



# Som, Vídeo, e Deep Zoom

José Antônio da Cunha  
IFRN

# Som

A plataforma Silverlight permite a interação com áudio e vídeo em seus aplicativos, facilitando assim a construção de aplicações que fornecem experiências mais ricas aos utilizadores finais.

# Som

## Formatos suportados

Atualmente, a plataforma Silverlight suporta vários formatos. A variedade de escolha obriga-nos a tomar certas decisões antes de implementarmos um userControl que apresenta este tipo de conteúdo. Assim, precisamos responder às seguintes perguntas:

- **Qual o método de entrega:** o método de entrega indica a forma como o conteúdo multimédia é entregue ao usuário final. Podemos optar por *streaming* tradicional, *smooth streaming* ou *downloads* progressivos.
- **Qual é o tipo usado:** qual é o formato no qual o conteúdo multimédia é distribuído. O “contentor”, sabe como os bytes foram armazenados. Como exemplos, temos MP3, MP4, Windows Media Files, etc.
- **Qual o codec usado:** os codecs são usados para codificar os dados do conteúdo a distribuir. Por exemplo, o Silverlight suporta o codec de vídeo H.264 e o codec de áudio AAC (lembre-se que o codec H.264 é usado em várias áreas de referência associadas a utilização de vídeo de alta definição – ex.: Blu-ray).

Mais informações em <http://bit.ly/cealrH>

# Som

## **Para áudio, o Silverlight suporta o seguinte:**

- Windows Media Audio (WMA) versões 7, 8 e 9;
- MP3 com taxa de bits fixa ou variável de 8-320 Kbs;
- VC-1;
- Audio Windows media 10 Pro

## **Quando se trata de vídeos, o Silverlight suporta as seguintes padrões:**

- Windows Media Vídeo 7 (WMVI);
- Windows Media vídeo 8 (WMV2);
- Windows Media vídeo 8 (WMV3);
- Windows Media vídeo Advanced Profile, non-VC-1 (WMVA);
- Windows Media vídeo Advanced Profile, VC-1 (WMVC1);
- H.264 vídeo e AAC áudio (também conhecido como MPEG-4 Part 10 ou MPEG-4 AVC).

# Som

## O MediaElement

No Silverlight, toda a funcionalidade de áudio e vídeo é construído em uma única classe: `MediaElement`.

Uma tag `MediaElement` simples é tudo que você precisa para reproduzir um som. Por exemplo, adicione esta marcação para sua interface de usuário:

```
<MediaElement Source="Kalimba.mp3"></MediaElement>
```

Quando a página for carregada, ele vai baixar o arquivo `Kalimba.mp3` e começará a tocar automaticamente.

Claro, para que isso funcione, seu aplicativo Silverlight precisa ser capaz de encontrar o arquivo `Kalimba.mp3`.

A classe **MediaElement** usa o sistema de URL, tal qual a classe **Image**. Isso significa que você pode inserir um arquivo de mídia em seu pacote XAP ou implantá-lo no mesmo sítio, juntamente com o arquivo XAP. Geralmente, é melhor manter os arquivos de mídia separado, a menos que eles sejam extremamente pequenos. Caso contrário, você vai aumentar o tamanho de sua aplicação e do download inicial.

# Som

## Controlando Playback

O exemplo anterior começa executar um arquivo de som imediatamente quando a página com o `MediaElement` é carregado. A reprodução continua até que o arquivo de som seja finalizado.

Geralmente, você precisa controlar a reprodução com maior precisão. Por exemplo, você pode querer que ele seja acionado em um momento específico, controlar a repetição, e assim por diante. Uma maneira de alcançar esse resultado é usar os métodos da classe **MediaElement** no momento oportuno.

# Som

## Controlando Playback

O comportamento de inicialização do **MediaElement** é determinado por sua propriedade de reprodução automática (AutoPlay). Se essa propriedade é definida como false, o arquivo de áudio é carregado, mas o código fica responsável para iniciar a reprodução, na hora certa:

```
<MediaElement x:Name="videoElement" AutoPlay="false" Source="Kalimba.mp3">  
</MediaElement>
```

Geralmente, a interação consiste em chamar os métodos **Play ()**, **Pause ()** e **Stop ()**.

# Som

## Controlando Playback

```
<MediaElement x:Name="videoElement" AutoPlay="False" Source="Kalimba.mp3">  
</MediaElement>
```

```
private void btnPlay_Click(object sender, RoutedEventArgs e)  
{  
    MediaElement media = new MediaElement();  
    media.Source = new Uri("http://localhost/Kalimba.mp3", UriKind.Absolute);  
    media.MediaEnded += new RoutedEventHandler(media_MediaEnded);  
    LayoutRoot.Children.Add(media);  
}
```

```
private void media_MediaEnded(object sender, RoutedEventArgs e)  
{  
    LayoutRoot.Children.Remove((MediaElement)sender);  
}
```

# Som

## Controlando Playback

### Manipulando erro

```
<MediaElement x:Name="videoElement" AutoPlay="False"  
              Source="http://localhost:49925/kalimba.mp3"  
              MediaFailed="videoElement_MediaFailed">  
</MediaElement>
```

```
private void videoElement_MediaFailed(object sender, ExceptionRoutedEventArgs e)  
{  
    lblError.Content = e.ErrorException.Message;  
}
```

# Som

## Propriedades de MediaElement para Audio

| Propriedade      | Descrição  |
|------------------|--|
| AUDIOSTREAMCOUNT | Verificar o número de faixas.                                    |
| AUDIOSTREAMINDEX | Determinar a faixa a ser reproduzida                             |
| BALANCE          | Balancear o som entre o ouvido esquerdo e o direito (-1.0 a 1.0) |
| ISMUTED          | Determinar se o som é audível ou não                             |
| VOLUME           | Determina o grau audível de um MediaElement (0.0 a 1.0)          |

Veja os exemplos de audio → projeto media

# Video

## Reproduzindo video

Tudo o que você aprendeu sobre o uso da classe **MediaElement** se aplica igualmente para arquivo de vídeo e de áudio.

A principal diferença com arquivos de vídeo é que as propriedades visuais e relacionados com a disposição do **MediaElement** são subitamente importante. O tamanho original do vídeo é fornecido por meio das propriedades **NaturalVideoHeight** e **NaturalVideoWidth** do **MediaElement**.

# Video

## Listas do lado do cliente (**Client-Side Playlists**)

O Silverlight também suporta metarquivos Windows Media, que são essencialmente listas que apontam para um ou mais outros arquivos de mídia. Metarquivos de mídia do Windows normalmente têm a extensão .wax, .wvx, .wmx, .wpl, ou .asx.

Aqui está uma lista básica que se refere a dois arquivos de vídeo:

```
<asx version="3.0">
<title>My Playlist</title>
<entry>
<title>Movie #1</title>
<ref href="test.wmv"/>
</entry>
<entry>
<title>Movie #2</title>
<ref href="Butterfly.wmv"/>
</entry>
</asx>
```

Advanced Stream Redirector (ASX) é um arquivo XML.

# Video

**Tabela 3. Os elementos ASX suportados pelo Silverlight.**

| Elemento  | Descrição  |
|-----------|--|
| Asx       | Elemento raiz de uma lista de reprodução na estação-cliente.                           |
| Author    | Especifica o(s) nome(s) do(s) individuo(s) que criou(criaram) a lista.                 |
| Base      | Representa um URL que será prefixada antes de começar a reproduzir na estação-cliente. |
| Copyright | Declara a informação de direito autoral para um elemento ASX ou Entry.                 |
| Entry     | Define um item em uma lista de reprodução no cliente.                                  |
| MoreInfo  | Permite especificar uma URL que proporcione informações mais detalhadas sobre a lista. |
| Ref       | Esse elemento é o item que especifica a qual arquivo se referir para um clip de mídia. |
| Title     | Título da lista  |

# Video

## Usando lista de reprodução no servidor

Lista de reprodução no servidor permitem que os administradores de conteúdo determinem de forma dinâmica quando e qual conteúdo será reproduzido. O servidor que faz o streaming do conteúdo tem controle total sobre como ele é distribuído. Essa abordagem proporciona diversas vantagens sobre listas de reprodução no cliente, incluindo:

- **Menor gasto de banda** – geralmente as listas de reprodução no cliente servem o conteúdo como fluxos separados para cada entrada. Isso faz com que a aplicação Silverlight se reconecte ao servidor muitas vezes, desperdiçando largura da banda. Como as listas de reprodução no servidor usam um fluxo contínuo, a aplicação Silverlight terá que se conectar uma só vez.
- **Criação de listas de reprodução dinâmicas** – listas de reprodução no servidor permitem modificar uma lista de reprodução mesmo depois que uma aplicação Silverlight tenha se conectado.

# Video

## Usando lista de reprodução no servidor (.WSX)

```
<?wsx version="1.0"?>
<smil>
  <seq id="sq1">
    <media id="anuncio1" src="advertisement1.wmv" />
    <media id="anuncio2" src="advertisement2.wmv" />
  </seq>
</smil>
```

# Video

**Tabela 4. Elementos SMIL suportados no Silverlight.**

| Elemento | Descrição   |
|----------|---|
| Excl     | Recipiente para itens de mídia. Esses itens podem ser reproduzidos em qualquer ordem, mas só um será reproduzido por vez. |
| Media    | Referencia um arquivo de áudio ou vídeo por um atributo src.  |
| Seq      | Recipiente para arquivos de mídia. Esses itens serão reproduzidos em ordem seqüencial.                                    |
| Smil     | Elemento raiz para uma lista de reprodução na estação servidora.  |
| Switch   | Recipiente para uma série de itens que podem ser intercambiados se um deles falhar.                                       |

# Video

## Referenciando listas de reprodução no servidor

Depois que o arquivo .wsx foi criado, ele pode ser publicado no servidor. Deve-se publicar uma lista de reprodução no servidor antes que a aplicação Silverlight possa utilizá-la.

```
<MediaElement Source="mms://www.silverlightinaction.com:1234/lista" />
```

Alternando um plug-in para modo tela-cheia.

```
Application.Current.Host.Content.IsFullScreen = true;
```