

# Feature Details

**Feature Id 50**

**Project Gallery Server Pro**

**Name Enable data backup and restore functionality**

<b>Build #</b> 2.1	<b>Start Date</b> 9/11/2007	<b>Priority</b> High
<b>Estimated Work</b> 24 hrs	<b>Due Date</b>	<b>Status</b> Closed
<b>Actual Work</b> 40 hrs	<b>Completion Date</b> 7/14/2008	<b>Percent Complete</b> 100
<b>Requested By</b> Roger Martin	<b>Assigned To</b> Roger Martin	

## Description

Add functionality to allow for the backup of gallery data to an XML file. This file can then be imported into a different gallery installed elsewhere. This provides backup and transfer functionality.

## Notes

**Feature Id 77**

**Project Gallery Server Pro**

**Name Add support for Silverlight**

<b>Build #</b> 2.1	<b>Start Date</b> 4/13/2008	<b>Priority</b> High
<b>Estimated Work</b> 32	<b>Due Date</b>	<b>Status</b> Closed
<b>Actual Work</b> 32 hrs	<b>Completion Date</b> 7/26/2008	<b>Percent Complete</b> 100
<b>Requested By</b> Roger Martin	<b>Assigned To</b> Roger Martin	

## Description

Allow Silverlight to be used to render any media types it supports (.wma, .wmv, .mp3, and .asx). Include support for hosting the media at <http://silverlight.live.com/>.

## Notes

Using the MediaPlayer control in the Silverlight 2 Beta 2 Tools package, I was able to determine the combination of javascript and javascript files needed to render a media object in Silverlight without a dependence on System.Web.Silverlight.dll or .NET 3.x. Requires .NET 2.0 and MS Ajax 1.0. If server is running .NET 3.5 with config files pointing to the 3.5 versions of the Ajax dll's then MS Ajax 1.0 is not necessary. The two javascript files below were extracted from System.Web.Silverlight.dll using Reflector.

```
<asp:ScriptManagerID="ScriptManager1" runat="server">
<Scripts>
<asp:ScriptReferencePath="~/script/SilverlightControl.js"/>
<asp:ScriptReferencePath="~/script/SilverlightMedia.js"/>
</Scripts>
</asp:ScriptManager>

<div id="mp1p"></div>

<script type="text/javascript">
//</pre></div><div data-bbox="29 902 936 940" data-label="Text"><pre>Sys.UI.Silverlight.Control.createObject('mp1p', &lt;object type="application/x-silverlight" data="data:application/x-silverlight," id="mp1" style="height:450px; width:599px;"&gt;\r\n\t&lt;a href="http://go2.microsoft.com/fwlink/?</pre></div><div data-bbox="35 960 158 974" data-label="Page-Footer"><p>Printed on 9/3/2008</p></div><div data-bbox="904 960 958 975" data-label="Page-Footer"><p>Page 1</p></div>
```

## Feature Details

```
LinkID=114576&v=1.0"></a>\r\n</object>');
Sys.Application.initialize();
Sys.Application.add_init(function() {
$create(Sys.UI.Silverlight.MediaPlayer, {
"mediaSource":"video/Rayco.wmv","scaleMode":1,"source":"skins/mediaplayer/Professional.xaml"}, null,null, $get("mp1p"));
});
//]]>
</script>
```

Surprisingly, it does not appear that the Silverlight.js file is required to assist the end user with a fresh Silverlight installation, but I should keep this in mind in case I encounter difficulty. Incorporating this logic into Gallery Server required the concept of associating ECMA script with a media object. This was accomplished by adding a scriptOutput attribute to the <browser> element in galleryserverpro.config. Silverlight can handle five file types: .wma, .wmv, .mp3, .asf, and .asx, so for each of these I added the following <mediaobject> section:

```
<mediaObjectmimeType="audio/x-ms-wma |audio/x-ms-wma | video/x-ms-wmv |video/x-ms-asf">
<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;data='&quot;
data:application/x-silverlight,&quot;id='&quot;mp1p&quot;style='&quot;height: {Height}px;width: {Width}px; &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.initialize();Sys.Application.add_init(function() { $create
(Sys.UI.Silverlight.MediaPlayer, {&quot;mediaSource&quot;:&quot;{MediaObjectUrl}&quot;,&quot;scaleMode&quot;:1,&quot;source
&quot;:&quot;skins/mediaplayer/Professional.xaml&quot;,&quot;autoPlay&quot;:{AutoStartMediaObjectText} }, null, null, $get(&quot;
mp1p&quot;)); });Array.add(_mediaObjectsToDispose,&quot;mp1p&quot;);"/>
</browsers>
</mediaObject>
```

Notice the expression `Array.add(_mediaObjectsToDispose,&quot;mp1p&quot;);` is specified at the end. This adds the newly created Ajax control to a page-level array containing objects that require disposal. When the user navigates to a new media object (e.g. clicks Next/Previous) javascript in mediaobjectview.js disposes of all objects in the array. This array was created specifically for this feature and is not currently used for any other purpose, but in theory it can be used any time an Ajax component requires disposal when navigating between objects.

For audio files (wma and mp3), I replaced Professional.xaml with AudioGray.xaml, as it is a more appropriate skin for audio.

### Observations:

- \* The implementation works in IE6, IE 7, and Firefox 2. It is intermittent in Firefox 3 when the user is not logged on, and almost never works when the user is logged on. Hopefully this will be resolved in future Firefox and/or Silverlight updates. I should keep an eye out for updates to SilverlightControl.js and SilverlightMedia.js.

- \* I added the line `Sys.Application.notifyScriptLoaded();` to the end of mediaobjectview.js because at one point during testing the variable `_mediaObjectsToDispose` array defined in this file was attempted to be written to by the line in the scriptOutput template before the array was initialized. I am not sure this is sufficient to ensure mediaobjectview.js finishes loading before the scriptTemplate runs.

- \* Even though Silverlight claims to support ASX files, I could not get it working. ASX files are simply XML files that reference other media, like this:

```
<ASX version = "3.0">
<TITLE>Simple ASX Demo</TITLE>
```

## Feature Details

```
<ENTRY>
  <TITLE>Xbox Fable Video</TITLE>
  <AUTHOR>Microsoft Corporation</AUTHOR>
  <COPYRIGHT>(c)2007 Microsoft Corporation</COPYRIGHT>
  <REF HREF = "http://rdog/dev/gs/mediaobjects/Silverlight/04%20Brian%20Wilson.wma" />
</ENTRY>
</ASX>
```

I noticed that using the default video rendering for the video/\* MIME type caused IE 7 to play the above ASX file, but Firefox 3 did not work.

\* An alternative method for dynamically updating the media object as the user clicks Next / Previous is this: Instead of disposing the original object and recreating it, grab a reference to the current DOM element containing the Silverlight object and update the source property on the MediaPlayer within it. The main reason this was rejected is that there is no guarantee that the object would exist. For example, if the user is browsing a photo and clicks Next to a Silverlight video, this technique won't work.

```
<script type="text/javascript">
function showNext()
{
  var host = $get("mp1");
  var mediaPlayer = host.content.findName("VideoWindow");
  mediaPlayer.source = "video/countyfair_lores.wmv";
}
</script>
```

### Feature Id 78

### Project Gallery Server Pro

#### Name Add support for externally hosted media objects (such as YouTube videos)

<b>Build #</b> 2.1	<b>Start Date</b> 4/13/2008	<b>Priority</b> Low
<b>Estimated Work</b> 40 hrs	<b>Due Date</b>	<b>Status</b> Closed
<b>Actual Work</b> 16 hrs	<b>Completion Date</b> 7/14/2008	<b>Percent Complete</b> 100
<b>Requested By</b> Roger Martin	<b>Assigned To</b> Roger Martin	

#### Description

Allow externally hosted content to be shown in Gallery Server.

#### Notes

Two fields were added to the table gs\_MediaObject: ExternalHtmlSource and ExternalType. The Add objects web page was modified to have a tab called External Content. This allows the user to paste embed code from other sites, such as YouTube. The embed code and the user-entered media type are stored in the new database fields.

### Feature Id 80

### Project Gallery Server Pro

#### Name Consolidate language resource files

<b>Build #</b> 2.1	<b>Start Date</b> 4/23/2008	<b>Priority</b> High
<b>Estimated Work</b> 4 hrs	<b>Due Date</b>	<b>Status</b> Closed
<b>Actual Work</b> 2 hrs	<b>Completion Date</b> 8/4/2008	<b>Percent Complete</b> 100
<b>Requested By</b> Roger Martin	<b>Assigned To</b> Roger Martin	

#### Description

Consolidate the language resource files into the minimum number of required files, probably one for the web

## Feature Details

project and at most one for each of the dependent libraries. This will make language translation easier.

### Notes

Eliminated the local resource files for changepassword.aspx, site.master, and login.aspx. Placed the resource strings into the global resource file instead. This results in - at most - one resource file per project.

**Feature Id 81**

**Project Gallery Server Pro**

**Name Add SQLite provider and make it the default data store**

**Build #** 2.1

**Start Date** 4/23/2008

**Priority** High

**Estimated Work** 80

**Due Date**

**Status** Closed

**Actual Work** 80 hrs

**Completion Date** 7/14/2008

**Percent** 100

**Requested By** Roger Martin

**Assigned To** Roger Martin

**Complete**

### Description

Add SQLite provider and make it the default data store

### Notes

SQLite requires the web application to run under full trust or for System.Data.SQLite.dll to be placed in the GAC.

**Feature Id 82**

**Project Gallery Server Pro**

**Name Modify install wizard to handle SQL Server and SQLite**

**Build #** 2.1

**Start Date** 4/23/2008

**Priority** High

**Estimated Work** 2 hrs

**Due Date**

**Status** Closed

**Actual Work** 24 hrs

**Completion Date** 8/28/2008

**Percent** 100

**Requested By** Roger Martin

**Assigned To** Roger Martin

**Complete**

### Description

Modify install wizard to handle SQL Server and SQLite.

### Notes

**Feature Id 83**

**Project Gallery Server Pro**

**Name Replace NeatUpload control with ComponentArt upload control**

**Build #** 2.1

**Start Date** 4/23/2008

**Priority** High

**Estimated Work** 4 hrs

**Due Date**

**Status** Closed

**Actual Work** 8 hrs

**Completion Date** 7/14/2008

**Percent** 100

**Requested By** Roger Martin

**Assigned To** Roger Martin

**Complete**

### Description

Replace NeatUpload control with ComponentArt upload control. Ensure that it works under IIS 7.0 Integrated Pipeline mode.

### Notes

## Feature Details

**Feature Id 84**

**Project Gallery Server Pro**

**Name Make it easier for users to contribute to Gallery Server Pro**

<b>Build #</b> 2.1	<b>Start Date</b> 4/23/2008	<b>Priority</b> High
<b>Estimated Work</b> 40 hrs	<b>Due Date</b>	<b>Status</b> Closed
<b>Actual Work</b> 40 hrs	<b>Completion Date</b> 8/28/2008	<b>Percent Complete</b> 100
<b>Requested By</b> Roger Martin	<b>Assigned To</b> Roger Martin	

### Description

Make it easier for users to contribute to Gallery Server Pro. Thirty days after installation Gallery Server should go into reduced functionality mode and inform the user that a free product key is required to continue. The user goes to the website and is asked to contribute to the project. Regardless of the amount of donation, the user is then given the product key.

### Notes

This was implemented by the following changes:

1. The column DateAdded was added to the table gs\_Gallery. It is populated by the installer wizard with the current date.
2. Modified the Site admin pages so that, if more than 30 days have passed after installation and no product key has been entered, they become read only.
3. Modified the ASHX handler getmediaobject.ashx to render a watermark on images if the trial period has expired.
4. Modified the default page of Site admin to accept a product key.
5. Added a Product Key Wizard to www.galleryserverpro.com to generate a product key for users. One step in the wizard asks the user to consider a donation.

**Feature Id 85**

**Project Gallery Server Pro**

**Name Allow Gallery Server Pro to run under .NET 3.5 without MS Ajax 1.0 installed**

<b>Build #</b> 2.1	<b>Start Date</b> 4/23/2008	<b>Priority</b> High
<b>Estimated Work</b> 8 hrs	<b>Due Date</b>	<b>Status</b> Closed
<b>Actual Work</b> 1 hrs	<b>Completion Date</b> 8/28/2008	<b>Percent Complete</b> 100
<b>Requested By</b> Roger Martin	<b>Assigned To</b> Roger Martin	

### Description

Allow source code to run under .NET 3.5 without MS Ajax 1.0 installed.

### Notes

This was accomplished by loading the source code into Visual Studio 2008 and modifying the gsweb project properties to target .NET 3.5. This caused web.config to be automatically updated to point to the 3.5 dll's. The same thing was done for the other config file - web\_sqlserver.config.

These modified config files were then packaged into a ZIP file, along with a Readme.txt file with instructions, and posted on www.galleryserverpro.com.

# Feature Details

**Feature Id 111**

**Project Gallery Server Pro**

**Name Increase robustness of synchronization**

<b>Build #</b> 2.1	<b>Start Date</b> 5/15/2008	<b>Priority</b> High
<b>Estimated Work</b> 24 hrs	<b>Due Date</b>	<b>Status</b> Closed
<b>Actual Work</b> 24 hrs	<b>Completion Date</b> 7/16/2008	<b>Percent Complete</b> 100
<b>Requested By</b> Roger Martin	<b>Assigned To</b> Roger Martin	

## Description

Corrupt images and other issues during synchronizing can cause the process to fail.

## Notes

Changes were made to handle more cases of invalid objects. Also, the synch UI displays the last synchronized filename. If an error occurs, the exception type and stack trace is displayed in addition to the message. If any files are skipped, the list of skipped objects are at the end of the synchronization.

**Feature Id 115**

**Project Gallery Server Pro**

**Name Improve performance for very large galleries**

<b>Build #</b> 2.1	<b>Start Date</b> 7/1/2008	<b>Priority</b> High
<b>Estimated Work</b> 8 hrs	<b>Due Date</b> 7/1/2008	<b>Status</b> Closed
<b>Actual Work</b> 1 hrs	<b>Completion Date</b> 7/1/2008	<b>Percent Complete</b> 100
<b>Requested By</b> Roger Martin	<b>Assigned To</b> Roger Martin	

## Description

The first time an album is viewed, Gallery Server iterates through all of its media objects and all of its child album's media objects. This can take a long time when it or its child albums contain a large number of objects (more than a few thousand).

Note that this performance hit occurs only the first time it is loaded or after the cache is purged, as the album is inserted into the cache when the initial processing is complete.

## Notes

The method Album.VerifyThumbnailsInflated() in the business layer was modified so that the media objects of the requested album's child albums are iterated through \*only\* if they have already been instantiated. If not, we load the media object for the thumbnail via its ID.

**Feature Id 116**

**Project Gallery Server Pro**

**Name Extract album title divider character from resource file**

<b>Build #</b> 2.1	<b>Start Date</b> 7/1/2008	<b>Priority</b> Medium
<b>Estimated Work</b> 15 min	<b>Due Date</b>	<b>Status</b> Closed
<b>Actual Work</b> 15 min	<b>Completion Date</b> 7/1/2008	<b>Percent Complete</b> 100
<b>Requested By</b> Roger Martin	<b>Assigned To</b> Roger Martin	

## Description

Currently the divider character (>>) is hard coded in gsweb/uc/albummenu.ascx.cs, in the function buildMenuString(), so it can't be changed unless you modify the source and recompile. It really should be stored in a resource file so that it can be changed by end users.

## Feature Details

---

### Notes

Extracted as resource string UC\_Album\_Menu\_Album\_Divider\_Text.