

## Defect Details

**Defect Id:** 80

**Project:** Gallery Server Pro

**Name:** Synchronizing video and audio files does not update the original width and height values

**Date Found:** 12/9/2007

**Build #:** 2.0.2898

**Severity:** Medium Impact

**Date Fixed:** 8/4/2008

**Build # of Fix:** 2.1.3138

**Status:** Closed

**Estimated Work:** 2 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 2 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

The width and height specified in the Site Admin section for video/audio/generic objects is assigned as the width and height for each object when it is added. The record for each object in the database (table gs\_MediaObject) has two fields - OriginalWidth and OriginalHeight. Once these are assigned, there is nothing you can do to change them. Changing the values in Site Admin only affects newly added objects, not existing ones.

This is a bug. One of the following should happen (I'm not sure which is the best approach at the moment):

1. The width and height setting from Site Admin is used when rendering the video window, and the width and height associated with the actual video is ignored.
2. When you synchronize, the OriginalWidth and OriginalHeight columns in the db should be updated with the most recent Site Admin width and height settings. This is probably the preferred approach, as it will allow differently sized objects on a per album basis, since albums can be individually synchronized.

### Replication Procedures

### Resolution

Refactored code in GalleryServerPro.Business.SynchronizationManager.UpdateExistingMediaObject so that the Width and Height properties of a media object's Original object is set to the width and height values specified in galleryserverpro.config. Applies only to audio, video, and generic media objects.

The resulting behavior is that the width and height will always be updated for audio, video and generic objects during a synch. In rare cases this may not be desirable, but the only probably alternative is to add an option on the synch page called "Update width and height of audio, video and generic objects" to control whether or not the update occurs. This is likely to be confusing.

### Notes

**Defect Id:** 82

**Project:** Gallery Server Pro

**Name:** Apostrophe in filename causes thumbnail to not render on rearrange page

**Date Found:** 12/31/2007

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 12/31/2007

**Build # of Fix:** 2.0.2921

**Status:** Closed

**Estimated Work:** 1 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 30 min

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

If the filename for a media object contained an apostrophe, the thumbnail for it would not render on the Rearrange objects page.

## Defect Details

### Replication Procedures

### Resolution

This occurred because the img tag that contained the thumbnail had the src attribute specified using apostrophe, unlike quotation marks that are used elsewhere. The code was change to specify the src in quotation marks instead of apostrophes, thus allowing the src value to contain one or more apostrophes.

### Notes

**Defect Id:** 83

**Project:** Gallery Server Pro

**Name:** Poor performance of manage users page when there are thousands of users

**Date Found:** 1/6/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 1/6/2008

**Build # of Fix:** 2.0.2927

**Status:** Closed

**Estimated Work:** 1 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 1 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

The manager users page (manageusers.aspx) performs slowly when there are thousands of users.

### Replication Procedures

### Resolution

Resolved by disabling view state on the grid gdUsers (EnableViewState="false"). View state wasn't being used anyway so no other change was required. I also disabled view state on the gdRoles grid on the manage roles page, since it is a similar situation.

### Notes

**Defect Id:** 84

**Project:** Gallery Server Pro

**Name:** IE not caching images on the client

**Date Found:** 1/6/2008

**Build #:** 2.0.2898

**Severity:** Medium Impact

**Date Fixed:** 1/6/2008

**Build # of Fix:** 2.0.2927

**Status:** Closed

**Estimated Work:** 1 hrs

**Assigned To:** Roger Martin

**Priority:** High

**Actual Work:** 1 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

Internet Explorer 7 does not cache images sent by the ASHX handler, even though the cacheability is set to `HttpCacheability.ServerAndPrivate`. Firefox properly caches images, and IE 6 was not tested.

### Replication Procedures

### Resolution

This is a bug in ASP.NET as described here:<http://support.microsoft.com/kb/836868>. There is a hotfix available, but I also found a workaround that I implemented. Instead of using the `ServerAndPrivate` enum value, I changed it to use `Public`. That causes it to cache correctly on the client, but - according to the documentation anyway - it will no longer be cached on the server. I think this is a worthwhile tradeoff.

Specifically, in `~/handler/getmediaobjects.ashx`, I changed the following line:

## Defect Details

```
cachePolicy.SetCacheability(HttpCacheability.ServerAndPrivate);
```

to this:

```
cachePolicy.SetCacheability(HttpCacheability.Public);
```

### Notes

---

**Defect Id:** 86

**Project:** Gallery Server Pro

**Name:** Install wizard does not show any databases in its dropdown list in certain hosting environments

**Date Found:** 1/27/2008

**Build #:** 2.0.2898

**Severity:** Medium Impact

**Date Fixed:** 1/27/2008

**Build # of Fix:** 2.0.2948

**Status:** Closed

**Estimated Work:** 10 min

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 10 min

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

Certain hosting companies, such as GoDaddy, have the ASP.NET engine run in a limited permission environment, causing the list of databases to be empty in the web install wizard.

### Replication Procedures

---

### Resolution

Modified the function BindDatabaseDropdownlist() in installer/default.aspx.cs to comment out the following line of code:

```
cn.ChangeDatabase("master");
```

It turns out this line isn't necessary for the subsequent SQL that extracts the list of databases ("select name from master..sysdatabases order by name asc"). In fact, removing this line allows this SQL to return the list of databases in environments such as GoDaddy.

### Notes

---

**Defect Id:** 87

**Project:** Gallery Server Pro

**Name:** Image metadata is sometimes extracted as the text "System.Int64[]"

**Date Found:** 2/9/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 8/4/2008

**Build # of Fix:** 2.1.3138

**Status:** Closed

**Estimated Work:** 4 hrs

**Assigned To:** Roger Martin

**Priority:** Low

**Actual Work:** 30 min

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

The metadata strings from images are sometimes extracted as the text "System.Int64[]". Attached is an example of an image where this occurs with the Comment metadata item.

### Replication Procedures

---

### Resolution

## Defect Details

Modified class GalleryServerPro.Business.Metadata.MetadataItem so that when the class encounters a property item that it cannot parse because it is an array, the Value property is assigned a null value instead of the unprocessed array. This allowed the calling method - GalleryServerPro.Business.Metadata.MediaObjectMetadataExtractor.FillRawMetadataDictionary - to check for a null value and then choose to not add it to the RawMetadataDictionary dictionary. This results in the metadata item not being added.

In my experiments with a particular photo that exhibited this behavior, I observed that all the bytes in the array were zero, which means there was no meaningful data. It is possible that other photos have valid data that theoretically can be parsed, but I do not know how to extract the data in this situation.

### Notes

**Defect Id:** 88

**Project:** Gallery Server Pro

**Name:** Hard coded English text in web pages

**Date Found:** 2/9/2008

**Build #:** 2.0.2898

**Severity:** Medium Impact

**Date Fixed:** 8/28/2008

**Build # of Fix:** 2.1.3162

**Status:** Closed

**Estimated Work:** 1 hrs

**Assigned To:** Roger Martin

**Priority:** High

**Actual Work:** 1 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

The following English text is hard coded into the web pages and were never extracted into resource files:

1. "My account" link text in site.master
2. "Search" text in search box (site.master watermark extender)
3. "Retrieve Password" in passwordrecovery.aspx
4. Description and Value heading in Image Metadata grid
5. "Save" and "Cancel" on the Edit album user control (albumedittemplate.ascx)
6. Error pages in error directory (need to convert error.htm to error.aspx so that language is localizable)

### Replication Procedures

### Resolution

### Notes

**Defect Id:** 89

**Project:** Gallery Server Pro

**Name:** Code references absolute URL when relative URL is sufficient

**Date Found:** 2/10/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 2/10/2008

**Build # of Fix:** 2.0.2962

**Status:** Closed

**Estimated Work:** 1 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 1 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

There are a few places where the code uses the absolute URL using Request.Url.AbsoluteUri. This returns an URL containing the server address/domain name. These instances can instead be handled by using a relative path.

## Defect Details

Absolute URLs may be causing issues with users using domain masking. Change these to use relative URLs.

### Replication Procedures

### Resolution

The following files were modified: site.master.cs and GspPage.cs.

### Notes

**Defect Id:** 90

**Project:** Gallery Server Pro

**Name:** An album title with a trailing period or space causes an exception

**Date Found:** 2/10/2008

**Build #:** 2.0.2898

**Severity:** Medium Impact

**Date Fixed:** 8/4/2008

**Build # of Fix:** 2.1.3138

**Status:** Closed

**Estimated Work:** 2 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 30 min

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

An InvalidMediaObjectExceptionexception is thrown when a user adds a media object to an album whose title contains a period (.) or space as the final character. The exception is thrown in the setter of the FileInfo property in DisplayObject.cs (line 217). Note this error also occurs if, when a long title is truncated for the directory name, if the final character is a period or space.

Although a period and space are valid characters in a directory name, Windows discards them when they are at the end of the directory name. This is a 8.3 DOS filename compatibility issue where most/all trailing periods and spaces are stripped from file and folder names by Windows, a holdover from the transition from 8.3 filenames where the dot is not stored but implied.

The code should be modified so that when an album is named (or renamed) with a trailing period, that it removes this period from the DirectoryName property so that the value stored in the database is the same as that of the directory name of the hard drive. An alternative approach is to not try to clean up the directory name before saving. Instead, just create the directory and then query the file system for the name that was created. I'm not quite sure if this approach is possible but it would be more robust.

### Replication Procedures

### Resolution

Modified function GalleryServerPro.Business.ValidateDirectoryName so that the following line is added at the end of the Parameter Validation region:

```
newDirName = newDirName.TrimEnd(newchar[] {'.'});
```

This strips any trailing periods ('.') from the directory name.

### Notes

## Defect Details

**Defect Id:** 92

**Project:** Gallery Server Pro

**Name:** Page times out when copying large numbers of objects

**Date Found:** 2/12/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 2/12/2008

**Build # of Fix:** 2.0.2962

**Status:** Closed

**Estimated Work:** 10 min

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 10 min

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

Copying and moving objects within a gallery occur on transferobject.aspx, and the page inherits the default timeout settings from ASP.NET. However, when copying a large number of objects, one can exceed this timeout value. Therefore, the timeout should be increased for this web page.

### Replication Procedures

### Resolution

The following section was added to web.config to increase the timeout to 60 minutes:

```
<location path="task/transferobject.aspx">  
<system.web>  
<httpRuntime executionTimeout="3600"/>  
</system.web>  
</location>
```

### Notes

**Defect Id:** 95

**Project:** Gallery Server Pro

**Name:** Out of memory when adding very large images

**Date Found:** 3/19/2008

**Build #:** 2.0.2898

**Severity:** Medium Impact

**Date Fixed:** 7/16/2008

**Build # of Fix:** 2.0.3119

**Status:** Closed

**Estimated Work:** 8 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 32 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

When adding very large images, Gallery Server throws an OutOfMemoryException exception when trying to create the thumbnail and optimized version. It may not be possible to prevent the exception, but it should degrade gracefully and use a generic thumbnail and no optimized image in this situation.

### Replication Procedures

### Resolution

Added error handling so that if an OutOfMemoryException occurs while processing an image, Gallery Server reverts to treating the file as a document, meaning it gets a generic thumbnail and cannot be previewed in the web browser. However, it can be downloaded. It was noticed that very large image files can cause more issues than just OutOfMemoryExceptions, such as System.Runtime.InteropServices.ExternalException and System.ArgumentException. The test image attached to this bug was used to resolve these issues so that the user can add or synchronize without an error occurring.

### Notes

## Defect Details

**Defect Id:** 96

**Project:** Gallery Server Pro

**Name:** Objects added to a private album may incorrectly appear in search results

**Date Found:** 5/6/2008

**Build #:** 2.0.2898

**Severity:** High Impact

**Date Fixed:** 5/6/2008

**Build # of Fix:** 2.0.3048

**Status:** Closed

**Estimated Work:** 4 hrs

**Assigned To:** Roger Martin

**Priority:** High

**Actual Work:** 2 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

When a media object is added to a private album, the IsPrivate property of the object is incorrectly set to false. This causes it to be searchable when anonymous users use the search functionality. Note that even though objects appear in the search result, they produce an "Insufficient permission" message if the user clicks on them.

The search function works correctly for logged on users.

### Replication Procedures

### Resolution

The HelperFunctions.SearchGallery() function in the business layer project uses the media object's IsPrivate property to determine whether it should appear in the search results for anonymous users. But media objects added to an album always have the IsPrivate property set to false, even when the parent album is private.

This was resolved by modifying one of the constructors in each of the media object classes: GenericMediaObject, Audio, Video, and Image. The original constructor looked like this:

```
internal Image(System.IO.FileInfo imageFile, IAlbum parentAlbum)
: this(int.MinValue, parentAlbum != null ? parentAlbum.Id : int.MinValue, parentAlbum, string.Empty,
string.Empty, string.Empty,
int.MinValue, int.MinValue, int.MinValue, string.Empty, int.MinValue, int.MinValue,
int.MinValue, string.Empty, int.MinValue, int.MinValue, int.MinValue, int.MinValue,
String.Empty, DateTime.MinValue, String.Empty, DateTime.MinValue, false, false, imageFile)
{
}
```

The modified constructor looks like this:

```
internal Image(System.IO.FileInfo imageFile, IAlbum parentAlbum)
: this(int.MinValue, parentAlbum != null ? parentAlbum.Id : int.MinValue, parentAlbum, string.Empty,
string.Empty, string.Empty,
int.MinValue, int.MinValue, int.MinValue, string.Empty, int.MinValue, int.MinValue,
int.MinValue, string.Empty, int.MinValue, int.MinValue, int.MinValue, int.MinValue,
String.Empty, DateTime.MinValue, String.Empty, DateTime.MinValue, parentAlbum != null ?
parentAlbum.IsPrivate : false, false, imageFile)
{
}
```

## Defect Details

Notice this constructor calls the other constructor. In that other constructor, we are changing the value passed to the third from last parameter, which is the isPrivate parameter. In short, when creating media objects from an file, which only happens for new media objects, we set the IsPrivate property to that of the album containing it.

### Notes

**Defect Id:** 98

**Project:** Gallery Server Pro

**Name:** File lock not released for some files

**Date Found:** 5/19/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 5/19/2008

**Build # of Fix:** 2.0.3061

**Status:** Closed

**Estimated Work:** 8 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 4 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

In certain, somewhat rare cases, media files added to Gallery Server will be locked by the IIS worker process. This will prevent the user from moving or deleting the object. The bug is triggered in the following circumstances:

- \* The web application is running under full trust.
- \* .NET 3.0 or higher is installed on the web server.
- \* The config setting enableWpfMetadataExtraction is true (which is the default).
- \* The media file being added has a mime type of image but the .NET Framework class BitmapDecoder throws an exception when the file is passed to its Create method. The mime type is determined from the mimeType section of galleryserverpro.config. For example, the wmf file type has a mime type of image/wmf.

This bug was initially reported through the post

at [http://www.galleryserverpro.com/forum/yaf\\_postsm1169\\_Deleting-Files-with-New-File-Extensions.aspx](http://www.galleryserverpro.com/forum/yaf_postsm1169_Deleting-Files-with-New-File-Extensions.aspx).

### Replication Procedures

### Resolution

I posted a message on Microsoft's General Discussions forum:

<http://www.microsoft.com/communities/newsgroups/en-us/default.aspx?dg=microsoft.public.dotnet.framework&tid=fe3fb82f-0191-40a3-b789-0602cc4445d3&cat=&lang=&cr=&sloc=&p=1>

That resulted in the following bug submission with MS:

<https://connect.microsoft.com/VisualStudio/feedback/ViewFeedback.aspx?FeedbackID=344914>

I modified the code to use a different overload of the BitmapDecoder.Create method, so instead of taking a URI, it now takes a FileStream that I ensure is properly closed even when an exception occurs:

```
BitmapDecoder fileBitmapDecoder;
using (Stream stream = new FileStream(imageFilePath, FileMode.Open, FileAccess.Read, FileShare.Read))
{
    try
    {
        fileBitmapDecoder = BitmapDecoder.Create(stream, BitmapCreateOptions.None,
        BitmapCacheOption.OnLoad);
    }
}
```

## Defect Details

```
// DO NOT USE: fileBitmapDecoder = BitmapDecoder.Create(new Uri(imageFilePath, UriKind.Absolute),  
BitmapCreateOptions.None, BitmapCacheOption.OnLoad);  
}  
catch (NotSupportedException)  
{  
    // Some image types, such as wmf, throw an exception. Let's skip trying to get the metadata.  
    return null;  
}  
}
```

### Notes

**Defect Id:** 99

**Project:** Gallery Server Pro

**Name:** Incomplete error information if an exception occurs when adding a user

**Date Found:** 5/19/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 5/20/2008

**Build # of Fix:** 2.0.3061

**Status:** Closed

**Estimated Work:** 2 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 2 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

If a MembershipCreateUserException exception occurs when adding a user (that is, after clicking Create User), a javascript alert message says "The data could not be loaded.". This is not very informative.

### Replication Procedures

### Resolution

This bug was caused by a problem with the error handling mechanism. Specifically, the catch block in the cbAddUser\_Callback method of manageusers.aspx.cs that handles the MembershipCreateUserException exception references the wrong text box, causing an unhandled exception. Instead of:

```
if ((ex.StatusCode != MembershipCreateStatus.DuplicateUserName) && (Membership.GetUser  
(txtNewUserEmail.Text) != null))
```

It was changed to this:

```
if ((ex.StatusCode != MembershipCreateStatus.DuplicateUserName) && (Membership.GetUser  
(txtNewUserUserName.Text) != null))
```

With this change, the original exception is correctly handled and its corresponding message is passed to the user.

### Notes

## Defect Details

**Defect Id:** 100

**Project:** Gallery Server Pro

**Name:** Adding a user sometimes never returns from "Communicating with server..." message

**Date Found:** 5/20/2008

**Build #:** 2.0.2898

**Severity:** Medium Impact

**Date Fixed:** 5/20/2008

**Build # of Fix:** 2.0.3062

**Status:** Closed

**Estimated Work:** 2 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 1 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

If an exception \*other\* than MembershipCreateUserException occurs when adding a user (that is, after clicking Create User), the message "Communicating with server" appears and never goes away. The user is forced to refresh the page and to wonder what happened.

### Replication Procedures

### Resolution

This bug is similar to #99, where the catch block referenced incorrect controls, causing an unhandled exception. Specifically, the catch block in the cbAddUser\_Callback method of manageusers.aspx.cs that handles the Exception exception had two errors. The original handler is this:

```
catch (Exception ex)
{
    // Just in case we created the user and the exception occurred at a later step, like adding the roles, delete the user.
    if (Membership.GetUser(txtNewUserEmail.Text) != null)
    {
        Membership.DeleteUser(txtNewUserUserName.Text, true);
    }
}
```

```
GalleryServerPro.Web.uc.usermessage msgBox = (GalleryServerPro.Web.uc.usermessage)LoadControl
("~/uc/usermessage.ascx");
msgBox.IconStyle = GalleryServerPro.Web.MessageStyle.Error;
msgBox.MessageTitle = Resources.GalleryServerPro.Admin_Manage_Users_Cannot_Create_User_Msg;
msgBox.MessageDetail = ex.Message;
pnlAddUserMessage.Controls.Add(msgBox);

FindControlRecursive(dgAddUser, "pnlAddUserDialogContent").RenderControl(e.Output);
}
```

The corrected handler is this:

```
catch (Exception ex)
{
    // Just in case we created the user and the exception occurred at a later step, like adding the roles, delete the user.
    if (Membership.GetUser(txtNewUserUserName.Text) != null)
    {
        Membership.DeleteUser(txtNewUserUserName.Text, true);
    }
}
```

## Defect Details

```
GalleryServerPro.Web.uc.usermessage msgBox = (GalleryServerPro.Web.uc.usermessage)LoadControl
("~/uc/usermessage.ascx");
msgBox.IconStyle = GalleryServerPro.Web.MessageStyle.Error;
msgBox.MessageTitle = Resources.GalleryServerPro.Admin_Manage_Users_Cannot_Create_User_Msg;
msgBox.MessageDetail = ex.Message;
pnlAddUserMessage.Controls.Add(msgBox);

FindControlRecursive(dgAddUser, "pnlAddUserMessage").RenderControl(e.Output);
}
```

After this fix, the catch block correctly handles the original exception and passes the exception message to the user.

### Notes

**Defect Id:** 101

**Project:** Gallery Server Pro

**Name:** Cannot manage users with ActiveDirectory provider

**Date Found:** 5/20/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 5/20/2008

**Build # of Fix:** 2.0.3062

**Status:** Closed

**Estimated Work:** 4 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 4 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

Due to differences between the SqlMembershipProvider and the ActiveDirectoryMembershipProvider, the Manage Users throws an exception or provides vague error messages in an Active Directory scenario.

### Replication Procedures

### Resolution

The vague error messages are addressed in bugs # 99 and 100.

Since the AD provider does not support the LastActivityDate and LastLoginDate properties, I wrapped calls to these properties in a try..catch. This change is inside the BindUserInfoControls method in manageusers.aspx.cs.

```
try
{
    lblCreationDate.Text = String.Format(CultureInfo.CurrentCulture, "{0} {1}",
user.CreationDate.ToLongDateString(), user.CreationDate.ToLongTimeString());
    lblLastActivityDate.Text = String.Format(CultureInfo.CurrentCulture, "{0} {1}",
user.LastActivityDate.ToLongDateString(), user.LastActivityDate.ToLongTimeString());
    lblLastLogOnDate.Text = String.Format(CultureInfo.CurrentCulture, "{0} {1}",
user.LastLoginDate.ToLongDateString(), user.LastLoginDate.ToLongTimeString());
    lblLastPasswordChangedDate.Text = String.Format(CultureInfo.CurrentCulture, "{0} {1}",
user.LastPasswordChangedDate.ToLongDateString(), user.LastPasswordChangedDate.ToLongTimeString());
}
catch (NotSupportedException) { /* Ignore if provider does not support one or more properties */}
```

The other change was inside the SaveUser method. The original code simply assigned the textbox value to the appropriate property:

## Defect Details

```
user.Email = txtEmail.Text;  
user.Comment = txtComment.Text;
```

This was changed so that the property was assigned only when the textbox was NOT null or empty.

```
user.Email = String.IsNullOrEmpty(txtEmail.Text) ? user.Email : txtEmail.Text;  
user.Comment = String.IsNullOrEmpty(txtComment.Text) ? user.Comment : txtComment.Text;
```

### Notes

**Defect Id:** 102

**Project:** Gallery Server Pro

**Name:** Album stats count hidden albums

**Date Found:** 5/22/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 8/4/2008

**Build # of Fix:** 2.0.3138

**Status:** Closed

**Estimated Work:** 2 hrs

**Assigned To:** Roger Martin

**Priority:** Low

**Actual Work:** 30 min

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

The album statistics that show in the top right corner (ex.(8 objects, created 9/30/2007)) count hidden albums, even when the user is not logged in and can't see them. It would be better if the count accurately reflected the number of objects the user was looking at.

### Replication Procedures

### Resolution

Modified function getChildGalleryStats() in albumheader.ascx.cs so that the following line:

```
int numObjects = album.GetChildGalleryObjects().Count;
```

is changed to this:

```
int numObjects = album.GetChildGalleryObjects(false, this.PageBase.IsAnonymousUser).Count;
```

This overload of getChildGalleryObjects excludes hidden objects for anonymous users, but includes them for logged on users. This results in the numObjects variable being assigned an accurate number.

### Notes

**Defect Id:** 103

**Project:** Gallery Server Pro

**Name:** URL occasionally incorrectly calculated when installed as root application

**Date Found:** 5/24/2008

**Build #:** 2.0.2898

**Severity:** Medium Impact

**Date Fixed:** 5/24/2008

**Build # of Fix:** 2.0.3065

**Status:** Closed

**Estimated Work:** 4 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 4 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

## Defect Details

When Gallery Server is installed in the root of the web site, the dynamically generated URLs may be occasionally incorrectly calculated. For example, the replacement parameter `MediaObjectAbsoluteUriNoHandler` may generate an URL like this: `"/mediaobjects/myalbum/myvideo.avi"`. The double slash at the beginning is interpreted by browsers as `http://mediaobjects/myalbum/myvideo.avi`, which is an invalid URL. The URL should have been `"/mediaobjects/myalbum/myvideo.avi"`.

### Replication Procedures

### Resolution

The `.NET Request.ApplicationPath` property is used in a handful of places throughout Gallery Server. It generates a value like `"/galleryserverpro"` when Gallery Server is installed in a subdirectory of the web site and `"/"` when installed in the root of the web site. This value is often concatenated with additional text to create an URL to a web resource, like this:

```
String.Concat(Request.ApplicationPath, "/", relativePath);
```

When Gallery Server is installed in the root, you end up with two leading slashes.

This was solved by funneling all requests for the application path to the following function in `WebController.cs`:

```
public static string GetAppRootPathUrl()
{
    string appPath = HttpContext.Current.Request.ApplicationPath; if (appPath == "/")
        appPath = String.Empty; return appPath;
}
```

After this change, concatenations that involve a hard-coded slash, such as above, will generate a correct URL.

As part of the refactoring process, I also added `"default.aspx"` to any calculation when the URL is pointing to the home page. Previously, there were places, such as the redirect URL after adding objects, where the generated URL looked like this:

```
http://www.site.com/galleryserverpro/?aid=25
```

After this change, the URL looks like this:

```
http://www.site.com/galleryserverpro/default.aspx?aid=25
```

I don't think this change actually has an impact, since the first URL still works in browsers (not sure if it would be considered valid, though), but I thought that explicitly referencing it would make it more consistent with the rest of the application.

### Notes

**Defect Id:** 104

**Project:** Gallery Server Pro

**Name:** User cannot change password if e-mail functionality is not configured

**Date Found:** 5/27/2008

**Build #:** 2.0.2898

**Severity:** Medium Impact

**Date Fixed:** 5/28/2008

**Build # of Fix:** 2.0.3070

**Status:** Closed

**Estimated Work:** 1 hrs

**Assigned To:** Roger Martin

**Priority:** High

**Actual Work:** 2 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

## Defect Details

### Description

If the SMTP server is not specified in galleryserverpro.config, an error occurs when the ChangePassword control attempts to send a confirmation e-mail after a password change.

### Replication Procedures

### Resolution

This was resolved by intercepting the OnSendingMail event of the ChangePassword control and manually routing the e-mail to the SendMail method of the WebSiteController class.

### Notes

**Defect Id:** 105

**Project:** Gallery Server Pro

**Name:** User cannot change e-mail address on account

**Date Found:** 5/27/2008

**Build #:** 2.0.2898

**Severity:** Medium Impact

**Date Fixed:** 5/27/2008

**Build # of Fix:** 2.0.3069

**Status:** Closed

**Estimated Work:** 1 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 15 min

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

The change e-mail address functionality on the "My account" page does not work. User can enter a new e-mail address but the user's account is never updated.

### Replication Procedures

### Resolution

This bug was introduced on November 17, 2007 and released as part of version 2.0.2886. I had incorrectly removed a check for IsPostBack in the page load event in an attempt to clean up the code. I reverted the code to the previous version.

### Notes

**Defect Id:** 106

**Project:** Gallery Server Pro

**Name:** Certain images that previously synchronized fail during subsequent synchronizations

**Date Found:** 6/11/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 6/11/2008

**Build # of Fix:** 2.0.3084

**Status:** Closed

**Estimated Work:** 4 hrs

**Assigned To:** Roger Martin

**Priority:** High

**Actual Work:** 2 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

Certain images, when synchronized after an initial synchronization, will cause the synch to fail with the following message:

Unsupported image: the file "C:\file\imagefilename.jpg" cannot be loaded into the .NET Framework's System.Drawing.Bitmap class. This is most likely because it is not an image supported by the .NET Framework. This file must be managed by the GalleryServerPro.Business.GenericObject class rather than the GalleryServerPro.Business.Image class.

## Defect Details

This message is generated when a media object's DisplayObject.Bitmap property is accessed, which causes a media object to be passed into the System.Drawing.Bitmap constructor. If the file is not a valid image, the Bitmap class will throw an ArgumentException, which is caught by Gallery Server and rethrown as GalleryServerPro.ErrorHandler.CustomExceptions.UnsupportedImageTypeException.

The logic path that leads to the error is such that it only occurs when synchronizing an invalid image that has been previously synchronized. The first time it was synchronized Gallery Server correctly characterized it as a Generic media object, but during subsequent synchs it is instantiated as an Image object, which results in the error when the Bitmap property is accessed.

### Replication Procedures

### Resolution

HelperFunctions.cs in the business layer was modified. Specifically, the method DetermineMediaObjectType(int mediaObjectId) was changed so that the OriginalWidth and OriginalHeight values in the gs\_MediaObject table are taken into account - in addition to the file's extension - when determining the type. If the file's extension indicates it is an image, then GalleryObjectType.Image is returned, but only if OriginalWidth and OriginalHeight are greater than int.MinValue. If they are int.MinValue, the method returns GalleryObjectType.Generic.

Also, a third overload of DetermineMediaObjectType(string fileName, int originalWidth, int originalHeight) was created with similar logic to the first one, such that the width and height are used to correctly calculate the type.

### Notes

**Defect Id:** 107

**Project:** Gallery Server Pro

**Name:** Dialog window with the text "true" appears when clicking the Show metadata toolbar icon

**Date Found:** 6/17/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 6/17/2008

**Build # of Fix:** 2.0.3090

**Status:** Closed

**Estimated Work:** 15 min

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 15 min

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

When clicking the 'Show metadata' toolbar command button while viewing an original (not compressed) image, a javascript alert box appears with the text 'true'. Clicking OK causes it to disappear and the page continues to render correctly.

### Replication Procedures

### Resolution

A line of javascript that helped with debuggin was accidentally left in the file mediaobjectview.js. The line was:  
alert(\$get('hr').value);

The line was removed.

### Notes

## Defect Details

**Defect Id:** 108

**Project:** Gallery Server Pro

**Name:** Error during app startup if a role in gs\_Role and aspnet\_Roles differ only by case

**Date Found:** 6/26/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 6/26/2008

**Build # of Fix:** 2.0.3099

**Status:** Closed

**Estimated Work:** 30 min

**Assigned To:** Roger Martin

**Priority:** Low

**Actual Work:** 30 min

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

If a role in the table gs\_Role has a name that differs only by case from a role in aspnet\_Roles, then the startup code in WebSiteController.ValidateRoles() will consider them different and subsequently attempt to create a new role in aspnet\_Roles with the second name. However, the SQL Server role provider does not allow roles to differ only by case, so an error is thrown.

This bug would not typically occur when Gallery Server manages the users and roles, but it can appear when connecting Gallery Server to existing users and roles in new databases.

### Replication Procedures

### Resolution

The methods GetRoleByRoleName and GetRolesByRoleNames in GalleryServerRoleCollection were modified to perform a case-insensitive search.

### Notes

**Defect Id:** 110

**Project:** Gallery Server Pro

**Name:** Timeout error for very large galleries (> 40,000 albums)

**Date Found:** 7/6/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 7/6/2008

**Build # of Fix:** 2.0.3109

**Status:** Closed

**Estimated Work:** 4 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 2 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

When there are a very large number of albums (> 40,000), the data access method Roles\_GetDataReaderRoleAllAlbums can sometimes take so long that the following timeout error is generated: "Timeout expired. The timeout period elapsed prior to completion of the operation or the server is not responding."

### Replication Procedures

### Resolution

To confirm the issue, I created a gallery with 454,000 albums. Sure enough, the stored procedure took more than 30 minutes to execute. I made two changes that brought the execution time to four seconds. Equivalent changes were made in the SQLite provider.

1. Create a non-clustered, non-unique index on the AlbumParentID column in gs\_Album.

SQL Server 2005:

**Code:**IF NOT EXISTS (SELECT \* FROM sys.indexes WHERE object\_id = OBJECT\_ID(N'[dbo].[gs\_Album]'))

## Defect Details

```
AND name = N'IDX_gs_Album_AlbumParentID')
```

```
CREATE NONCLUSTERED INDEX [IDX_gs_Album_AlbumParentID] ON [dbo].[gs_Album]
([AlbumParentID] ASC )
```

```
WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON)
```

SQL Server 2000:

```
Code:IF NOT EXISTS (SELECT * FROM dbo.sysindexes WHERE id = OBJECT_ID(N'[dbo].[gs_Album]') AND
name = N'IDX_gs_MediaObject_FKAlbumID')
```

```
CREATE NONCLUSTERED INDEX [IDX_gs_Album_AlbumParentID] ON [dbo].[gs_Album]
([AlbumParentID] ASC )
```

2. Modify the stored procedure gs\_Role\_AlbumSelectAllAlbumsByRoleName (SQL Server 2000 & 2005). Instead of always building the Album

ID list by drilling down, album by album, first check to see if the role is applied to the root album. If so, then we know the role applies to

all albums, so just return a list of all album ID's for the current gallery.

```
Code:ALTER PROCEDURE [dbo].[gs_Role_AlbumSelectAllAlbumsByRoleName]
```

```
( @RoleName nvarchar(256), @GalleryId int )
```

```
AS
```

```
SET NOCOUNT ON
```

```
/* Retrieve all the album IDs that are affected by the specified role name. The album IDs that are stored in
the gs_Role_Album table only hold the highest ranking album ID. */
```

```
/* If the role is applied to the root album, then we can just return all albums in the gallery. This is
much cheaper than drilling down by album. */
```

```
DECLARE @RootAlbumId int
```

```
SELECT @RootAlbumId = AlbumID FROM gs_Album WHERE (AlbumParentID = 0)
```

```
IF EXISTS (SELECT * FROM gs_Role_Album WHERE FKRoleName = @RoleName AND FKAlbumId =
@RootAlbumId)
```

```
BEGIN
```

```
SELECT AlbumID FROM gs_Album WHERE FKGalleryID = @GalleryId
```

```
END
```

```
ELSE
```

```
BEGIN
```

```
/* The role applies to an album or albums below the root album, so we need to drill down and retrieve all
the children. Start by creating a temporary table to hold our data. */
```

```
DECLARE @AlbumList table
```

```
(AlbumID int not null,
```

```
AlbumParentID int not null,
```

```
AlbumDepth int not null)
```

```
/* Insert the top level album IDs. */
```

```
INSERT @AlbumList (AlbumID, AlbumParentID, AlbumDepth)
```

```
SELECT FKAlbumID, 0, 1
```

```
FROM gs_Role_Album INNER JOIN gs_Album ON gs_Role_Album.FKAlbumId = gs_Album.AlbumID
```

```
WHERE (gs_Role_Album.FKRoleName = @RoleName) AND (gs_Album.FKGalleryID = @GalleryId)
```

## Defect Details

```
/* Continue drilling down, level by level, until we reach a level where there are no more child albums. */
WHILE (@@rowcount > 0) BEGIN
    INSERT @AlbumList (AlbumId, AlbumParentId, AlbumDepth)
    SELECT a.AlbumId, a.AlbumParentId, al.AlbumDepth + 1
    FROM gs_Album a JOIN @AlbumList al ON a.AlbumParentId = al.AlbumId
    WHERE al.AlbumDepth = (SELECT MAX(AlbumDepth) FROM @AlbumList)
END

/* Retrieve the list of album IDs. */
SELECT AlbumId
FROM @AlbumList
END

RETURN
```

### Notes

**Defect Id:** 111

**Project:** Gallery Server Pro

**Name:** Receive InvalidMediaObjectException during synchronization

**Date Found:** 7/6/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 7/6/2008

**Build # of Fix:** 2.03109

**Status:** Closed

**Estimated Work:** 30 min

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 30 min

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

If the path specified in the mediaObjectPath setting in galleryserverpro.config differs in case from the actual path, a InvalidMediaObjectException exception may be thrown during synchronization. For example, if mediaObjectPath = "D:\Pictures" and the actual path is "D:\pictures", this exception is thrown during the validation check of the FileInfo property setter of GalleryServerPro.Business.DisplayObject.

Limited testing seems to indicate the error is thrown only for objects that have been shortened to the older Windows 8.3 naming style. Specifically, the error occurred only for a zip file named 22OBJE~1.ZIP. Other files in the same directory did not cause the error. This probably has something to do with how .NET or Win32 processes the files internally.

Also, it may be relevant that, although the directory was named "D:\pictures", this happened to be the directory that I instructed Vista to treat as my photo directory. This caused Vista to display the directory as "Pictures" in certain Windows views.

### Replication Procedures

### Resolution

The property setter for GalleryServerPro.Business.DisplayObject.FileInfo was modified so that the directory string comparison is done in a case-insensitive way.

### Notes

## Defect Details

**Defect Id:** 112

**Project:** Gallery Server Pro

**Name:** Message not cleared after logging in

**Date Found:** 7/14/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 8/4/2008

**Build # of Fix:** 2.1.3138

**Status:** Closed

**Estimated Work:** 1 hrs

**Assigned To:** Roger Martin

**Priority:** Low

**Actual Work:** 1 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

When the home page (default.aspx) receives the query string parameter "msg", it displays the relevant message corresponding to the query string value. If this message is displayed to an anonymous user, and then the user logs on, the message is still displayed. In some cases this is confusing.

For example, a logged on user may navigate to an album that is not visible to anonymous users. When that user logs off, she is redirected to the home page where the following message is displayed:

"Insufficient permissionYou do not have permission to view the specified album. The albums for which you do have access are shown instead."

That is appropriate, because the user no longer has permission to see the album they were previously viewing. However, if the user then logs on using the username and password boxes in the top right, the original message remains after logon. It would be less confusing if the message was no longer displayed.

### Resolution

This bug was introduced when I changed all the Redirects in the web app to use the overload with the endResponse parameter set to false. In this case, the master page site.Master redirects the user after a login or logout to the current URL with any msg query string parameters removed. However, this overload of Redirect does not work from within the LoggedIn and LoggedOut event handlers. It is simply ignored and the page continues normal postback rendering, which will display the current page with any previous messages still showing.

In order to force the Redirect to work, I reverted to the original technique:

```
Response.Redirect(GetRedirectUrlAfterLoginOrLogout());
```

**Defect Id:** 113

**Project:** Gallery Server Pro

**Name:** Thumbnail / optimized images sometimes created with wrong width/height

**Date Found:** 7/16/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 7/16/2008

**Build # of Fix:** 2.0.3119

**Status:** Closed

**Estimated Work:** 1 hrs

**Assigned To:** Roger Martin

**Priority:** Low

**Actual Work:** 15 min

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

Images with extreme aspect ratios - for example, 400px width by 2px height may result in thumbnail or optimized images with a different aspect ratio.

### Resolution

This is caused by a bug in GalleryServerPro.Business.ImageHelper.CreateResizedBitmap where two integers are divided and assigned to a double. In C#, when two integers are divided, the result is an integer, resulting in the loss

## Defect Details

of any decimal portion. The solution is to cast the integers before performing the division:

```
xScaleFactor = (float)calculatedNewWidth / (float)sourceBmpWidth;// Update the scale factor to reflect the new width
```

```
yScaleFactor = (float)calculatedNewHeight / (float)sourceBmpHeight;// Update the scale factor to reflect the new height
```

**Defect Id:** 114

**Project:** Gallery Server Pro

**Name:** Installer does not install data scripts to SQL Server 2008

**Date Found:** 7/31/2008

**Build #:** 2.0.2898

**Severity:** High Impact

**Date Fixed:** 7/31/2008

**Build # of Fix:** 2.1.3134

**Status:** Closed

**Estimated Work:** 1 hrs

**Assigned To:** Roger Martin

**Priority:** High

**Actual Work:** 1 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

A bug in the function GetSqlVersion() in the Installer caused SQL Server 2008 to not be correctly identified. Because of this error, a path to a directory instead of a SQL script was passed to the ExecuteSqlInFile function, resulting in an exception.

### Resolution

The function GetSqlVersion() was fixed so that SQL Server 2008 is correctly identified. A separate bug also caused versions of SQL Server prior to 7 to also not be identified. This was fixed so that if the installer encountered an unsupported early version, a NotSupportedException exception is thrown with a special message describing the issue.

**Defect Id:** 115

**Project:** Gallery Server Pro

**Name:** User does not receive friendly message when Gallery Server does not have delete permission to

**Date Found:** 8/1/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 8/1/2008

**Build # of Fix:** 2.1.3135

**Status:** Closed

**Estimated Work:** 1 hrs

**Assigned To:** Roger Martin

**Priority:** Low

**Actual Work:** 1 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

Each time Gallery Server starts some test code runs to make sure it has read/write/delete permission to the media objects directory. If it doesn't, a friendly message informs the user of the issue and provides guidance how to solve it. However, the friendly message does not appear if the process has write permission but not delete permission. In this case, a "hard" error occurs, and if customErrors is set to On or RemoteOnly, the user receives a generic "Gallery Server is feeling ill" message, which is not very informative.

### Resolution

The function GalleryServerPro.Business.HelperFunctions.ValidatePhysicalPathExistsAndIsReadWritable was modified so that the command to delete the test file is contained within a try..catch, as follows:

try

## Defect Details

```
{  
  File.Delete(testFilePath);  
}  
catch(Exception ex)  
{  
  throw new GalleryServerPro.ErrorHandler.CustomExceptions.CannotWriteToMediaObjectDirectoryException  
(fullPhysicalPath, ex);  
}
```

Note that in this situation, the test file (a tiny text file) is left in the media objects directory. A future enhancement might be to inform the user about its existence so that it can be manually deleted.

**Defect Id:** 117

**Project:** Gallery Server Pro

**Name:** Installer cannot continue after failed attempt at creating a login

**Date Found:** 8/22/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 8/22/2008

**Build # of Fix:** 2.1.3156

**Status:** Closed

**Estimated Work:** 1 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 1 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

On the SQL Server Connection - Runtime step of the installer wizard, the user can specify a new login to be created. Upon clicking Next, the installer checks that the login can be created by executing the SQL and then rolling back the transaction. If an error occurs, (e.g. the password is not complex enough), the transaction is left open, and subsequent SQL statements time out.

### Resolution

This was solved by adding logic to the create user SQL that checked for an error condition after each statement and rolls back the transaction if necessary. The affected files were AddSqlLoginSql2005.sql and AddWinLoginSql2005.sql.

The SQL Server 2000 files were not modified because the installer does not test that the user can be created under SQL Server 2000 (it does not support rolling back the creation of a user).

**Defect Id:** 118

**Project:** Gallery Server Pro

**Name:** Image occasionally not rendered

**Date Found:** 8/22/2008

**Build #:** 2.0.2898

**Severity:** Medium Impact

**Date Fixed:** 8/22/2008

**Build # of Fix:** 2.1.3156

**Status:** Closed

**Estimated Work:** 1 hrs

**Assigned To:** Roger Martin

**Priority:** High

**Actual Work:** 1 hrs

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

Occasionally, certain media objects would not render in the browser. No error would be displayed, but an exception occurred in the ASHX handler with this message: Invalid length for a Base-64 char array.

### Resolution

This was caused by the encrypted query string parameter not being URL encoded, and thus failing as a valid URL in some cases, such as when the encryption string contained a backslash.

The solution was to URL encode the query string when it is sent to the browser, and then URL decode it in the

## Defect Details

ASHX handler.

Also, the following line in `MediaObjectHtmlBuilder.GetMediaObjectUrl`:

```
return string.Format(CultureInfo.CurrentCulture, "{0}/handler/getmediaobject.ashx?{1}",  
WebsiteController.GetAppRootPathUrl(), queryString);
```

was changed to this:

```
return string.Concat(WebsiteController.GetAppRootPathUrl(), "/handler/getmediaobject.ashx?", queryString);
```

This was changed because the `string.Format` method somehow messed up the query string value. A bonus is that the `Concat` function is a little faster, so it should have been used anyway.

**Defect Id:** 120

**Project:** Gallery Server Pro

**Name:** Watermark text sometimes extends beyond image border

**Date Found:** 8/26/2008

**Build #:** 2.0.2898

**Severity:** Low Impact

**Date Fixed:** 8/26/2008

**Build # of Fix:** 2.1.3160

**Status:** Closed

**Estimated Work:** 1 hrs

**Assigned To:** Roger Martin

**Priority:** Medium

**Actual Work:** 30 min

**Reported By:** Roger Martin

**Percent Complete:** 100

### Description

When the watermark text is specified to be 100% of the width of the image, the text sometimes does not fit on the image and is clipped.

### Resolution

This occurred because the algorithm in `Watermark.ApplyTextWatermark` determines the font size to use by setting up a loop where the font size is increased until its width exceeds that of the image. When that happens, the font size is actually one bigger than it should be, since by the time it breaks out of the loop the text width exceeds the width of the image.

The solution was to add the following code after breaking out of the loop so that the font size is one less:

```
fontSize -= 1;
```

```
font.Dispose();
```

```
font = new Font(this.TextFontName, fontSize);
```