変更されてしまうのは、SmartFilter ソフトウェアの各新規バージョンで設定の詳細が変更されている可能性があるからです。SmartFilter プラグインのアップグレードまたはダウングレードの後、SmartFilter 管理コンソールから Content Engine へ最新データベースをダウンロードする必要があります。

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 個の カテゴリがあらかじめ設定されていて、広範囲に及ぶ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 個のカテゴリは、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）サイト含むカテゴリもあります。これら 30 個のカテゴリ以外は、ビジネスや教育にとって役立たないか不適切と考えられます。

また、SmartFilter ソフトウェアは、10 個のユーザ定義カテゴリを提供し、SmartFilter コントロール リストに含まれていないサイトを拒否およびフィルタ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 か月間更新しないと、SmartFilter ソフトウェアは、コントロール リストが「失効」したと見なし、SmartFilter License ウィンドウで指定したアクションを実行します。

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

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

ACNS 5.2.3 ソフトウェア以降では、特定の HTTP および HTTPS 要求に対しての URL フィルタリングをバイパスするように Content Engine を設定できます。この機能は、Websense、SmartFilter、N2H2 などの URL フィルタリングと同様、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）でもサポートされています。

たとえば、Content Engine 上のローカル フィルタリングを有効にして、悪質サイト拒否機能（たとえば、ブロックすべき URL を含む badfile.txt ファイル）を有効にし、**rule no-url-filtering** アクションがヒット（一致）すると、Content Engine は特定の要求に対して URL フィルタリングをバイパスします。ヒットしなければ、URL 要求を URL フィルタリングで処理し、ブロックします。

スタンドアロン Content Engine でこの機能を設定するには、**rule action no-url-filtering** グローバル設定コマンドを使用します。このトピックに関する詳細は、「no-url-filtering アクションの例」（P. 13-12）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
```

次の例では、**show url-filter http** コマンドを使用して、スタンドアロン Content Engine 上で現在設定されている、HTTP 要求に対する、すべての TTP URL フィルタリング方式すべての状況を表示します。

```
ContentEngine# show url-filter http
URL filtering is set to use bad-list

Local list configurations
=================================
Good-list file name :
Bad-list file name : /local1/url-filter/badlist.http
Custom message directory :

Websense server configuration
=================================
Websense server IP     : 172.16.193.165
Websense server port   : 15868
Websense server timeout: 20 (in seconds)
Websense server connections : 40
Websense allow mode is ENABLED

N2H2 server configuration
============================
N2H2 server IP       : 172.16.193.165
N2H2 server port     : 4005
N2H2 server timeout  : 5 (in seconds)
N2H2 allow mode is ENABLED
ContentEngine#
```

注意　ローカル リスト ファイルが大きくなると、プロキシのパフォーマンスが悪くなります。これは、ローカル リスト ファイル フィルタリングが有効になっていると、そのローカル リスト ファイルは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 ソフトウェアはローカル リスト ファイルにサイズ制限を強制しません。ローカル リスト ファイルのサイズを管理し、パフォーマンスに悪影響を及ぼす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 上で設定されるさまざまな 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 mms and rtsp requests
```

WMT 要求（MMS [MMSU および MMST] 要求と、Windows Media 9 Player からの RTSP 要求）のローカル リスト URL フィルタリングの統計情報を表示するには、**show statistics url-filter wmt local-list** EXEC コマンドを入力します。

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

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

（注）　WMT 要求の場合は、ローカル リスト ファイルは、サポートされる唯一の URL フィルタリング方式です。

コマンド出力例にあるように、許可された WMT 要求、ブロックされた WMT 要求、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
Lodal 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 コマンドを入力します。RealMedia 要求は、スタンドアロン Content Engine で実行している 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 要求の場合は、ローカル リスト ファイルは、サポートされる唯一の URL フィルタリング方式です。登録済みの Content Engine（ACNS ネットワーク内に配置されていても、初期状態では Content Distribution Manager には登録されていないスタンドアロン Content Engine に対して、Content Distribution Manager に登録されている Content Engine）では、**show statistics url-filter rtsp local-list** EXEC コマンドから出力されるには、2 つのバックエンド RTSP サーバが登録されている Content Engine 上で有効になっている場合に RealSubscriber と Cisco Streaming Engine がクライアントにサービスを提供した統計情報も含まれています。

コマンド出力例にあるように、許可された要求およびブロックされた要求の数と、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数が表示されます。

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

HTTP 要求に対する 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 要求を URL フィルタリングする方法には、ローカル リスト ファイルと、サードパーティ製ソフトウェア（たとえば、N2H2 や Websense ソフトウェア）を介した 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 フィルタリング統計情報のクリア

スタンドアロン Content Engines 上の URL フィルタリング統計情報をクリアするには、**lear statistics url-filter** EXEC コマンドを入力します。

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

たとえば、スタンドアロン Content Engines 上の WMT URL フィルタリング統計情報をクリアするには、次のように行います。

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