

MediaPlayer for WPF/Silverlight

2018.04.11 更新

グレースィティ株式会社

目次

製品の概要	3
ComponentOne for WPF/Silverlight のヘルプ	3
主な特長	4
WPF クイックスタート	5
手順 1: アプリケーションの作成	5
手順 2: コンテンツの追加	5-6
手順 3: アプリケーションの実行	6-7
Silverlight クイックスタート	8
手順 1: アプリケーションの作成	8-9
手順 2: コンテンツの追加	9-10
手順 3: アプリケーションの実行	10-12
XAML クイックリファレンス	13-15
MediaPlayer の使い方	16
メディアプレイヤーの要素	16-18
画面	18-19
項目リスト	19-20
チャプタリスト	20-21
ボリューム調整	21-22
タイムプレゼンタ	22
サポートされているファイルの種類	22-23
レイアウトおよび外観	24
ComponentOne ClearStyle 技術	24
ClearStyle の仕組み	24
ClearStyle プロパティ	24-25
テーマ	25-31
テンプレート	31-32
タスク別ヘルプ	33
メディアコンテンツを追加する	33-35
チャプタを作成する	35-36
自動再生を無効化する	36-37
メディアファイルをループ再生する	37-38

初期音量を設定する	38-39
ページロード時にチャプタリストを表示する	39-40
ページロード時に項目リストを表示する	40
テーマを使用する (Silverlight のみ)	40-42

製品の概要

MediaPlayer for WPF/Silverlight は、WPF/Silverlight がサポートするすべてのオーディオ/ビデオのメディア形式を再生するプレイヤーを提供します。プレイリスト、全画面表示モード、オーバーレイサポートなどの高度な機能を簡単に実装します。

 **メモ:** 説明内に含まれるクラスおよびメンバーに対するリファレンスへのリンクは、原則としてWPF版のリファレンスページを参照します。Silverlight版については、目次から同名のメンバーを参照してください。

ComponentOne for WPF/Silverlight のヘルプ

ComponentOne for WPF/Silverlight のすべてのコンポーネントで共通の使用方法については、「[ComponentOne for WPF/Silverlight ユーザーガイド](#)」を参照してください。

主な特長

MediaPlayer for WPF/Silverlight を使用して、機能豊富でカスタマイズされたアプリケーションを作成できます。以下の主要な機能をうまく利用して、MediaPlayer for WPF/Silverlight を最大限に活用してください。

- **プレイリストの作成**
独自のプレイリストをインラインで追加するか、またはその背後にあるコードに追加します。
- **フルスクリーンモード**
プレイヤーをフルスクリーンモードで表示するように簡単に設定します。
- **オーバーレイのサポート**
WPF/Silverlight メディアプレイヤーにオーバーレイを追加します。たとえば、ロゴ、バナー、および広告をオーバーレイできます。

WPF クイックスタート

このクイックスタートは、**MediaPlayer for WPF** を初めて使用するユーザーのために用意されています。このクイックスタートでは、最初に Visual Studio で新しいプロジェクトを作成し、アプリケーションに **C1MediaPlayer** コントロールを追加し、**C1MediaPlayer** コントロールにチャプタ分割されたコンテンツを追加します。

手順 1: アプリケーションの作成

この手順では、最初に Visual Studio で **MediaPlayer for WPF** を使用する WPF アプリケーションを作成します。次の手順に従います。

1. Visual Studio で、**[ファイル]**→**[新しいプロジェクト]**を選択します。
2. **[新しいプロジェクト]**ダイアログボックスの左ペインから言語を選択し、テンプレートリストから**[WPF アプリケーション]**を選択します。プロジェクトの名前を入力し、**[OK]**をクリックします。
3. **[分割]**ボタンをクリックして、分割ビューを表示します。
4. ツールボックスに移動し、**C1MediaPlayer**アイコンをダブルクリックして、**C1MediaPlayer** コントロールをプロジェクトに追加します。
5. **C1MediaPlayer** コントロールをクリックして選択します。
6. **[プロパティ]**ウィンドウで、次のプロパティを設定します。
 - **Height** プロパティを「300」に設定します。
 - **Width** プロパティを「350」に設定します。

これで、**C1MediaPlayer** コントロールを含む WPF アプリケーションを作成できました。次の手順では、**C1MediaPlayer** コントロールにビデオコンテンツを追加します。

手順 2: コンテンツの追加

クイックスタートチュートリアルはこのセクションでは、**C1MediaPlayer** コントロールに1つのチャプタから成るビデオを追加します。

次の手順に従います。

1. **[プロパティ]**ウィンドウで、**[Items]**の省略符ボタン  をクリックします。
2. **[コレクション エディター:Items]**ダイアログボックスが表示されます。
3. **[追加]**をクリックして、**C1MediaItem** 項目を **C1MediaPlayer** コントロールに追加します。
4. **[プロパティ]**グリッドで、次のプロパティを設定します。
5. **MediaSource** プロパティを「<http://ftp.grapecity.com/product/dotnet/xapoptimizer/movie/clientbin/xapoptimizerintroduce.wmv>」に設定します。これは、メディアソースの URL を設定します。
6. **Title** プロパティを「XapOptimizer 機能紹介」に設定します。これは、メディアのタイトルを指定します。
7. **NaturalDuration** プロパティを「00:00:33」に設定します。これは、再生時間のスタンプを 33秒に設定します。
8. **Chapters** の省略符ボタン  をクリックして、**[コレクション エディター:Chapters]**ダイアログボックスを開きます。

9. **[追加]**をクリックして、**C1MediaChapter** 項目を **C1MediaItem** 項目に追加します。
10. **[プロパティ]**グリッドで、**Position** プロパティを「00:02:48」に設定して、チャプタを開始する位置を指定します。
11. **[プロパティ]**グリッドで、**Title** プロパティを「クラスの解析」に設定して、チャプタのタイトルを作成します。

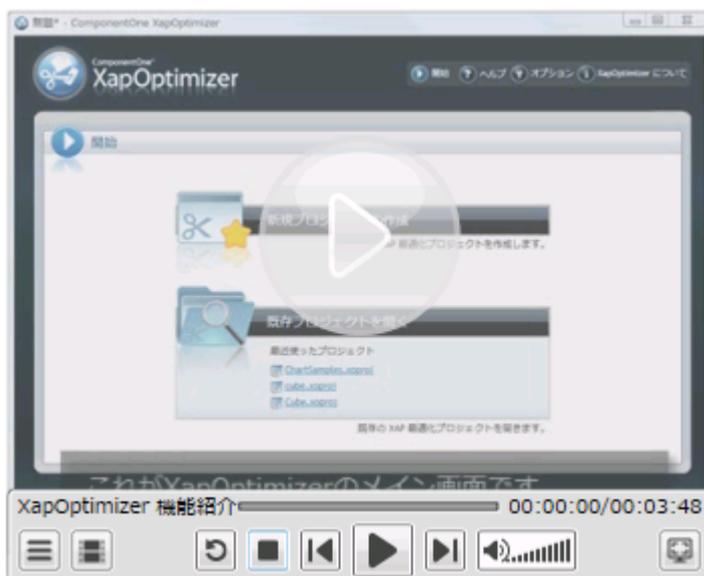
これで、1つのチャプタから成るビデオが **C1MediaPlayer** コントロールに追加されました。次の手順では、プロジェクトを実行し、**C1MediaPlayer** コントロールのいくつかの機能を確認します。

手順 3: アプリケーションの実行

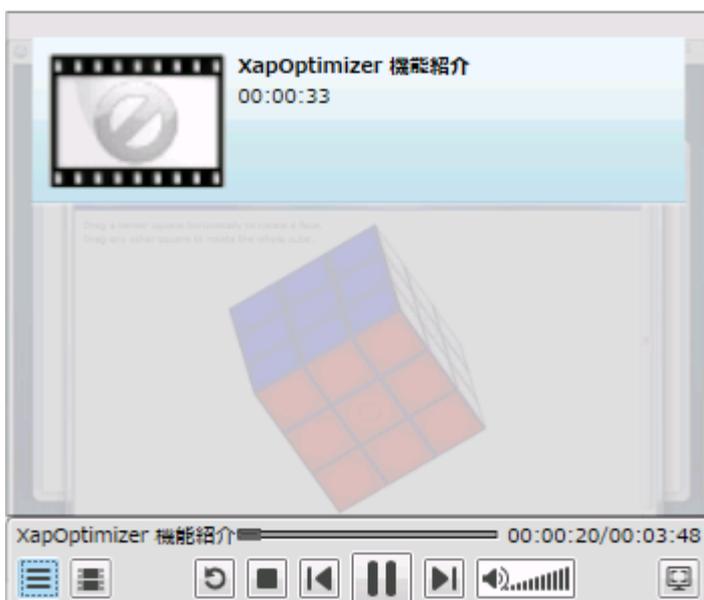
前の手順では、1つのチャプタから成るビデオを **C1MediaPlayer** コントロールに追加しました。この手順では、プロジェクトを実行し、**C1MediaPlayer** コントロールの実行時機能をいくつか確認します。

次の手順に従います。

1. **[デバッグ]**メニューから**[デバッグ開始]**を選択し、実行時にアプリケーションがどのように表示されるかを確認します。ビデオコンテンツが自動的に再生され、次のようなアプリケーションが表示されることを確認します。



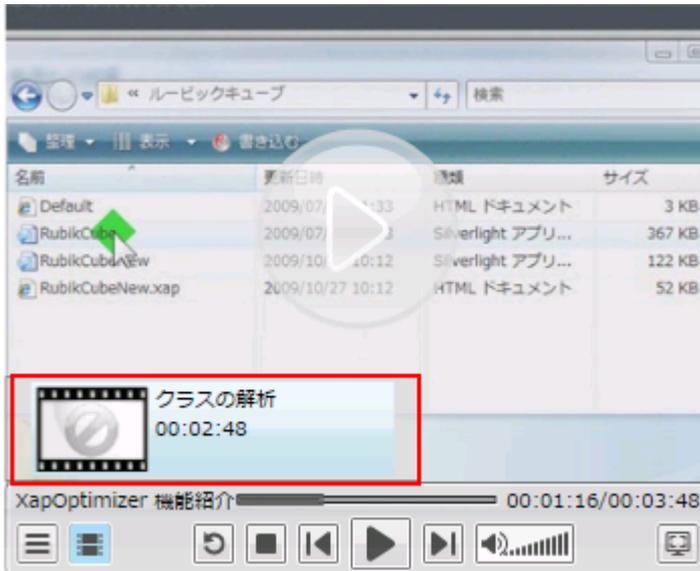
2. **[再生リスト]**ボタン  をクリックして項目リストを開きます。次のようになります。



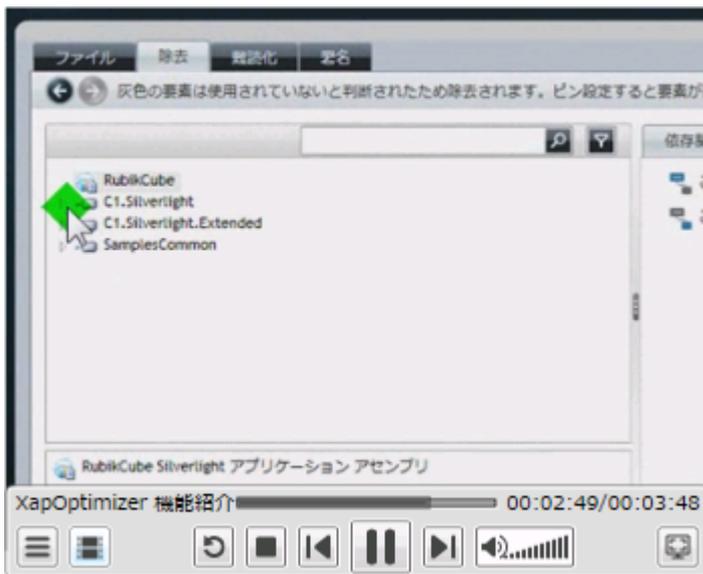
MediaPlayer for WPF/Silverlight

ビデオのタイトルと再生時間は、「手順 2: コンテンツの追加」の手順2と3で追加しました。

3. [チャプタリスト]ボタン  をクリックしてチャプタリストを開きます。次のようになります。



4. [クラスの解析]をクリックすると、ビデオがそのチャプタに早送りされることを確認します。メディアプレイヤーのコンテンツが次のように表示されます。



おめでとうございます。これで、**MediaPlayer for WPF** クイックスタートは終了です。このクイックスタートでは、**MediaPlayer for WPF** のアプリケーションを作成し、チャプタ分割されたビデオコンテンツを **C1MediaPlayer** コントロールに追加し、コントロールの実行時機能をいくつか確認しました。

Silverlight クイックスタート

このクイックスタートは、**MediaPlayer for Silverlight** を初めて使用するユーザーのために用意されています。このクイックスタートでは、最初に Visual Studio で新しいプロジェクトを作成し、アプリケーションに **C1MediaPlayer** コントロールを追加し、**C1MediaPlayer** コントロールにチャプタ分割されたコンテンツを追加します。

手順 1: アプリケーションの作成

この手順では、最初に Visual Studio で **MediaPlayer for Silverlight** を使用する Silverlight アプリケーションを作成します。次の手順に従います。

1. Visual Studio で、**[ファイル]→[新しいプロジェクト]**を選択します。
2. **[新しいプロジェクト]**ダイアログボックスの左ペインから言語を選択し、テンプレートリストから**[Silverlight アプリケーション]**を選択します。プロジェクトの名前を入力し、**[OK]**をクリックします。**[新しい Silverlight アプリケーション]**ダイアログボックスが表示されます。
3. **[OK]**をクリックすると、**[新しい Silverlight アプリケーション]**ダイアログボックスが閉じ、プロジェクトが作成されます。
4. プロジェクトの XAML ウィンドウで、`<UserControl>` タグの **DesignWidth="400" DesignHeight="300"** を **DesignWidth="Auto" DesignHeight="Auto"** に変更して、**UserControl** をサイズ変更します。次のようになります。

XAML

```
<UserControl x:Class="SilverlightApplication1.MainPage"
  xmlns=http://schemas.microsoft.com/winfx/2006/xaml/presentation
  xmlns:x=http://schemas.microsoft.com/winfx/2006/xaml
  xmlns:d=http://schemas.microsoft.com/expression/blend/2008
  xmlns:mc=http://schemas.openxmlformats.org/markup-compatibility/2006
  mc:Ignorable="d" d:DesignWidth="Auto" d:DesignHeight="Auto">
```

これで、**UserControl** は、中に置かれた内容に合わせてサイズ変更されるようになります。

5. ツールボックスに移動し、**C1MediaPlayer** アイコンをダブルクリックして、コントロールをグリッドに追加します。XAML マークアップは次のようになります。

XAML

```
<Grid x:Name="LayoutRoot">
  <c1:C1MediaPlayer></c1:C1MediaPlayer>
</Grid>
</UserControl>
```

6. `<c1:C1MediaPlayer>` タグに `Height="300"` を追加し、コントロールの高さを設定します。XAML マークアップは次のようになります。

XAML

```
<c1:C1MediaPlayer Height="300">
```

7. `<c1:C1MediaPlayer>` タグに `Width="350"` を追加し、コントロールの幅を設定します。XAML マークアップは次のようになります。

XAML

```
<c1:C1MediaPlayer Height="300" Width="350">
```

これで、**C1MediaPlayer** コントロールを含む Silverlight アプリケーションを作成できました。次の手順では、**C1MediaPlayer** コントロールにビデオコンテンツを追加します。

手順 2: コンテンツの追加

クイックスタートチュートリアルのこのセクションでは、**C1MediaPlayer** コントロールに1つのチャプタから成るビデオを追加します。

次の手順に従います。

1. `<c1:C1MediaPlayer>` タグと `</c1:C1MediaPlayer>` タグの間に次の XAML マークアップを配置して、ビデオコンテンツを **C1MediaPlayer** コントロールに追加します。

XAML

```
<c1:C1MediaItem  
MediaSource="http://ftp.grapecity.com/product/dotnet/xapoptimizer/  
movie/clientbin/xapoptimizerintroduce.wmv"></c1:C1MediaItem>
```

2. `<c1:C1MediaItem>` タグに **Title="XapOptimizer 機能紹介"** を追加して、ビデオのタイトルを追加します。マークアップは、次のようになります。

XAML

```
<c1:C1MediaItem  
MediaSource="http://ftp.grapecity.com/product/dotnet/xapoptimizer/  
movie/clientbin/xapoptimizerintroduce.wmv" Title="XapOptimizer 機能紹介">
```

3. `<c1:C1MediaItem>` タグに **NaturalDuration="00:00:33"** を追加して、**NaturalDuration** プロパティを 33 秒に設定します。マークアップは、次のようになります。

XAML

```
<c1:C1MediaItem  
MediaSource="http://ftp.grapecity.com/product/dotnet/xapoptimizer/  
movie/clientbin/xapoptimizerintroduce.wmv" Title="XapOptimizer 機能紹介"  
NaturalDuration="00:00:33">
```

NaturalDuration プロパティには、ユーザーが項目リストを表示したときに表示される再生時間を設定します。

4. `<c1:C1MediaItem>` タグと `</c1:C1MediaItem>` タグの間に次の XAML マークアップを配置して、ビデオコンテンツにチャプタを追加します。

XAML

```
<c1:C1MediaChapter></c1:C1MediaChapter>
```

5. `<c1:C1MediaChapter>` タグに **Position="00:02:48"** を追加して、チャプタの位置を指定します。マークアップは、次のようになります。

XAML

```
<c1:C1MediaChapter Position="00:02:48">
```

6. <c1:C1MediaChapter> タグに **Title="クラスの解析"** を追加して、チャプタのタイトルを作成します。マークアップは、次のようになります。

XAML

```
<c1:C1MediaChapter Position="00:02:48" Title="クラスの解析">
```

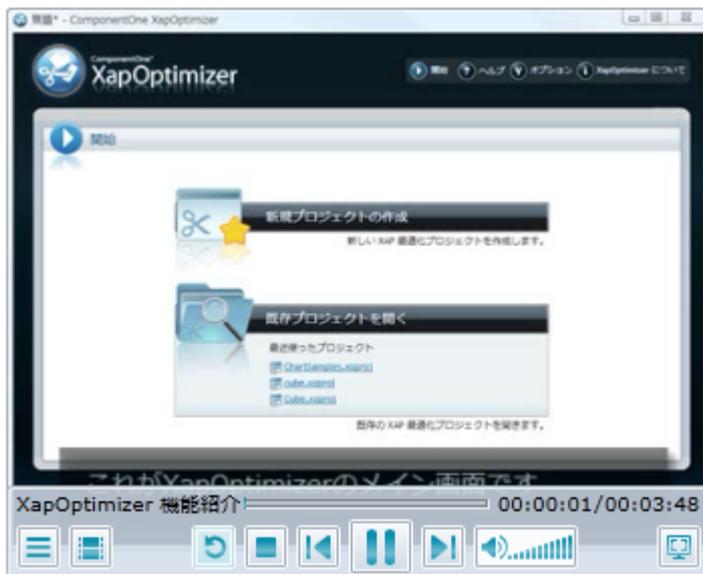
これで、1つのチャプタから成るビデオが **C1MediaPlayer** コントロールに追加されました。次の手順では、プロジェクトを実行し、**C1MediaPlayer** コントロールのいくつかの機能を確認します。

手順 3: アプリケーションの実行

前の手順では、1つのチャプタから成るビデオを **C1MediaPlayer** コントロールに追加しました。この手順では、プロジェクトを実行し、**C1MediaPlayer** コントロールの実行時機能をいくつか確認します。

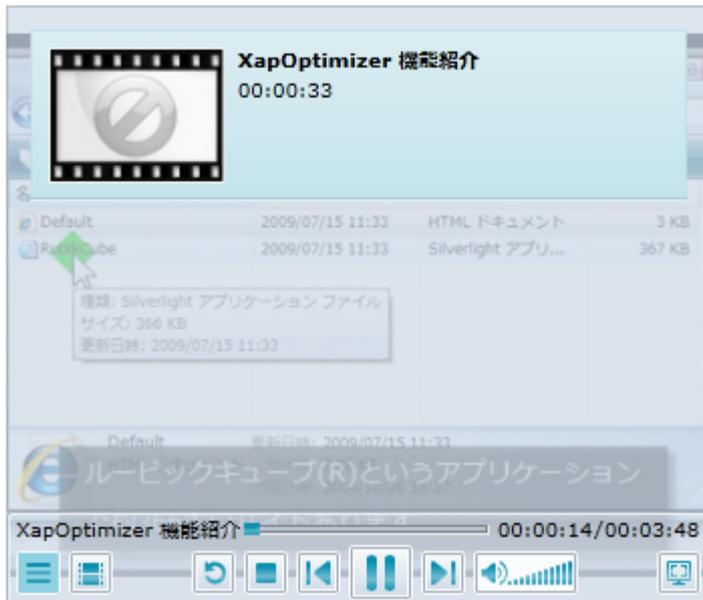
次の手順に従います。

1. **[デバッグ]**メニューから**[デバッグ開始]**を選択し、実行時にアプリケーションがどのように表示されるかを確認します。ビデオコンテンツが自動的に再生され、次のようなアプリケーションが表示されることを確認します。



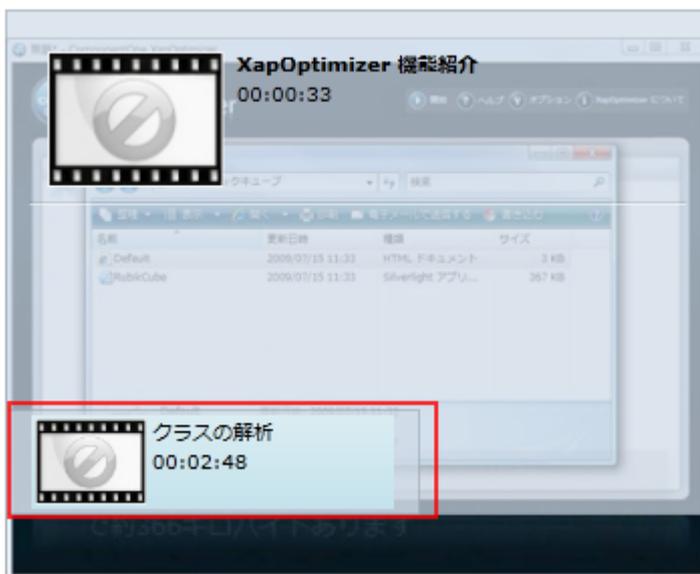
ビデオの再生開始から数秒後に、ツールバーは消えます。

2. **[再生リスト]**ボタン  をクリックして項目リストを開きます。次のようになります。

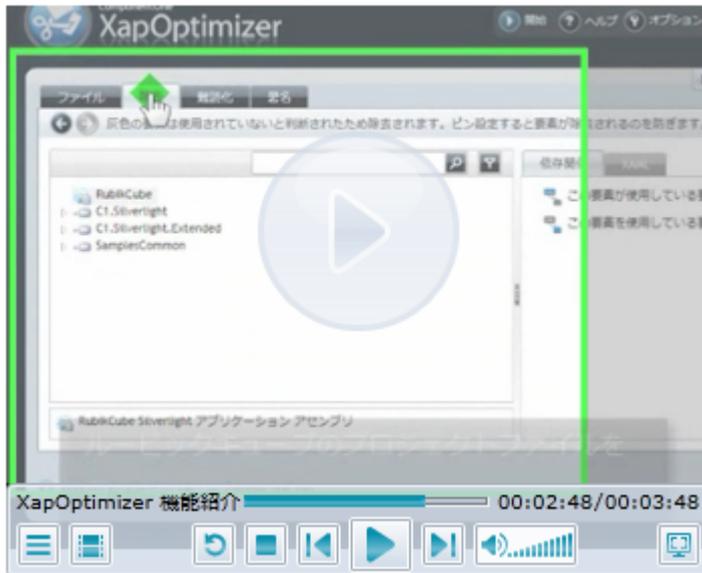


ビデオのタイトルと再生時間は、「手順 2: コンテンツの追加」の手順 2 と 3 で追加しました。

3. [チャプターリスト] ボタン  をクリックしてチャプターリストを開きます。次のようになります。



4. [クラスの解析] をクリックすると、ビデオがそのチャプタに早送りされることを確認します。メディアプレイヤーのコンテンツが次のように表示されます。



おめでとうございます。これで、**MediaPlayer for Silverlight** クイックスタートは終了です。このクイックスタートでは、**MediaPlayer for Silverlight** のアプリケーションを作成し、チャプタ分割されたビデオコンテンツを **C1MediaPlayer** コントロールに追加し、コントロールの実行時機能をいくつか確認しました。

XAML クイックリファレンス

このトピックでは、さまざまな **C1MediaPlayer** タスクの実行に使用される XAML の概要を提供します。詳細については、「[タスク別ヘルプ](#)」セクションを参照してください。

このセクションの XAML マークアップでは、3つのメディア項目を含む **C1MediaPlayer** コントロールを作成する方法を示します。各メディア項目には、3つのチャプタが含まれます。このコントロールには、ComponentOne パッケージに付属する **RainierOrange** テーマが設定されます。

XAML

```
<!--テーマ-->
<c1:C1ThemeRainierOrange>
<!--MediaPlayer マークアップ-->
    <c1:C1MediaPlayer HorizontalAlignment="Left" Margin="10,10,0,0"

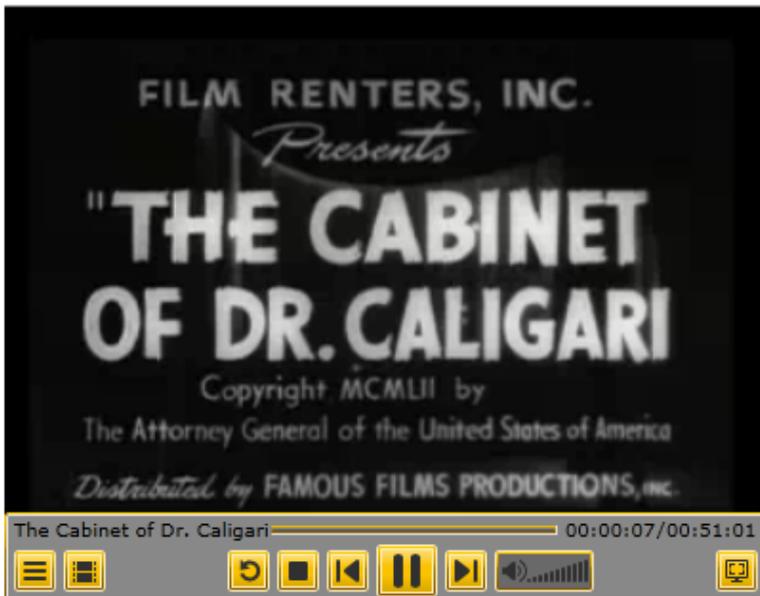
Name="c1MediaPlayer1" VerticalAlignment="Top" Height="329" Width="400">
<!--最初の MediaItem-->
    <c1:C1MediaItem MediaSource="http://ia600300.us.archive.org/22/items/

DasKabinettdesDoktorCaligariTheCabinetofDrCaligari/The_Cabinet_of_Dr._Caligari_512kb.mp4"

Title="The Cabinet of Dr. Caligari" NaturalDuration="02:00:00">
<!--最初の MediaItem のチャプタ-->
    <c1:C1MediaChapter Title="Title Sequence" Position="00:01:00" />
    <c1:C1MediaChapter Title="Intermission" Position="00:35:00" />
    <c1:C1MediaChapter Title="End Credits" Position="01:55:00" />
    </c1:C1MediaItem >
<!--2番目の MediaItem-->
    <c1:C1MediaItem MediaSource="Dracula.mp4" Title="Dracula" NaturalDuration="02:35:00">
<!--2番目の MediaItem のチャプタ-->
    <c1:C1MediaChapter Title="Title Sequence" Position="00:00:20" />
    <c1:C1MediaChapter Title="Intermission" Position="00:32:00" />
    <c1:C1MediaChapter Title="End Credits" Position="02:32:00" />
    </c1:C1MediaItem>
<!--3番目の MediaItem-->
    <c1:C1MediaItem MediaSource="Frankenstein.mp4" Title="Frankenstein"

NaturalDuration="01:38:00">
<!--3番目の MediaItem のチャプタ-->
    <c1:C1MediaChapter Title="Title Sequence" Position="00:00:30" />
    <c1:C1MediaChapter Title="Intermission" Position="00:42:00" />
    <c1:C1MediaChapter Title="End Credits" Position="01:35:00" />
    </c1:C1MediaItem>
</c1:C1MediaPlayer>
</c1:C1ThemeRainierOrange>
```

起動時に、**C1MediaPlayer** は次のように表示されます。



[項目リスト]ボタンをクリックすると、リストに追加された3つの項目が表示されます。



次に、[チャプタリスト]ボタンをクリックすると、現在オンになっている項目のチャプタが表示されます。

MediaPlayer for WPF/Silverlight



MediaPlayer の使い方

C1MediaPlayer コントロールは、ユーザーにオーディオ/ビデオを配信するために使用するメディアデバイスです。**C1MediaPlayer** コントロールを使用すると、コンテンツのプレイリストを作成できるほか、長いビデオを管理が容易なチャプタに分割することもできます。このコントロールは、WPF/Silverlight がサポートするすべてのメディア形式を扱うことができます。



C1MediaPlayer コントロールをプロジェクトに追加した後は、その外観と動作をカスタマイズできます。それにより、テーマやグラデーションの適用、ビデオのオーバーレイの作成、フルスクリーンビューの選択、ボタンの追加や削除、その他の多くの機能を実行できます。

C1MediaItem 項目を作成し、その **MediaSource** プロパティに対して画像、ビデオ、またはオーディオのファイルの URL を設定すると、**C1MediaPlayer** コントロールにコンテンツを追加できます。

以下の一連のトピックでは、この作業を開始できるように、**C1MediaPlayer** コントロールの一般的な要素や機能の概要について説明します。

メディアプレイヤーの要素

このセクションでは、**C1MediaPlayer** コントロールを構成する要素について画像を使用してわかりやすく説明します。次の図に、**C1MediaPlayer** コントロールのデフォルトの要素を示します。

MediaPlayer for WPF/Silverlight



C1MediaPlayer の各要素の説明については、次の表を参照してください。

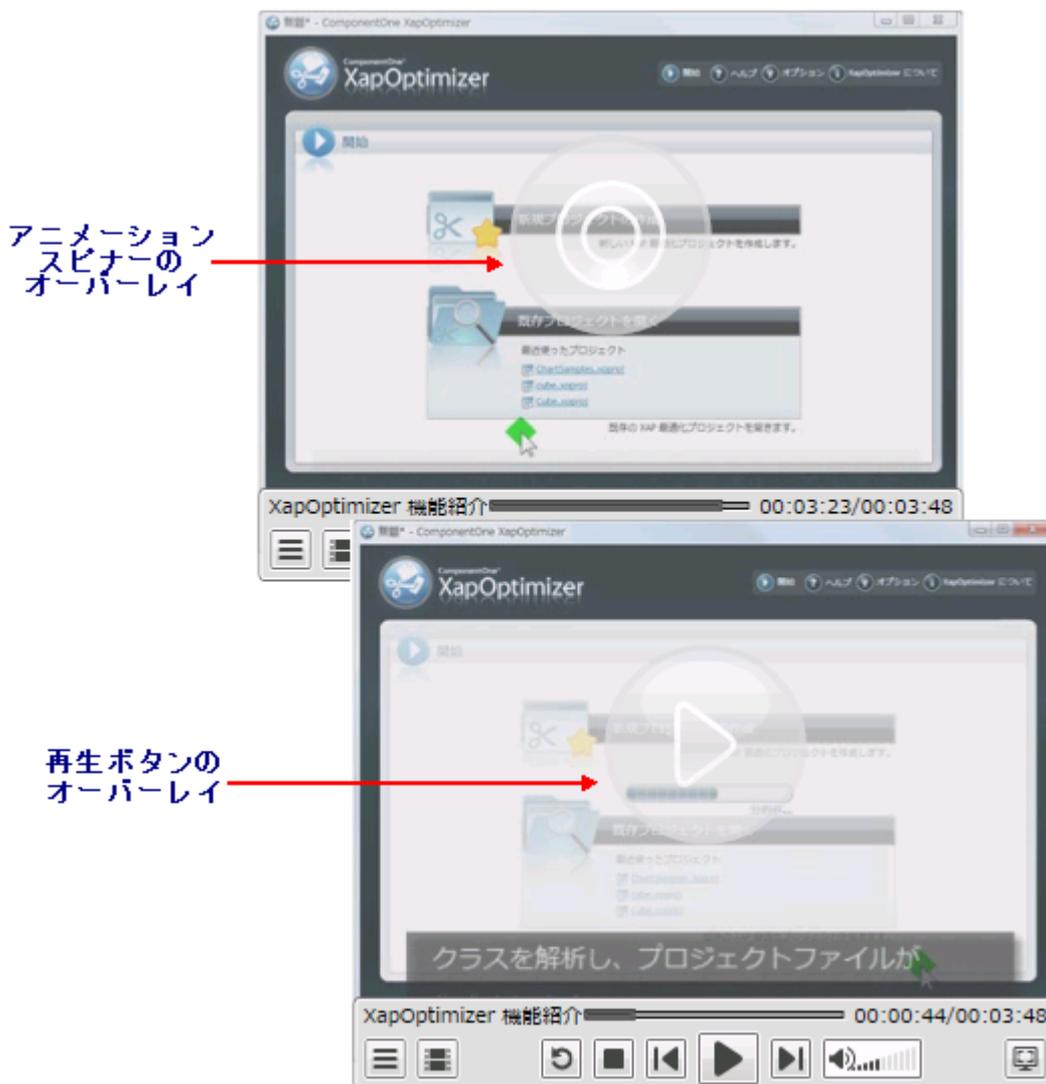
要素	説明
画面	C1MediaPlayer のコンテンツ領域には、ビデオが表示されます。コンテンツ領域の詳細については、「画面」を参照してください。
ツールバー	ツールバーには、すべてのメディアプレイヤー Title プロパティコントロールが表示されます。ツールバーは、起動時と、ツールバーの上にカーソルが置かれたときにのみ表示されます。
タイトル	タイトルは、Title プロパティを使って設定され、画像、ビデオ、または音楽のファイルのタイトルを表示するために使用されます。タイトルはデフォルトで表示されますが、IsTitleVisible プロパティを False に設定して非表示にすることもできます。
再生リスト	[再生リスト] ボタンにより、コントロールに追加されているすべてのメディア項目を含むプレイリストである項目リストを表示または非表示にできます。IsItemListButtonVisible プロパティを False に設定すると、[再生リスト] ボタンを非表示にできます。
チャプターリスト	[チャプターリスト] ボタンは、現在のメディア項目に追加されているすべてのチャプタを表示するチャプターリストを表示または非表示にできます。IsChapterListButtonVisible プロパティを False に設定すると、[チャプターリスト] ボタンを非表示にできます。
連続再生	[連続再生] ボタンは、ループのオン/オフを行います。ループ再生はデフォルトでオフですが、IsLooping プロパティを True に設定してオンにすることもできます。IsLoopButtonVisible プロパティを False に設定すると、[連続再生] ボタンを非表示にできます。
停止	[停止] ボタンは、C1MediaPlayer コントロールによるコンテンツの再生を停止します。[停止] ボタンを非表示にするには、IsStopButtonVisible プロパティを False に設定します。
前へ	[前へ] ボタンによって前のメディアファイルに巻き戻すことができます。IsPreviousButtonVisible プロパティを False に設定すると、[前へ] ボタンを非表示にできます。
再生/一時停止	[再生/一時停止] ボタンにより、開いているメディアファイルの再生と一時停止を行うことができます。このボタンは、メディアファイルが停止中、再生中、一時停止中のどれであるかによって変化します。メディアプレイヤーが停止中または一時停止中の場合は[再生] ボタンが表示され、メディアプレイヤーが再生

	中の場合は[一時停止]ボタンが表示されます。
次へ	[次へ]ボタンによって次のメディアファイルに早送りできます。 IsNextButtonVisible プロパティを False に設定すると、[次へ]ボタンを非表示にできます。
音量	音量スライダは、表示中のメディアファイルのコンテンツを制御します。ミュートボタン をクリックすると、サウンドをミュートできます。
全画面表示	[全画面表示]ボタンにより、C1MediaPlayer コントロールのフルスクリーン表示のオン/オフを切り替えることができます。コントロールがフルスクリーンモードの場合は、もう一度[全画面表示]ボタンをクリックするか、キーボードの[ESC]キーを押すことにより、通常のサイズに戻すことができます。[全画面表示]ボタンを非表示にするには、 IsFullScreenButtonVisible プロパティを False に設定します。
タイムプレゼンタ	タイムプレゼンタには、現在のメディアファイルの経過時間と残り時間の両方が表示されます。タイムプレゼンタを非表示にするには、 IsTimePresenterVisible プロパティを False に設定します。
ステータスバー	ステータスバーには、現在のメディアファイルの経過時間と残り時間の両方が示されます。これは、現在のメディアファイルの再生位置を変更するためのスライダバーでもあります。ステータスバーは、 IsPositionSliderVisible プロパティを False 設定すると非表示になります。

画面

C1MediaPlayer コントロールの画面にはビデオが表示されます。**C1MediaPlayer** コントロールで MP3、WMA ファイルなどの視覚トラックを含まないメディアファイルを再生する場合、コンテンツ領域は黒の背景色で表示されます。最初、画面の下端はツールバーで隠されますが、その後ツールバーは消えて、ユーザーがツールバーの上にカーソルを置くまで再表示されません。

画面の全体または一部に複数のオーバーレイを表示することもできます。ファイルのバッファリング中は、コンテンツ領域にアニメーションスピナーのオーバーレイが表示されます。ファイルが一時停止または停止されると、コンテンツ領域に再生ボタンのオーバーレイが表示されます。ユーザーはこのオーバーレイをクリックしてファイルの再生を開始できます。

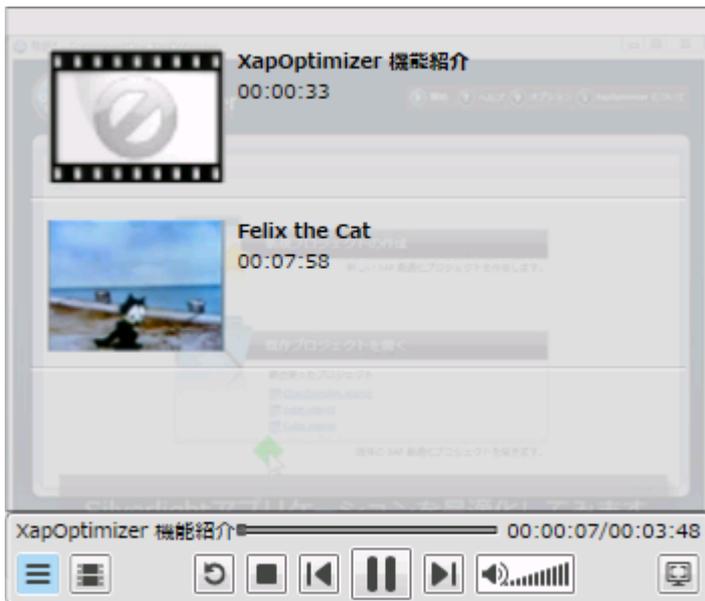


画面には、項目リストとチャプタリストを表示することもできます。再生可能なメディアファイルの一覧である項目リストは、画面全体に表示されます。現在のメディアファイル内のチャプタを表示するチャプタリストは、長方形のオーバーレイで表示されます。

ScreenExtension テンプレートを追加してからカスタマイズすることにより、カスタムのオーバーレイを作成できます。この種のオーバーレイは、著作権情報、ビデオの説明、または広告を表示するために使用できます。

項目リスト

項目リストは、メディアプレイヤーに追加されたコンテンツの一覧です。アップ/ダウンボタンを使ってリストをスクロールし、メディアファイルの1つをクリックして選択できます。項目リストは、次のようになります。



上の図のような項目リストの外観や動作を実現するには、いくつかのプロパティを設定する必要があります。メディア項目にタイトルを付けるには、**C1MediaItem** の **Title** プロパティに文字列を設定する必要があります。ビデオにタイムスタンプを追加するには、**NaturalDuration** プロパティを設定します。**ThumbnailSource** プロパティに画像ファイルの場所を設定すると、サムネイルを追加できます。

このリストの最初の項目を作成するために、次の XAML を使用しました。

XAML

```
<clmediaplayer:C1MediaItem
  MediaSource="http://www.acme.org/download/public_domain_cartoon.mp4"
  ThumbnailSource="FelixMiddle.png" NaturalDuration="00:07:58" Title="Felix the Cat" >
```

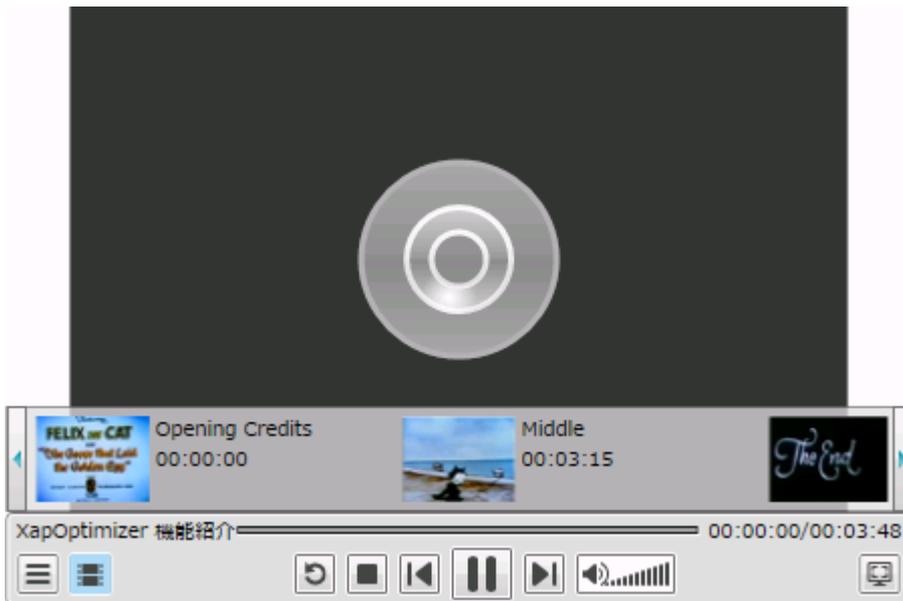
メディア項目を作成した後は、チャプタを作成してチャプタリストに追加できます。チャプタリストの詳細については、「[チャプタリスト](#)」を参照してください。

項目リストへのアクセス

項目リストには、**C1MediaPlayer** コントロールの [**再生リスト**] ボタン  をクリックしてアクセスできます。起動時に項目リストを開く場合は、**C1MediaPlayer.IsItemVisible** プロパティを **True** に設定します。

チャプタリスト

チャプタリストには、現在のメディア項目に追加されたすべてのチャプタが表示されます。左右の矢印ボタンを使用してチャプタリストをスクロールし、チャプタをクリックして選択できます。チャプタリストは、次のようになります。



上の図のようなチャプタリストの外観や動作を実現するには、いくつかのプロパティを設定する必要があります。メディア項目にタイトルを付けるには、**C1MediaChapter.Title** プロパティに文字列を設定する必要があります。**C1MediaChapter.ThumbnailSource** プロパティに画像ファイルを設定すると、サムネイルを追加できます。最後に、**C1MediaChapter.Position** プロパティに時間を設定して、チャプタが開始されるまでの時間を指定する必要があります。上の図のようなチャプタリストを作成するには、次の XAML を使用します。

XAML

```
<c1:clmediaplayer:C1MediaChapter
Title="Opening Credits"
Position="00:00:00"
ThumbnailSource="FelixOpen.png"/>
<c1:C1MediaChapter
Title="Middle"
Position="00:03:15"
ThumbnailSource="FelixMiddle.png"/>
<c1:C1MediaChapter
Title="End Credits"
Position="00:07:04"
ThumbnailSource="FelixEnd.png"/>
```

メディア項目へのチャプタの追加に関するタスク別ヘルプについては、「[チャプタを作成する](#)」を参照してください。

チャプタリストへのアクセス

チャプタリストには、**C1MediaPlayer** コントロールの [チャプターリスト] ボタン  をクリックしてアクセスできます。起動時に項目リストを開く場合は、**IsChapterListVisible** プロパティを **True** に設定します。

ボリューム調整

メディアのボリュームは、ボリューム調整スライダを操作して制御できます。ボリューム調整スライダは、Windows Media Player のスライダと同様に機能します。左にスライドするとボリュームが小さくなり、右にスライドすると大きくなり、ミュートボタン  をクリックするとサウンドがミュートされます。

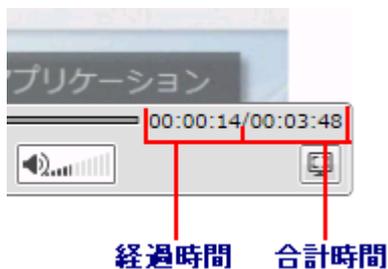


初期ボリュームの制御

C1MediaPlayer コントロールの初期ボリュームは、**Volume** プロパティを0と1の間の値に設定して指定できます。0に設定するとボリュームはオフになり、1に設定するとボリュームは最大になり、0.5に設定するとボリュームはスケールの中間値になります。実行時にコントロールをミュートする場合は、**IsMuted** プロパティを **True** に設定することもできます。

タイムプレゼンタ

位置要素は、経過時間と合計時間の2つの時間マーカーを提供します。経過時間は、タイムプレゼンタの左側に表示され、合計時間は右側に表示されます。



デフォルトでは、経過時間はゼロから始まります。ただし、**Position** プロパティに時間を設定すると、メディアファイルの開始位置を変更できます。プロジェクトを実行する場合、メディアファイルは指定したポイントから再生され、タイムプレゼンタの経過時間部分は、その位置から時間計測を開始します。

サポートされているファイルの種類

現在、**C1MediaPlayer** コントロールは、次のメディアタイプをサポートします。

ビデオ

サポートされているビデオ形式は次のとおりです。

- 未加工ビデオ
- YV12 - YCrCb (4:2:0)
- RGBA - 32 ビット (Alpha, Red, Green, Blue)
- WMV、MV2、MV3 (Windows Media Video 7、8、9)
- シンプルプロファイルとメインプロファイルをサポート
- プログレッシブ (ノンインターレース) コンテンツのみをサポート
- WMVA: Windows Media Video のアドバンスドプロファイル、非 VC-1
- WVC1: Windows Media Video のアドバンスドプロファイル、VC-1
- アドバンスドプロファイルをサポート
- プログレッシブ (ノンインターレース) コンテンツのみをサポート
- h164 (ITU-T H.264/ISO MPEG-4 AVC)

オーディオ

サポートされているオーディオ形式は次のとおりです。

- "1" - WAV 形式
- "353" - Microsoft Windows Media Audio v7、v8、v9.x Standard (WMA Standard)
- "354" - Microsoft Windows Media Audio v9.x、v10 Professional (WMA Professional)
- "85" - ISO MP3 (MPEG-1 Layer III)
- "255" - ISO AAC (Advanced Audio Coding)

レイアウトおよび外観

以下のトピックでは、**C1MediaPlayer** コントロールのレイアウトと外観をカスタマイズする方法について詳しく説明します。組み込みのレイアウトオプションを使用して、グリッドやキャンバスなどのコントロールをパネル内でレイアウトできます。テーマを使用することで、グリッドの外観をカスタマイズしたり、WPF/Silverlight の XAML ベースのスタイル設定を活用することができます。また、テンプレートを使用して、コントロールを書式設定およびレイアウトしたり、コントロールの操作をカスタマイズすることもできます。

ComponentOne ClearStyle 技術

ClearStyle は、WPF コントロールと Silverlight コントロールのスタイル設定をすばやく簡単に実行できる新技術です。ClearStyle を使用すると、面倒な XAML テンプレートやスタイルリソースを操作しなくても、コントロールのカスタムスタイルを作成できます。

現在のところ、すべての標準 WPF/Silverlight コントロールにテーマを追加するには、スタイルリソーステンプレートを作成する必要があります。Microsoft Visual Studio ではこの処理は困難であるため、Microsoft は、このタスクを簡単に実行できるように Expression Blend を導入しました。ただし、Blend に不慣れであったり、十分な学習時間を取れない開発者にとっては、この2つの環境を行き来することはかなり困難な作業です。デザイナーに作業を任せるとも考えられますが、デザイナーと開発者が XAML ファイルを共有すると、かえって煩雑になる可能性があります。

このような場合に、ClearStyle を使用します。ClearStyle は、Visual Studio を使用して直感的な方法でスタイル設定を実行できるようにします。ほとんどの場合は、アプリケーション内のコントロールに対して単純なスタイル変更を行うだけなので、この処理は簡単に行えるべきです。たとえば、データグリッドの行の色を変更するだけであれば、1つのプロパティを設定するだけで簡単に行えるようにする必要があります。一部の色を変更するためだけに、完全に複雑なテンプレートを作成する必要はありません。

ClearStyle の仕組み

コントロールのスタイルの主な要素は、それぞれ単純な色プロパティとして表されます。これが集まって、コントロール固有のスタイルプロパティセットを形成します。たとえば、**Gauge** には **PointerFill** プロパティや **PointerStroke** プロパティがあり、**DataGrid** の行には **SelectedBrush** や **MouseOverBrush** があります。

たとえば、フォーム上に ClearStyle をサポートしていないコントロールがあるとします。その場合は、ClearStyle によって作成された XAML リソースを使用して、フォーム上の他のコントロールを調整して合わせるすることができます（正確な色合わせなど）。また、スタイルセットの一部を ClearStyle (カスタムスクロールバーなど) で上書きしたいとします。ClearStyle は拡張可能なのでこれも可能です。必要な場所でスタイルを上書きできます。

ClearStyle は、すばやく簡単にスタイルを変更することを意図したソリューションですが、ComponentOne のコントロールには引き続き従来の方法を使用して、必要なスタイルを細かく指定して作成できます。完全なカスタム設計が必要になる特別な状況で ClearStyle が邪魔になることはありません。

ClearStyle プロパティ

MediaPlayer for WPF/Silverlight は、コントロールのテンプレートを変更することなくコントロールの色を簡単に変更できる ComponentOne の新しい ClearStyle 技術をサポートします。色のプロパティをいくつか設定するだけで、グリッド全体のスタイルを簡単に設定できます。

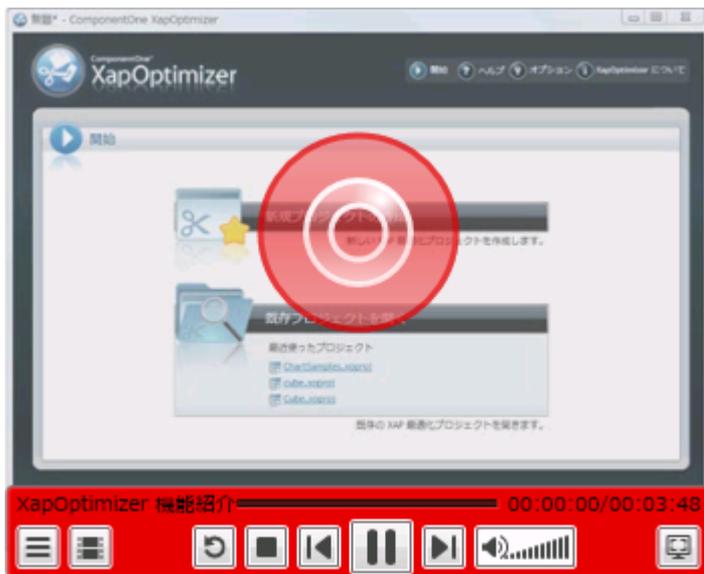
次の表に、**C1MediaPlayer** コントロールのブラシのプロパティの概要を示します。

ブラシ	説明
Background	コントロールの背景のブラシを取得または設定します。
ButtonBackground	ボタンの背景色のブラシを取得または設定します。

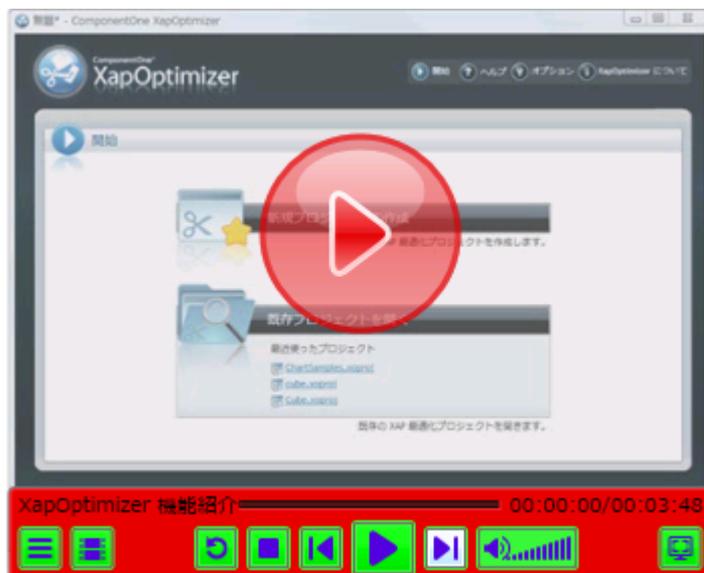
MediaPlayer for WPF/Silverlight

ButtonForeground	ボタンの前景色(ボタン上のシンボルなど)のブラシを取得または設定します。
MouseOverBrush /td>	マウスポインタが置かれたボタンを強調表示するために使用される System.Windows.Media.Brush を取得または設定します。
PressedBrush	クリックされたボタンを強調表示するために使用される System.Windows.Media.Brush を取得または設定します。<

いくつかのプロパティを設定することで、**C1MediaPlayer** コントロールの外観を完全に変更できます。たとえば、**Background** は、メディアプレイヤーの背景色を設定します。**Background** プロパティを "#FFE40005" に設定すると、**C1MediaPlayer** コントロールは次のようになります。



赤色に対してよく映えるようにボタンを黄緑色に表示する場合は、**ButtonForeground** プロパティと **ButtonBackground** プロパティを設定します。次の例では、**ButtonForeground** プロパティを "#FF5500DE" に設定し、**ButtonBackground** プロパティを "#FF00F500" に設定しています。



ComponentOne の ClearStyle 技術は、このように簡単に使用できます。ClearStyle の詳細については、「[ComponentOne ClearStyle 技術](#)」のトピックを参照してください。

テーマ

MediaPlayer for WPF/Silverlight には、グリッドの外観をカスタマイズできるいくつかのテーマが組み込まれています。**C1MediaPlayer** コントロールを初めてページに追加すると、次の図のように表示されます。

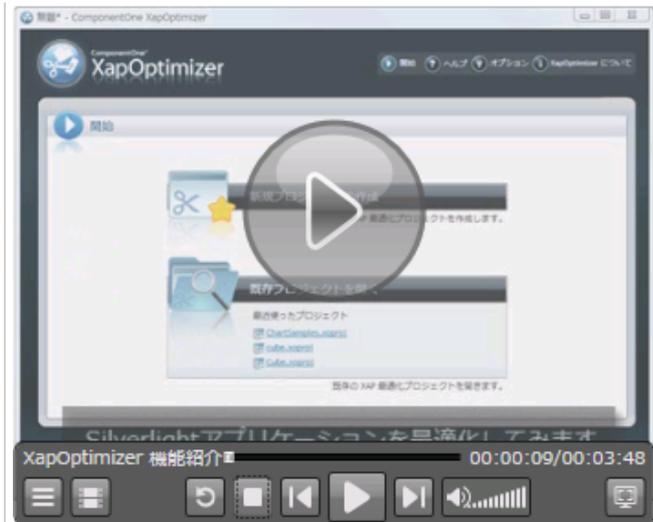


これは、このコントロールのデフォルトの外観です。この外観は、組み込みテーマの1つを使用したり、独自のカスタムテーマを作成することで変更できます。すべての組み込みテーマは、WPF Toolkit テーマに基づいています。以下に、組み込みのテーマについて説明および図示します。

完全なテーマ名	外観
C1ThemeBureauBlack	
C1Blue (WPF のみ)	

MediaPlayer for WPF/Silverlight

C1ThemeExpressionDark



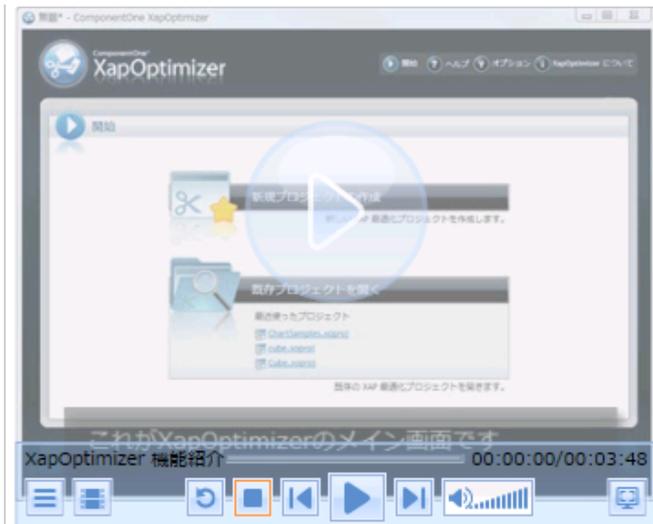
C1ThemeExpressionLight



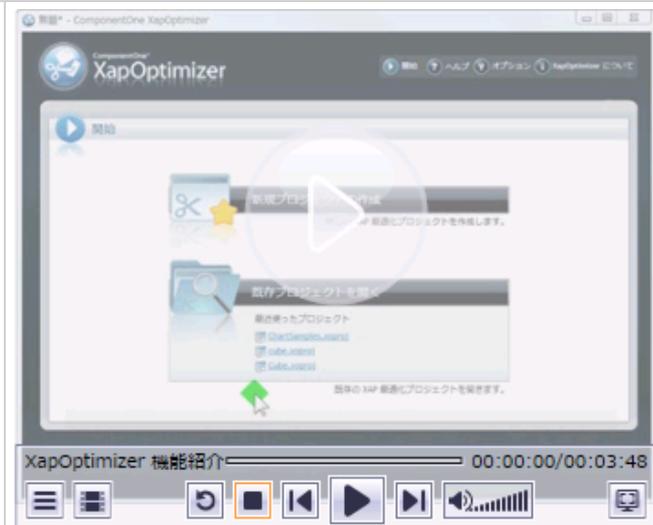
C1ThemeOffice2007Black



C1ThemeOffice2007Blue



C1ThemeOffice2007Silver



C1ThemeOffice2010Black



MediaPlayer for WPF/Silverlight

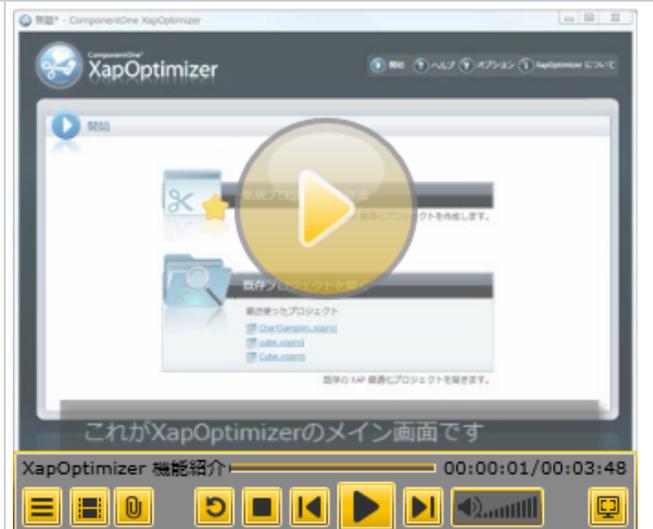
C1ThemeOffice2010Blue



C1ThemeOffice2010Silver



C1ThemeRainierOrange (Silverlight のみ)



C1ThemeShinyBlue



C1ThemeWhistlerBlue



要素のテーマを設定するには、**ApplyTheme** メソッドを使用します。最初に、テーマアセンブリへの参照をプロジェクトに追加し、次のようにコードでテーマを設定します。

VisualBasic

```
Private Sub Window_Loaded(sender As System.Object, e As System.Windows.RoutedEventArgs)
    Handles MyBase.Loaded
    Dim theme As New C1ThemeExpressionDark
    ' ApplyTheme の使用
    C1Theme.ApplyTheme(LayoutRoot, theme)
End Sub
```

C#

```
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    C1ThemeExpressionDark theme = new C1ThemeExpressionDark();
    //ApplyTheme の使用
    C1Theme.ApplyTheme(LayoutRoot, theme);
}
```

アプリケーション全体にテーマを適用するには、**System.Windows.ResourceDictionary.MergedDictionaries** プロパティを使用します。最初に、テーマアセンブリへの参照をプロジェクトに追加し、次のようにコードでテーマを設定します。

VisualBasic

```
Private Sub Window_Loaded(sender As System.Object, e As System.Windows.RoutedEventArgs)
    Handles MyBase.Loaded
    Dim theme As New C1ThemeExpressionDark
    ' MergedDictionaries の使用
    Application.Current.Resources.MergedDictionaries.Add(C1Theme.GetCurrentThemeResources(theme))
End Sub
```

C#

```
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    C1ThemeExpressionDark theme = new C1ThemeExpressionDark();
    //MergedDictionaries の使用
    Application.Current.Resources.MergedDictionaries.Add(C1Theme.GetCurrentThemeResources(theme));
}
```

この方法は、初めてテーマを適用する場合にのみ使用できることに注意してください。別の ComponentOne テーマに切り替える場合は、最初に、**Application.Current.Resources.MergedDictionaries** から前のテーマを削除します。

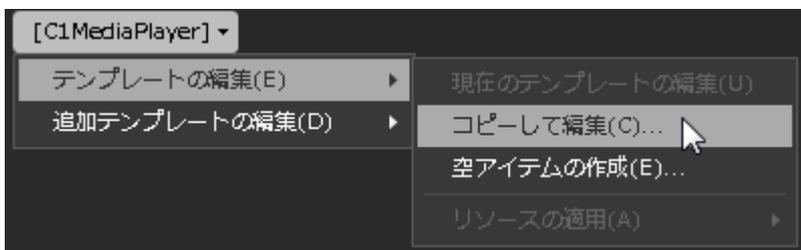
 **メモ:** Silverlight の場合は、BureauBlack、ExpressionDark、ExpressionLight、RainierOrange、ShinyBlue、WhistlerBlue の6つのテーマが提供されています。

テンプレート

WPF/Silverlight コントロールを使用する主な利点の1つは、これが自由にカスタマイズできるユーザーインターフェイスを持つ「外観のない」コントロールであることです。WPF/Silverlight アプリケーションのユーザーインターフェイスであるルックアンドフィールを独自に設計するのと同様に、**MediaPlayer for WPF/Silverlight** で管理されるデータに関して独自の UI を提供できます。Extensible Application Markup Language (XAML; 「ザムル」と発音する) は、コードを記述することなく独自の UI を設計するための簡単な方法を提供する XML ベースの宣言型言語です。

テンプレートへのアクセス

テンプレートにアクセスするには、Microsoft Expression Blend で、**C1MediaPlayer** コントロールを選択し、メニューから **[テンプレートの編集]** を選択します。[コピーして編集] を選択して現在のテンプレートの **コピーを作成して編集するか**、**[空アイテムの作成]** を選択して新しい空のテンプレートを作成します。

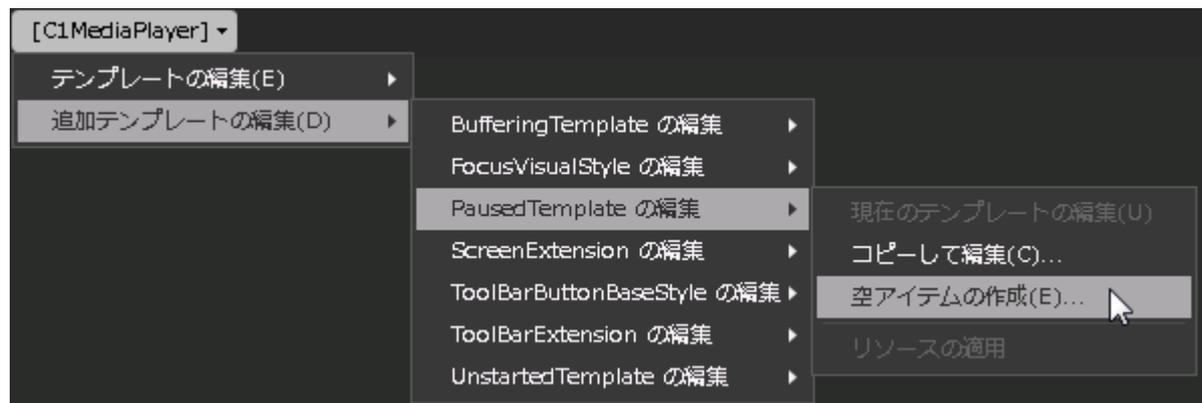


新しく作成されたテンプレートは、[オブジェクトとタイムライン] ウィンドウに表示されます。Template プロパティを使用してテンプレートをカスタマイズできます。

 メニューを使用して新しいテンプレートを作成する場合、テンプレートはそのテンプレートのプロパティに自動的にリンクされます。手作業でテンプレートの XAML を作成する場合は、作成したテンプレートに適切な Template プロパティをリンクする必要があります。

追加のテンプレート

デフォルトテンプレートのほかに、**C1MediaPlayer** コントロールには追加のテンプレートがいくつかあります。これらの追加テンプレートには、Microsoft Expression Blend からアクセスできます。Blend で **C1MediaPlayer** コントロールを選択し、メニューから**追加テンプレートの編集**を選択します。テンプレートを選択し、**空アイテムの作成**を選択します。



次の表では、追加テンプレートについて説明します。

テンプレート	説明
BufferingTemplate	コンテンツの状態がバッファリングのときに画面に表示されるオーバーレイのテンプレートです。
PausedTemplate	コンテンツの状態が一時停止のときに画面に表示されるオーバーレイのテンプレートです。
ScreenExtension	ビデオ領域のオーバーレイのためのテンプレートです(オプション)。
ToolBarExtension	項目リスト/チャプタリストの表示/非表示など、ツールバーに対するオプションの拡張機能のテンプレートです。
UnstartedTemplate	コンテンツの状態が再生開始前のときに画面に表示されるオーバーレイのテンプレートです。

タスク別ヘルプ

タスク別ヘルプは、ユーザーの皆様が Visual Studio のプログラミングに精通しており、**C1MediaPlayer** コントロールを使用する方法を理解していることを前提としています。**MediaPlayer for WPF/Silverlight** 製品を初めて使用される場合は、まず [WPF クイックスタート](#) および [Silverlight クイックスタート](#) を参照してください。

このセクションの各トピックは、**MediaPlayer for WPF/Silverlight** 製品を使用して特定のタスクを行うための方法を提供します。

また、タスク別ヘルプトピックは、新しい WPF/Silverlight プロジェクトが既に作成されていることを前提としています。

メディアコンテンツを追加する

メディアコンテンツを追加するには、**C1MediaItem** オブジェクトを作成し、**MediaSource** プロパティにメディアファイルの場所を設定します。このトピックでは、**[プロパティ]** ウィンドウ、XAML、およびコードで、ビデオファイルをメディアプレイヤーに追加します。

Visual Studio での設計時

ビデオコンテンツを追加するには、次の手順に従います。

1. **C1MediaPlayer** コントロールをクリックして選択します。
2. **Items** プロパティを見つけ、その省略符ボタン  をクリックします。
[コレクション エディター:Items] ダイアログボックスが開きます。
3. **[追加]** をクリックして、**C1MediaItem** オブジェクトを **C1MediaPlayer** コントロールに追加します。
4. **[プロパティ]** グリッドで、次の内容を設定します。
 - **MediaSource** プロパティを「<http://ftp.grapecity.com/product/dotnet/xapoptimizer/movie/clientbin/xapoptimizerintroduce.wmv>」に設定します。
 - **Title** プロパティを「XapOptimizer 機能紹介」に設定します。
5. **[OK]** をクリックして、**[コレクション エディター:Items]** ダイアログボックスを閉じます。
6. プログラムを実行し、コントロールにコンテンツがロードされていることを確認します。

Blend での設計時

Blend を使ってビデオコンテンツを追加するには、次の手順に従います。

1. **C1MediaPlayer** コントロールをクリックして選択します。
2. **[プロパティ]** ウィンドウで、**Items** プロパティを見つけ、省略符ボタンをクリックします。
[C1MediaItem コレクション エディター: Items] ダイアログボックスが開きます。
3. **[別のアイテムを追加]** をクリックして、**C1MediaPlayer** コントロールに **C1MediaItem** オブジェクトを追加します。
4. **[プロパティ]** グリッドで、次の内容を設定します。
 - **MediaSource** プロパティを「<http://ftp.grapecity.com/product/dotnet/xapoptimizer/movie/clientbin/xapoptimizerintroduce.wmv>」に設定します。
 - **Title** プロパティを「XapOptimizer 機能紹介」に設定します。

5. **[OK]**をクリックして、**[C1MediaItem コレクション エディター: Items]**ダイアログボックスを閉じます。
6. プロジェクトを実行します。

XAML の場合

XAML を使用してビデオコンテンツを追加するには、次の手順に従います。

1. 次のマークアップを `<c1:C1MediaPlayer>` タグと `</c1:C1MediaPlayer>` タグの間に追加します。

XAML

```
<c1:C1MediaItem
MediaSource="http://ftp.grapecity.com/product/dotnet/xapoptimizer/movie/
clientbin/xapoptimizerintroduce.wmv" Title="XapOptimizer 機能紹介"
Name="C1MediaPlayer1" />
```

2. プログラムを実行し、コントロールにコンテンツがロードされていることを確認します。

コードの場合

コードを使用してビデオコンテンツを追加するには、次の手順に従います。

1. **Window1.xaml.cs** ページを開き、次の名前空間をインポートします。

WPF

VisualBasic

```
Imports Cl.WPF.MediaPlayer
```

C#

```
using Cl.WPF.MediaPlayer;
```

Silverlight

VisualBasic

```
Imports Cl.Silverlight.MediaPlayer
```

C#

```
using Cl.Silverlight.MediaPlayer;
```

2. **InitializeComponent()** メソッドの下に次のコードを追加します。

VisualBasic

```
'C1MediaItem オブジェクトを作成する
Dim C1MediaItem1 As New C1MediaItem()
'メディアファイルのパスを含む URI オブジェクトを作成する
Dim Uri1 As New Uri("http://ftp.grapecity.com/product/dotnet/xapoptimizer/movie/
```

MediaPlayer for WPF/Silverlight

```
clientbin/xapoptimizerintroduce.wmv")
'C1MediaItems コンテンツのソースを URI オブジェクトに設定する
C1MediaItem1.MediaSource = Uri1
'メディア項目に名前を付ける
C1MediaItem1.Title = "XapOptimizer 機能紹介"
'メディア項目をメディアプレイヤーに追加する
C1MediaPlayer1.Items.Add(C1MediaItem1)
```

C#

```
//C1MediaItem オブジェクトを作成する
C1MediaItem C1MediaItem1 = new C1MediaItem();
//メディアファイルのパスを含む URI オブジェクトを作成する
Uri Uri1 = new Uri("http://ftp.grapecity.com/product/dotnet/xapoptimizer/movie/
clientbin/xapoptimizerintroduce.wmv");
//C1MediaItems コンテンツのソースを URI オブジェクトに設定する
C1MediaItem1.MediaSource = Uri1;
//メディア項目に名前を付ける
C1MediaItem1.Title = "XapOptimizer 機能紹介";
//メディア項目をメディアプレイヤーに追加する
c1MediaPlayer1.Items.Add(C1MediaItem1);
```

3. プログラムを実行し、コントロールにコンテンツがロードされていることを確認します。

チャプタを作成する

チャプタを作成するには、**C1MediaChapter** 項目を作成し、**Position** プロパティを設定します。このトピックでは、「[メディアコンテンツを追加する](#)」というタスク別ヘルプトピックを完了していることを前提としています。

Visual Studio での設計時

チャプタを作成するには、次の手順に従います。

1. **C1MediaPlayer** コントロールをクリックして選択します。
2. [プロパティ] ウィンドウで、**Items** プロパティの省略符ボタン  をクリックします。
[コレクション エディター:Items] ダイアログボックスが開きます。
3. [項目] ペインで、チャプタを追加する **C1MediaItem** を選択します。
4. [プロパティ] ペインで、**Chapters** プロパティの省略符ボタン  をクリックします。
5. [コレクション エディター:Chapters] ダイアログボックスが開きます。
6. 次の手順に従って、チャプタを作成します。
 - **[追加]** をクリックしてチャプタを追加します。
 - **[項目の選択]** ペインに **C1MediaChapter** 項目が追加されます。
 - **Position** プロパティに時間を設定します。この例では、メディアファイルの1分 25 秒にチャプタを作成するために、「00:01:25」に設定します。
 - **Title** プロパティに名前を設定します。この例では、「クラスの解析」に設定します。

7. **[OK]**をクリックして、**[コレクション エディター: Chapters]**ダイアログボックスを閉じます。
8. **[OK]**をクリックして、**[コレクション エディター: Items]**ダイアログボックスを閉じます。
9. プログラムを実行し、**[チャプタリスト]**ボタン をクリックします。「**クラスの解析**」というチャプタがリストに表示されることを確認します。これをダブルクリックすると、メディアファイルのその部分までスキップできます。

Blend での設計時

チャプタを作成するには、次の手順に従います。

1. **C1MediaPlayer** コントロールをクリックして選択します。
2. **[プロパティ]**ウィンドウで、**Items** プロパティを見つけ、省略符ボタンをクリックします。**[C1MediaItem コレクション エディター: Items]**ダイアログボックスが開きます。
3. **[Items]**ペインで、チャプタを追加する **C1MediaItem** を選択します。
4. **[プロパティ]**ペインで、**Chapters** プロパティを見つけ、省略符ボタンをクリックします。
5. **[C1MediaItem コレクション エディター: Chapters]**ダイアログボックスが開きます。
6. 次の手順に従ってチャプタを作成します。
 - **[別のアイテムを追加]**をクリックしてチャプタを追加します。**[Items]**ペインに **C1MediaChapter** 項目が追加されます。
 - **Position** プロパティに時間を設定します。この例では、メディアファイルの1分 25 秒にチャプタを作成するために、「00:01:25」に設定します。
 - **Title** プロパティに名前を設定します。この例では、「クラスの解析」に設定します。
7. **[OK]**をクリックして、**[C1MediaItem コレクション エディター: Chapters]**ダイアログボックスを閉じます。
8. **[OK]**をクリックして、**[C1MediaItem コレクション エディター: Items]**ダイアログボックスを閉じます。
9. プログラムを実行し、**[チャプタリスト]**ボタン  をクリックします。「**クラスの解析**」というチャプタがリストに表示されることを確認します。これをダブルクリックすると、メディアファイルのその部分までスキップできます。

XAML の場合

チャプタを作成するには、次の手順に従います。

1. 次のマークアップを `<c1:C1MediaItem>` タグと `</c1:C1MediaItem>` タグの間に配置します。

XAML

```
<c1:C1MediaChapter Position="00:01:25" Title="クラスの解析"/>
```

2. プログラムを実行し、**[チャプタリスト]**ボタン  をクリックします。「**クラスの解析**」というチャプタがリストに表示されることを確認します。これをダブルクリックすると、メディアファイルのその部分までスキップできます。

自動再生を無効化する

デフォルトでは、**C1MediaPlayer** コントロールはオーディオファイルを自動再生しますが、**AutoPlay** プロパティを **False** に設定すると、この機能をオフにできます。

MediaPlayer for WPF/Silverlight

設計時

自動再生をオフにするには、次の手順に従います。

1. **C1MediaPlayer** コントロールをクリックして選択します。
2. **[プロパティ]** ウィンドウで、**[AutoPlay]** チェックボックスを見つけてオフにします。

XAML の場合

自動再生をオフにするには、**AutoPlay="False"** を `<c1:C1MediaPlayer>` タグに追加します。マークアップは次のようになります。

XAML

```
<c1:C1MediaPlayer Name="C1MediaPlayer1" AutoPlay="False">
```

コードの場合

自動再生をオフにするには、次の手順に従います。

1. Window1.xaml.cs ページを開きます。
2. **x:Name="C1MediaPlayer1"** を `<c1:C1MediaPlayer>` タグに追加します。これにより、このコントロールをコード内で呼び出すための一意の識別子が指定されます。
3. **InitializeComponent()** メソッドの下に次のコードを追加します。

Visual Basic

```
C1MediaPlayer1.AutoPlay = False
```

C#

```
c1MediaPlayer1.AutoPlay = false;
```

4. プログラムを実行します。

メディアファイルをループ再生する

デフォルトでは、**C1MediaPlayer** コントロールは、その項目リストを使用して1回だけ自動再生されます。ただし、**IsLooping** プロパティを **True** に設定すると、項目を繰り返してループ再生できます。

設計時

ループ再生をオンにするには、次の手順に従います。

1. **C1MediaPlayer** コントロールをクリックして選択します。
2. **[プロパティ]** ウィンドウで、**[IsLooping]** チェックボックスを見つけてオンにします。

XAML の場合

ループ再生をオンにするには、**IsLooping="True"** を `<c1:C1MediaPlayer>` タグに追加します。マークアップは次のようになります。

XAML

```
<c1:C1MediaPlayer Name="C1MediaPlayer1" IsLooping="True">
```

コードの場合

ループ再生をオンにするには、次の手順に従います。

1. Window1.xaml.cs ページを開きます。
2. **x>Name="C1MediaPlayer1"** を `<c1:C1MediaPlayer>` タグに追加します。これにより、このコントロールをコード内で呼び出すための一意の識別子が指定されます。
3. **InitializeComponent()** メソッドの下に次のコードを追加します。

VisualBasic

```
C1MediaPlayer1.IsLooping = True
```

C#

```
c1MediaPlayer1.IsLooping = true;
```

4. プログラムを実行します。

初期音量を設定する

C1MediaPlayer コントロールの初期ボリュームは、**Volume** プロパティを0と1の間の値に設定して指定できます。このトピックでは、ページのロード時にボリュームが中間点に設定されるように設定します。

設計時

ボリュームを設定するには、次の手順に従います。

1. **C1MediaPlayer** コントロールをクリックして選択します。
2. **[プロパティ]** ウィンドウで、**Volume** プロパティを「0.5」に設定します。
3. プログラムを実行し、**C1MediaPlayer** コントロールのボリュームが中間点に設定されることを確認します。

XAML の場合

音量を設定するには、次の手順に従います。

1. **Volume="0.5"** を `<c1:C1MediaPlayer>` タグに追加します。マークアップは次のようになります。

XAML

```
<c1:C1MediaPlayer x>Name="C1MediaPlayer1" Volume="0.5">
```

2. プログラムを実行し、**C1MediaPlayer** コントロールの音量が中間点に設定されることを確認します。

コードの場合

音量を設定するには、次の手順に従います。

MediaPlayer for WPF/Silverlight

1. Window1.xaml.cs ページを開きます。
2. **x:Name="C1MediaPlayer1"** を `<c1:C1MediaPlayer>` タグに追加します。これにより、このコントロールをコード内で呼び出すための一意の識別子が指定されます。
3. **InitializeComponent()** メソッドの下に次のコードを追加します。

Visual Basic

```
C1MediaPlayer1.Volume = 0.5
```

C#

```
C1MediaPlayer1.Volume = 0.5;
```

4. プログラムを実行し、**C1MediaPlayer** コントロールの音量が中間点に設定されることを確認します。

ページロード時にチャプタリストを表示する

デフォルトでは、**C1MediaPlayer** コントロールのロード時にチャプタリストは表示されません。ユーザーは[チャプタリスト]ボタンをクリックして、リストを開く必要があります。ただし、**IsChapterListVisible** プロパティを True に設定すると、ページのロード時にチャプタリストを表示できます。

設計時

ページのロード時にチャプタリストを開くには、次の手順に従います。

1. **C1MediaPlayer** コントロールをクリックして選択します。
2. [プロパティ]ウィンドウで、[**IsChapterListVisible**]チェックボックスを見つけてオンにします。
3. プログラムを実行し、実行時にチャプタリストが開くことを確認します。

XAML の場合

ページのロード時にチャプタリストを開くには、次の手順に従います。

1. **IsChapterListVisible="True"** を `<c1:C1MediaPlayer>` タグに追加します。マークアップは次のようになります。

XAML

```
<c1:C1MediaPlayer x:Name="C1MediaPlayer1" IsChapterListVisible="True">
```

2. プログラムを実行し、実行時にチャプタリストが開くことを確認します。

コードの場合

ページのロード時にチャプタリストを開くには、次の手順に従います。

1. **Window1.xaml.cs** ページを開きます。
2. **x:Name="C1MediaPlayer1"** を `<c1:C1MediaPlayer>` タグに追加します。これにより、このコントロールをコード内で呼び出すための一意の識別子が指定されます。
3. **InitializeComponent()** メソッドの下に次のコードを追加します。

Visual Basic

```
C1MediaPlayer1.IsChapterListVisible = True
```

C#

```
C1MediaPlayer1.IsChapterListVisible = true;
```

4. プログラムを実行し、実行時にチャプタリストが開くことを確認します。

ページロード時に項目リストを表示する

デフォルトでは、**C1MediaPlayer** コントロールのロード時に項目リストは表示されません。ユーザーは[項目リスト]ボタンをクリックして、リストを開く必要があります。ただし、**IsItemVisible** プロパティを **True** に設定すると、ページのロード時に項目リストを表示できます。

設計時

ページのロード時に項目リストを開くには、次の手順に従います。

1. **C1MediaPlayer** コントロールをクリックして選択します。
2. [プロパティ]ウィンドウで、[**IsItemVisible**]チェックボックスを見つけてオンにします。
3. プログラムを実行し、実行時に項目リストが開くことを確認します。

XAML の場合

ページのロード時に項目リストを開くには、次の手順に従います。

1. **IsItemVisible="True"** を <c1:C1MediaPlayer> タグに追加します。マークアップは次のようになります。

XAML

```
<c1:C1MediaPlayer x:Name="C1MediaPlayer1" IsItemVisible="True">
```

2. プログラムを実行し、実行時に項目リストが開くことを確認します。

コードの場合

ページのロード時に項目リストを開くには、次の手順に従います。

1. **Window1.xaml.cs** ページを開きます。
2. **x:Name="C1MediaPlayer1"** を <c1:C1MediaPlayer> タグに追加します。これにより、このコントロールをコード内で呼び出すための一意の識別子が指定されます。
3. **InitializeComponent()** メソッドの下に次のコードを追加します。

Visual Basic

```
C1MediaPlayer1.IsItemVisible = True
```

C#

```
C1MediaPlayer1.IsItemVisible = true;
```

4. プログラムを実行し、実行時に項目リストが開くことを確認します。

テーマを使用する (Silverlight のみ)

C1MediaPlayer コントロールには、デフォルトで明るい青色のテーマが設定されていますが、全部で複数のテーマ(「[テーマ](#)」を参照)を適用できます。このトピックでは、**C1MediaPlayer** コントロールのテーマを **C1ThemeRainierOrange** に変更します。

Visual Studio での設計時

次の手順を実行します

1. Visual Studio で、.xaml ページを開きます。
2. <Grid></Grid>タグの間にカーソルを置きます。
3. ツールボックスで、**C1ThemeRainierOrange** アイコンをダブルクリックして、テーマを宣言します。このタグは次のようになります。

XAML

```
<my:C1ThemeRainierOrange></my:C1ThemeRainierOrange>
```

4. <my:C1ThemeRainierOrange> タグと </my:C1ThemeRainierOrange> タグの間にカーソルを置きます。
5. ツールボックスで、**C1MediaPlayer** アイコンをダブルクリックして、このコントロールをプロジェクトに追加します。そのタグは <my:C1ThemeRainierOrange> タグの子として表示され、マークアップは次のようになります。

XAML

```
<my:C1ThemeRainierOrange>  
    <c1:C1MediaPlayer></c1:C1MediaPlayer>  
</my:C1ThemeRainierOrange>
```

6. プログラムを実行します。

Blend での設計時

次の手順を実行します。

1. [アセット]パネルをクリックします。
2. 検索バーで「C1ThemeRainierOrange」と入力します。**C1ThemeRainierOrange** アイコンが表示されます。
3. **C1ThemeRainierOrange** アイコンをダブルクリックしてプロジェクトに追加します。
4. 検索バーで「C1MediaPlayer」と入力して **C1MediaPlayer** コントロールを検索します。
5. **C1MediaPlayer** アイコンをダブルクリックしてプロジェクトに追加します。
6. [オブジェクトとタイムライン]タブで[C1MediaPlayer]を選択して、[C1ThemeRainierOrange]をドラッグ & ドロップして配置します。
7. プロジェクトを実行します。

このトピックの作業結果

次の画像は **C1ThemeRainierOrange** テーマを適用した **C1MediaPlayer** コントロールを表しています。

