

# Feature Details

Feature Id 166

Project Gallery Server Pro

Name Allow a membership or role provider to be specified by name rather than assuming the default provider

Build #	2.3.3505	Start Date		Priority	Medium
Estimated Work	4 hrs	Due Date		Status	Closed
Actual Work	4 hrs	Completion Date	7/11/2009	Percent Complete	100
Requested By	Roger Martin	Assigned To	Roger Martin		

## Description

Some users may specify multiple membership or role providers in the web.config file. GSP always assumes the default provider is the correct one, but this may not always be the case. Provide for the ability to tell GSP to use a provider by name.

For example, web.config may specify two role providers:

```
<roleManager defaultProvider="YafSqlRoleProvider" ...>
<providers>
<clear />
<add applicationName="YAF" name="YafSqlRoleProvider" ... />
<add applicationName="Gallery Server Pro" name="GspSqlRoleProvider" ... />
</providers>
</roleManager>
```

It would be useful to be able to tell GSP to use the provider named GspSqlRoleProvider.

## Notes

Added two configuration settings to galleryserverpro.config: membershipProviderName and roleProviderName. When blank, which is the default, Gallery Server assumes the default providers specified in web.config are the correct ones. A user can override this behavior by specifying a provider name. For example, the sample role provider specified earlier was this:

```
<roleManager defaultProvider="YafSqlRoleProvider" ...>
<providers>
<clear />
<add applicationName="YAF" name="YafSqlRoleProvider" ... />
<add applicationName="Gallery Server Pro" name="GspSqlRoleProvider" ... />
</providers>
</roleManager>
```

An administrator can specify roleProviderName="GspSqlRoleProvider" in galleryserverpro.config to use that role provider.

Note: The user may need to re-run the install wizard after making this change so that the new roles are correctly configured.

Note: I attempted to allow for configuring a non-default profile provider as well, but I could not determine a programmatic way to access a non-default profile provider.

# Feature Details

Feature Id 167

Project Gallery Server Pro

Name Add support for Adobe Flash Video (.flv) files

Build # 2.3.3505

Start Date 7/3/2009

Priority Medium

Estimated Work 4 hrs

Due Date

Status Closed

Actual Work 4 hrs

Completion Date 7/3/2009

Percent Complete 100

Requested By Roger Martin

Assigned To Roger Martin

## Description

Allow users to play videos in the Adobe Flash Video format (.flv).

## Notes

The Flash player Flowplayer at <http://flowplayer.org> was chosen for playing Flash video. It is GPL open source and can therefore be re-distributed with GSP. (I had initially played with JW FLV Media Player from <http://www.longtailvideo.com> and had it working, but its license was not compatible with GSP's GPL.)

Implementing this required three files to be added to the script directory:

- \* flowplayer-3.1.1.min.js
- \* flowplayer-3.1.1.swf
- \* flowplayer.controls-3.1.1.swf

I added the .flv file type to galleryserverpro.config:

```
<mimeFileTypeExtension=".flv"browserId="default"type="video/x-flv"allowAddToGallery="false"/>
```

I added an HTML and script template in galleryserverpro.config:

```
<mediaObjectmimeType="video/x-flv">
<browsers>
<browserid="default"
htmlOutput="&lt;script type="&quot;text/javascript&quot;src="&quot;{GalleryPath}/script/flowplayer-3.1.1.min.js &quot;&gt;&lt;/script
&gt;&lt;a href="&quot;{MediaObjectUrl} &quot;style="&quot;display:block;width: {Width}px;height: {Height}px &quot;id="&quot;player
&quot;&gt;&lt;/a&gt;"
scriptOutput="flowplayer("amp;quot;player&quot;, { src:"amp;quot;{GalleryPath}/script/flowplayer-3.1.1.swf &quot;, wmode:"amp;quot;opaque
&quot;}, { clip: { autoPlay: {AutoStartMediaObjectText}, scaling:"amp;quot;fit&quot; } });"/>
</browsers>
</mediaObject>
```

This template respects the autostart setting for videos. The script is downloaded only when a .flv video is played, so there is no additional page weight for existing users. The Flowplayer offers a lot of skinning options and commercial licenses.

# Feature Details

Feature Id 168

Project Gallery Server Pro

Name Ignore hidden directories during a synchronization

Build # 2.3.3505

Start Date 7/10/2009

Priority Medium

Estimated Work 1 hrs

Due Date

Status Closed

Actual Work 30 min

Completion Date 7/10/2009

Percent Complete 100

Requested By Roger Martin

Assigned To Roger Martin

## Description

Gallery Server Pro should ignore hidden directories during a synchronization.

## Notes

The function `GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories` was modified to ignore hidden directories. Original function:

```
private
```

```
void SynchronizeChildDirectories(DirectoryInfo parentDirectory, IAlbum parentAlbum)
```

```
{  
    /* Irrelevant code not shown... */  
  
    // Recursively traverse all subdirectories and their files and synchronize each object we find.  
    foreach(DirectoryInfo subdirectory in parentDirectory.GetDirectories())  
    {  
        IAlbum childAlbum = SynchronizeDirectory(subdirectory, parentAlbum);  
        SynchronizeMediaObjectFiles(subdirectory, childAlbum);  
        SynchronizeExternalMediaObjects(childAlbum);  
        SynchronizeChildDirectories(subdirectory, childAlbum);  
    }  
    /* Irrelevant code not shown... */  
}
```

Modified function:

```
private
```

```
void SynchronizeChildDirectories(DirectoryInfo parentDirectory, IAlbum parentAlbum)
```

```
{  
    /* Irrelevant code not shown... */  
  
    // Recursively traverse all subdirectories and their files and synchronize each object we find.  
    // Skip any hidden directories.  
    foreach(DirectoryInfo subdirectory in parentDirectory.GetDirectories())  
    {  
        if((subdirectory.Attributes & FileAttributes.Hidden) == FileAttributes.Hidden)  
        {  
            this._synchStatus.SkippedMediaObjects.Add(new KeyValuePair<string, string>(subdirectory.FullName.Remove(0, mediaObjectPhysicalPathLength + 1), Resources.SynchronizationStatus_Hidden_Directory_Msg));  
        }  
    }  
}
```

# Feature Details

```
    continue;
}

IAlbum childAlbum = SynchronizeDirectory(subdirectory, parentAlbum);
SynchronizeMediaObjectFiles(subdirectory, childAlbum);
SynchronizeExternalMediaObjects(childAlbum);
SynchronizeChildDirectories(subdirectory, childAlbum);
}
}
```

**Feature Id** 169

**Project** Gallery Server Pro

**Name** Add support for H.264 audio and video

<b>Build #</b> 2.3.3505	<b>Start Date</b>	<b>Priority</b> Medium
<b>Estimated Work</b> 2 hrs	<b>Due Date</b>	<b>Status</b> Closed
<b>Actual Work</b> 2 hrs	<b>Completion Date</b> 7/20/2009	<b>Percent Complete</b> 100
<b>Requested By</b> Roger Martin	<b>Assigned To</b> Roger Martin	

## Description

Allow users to play H.264 audio and video using the new capability in Silverlight 3.0

## Notes

I modified galleryserverpro.config so that .m4a and .mp4 files are played using Silverlight. The template specifies a minimum Silverlight version of 3.0.

I added the .m4a file type to galleryserverpro.config (.mp4 was already present):

```
<mimeTypefileExtension=".m4a"browserId="default"type="audio/m4a"allowAddToGallery="false"/>
```

The .m4a and .mp4 file extensions were added to the list of file extensions that require the Silverlight javascript files:

```
silverlightFileTypes=".mp3,.wma,.wmv,.asf,.asx,.mp4,.m4a "
```

I added an HTML and script template in galleryserverpro.config:

```
<mediaObjectmimeType="audio/m4a">
  <browsers>
    <browserid="default"htmlOutput="&lt;div id='mp1p'&gt;&lt;/div&gt;"
scriptOutput="Sys.UI.Silverlight.Control.createObject('mp1p', '&lt;object type='&quot;application/x-silverlight&quot;id='&quot;mp1
&quot;style='&quot;height: {Height}px;width: {Width}px; &quot;&gt;&lt;param name='&quot;minRuntimeVersion &quot;value='&quot;
3.0.40624.0&quot;/&gt;&lt;param name='&quot;Windowless&quot;value='&quot;True&quot;/&gt;&lt;a href='&quot;
http://go2.microsoft.com/fwlink/?LinkID=114576 &amp;v=1.0&quot;&gt;&lt;img src='&quot;
http://go2.microsoft.com/fwlink/?LinkID=108181 &quot;alt='&quot;Get Microsoft Silverlight &quot;style='&quot;border-width:0;&quot;/
&gt;&lt;/a&gt;&lt;/object&gt;'); Sys.Application.add_init(function() { Screate(Sys.UI.Silverlight.MediaPlayer, { &quot;mediaSource
&quot;:&quot;{MediaObjectUrl}&quot;,&quot;scaleMode&quot;: 1,&quot;source&quot;:&quot;
{GalleryPath}/skins/mediaplayer/AudioGray.xaml &quot;,&quot;autoPlay&quot;:{AutoStartMediaObjectText} }, null, null, $get(&quot;
mp1p&quot;)); }); Sys.Application.initialize();Array.add(_mediaObjectsToDispose,&quot;mp1&quot;);"/>
  </browsers>
</mediaObject>
```

<

```
mediaObjectmimeType="video/mp4">
```

## Feature Details

---

```
<browsers>
<browserid="default"htmlOutput="&lt;div id='mp1p'&gt;&lt;/div&gt;"
scriptOutput="Sys.UI.Silverlight.Control.createObject('mp1p', '&lt;object type='&quot;application/x-silverlight&quot;id='&quot;mp1
&quot;style='&quot;height: {Height}px;width: {Width}px; &quot;&gt;&lt;param name='&quot;minRuntimeVersion &quot;value='&quot;
3.0.40624.0&quot;/&gt;&lt;param name='&quot;Windowless&quot;value='&quot;True&quot;/&gt;&lt;a href='&quot;
http://go2.microsoft.com/fwlink/?LinkID=114576 &amp;v=1.0&quot;&gt;&lt;img src='&quot;
http://go2.microsoft.com/fwlink/?LinkID=108181 &quot;alt='&quot;Get Microsoft Silverlight &quot;style='&quot;border-width:0;&quot;/
&gt;&lt;/a&gt;&lt;/object&gt;'); Sys.Application.add_init(function() { Screate(Sys.UI.Silverlight.MediaPlayer, { &quot;mediaSource
&quot;:&quot;{MediaObjectUrl}&quot;,&quot;scaleMode&quot;: 1,&quot;source&quot;:&quot;
{GalleryPath}/skins/mediaplayer/Professional.xaml &quot;,&quot;autoplay&quot;: {AutoStartMediaObjectText} }, null, null, $get(&quot;
mp1p&quot;)); }); Sys.Application.initialize();Array.add(_mediaObjectsToDispose,&quot;mp1&quot;);"/>
</browsers>
</mediaObject>
```