



拡張ルーティング設定

この付録では、WCCP 対応ルータまたは Content Router のいずれかを使用して、複数のオリジンサーバから取得したコンテンツを事前配信し、処理する方法を説明します。次の 5 つの事例で、この情報を記述します。

- ケース 1：要求をリダイレクトするために Content Router を使用した場合の Windows Media コンテンツの保存方法の概要 (p.C-2)
- ケース 2：Content Router を使用した動的作成コンテンツの取得と配信 (p.C-4)
- ケース 3：Content Router を使用した静的作成コンテンツの取得と配信 (p.C-9)
- ケース 4：WCCP ルーティングを使用した動的作成コンテンツの取得と配信 (p.C-12)
- ケース 5：WCCP ルーティングを使用した静的作成コンテンツの取得と配信 (p.C-17)

ケース 1：要求をリダイレクトするために Content Router を使用した場合の Windows Media コンテンツの保存方法の概要

ここでは、Windows Media ファイルに対して要求が行われると、クライアント、Content Router、Content Engine の間で発生する基本的なやりとりを説明します。このやりとりを説明するには、次の一般的な方法を考慮します。

ビデオ トレーニング モジュールを社員に配信すると仮定します。ビデオが Active Streaming Format (ASF) ファイルであり、社員のデスクトップの Windows Media Player にオンデマンドで配信されます。配信には Microsoft Media Server (MMS) プロトコルを使用します。ネットワークは Content Router を使用して要求を代行受信し、配信を加速するためビデオファイルを事前配信した、最適な Content Engine へ要求をリダイレクトします。MMS プロトコルはリダイレクションを認識できないので、社内 Web サイト上で次の URL を正しく公開します。

`http://cr-fqdn/video.asf.asx`

ネットワークに ACNS Content Router があり、MMS を使用してファイルをストリーミングする場合、拡張子「.asx」をパブリッシング URL の最後に追加する必要があります。.asx 拡張子がないと、クライアントへのコンテンツ配信は行われません。.asx 拡張子は Microsoft Advanced Streaming Redirector (ASX) ファイルのファイル拡張子です。ASX は、埋め込まれたリダイレクション URL を含んだメタデータ ファイルです。ASX ファイルは、MMS プロトコルを使用してファイルをストリーミングする際に、クライアント、Content Router、Content Engine の間のやりとりで重要な部分です。



(注)

HTTP または RTSP を使用して再生するファイルに対し、.asx 拡張子を追加する必要はありません。

社員が Web ブラウザのパブリッシング URL へのリンクをクリックすると、次のアクションが発生します。

1. Web ブラウザクライアントは、ファイル (video.asf) の要求を Content Router (cr-fqdn) へ送信します。

`http://cr-fqdn/video.asf.asx`

2. キャッシュに保存された要求ビデオファイルを持つ、最適な Content Engine のリダイレクション URL を含む ASX ファイルをクライアントに送信することで、Content Router はクライアントに応答します。

`http://ce-name.ce.crfqdn/video.asf.asx`

3. クライアントはこの新しい URL を使用して、要求を Content Engine に再送信します。
4. Content Engine は、コンテンツ用の実際の URL が含まれた ASX ファイルを戻します。

— ACNS 5.1 および 5.2 ソフトウェアでは、ASX ファイルには MMS (mms://) 用に 1 つの URL が、HTTP (http://) 用に 1 つの URL が含まれます。クライアントは最初に MMS URL を試行しますが、これが失敗すると HTTP URL を試行します。

— ACNS 5.3 以降のソフトウェア リリースでは、ASX ファイルには RTSP (rtsp://) 用の 3 番目の URL が含まれます。

RTSP プロトコルを使用してストリーミング コンテンツを送受信するには、Windows Media Client と Windows Media Server 両方で Windows Media バージョン 9 を使用している必要があります。

5. クライアントは、クライアント PC 上で表示される Windows Media Player のバージョンに基づいて使用するプロトコルを選択し、選択した URL を Content Engine に返してコンテンツを取得します。

mms://ce-name.ce.crfqdn:1755/video.asf



(注) ライブ ストリームの場合、HTTP URL への自動フェールオーバーは発生しません。ACNS 5.1 または 5.2 ソフトウェアでは、ライブ イベント用の MMS URL が失敗すると、ストリームが失敗します。ACNS 5.3 以降のソフトウェアでは、ライブ イベント用の MMS URL が失敗すると、RTSP URL が試行されます。両方のストリーミングプロトコルが失敗すると、ストリームは失敗します。

6. Content Engine は MMS プロトコルを使用して、ビデオ ファイルをクライアント Windows Media Player に配信します。

ケース 2: Content Router を使用した動的作成コンテンツの取得と配信

ケース 1 では、オリジンサーバ上の Web ページには、JavaScript や類似のスクリプトを使用して記述されたスクリプトなどを通じて動的に生成されたコンテンツへのリンクが含まれます。JavaScript のスクリプト内で参照されているリンクをたどって、動的コンテンツをダウンロードすることは不可能です。リンクの参照は ACNS ソフトウェアの取得機能であり、動的コンテンツの取得はサポートしないためです。ここでは、要求が Content Router によってルーティングされるような環境で、動的コンテンツを取得する指定方法を説明します。このシナリオでは、オリジンサーバ上の Web ページには、他の「外部」オリジンサーバ上でホスティングされている Web ページへのリンクもあります。この例では、読者が、Content Router を使用して要求するルーティングに必要なセットアップを十分理解していることを前提としています(コンテンツルーティングの詳細および背景情報については、第 4 章「ACNS ネットワークのコンテンツ要求ルーティング設定」を参照)。

この例では、コンテンツを公開している 3 つのオリジンサーバ (www-server、mms-server、file-server) があることを前提とします。オリジンサーバ (www-server) がメインサーバであり、他の 2 つの「外部」サーバ上の Web ページへリンクされています。メインオリジンサーバ (www-server) には、ACNS ネットワーク内で事前配信されることが望ましい 2 つのファイル (01.asx と index.html) が含まれています。index.html ファイルには、同じサーバ (www-server) 上の 01.asx ファイルへのリンク、mms-server と file-server の URL、および mms-server の URL へのリンクをもつ JavaScript があります。01.asx ファイルには、mms-server の URL へのリンクがあります。

ファイル :index.html

```
<HTML>
<HEAD>
<TITLE>Two-server combo</TITLE>
</HEAD>
<BODY>
<H1>Index </H1>
<ul>
<li>
<a href="01.asx">
Start Windows Media Presentation1
</a>
<li>
<a href="http://mms-server/DCARoot/wmload.asf">
One Windows Media file
</a>
<li>
<a href="http://file-server.cisco.com/biff/http10k.txt">
Regular http file
</a>
<script language="JavaScript">
function doRedir(url)
{
document.location.href = url;
}
</script>
<li>
Click <a href="javascript:
doRedir('mms://mms-server/DCARoot/mbr/MBR_BillGInterview_Low.wmv');">here</a> for
BillG Interview.
<br>
<li>
Click <a href="javascript:
doRedir('mms://mms-server/DCARoot/mbr/MBR_BillGSpeech_Low.wmv');">here</a> for BillG
Speech.
</BODY></HTML>
```

ファイル :01.asx

```
<ASX Version = "3.0">
<AUTHOR> Biff</Author>
<Entry><Ref href = "http://mms-server/DCARoot/Consumer_100K.wmv" />
<Entry><Ref href = "mms://mms-server/DCARoot/ColorBars1_500K.wmv" />
</Entry>
</ASX>
```

Content Router を使用してこの動的コンテンツを取得する手順は、次のとおりです。

ステップ 1 メイン オリジン サーバ `www-server` 上の公開コンテンツのコピーを作成し、新しいフォルダに入れます（この例では、複製フォルダは `Folder2` です）。

コンテンツを配信する Content Router に使用される Request Routed FQDN は、`www-server` にリンクするページのオリジン サーバ FQDN とは異なります。したがって、事前配信のために公開コンテンツのコピーを作成する必要があります。

さらに、Content Router を使用してコンテンツを適切に配信するには、コンテンツを事前配信する前に、コンテンツ自体に一定の変更を加える必要があります。

ステップ 2 Window Media メタファイル（`.asx`、`.wax`、`.wvx` の拡張子で終わるファイル）を除く `Folder2` 内のすべてのファイルに、次の変更を加えます。

- a. 外部サーバへの参照を、すべて対応する Request Routed FQDN に置き換えます（この例では、Request Routed FQDN は `ourtest.unicorn.com` です）。
- b. Windows Media ファイル（拡張子 `.wma`、`.wmv`、`.asf` を持つファイル）を参照するすべての URL で、ファイル名に拡張子 `.asx` を付けます。
- c. Windows Media ファイルを参照するすべての URL で、`mms://` を `http://` に変換します。

Windows Media メタファイル（拡張子 `.asx`、`.wax`、または `.wvx` を持つファイル）を参照する場合、そのメタファイルが参照するコンテンツは、メイン オリジン サーバに置かれているコンテンツと同じチャンネルを使用して事前配信されなければなりません。ACNS ソフトウェアは、他のチャンネル内でコンテンツを事前配信した ASX ファイルをサポートしません。

URL へのリンクが JavaScript を使用して指定される場合、それらのファイルが置かれているサーバ上で、ディレクトリ索引付け機能をイネーブルにできます。その結果、必要なファイルをすべて、そのオリジンサーバから簡単に取得できるようになります。

セカンダリ オリジン サーバ上で、異なるポートで受信する複製仮想ホストを作成できます。このホストは、ディレクトリ索引付け機能をイネーブルにし、特定のアクセス権を持ちます。この設定では、一般の顧客にファイルを公開せずに、メイン Web ページは引き続き機能できます。

■ ケース 2 : Content Router を使用した動的作成コンテンツの取得と配信

ステップ 2 で説明した変更が加えられると、Folder2 内のコンテンツは、次のように変更されます (変更箇所は太字表記)。

```
[root@PC-45 Folder1]# cd ../Folder2
[root@PC-45 Folder2]# ls
01.asx  index.html

1. index.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
  <HEAD>
    <TITLE>Two-server combo</TITLE>
  </HEAD>
  <BODY>
    <H1>Index </H1>
    <ul>
    <li>
      <a href="01.asx">
Start Windows Media Presentation1
      </a>

    <li>
      <a href="http://ourtest.unicorn.com/DCARoot/wmload.asf.asx">
One Windows Media file
      </a>

    <li>
      <a href="http://ourtest.unicorn.com/biff/http10k.txt">
Regular http file
      </a>

      <script language="JavaScript">
        function doRedir(url)
        {
          document.location.href = url;
        }
      </script>
    <li>
      Click <a href="javascript:
doRedir('http://ourtest.unicorn.com/DCARoot/mbr/MBR_BillGInterview_Low.wmv.asx');">her
e</a> for BillG Interview.
      <br>
    <li>
      Click <a href="javascript:
doRedir('http://ourtest.unicorn.com/DCARoot/mbr/MBR_BillGSpeech_Low.wmv.asx');">here</
a> for BillG Speech.

    </BODY></HTML>

2. 01.asx:
(remains unchanged)
<ASX Version = "3.0">
<AUTHOR> Biff</Author>
<Entry><Ref href = "http://mms-server/DCARoot/Consumer_100K.wmv" />
<Entry><Ref href = "mms://mms-server/DCARoot/ColorBars1_500K.wmv" />
</Entry>
</ASX>
```

これで、コンテンツは ACNS ネットワークで取得できる状態になりました。次に必要な作業は、マニフェストファイルの作成です。

ステップ 3 チャンネルのマニフェスト ファイルで、次のように複数のジョブを設定する必要があります。

- 外部サーバをクローलせず、メイン オリジン サーバの Folder2 フォルダからファイルを取得するために、クローल タスクまたは単一項目（どちらか便利な方）を設定します。
- 2 番めのオリジン サーバからファイルを取得するために、ディレクトリ クローलまたは単一項目を設定します。
- 3 番めのオリジン サーバからファイルを取得するために、ディレクトリ クローलまたは単一項目（どちらか便利な方）を設定します。

サンプル マニフェスト ファイル

```
<?xml version="1.0" ?>
<CdnManifest>
  <server name="cr-test1">
    <host name="http://www-server/biff/Folder2" proto="http"/>
  </server>

  <!-- get content from first origin server. Step 2a above-->
  <crawler server="cr-test1" start-url="index.html" depth="2" ttl="5" >
  </crawler>

  <!-- get content from the second origin server. Step 2b above-->
  <item src="http://mms-server/DCARoot/wmload.asf" />
  <!-- referenced in asx file -->
  <item src="http://mms-server/DCARoot/Consumer_100K.wmv" />
  <item src="mms://mms-server/DCARoot/ColorBars1_500K.wmv" />
  <!-- These two ensure that the links from the JavaScript are acquired-->
  <item src="mms://mms-server/DCARoot/mbr/MBR_BillGInterview_Low.wmv" />
  <item src="mms://mms-server/DCARoot/mbr/MBR_BillGSpeech_Low.wmv" />

  <!-- get content from the third origin server. Step 2c above-->
  <item src="http://file-server.cisco.com/biff/http10k.txt" />

</CdnManifest>
~
```

別の方法として、一部のリンクが JavaScript からのものではないので、追加のクローल タスクを定義することもできます。マニフェスト ファイルは次のようになります。

```
<?xml version="1.0" ?>
<CdnManifest>
  <server name="cr-test1-1">
    <host name="http://www-server/biff/Folder2" proto="http"/>
  </server>

  <server name="cr-test1-2">
    <host name="http://www-server/biff/Folder1" proto="http"/>
  </server>

  <!--get content from first origin server-->
  <crawler server="cr-test1-1" start-url="index.html" depth="2" ttl="5" >
  </crawler>

  <!-- These two ensure that the links from the JavaScript which cannot be crawled are
  acquired-->
  <item src="mms://mms-server/DCARoot/mbr/MBR_BillGInterview_Low.wmv" />
  <item src="mms://mms-server/DCARoot/mbr/MBR_BillGSpeech_Low.wmv" />

  <!-- get content from the second and third origin servers using a crawl-->
  <crawler server="cr-test1-2" start-url="index2.html" depth="2" ttl="5"

externalPrefixes="http://file-server.cisco.com/|mms://mms-server/|http://mms-server/">
  <matchRule>
    <match match-url="http://file-server.cisco.com/.*/>
    <match match-url="mms://mms-server/.*/>
    <match match-url="http://mms-server/.*/>
  </matchRule>
</crawler>
</CdnManifest>
```

■ ケース 2 : Content Router を使用した動的作成コンテンツの取得と配信

- ステップ 4** Content Distribution Manager GUI 内で、Content Router が使用する適切な Request Routed FQDN を持つ Web サイトを作成します（この例では、Request Routed FQDN は *ourtest.unicorn.com* です）。
- ステップ 5** DNS サーバ上で、ドメイン *ourtest.unicorn.com* 内のすべてのものを、要求の送信経路として指定された Content Router に送信するレコードを作成します。Content Router が、ネットワーク内の Content Engine と同じ Content Distribution Manager に登録されていることを確認してください。
- ステップ 6** Content Distribution Manager GUI 内で、作成した Web サイト、および上記で作成したマニフェストの 1 つを使用して、チャンネルを作成します。
- ステップ 7** Content Engine がチャンネルに割り当てるのに十分な帯域幅があることを確認してから、必要な Content Engine をチャンネルに割り当てます。
- ステップ 8** 複製の状況を調べ、複製が完了したことを確認します。
- ステップ 9** 事前配信コンテンツを再生するためのセットアップは、Content Router 要求ルーティングのセットアップとほぼ同じです。クライアント上の DNS サーバを、Content Router 情報を使用して設定した DNS サーバを指すように設定します（特に Windows システム上では、設定されたプライマリ DNS サーバだけに要求が送信されるように、セカンダリ DNS サーバのエントリを削除しなければならない場合があります）。
- ステップ 10** Request Routed FQDN URL を使用して、コンテンツをダウンロードまたは再生します。この例では、Request Routed FQDN は次のとおりです。

<http://ourtest.unicorn.com/index.html>



(注) 異なるオリジン サーバから取得されるコンテンツを同一 Request Routed FQDN の下に置くと、オーバーラップしてしまう可能性がある場合は、異なる Request Routed FQDN および異なる対応チャンネルを使用して、各オリジン サーバからコンテンツを取得してください。Web ページのリンクは、適切な Request Routed FQDN を参照するように変更する必要があります（上記のステップ 2a を参照）。

ケース 3: Content Router を使用した静的作成コンテンツの取得と配信

ケース 2 では、オリジン サーバには、同一オリジン サーバ上と異なるオリジン サーバ上のコンテンツへの静的リンク（クロール可能）があります。Content Router を使用してこのコンテンツを取得する手順は、次のとおりです。

ステップ 1 メイン オリジン サーバ上の公開コンテンツのコピーを作成し、新しいフォルダに入れます（この例では、複製フォルダは Folder3 です）。

ステップ 2 Folder3 内で、次の作業を行います。

- a. リモート サーバに対するすべての参照を、対応する Request Routed FQDN（この例では、*ourtest.unicorn.com*）に置き換えます。
- b. WMT ファイルに対するすべての参照で、ファイル名に拡張子 *.asx* を付けます。
- c. WMT ファイルに対するすべての参照で、*mms://* を *http://* に変換します。



(注) このステップと、ケース 1 の **ステップ 2** との主な相違点は、特にマニフェスト ファイル内で JavaScript のスクリプトからリンクされている項目を指定する必要があることです。ディレクトリ索引クロールか、単一項目のいずれか便利な方を使用できます。どちらのケースにも、メイン Web ページからの他のリンクの MMS クロールが適用できます。

ASX ファイルがある場合、そのファイルが参照するコンテンツは、メイン オリジン サーバからのコンテンツと同じチャンネルに属していなければなりません。このような場合は、ASX ファイルを変更しないでください。ACNS ソフトウェアは、他のチャンネル内でコンテンツを事前配信した ASX ファイルをサポートしません。

■ ケース 3 : Content Router を使用した静的作成コンテンツの取得と配信

ステップ 2 で説明した変更が加えられると、Folder3 内のコンテンツは、次のように変更されます (変更箇所は太字表記)。

```
[root@PC-45 Folder1]# cd ../Folder3
[root@PC-45 Folder3]# ls
01.asx  index.html

1. index.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
  <HEAD>
    <TITLE>Two-server combo</TITLE>
  </HEAD>
  <BODY>
    <H1>Index </H1>
    <ul>
      <li>
        <a href="01.asx">
          Start Windows Media Presentation1
        </a>

      <li>
        <a href="http://ourtest.unicorn.com/biff/http10k.txt">
          Regular http file
        </a>
      </li>
    </ul>
  </BODY></HTML>

2. 01.asx (unchanged):
<ASX Version = "3.0">
<AUTHOR> Biff</Author>
<Entry><Ref href = "http://mms-server/DCARoot/Consumer_100K.wmv" />
<Entry><Ref href = "mms://mms-server/DCARoot/ColorBars1_500K.wmv" />
</Entry>
</ASX>
```

ステップ 3 チャネルのマニフェスト ファイルで、次のように複数のジョブを設定します。

- a. メイン オリジン サーバの Content Router フォルダからファイルをクロールし、外部サーバからのコンテンツだけを取得するように、複数のジョブを設定します。
- b. メイン オリジン サーバの Folder3 フォルダからファイルをクロールし、メイン オリジン サーバからのコンテンツだけを取得するように、複数のジョブを設定します (次の例を参照)。

ケース 2 のサンプル マニフェスト ファイル

```
-----
<?xml version="1.0" ?>
<CdnManifest>

  <server name="cr-test2-1">
    <host name="http://www-server/biff/Folder3" proto="http"/>
  </server>

  <server name="cr-test2-2">
    <host name="http://www-server/biff/Folder" proto="http"/>
  </server>

  <crawler server="cr-test2-1" start-url="index.html" depth="2" ttl="5" >
  </crawler>

  <crawler server="cr-test2-2" start-url="index1.html" depth="2" ttl="5" >
  </crawler>

externalPrefixes="http://file-server.cisco.com/|mms://mms-server/|http://mms-server/">
  <matchRule>
    <match match-url="http://file-server.cisco.com/.*/>
    <match match-url="mms://mms-server/.*/>
    <match match-url="http://mms-server/.*/>
  </matchRule>
</crawler>
</CdnManifest>
~
```

- ステップ 4** たとえば、*ourtest.unicorn.com* を Request Routed FQDN として、Content Distribution Manager GUI 内で Web サイトを作成します。
- ステップ 5** **ステップ 4** で作成した Web サイトを使用して、Content Distribution Manager GUI でチャンネルを作成します。
- ステップ 6** DNS サーバ上で、ドメイン *ourtest.unicorn.com* 内のすべてのものを、要求の送信経路として指定された Content Router に送信するレコードを作成します。Content Router が、チャンネルに登録された Content Engine と同じ Content Distribution Manager に登録されていることを確認してください。
-

ケース 4 : WCCP ルーティングを使用した動的作成コンテンツの取得と配信

ケース 3 では、オリジン サーバは、JavaScript や類似のスクリプトを使用して記述されたスクリプトを介して、動的に作成されたコンテンツへのリンクを生成します。JavaScript のスクリプト内で参照されているリンクをたどって、動的コンテンツをダウンロードすることは不可能です。リンクの参照は、ACNS ソフトウェア内の取得機能であり、動的コンテンツの取得をサポートしないためです。

このケースでは、読者が WCCP 用のルータの設定、および WCCP を使用して要求される処理に必要なセットアップを十分理解していることを前提としています。WCCP ルーティングの詳細および背景情報については、第 4 章「ACNS ネットワークのコンテンツ要求ルーティング設定」を参照してください。

メイン オリジン サーバ上のファイル例

```
[root@PC-45 WCCP]# pwd
/var/ftp/biff/WCCP
[root@PC-45 WCCP]# ls
01.asx index.html
[root@PC-45 WCCP]#
1. index.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
<HEAD>
<TITLE>Two-server combo</TITLE>
</HEAD>
<BODY>
<H1>Index </H1>
<ul>
<li>
<a href="01.asx">
Start Windows Media Presentation1
</a>
<li>
<a href="http://mms-server/DCARoot/wmload.asf">
One Windows Media file
</a>
<li>
<a href="http://file-server.cisco.com/biff/http10k.txt">
Regular http file
</a>
<script language="JavaScript">
function doRedir(url)
{
document.location.href = url;
}
</script>
<li>
Click <a href="javascript:
doRedir('mms://mms-server/DCARoot/mbr/MBR_BillGInterview_Low.wmv');">here</a> for
BillG Interview.
<br>
<li>
Click <a href="javascript:
doRedir('mms://mms-server/DCARoot/mbr/MBR_BillGSpeech_Low.wmv');">here</a> for BillG
Speech.
</BODY></HTML>
2. 01.asx:
<ASX Version = "3.0">
<AUTHOR> Biff</Author>
<Entry><Ref href = "http://mms-server/DCARoot/Consumer_100K.wmv" />
<Entry><Ref href = "mms://mms-server/DCARoot/ColorBars1_500K.wmv" />
</Entry>
</ASX>
```

WCCP ルーティングを使用して、このような事前配信コンテンツを配信する手順は、次のとおりです。

-
- ステップ 1** オリジン サーバ上で、オリジナル フォルダの複製を作成します。たとえば、Folder2 にコピーします。
- ステップ 2** Folder2 内で、リモート サーバへのすべての参照を、対応する FQDN(この例では、www1.unicorn.com) に置き換えます。これ以外の変更は必要ありません。

同じオリジン サーバ上の他のファイルへのリンクが JavaScript を使用する場合、そのサーバ上でディレクトリ索引付け機能をイネーブルにする必要があります。これにより、必要なファイルがすべてこのオリジン サーバから取得できるようになります。

■ ケース 4 : WCCP ルーティングを使用した動的作成コンテンツの取得と配信

セカンダリ オリジン サーバ上で、異なるポートで受信する複製仮想ホストを作成できます。このホストはディレクトリ索引付け機能をイネーブルにし、特定のアクセス権を持ちます。この設定では、メイン Web ページは引き続き機能します。

ステップ 2 に記述されている変更を加えると、Folder2 内のコンテンツは次のように変更されます。

```
[root@PC-45 WCCP]# cd ../Folder2
[root@PC-45 Folder2]# ls
01.asx  index.html

1. index.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
  <HEAD>
    <TITLE>Two-server combo</TITLE>
  </HEAD>
  <BODY>
<H1>Index </H1>
<ul>
<li>
<a href="01.asx">
Start Windows Media Presentation1
</a>

<li>
<a href="http://www1.unicorn.com/DCARoot/wmload.asf">
One Windows Media file
</a>

<li>
<a href="http://www1.unicorn.com/biff/http10k.txt">
Regular http file
</a>

  <script language="JavaScript">
    function doRedir(url)
    {
      document.location.href = url;
    }
  </script>
<li>
  Click <a href="javascript:
doRedir('mms://www1.unicorn.com/DCARoot/mbr/MBR_BillGInterview_Low.wmv');">here</a>
for BillG Interview.
  <br>
<li>
  Click <a href="javascript:
doRedir('mms://www1.unicorn.com/DCARoot/mbr/MBR_BillGSpeech_Low.wmv');">here</a> for
BillG Speech.

</BODY></HTML>
2. 01.asx:
<ASX Version = "3.0">
<AUTHOR> Biff</Author>
<Entry><Ref href = "http://www1.unicorn.com/DCARoot/Consumer_100K.wmv" />
<Entry><Ref href = "mms://www1.unicorn.com/DCARoot/ColorBars1_500K.wmv" />
</Entry>
</ASX>
```

ステップ 3 チャンネルのマニフェスト ファイルで、次のように複数のジョブを設定します。

- a. 外部サーバをクロールせずに、メイン オリジン サーバの Folder2 からファイルを取得するために、クロール タスクまたは単一項目（どちらか便利な方）を設定します。
- b. 2 番めと 3 番めのオリジン サーバからファイルを取得するために、ディレクトリ クロールまたは単一項目を設定するか、外部サーバのクロールを許可したメイン オリジン サーバ Web サイトのクロールを設定します。

サンプル マニフェスト ファイル

```

-----
<?xml version="1.0" ?>
<CdnManifest>
  <server name="wccp-test1-1">
    <host name="http://www-server/biff/Folder2" proto="http"/>
  </server>

  <server name="wccp-test1-2">
    <host name="http://www-server/biff/WCCP" proto="http"/>
  </server>

  <crawler server="wccp-test1-1" start-url="index.html" depth="2" ttl="5" >
</crawler>

  <!-- These two come from the links from the JavaScript which cannot be crawled -->
  <!-- We can also use directory index crawling here -->
  <item src="mms://mms-server/DCARoot/mbr/MBR_BillGInterview_Low.wmv" />
  <item src="mms://mms-server/DCARoot/mbr/MBR_BillGSpeech_Low.wmv" />

  <crawler server="wccp-test1-2" start-url="index2.html" depth="2" ttl="5"
externalPrefixes="http://file-server.cisco.com/|mms://mms-server/|http://mms-server/">
  <matchRule>
    <match match-url="http://file-server.cisco.com/.*/>
    <match match-url="mms://mms-server/.*/>
    <match match-url="http://mms-server/.*/>
  </matchRule>
</crawler>
</CdnManifest>
~

```

ステップ 4 *www1.unicorn.com* (または適切な名前) をオリジン サーバ FQDN として、Content Distribution Manager GUI で Web サイトを作成します。



(注) たとえば、メイン Web サイトが *abc.xyz.com* でホスティングされる場合、単に Web サイト FQDN を *abc.xyz.com* として設定できます。

ステップ 5 **ステップ 4** で作成した Web サイトを使用して、Content Distribution Manager GUI でチャンネルを作成します。

ステップ 6 ルータと Content Engine を WCCP 用に設定します。

■ ケース 4 : WCCP ルーティングを使用した動的作成コンテンツの取得と配信

ステップ 7 新しいドメイン名の場合、**ステップ 4** で Content Distribution Manager GUI で設定した FQDN 用の DNS 解決を組み込むように、クライアントの DNS サーバを設定します。ドメイン名がメイン Web サイト FQDN と同じである場合、DNS サーバを設定する必要はありません。

コンテンツ複製が正常に完了すると、コンテンツは、すべて Content Engine によって処理されるので、FQDN が解決する Web サイトは問題ではありません。

ステップ 8 Content Engine が使用できる十分な帯域幅が、チャンネルに割り当てられていることを確認します。次に、必要な Content Engine をチャンネルに割り当てます。

ステップ 9 コンテンツ複製の状況を調べ、複製が完了したことを確認します。

ステップ 10 クライアント上の DNS サーバを、FQDN 情報を使用して設定した DNS サーバを指すように設定します（特に Windows システム上では、設定されたプライマリ DNS サーバだけに要求が送信されるように、セカンダリ DNS サーバのエントリを削除しなければならない場合があります）。

ステップ 11 WCCP がイネーブルになっているルータの適切なインターフェイスになるように、デフォルトゲートウェイを設定します。

ステップ 12 この FQDN URL を使用して、コンテンツをダウンロードし、再生します。

<http://www1.unicorn.com/index.html>



(注) 同じ FQDN の下に置くことで、別のオリジンサーバから取得されたコンテンツがオーバーラップする可能性がある場合は、別の FQDN および別の対応チャンネルを使用して、各オリジンサーバからコンテンツを取得してください。

ケース 5 : WCCP ルーティングを使用した静的作成コンテンツの取得と配信

ケース 4 では、オリジン サーバには、同一オリジン サーバ上と異なるオリジン サーバ上のコンテンツへの静的リンク（クローリング可能）があります。

完全を期するには、複数のチャンネルが設定されているケースを検討してください。取得コンテンツ保存元の各オリジン サーバごとにチャンネルを 1 つずつ設定します。ただし、異なるオリジン サーバから同一相対 URL が取得されるため、異なるオリジン サーバから取得されるコンテンツが重複する恐れがない場合、この作業は必要ありません。

メインオリジンサーバ上のファイル例

Original files:

```
[root@PC-45 WCCP]# pwd
/var/ftp/biff/WCCP
[root@PC-45 WCCP]# ls
01.asx index.html
[root@PC-45 WCCP]#
```

1. index.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
<HEAD>
<TITLE>Two-server combo</TITLE>
</HEAD>
<BODY>
<H1>Index </H1>
<ul>
<li>
<a href="01.asx">
Start Windows Media Presentation1
</a>

<li>
<a href="http://mms-server/DCARoot/wmload.asf">
One Windows Media file
</a>

<li>
<a href="http://file-server.cisco.com/biff/http10k.txt">
Regular http file

</a>

</BODY></HTML>
```

2. 01.asx:

```
<ASX Version = "3.0">
<AUTHOR> Biff</Author>
<Entry><Ref href = "http://mms-server/DCARoot/Consumer_100K.wmv" />
<Entry><Ref href = "mms://mms-server/DCARoot/ColorBars1_500K.wmv" />
</Entry>
</ASX>
```

WCCP ルーティングを使用して、このような事前配信コンテンツを配信する手順は、次のとおりです。

- ステップ 1** 各オリジン サーバからコンテンツを取得するためのマニフェスト ファイルを作成します。事前配信コンテンツ ファイルを変更する必要はありません。

サンプル マニフェスト ファイル

Manifest file for first channel (corresponding to the main origin server that is the starting point):

```
-----
<?xml version="1.0" ?>
<CdnManifest>
  <server name="wccp-test2-1">
    <host name="http://www-server/biff/WCCP" proto="http"/>
  </server>

  <crawler server="wccp-test2-1" start-url="index.html" depth="2" ttl="5" >
  </crawler>

</CdnManifest>
~
```

Manifest file for www2.unicorn.com:

```
-----
<?xml version="1.0" ?>
<CdnManifest>
  <server name="wccp-test2-2">
    <host name="http://www-server/biff/WCCP" proto="http"/>
  </server>

  <crawler server="wccp-test2-2" start-url="index.html" depth="2" ttl="5"
    externalPrefixes="http://file-server.cisco.com/">
    <matchRule>
      <match match-url="http://file-server.cisco.com/.*/>
    </matchRule>
  </crawler>
</CdnManifest>
~
```

Manifest file for www3.unicorn.com:

```
-----
<?xml version="1.0" ?>
<CdnManifest>
  <server name="wccp-test2-3">
    <host name="http://www-server/biff/WCCP" proto="http"/>
  </server>

  <crawler server="wccp-test2-3" start-url="index.html" depth="2" ttl="5"
    externalPrefixes="mms://mms-server/|http://mms-server/">
    <matchRule>
      <match match-url="mms://mms-server/.*/>
      <match match-url="http://mms-server/.*/>
    </matchRule>
  </crawler>
</CdnManifest>
~
```

- ステップ 2** Content Distribution Manager GUI 内で、1 つ以上の Web サイトを作成します。取得されたコンテンツが元々あったオリジン サーバのホスト名または IP アドレスを、オリジンサーバ FQDN として使用します。
- ステップ 3** ルータと Content Engine を WCCP 用に設定します。
- ステップ 4** クライアントの DNS サーバを設定し、そのサーバに、Content Distribution Manager GUI 内で設定した FQDN 用の DNS 解決が組み込まれていることを確認します。

- ステップ 5** ステップ 2 で作成した Web サイトを使用して、Content Distribution Manager GUI 内でチャンネルを作成します。
- ステップ 6** Content Engine が使用できる十分な帯域幅が、チャンネルに割り当てられていることを確認します。次に、必要な Content Engine をチャンネルに割り当てます。
- ステップ 7** コンテンツ複製の状況を調べ、すべてのチャンネルの複製が完了したことを確認します。
- ステップ 8** クライアント上の DNS サーバを、FQDN 情報を使用して設定した DNS サーバを指すように設定します（特に Windows システム上では、設定されたプライマリ DNS サーバだけに要求が送信されるように、セカンダリ DNS サーバのエントリを削除しなければならない場合があります）。
- ステップ 9** WCCP がイネーブルになっているルータの適切なインターフェイスになるように、デフォルトゲートウェイを設定します。
- ステップ 10** FQDN URL を使用して、コンテンツのダウンロードおよび再生を行います。

この情報は、CLI から **show statistics acquirer content channel-id channel-id** コマンドを使用して、ルート Content Engine 上で取得できます。この場合、<http://www-server/index.html> から始めます。

■ ケース 5 : WCCP ルーティングを使用した静的作成コンテンツの取得と配信