

# Gallery Server Pro - Fixed Defect Report

Version 2.6.0

**Defect:** Uploading 3 media files results in 4 media objects

**ID:** 337

**Created:** 14-Jul-10

**State:** Closed

**Resolved:** 21-Oct-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** When a user selects three media files and uploads them, the first one is repeated so that four media objects are added. This applies only to the ComponentArt Upload control.

By using the debugger I can see that in the Uploaded event, the UploadUploadedEventArgs.UploadedFiles property has four items when it should only have 3.

**Resolution:** Resolved by replacing control with Plupload.

---

**Defect:** Viewing error log generates "String was not recognized as a valid DateTime" error

**ID:** 486

**Created:** 21-Jun-11

**State:** Closed

**Resolved:** 21-Jun-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** When the server has a non-US culture, viewing the error log can result in this error:  
System.ArgumentException - String was not recognized as a valid DateTime.Couldn't store <20/06/2011 15:54:44> in TimeStamp Column. Expected type is DateTime.

Stack Trace:

```
at System.Data.DataColumn.set_Item(Int32 record, Object value)
at System.Data.DataRow.set_Item(DataColumn column, Object value)
at GalleryServerPro.Web.Controller.AppErrorController.GetAppErrorsDataSet(Int32 galleryId, Boolean includeSystemErrors)
at GalleryServerPro.Web.Pages.Admin.errorlog.BindGrid()
at GalleryServerPro.Web.Pages.Admin.errorlog.Page_Load(Object sender, EventArgs e)
at System.Web.Util.CalliHelper.EventArgFunctionCaller(IntPtr fp, Object o, Object t, EventArgs e)
at System.Web.Util.CalliEventHandlerDelegateProxy.Callback(Object sender, EventArgs e)
at System.EventHandler.Invoke(Object sender, EventArgs e)
at System.Web.UI.Control.OnLoad(EventArgs e)
at System.Web.UI.Control.LoadRecursive()
at System.Web.UI.Control.LoadRecursive()
at System.Web.UI.Control.LoadRecursive()
at System.Web.UI.Control.LoadRecursive()
at System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
```

Inner Exception:

System.FormatException - String was not recognized as a valid DateTime.

Stack Trace:

```
at System.DateTimeParse.Parse(String s, DateTimeFormatInfo dtfi, DateTimeStyles styles)
at System.Convert.ToDateTime(String value, IFormatProvider provider)
at System.String.System.IConvertible.ToDateTime(IFormatProvider provider)
at System.Data.Common.DateTimeStorage.Set(Int32 record, Object value)
at System.Data.DataColumn.set_Item(Int32 record, Object value)
```

**Resolution:** This was caused by the code attempting to store a culture-specific DateTime string in a DataColumn whose table has its Locale property set to CultureInfo.InvariantCulture. The Locale property was assigned per recommendation from Code Analysis performed prior to releasing 2.5.0, but the timestamp should have been formatted in an invariant culture format to match.

The fix was to format the timestamp when assigning it to the DataColumn. The original code in AppErrorController.GetAppErrorsDataSet():

```
errRow[2] = err.Timestamp.ToString();
```

The fixed code:

```
errRow[2] = err.Timestamp.ToString(CultureInfo.InvariantCulture);
```

**Defect:** (DotNetNuke only) Error when anonymous user views gallery that is configured to require authentication for all users

**ID:** 487

**Created:** 21-Jun-11

**State:** Closed

**Resolved:** 21-Jun-11

**Area:** GSP-DNN

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** When anonymous browsing is disabled for a gallery (on the gallery settings page), the following error may appear when an unauthenticated user attempts to view the gallery:

"The Gallery Server Pro DotNetNuke module does not support specifying a true value for the includeQueryString parameter of Util.GetCurrentPageUrl"

**Resolution:** The logic that redirects the user to the login page passes a true value for the includeQueryString parameter to the function Utils.GetCurrentPageUrl(). Since DotNetNuke may use query strings to discriminate between pages, GSP should never strip the query string from the URL. Because of this, validation code was added that threw an exception when this situation was detected. However, this validation was unnecessarily strict. A better solution is to ignore the includeQueryString parameter, which is what the fix does.

---

## Defect: An ArgumentNullException may be thrown during an auto-sync

**ID:** 488

**Created:** 21-Jun-11

**State:** Closed

**Resolved:** 21-Jun-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** Auto-syncs are run on a background thread, which does not have a current HTTP context. This can lead to the following error when syncing an image containing GPS data:  
ArgumentNullException - Value cannot be null. Parameter name: url  
Stack trace:  
at GalleryServerPro.Web.Utils.RemoveQueryStringParameter(String url, String queryStringParameterName)  
at GalleryServerPro.Web.Utils.AddQueryStringParameter(String url, String queryStringParameterNameValue)  
at GalleryServerPro.Web.Utils.GetUrl(PagelId page, String format, Object[] args)  
at GalleryServerPro.Web.Controller.GalleryController.GetGpsMapUrl(IGalleryObject galleryObject, String latitude, String longitude, String gpsLocation)  
at GalleryServerPro.Web.Controller.GalleryController.AddGpsLocationWithMapLink(IGalleryObject galleryObject, IMetadataDefinitionCollection metadataDisplaySettings)  
at GalleryServerPro.Web.Controller.GalleryController.GalleryObjectMetadataLoaded(Object sender, EventArgs e)  
at GalleryServerPro.Business.GalleryObject.set\_IsMetadataLoaded(Boolean value)  
at GalleryServerPro.Business.Factory.AddMediaObjectMetadata(IGalleryObject go)  
at GalleryServerPro.Business.Factory.RetrieveMediaObjectFromDataStore(Int32 id, GalleryObjectType galleryObjectType, IAlbum parentAlbum)  
at GalleryServerPro.Business.Factory.RetrieveMediaObject(Int32 mediaObjectId, GalleryObjectType galleryObjectType, IAlbum parentAlbum)  
at GalleryServerPro.Business.Album.VerifyThumbnailIsInflated(IDisplayObject thumbnail)  
at GalleryServerPro.Business.GalleryObject.get\_Thumbnail()  
at GalleryServerPro.Business.Album.AssignAlbumThumbnail(IAlbum album, Boolean recursivelyAssignParentAlbums, Boolean recursivelyAssignChildrenAlbums, String userName)  
at GalleryServerPro.Business.Album.AssignAlbumThumbnail(IAlbum album, Boolean recursivelyAssignParentAlbums, Boolean recursivelyAssignChildrenAlbums, String userName)  
at GalleryServerPro.Business.Album.AssignAlbumThumbnail(IAlbum album, Boolean recursivelyAssignParentAlbums, Boolean recursivelyAssignChildrenAlbums, String userName)  
at GalleryServerPro.Business.SynchronizationManager.Synchronize(String synchId, IAlbum album, String userName)  
at GalleryServerPro.Web.Controller.GalleryController.Synchronize(Object syncSettingsObject)

**Resolution:** Utils.GetUrl() was modified to return null when no HTTP context is present. The calling function is already able to gracefully handle a null result.

## Defect: Error "Property cannot be found" when adding or syncing an image

**ID:** 490

**Created:** 23-Jun-11

**State:** Closed

**Resolved:** 23-Jun-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** The following error can occur when syncing or adding an image:  
System.ArgumentException: Property cannot be found.  
Stack trace:  
at System.Windows.Media.Imaging.BitmapDecoder.SetupDecoderFromUriOrStream(Uri uri, Stream stream, BitmapCacheOption cacheOption, Guid& clsId, Boolean& isOriginalWritable, Stream& uriStream, UnmanagedMemoryStream& unmanagedMemoryStream, SafeFileHandle& safeFilehandle) at System.Windows.Media.Imaging.BitmapDecoder.CreateFromUriOrStream(Uri baseUri, Uri uri, Stream stream, BitmapCreateOptions createOptions, BitmapCacheOption cacheOption, RequestCachePolicy uriCachePolicy, Boolean insertInDecoderCache) at System.Windows.Media.Imaging.BitmapDecoder.Create(Stream bitmapStream, BitmapCreateOptions createOptions, BitmapCacheOption cacheOption) at GalleryServerPro.Business.WpfMetadataExtractor.GetBitmapMetadata(String imagePath) at GalleryServerPro.Business.WpfMetadataExtractor.AddWpfBitmapMetadata(String imagePath, IGalleryObjectMetadataItemCollection metadataItems) at GalleryServerPro.Business.Metadata.MediaObjectMetadataExtractor.GetGalleryObjectMetadataItemCollection() at GalleryServerPro.Business.Image..ctor(Int32 id, IAlbum parentAlbum, String title, String hashKey, String thumbnailFilename, Int32 thumbnailWidth, Int32 thumbnailHeight, Int32 thumbnailSizeKb, String optimizedFilename, Int32 optimizedWidth, Int32 optimizedHeight, Int32 optimizedSizeKb, String originalFilename, Int32 originalWidth, Int32 originalHeight, Int32 originalSizeKb, Int32 sequence, String createdByUsername, DateTime dateAdded, String lastModifiedByUsername, DateTime dateLastModified, Boolean isPrivate, Boolean isInflated, FileInfo imageFile) at GalleryServerPro.Business.Image..ctor(FileInfo imageFile, IAlbum parentAlbum) at GalleryServerPro.Business.Factory.CreateImageInstance(FileInfo imageFile, IAlbum parentAlbum) at GalleryServerPro.Business.Factory.CreateLocalMediaObjectInstance(FileInfo mediaObjectFile, IAlbum parentAlbum) at GalleryServerPro.Business.Factory.CreateMediaObjectInstance(FileInfo mediaObjectFile, IAlbum parentAlbum, String externalHtmlSource, MimeTypeCategory mimeTypeCategory) at GalleryServerPro.Business.SynchronizationManager.CreateNewMediaObject(IAlbum album, FileInfo file) at GalleryServerPro.Business.SynchronizationManager.SynchronizeMediaObjectFiles(DirectoryInfo directory, IAlbum album) at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo parentDirectory, IAlbum parentAlbum) at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo parentDirectory, IAlbum parentAlbum) at GalleryServerPro.Business.SynchronizationManager.Synchronize(String synchId, IAlbum album, String userName) at Gsp.Gallery.Synchronize(Int32 albumId, String synchId, Boolean isRecursive, Boolean overwriteThumb, Boolean overwriteOpt, Boolean regenerateMetadata)

**Resolution:** The problem is caused by the .NET Framework having trouble extracting metadata from an image file, most likely due to the metadata not conforming to the expected format or syntax. Added a catch block around the call to BitmapDecoder.Create() in GalleryServerPro.Business.WpfMetadataExtractor.GetBitmapMetadata() to silently handle ArgumentException exceptions.

Versions earlier than 2.5.0 had an exception handler around the reflection-based call to the WPF metadata extractor that caught exceptions like this (since any exception would be wrapped in a `System.Reflection.TargetInvocationException` that was handled).

---

**Defect:** Error "Object must be initialized before operation can be performed" when adding or syncing an image

**ID:** 491

**Created:** 23-Jun-11

**State:** Closed

**Resolved:** 23-Jun-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** The following error can occur when syncing or adding an image:  
System.InvalidOperationException: Object must be initialized before operation can be performed.  
Stack Trace:  
at System.Windows.Media.Imaging.BitmapMetadata.GetQuery(String query)  
at GalleryServerPro.Business.WpfMetadataExtractor.AddIptcMetadata(BitmapMetadata bmpMetadata, IGalleryObjectMetadataItemCollection metadataItems)  
at GalleryServerPro.Business.WpfMetadataExtractor.AddWpfBitmapMetadata(String imagePath, IGalleryObjectMetadataItemCollection metadataItems)  
at  
GalleryServerPro.Business.Metadata.MediaObjectMetadataExtractor.GetGalleryObjectMetadataItemCollection()  
at GalleryServerPro.Business.Image..ctor(Int32 id, IAlbum parentAlbum, String title, String hashKey, String thumbnailFilename, Int32 thumbnailWidth, Int32 thumbnailHeight, Int32 thumbnailSizeKb, String optimizedFilename, Int32 optimizedWidth, Int32 optimizedHeight, Int32 optimizedSizeKb, String originalFilename, Int32 originalWidth, Int32 originalHeight, Int32 originalSizeKb, Int32 sequence, String createdByUsername, DateTime dateAdded, String lastModifiedByUsername, DateTime dateLastModified, Boolean isPrivate, Boolean isInflated, FileInfo imageFile)  
at GalleryServerPro.Business.Image..ctor(FileInfo imageFile, IAlbum parentAlbum)  
at GalleryServerPro.Business.Factory.CreateImageInstance(FileInfo imageFile, IAlbum parentAlbum)  
at GalleryServerPro.Business.Factory.CreateLocalMediaObjectInstance(FileInfo mediaObjectFile, IAlbum parentAlbum)  
at GalleryServerPro.Business.Factory.CreateMediaObjectInstance(FileInfo mediaObjectFile, IAlbum parentAlbum, String externalHtmlSource, MimeTypeCategory mimeTypeCategory)  
at GalleryServerPro.Web.Pages.Task.addobjects.AddUploadedFilesLessThanFullTrust()  
at GalleryServerPro.Web.Pages.Task.addobjects.btnUploadMediaObjects\_Click(Object sender, EventArgs e)  
at System.Web.UI.WebControls.Button.OnClick(EventArgs e)  
at System.Web.UI.WebControls.Button.RaisePostBackEvent(String eventArgument)  
at System.Web.UI.WebControls.Button.System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(String eventArgument)  
at System.Web.UI.Page.RaisePostBackEvent(IPostBackEventHandler sourceControl, String eventArgument)  
at System.Web.UI.Page.RaisePostBackEvent(NameValueCollection postData)  
at System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)  
Inner exception:  
System.Runtime.InteropServices.COMException: Exception from HRESULT: 0x88982F0C

**Resolution:** The problem is caused by the .NET Framework having trouble extracting IPTC metadata from an image file, most likely due to the metadata not conforming to the expected format or syntax. Added a catch block around the call to bmpMetadata.GetQuery() in GalleryServerPro.Business.WpfMetadataExtractor.AddIptcMetadata() to silently handle InvalidOperationException exceptions.

Versions earlier than 2.5.0 had an exception handler around the reflection-based call to the WPF metadata extractor that caught exceptions like this (since any exception would be wrapped in a System.Reflection.TargetInvocationException that was handled).

**Defect:** Error: Collection was modified; enumeration operation may not execute

**ID:** 492

**Created:** 25-Jun-11

**State:** Closed

**Resolved:** 25-Jun-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** The following error can occur:  
System.InvalidOperationException - Error: Collection was modified; enumeration operation may not execute.

Stack trace:

```
at System.ThrowHelper.ThrowInvalidOperationException(ExceptionResource resource)
at System.Collections.Generic.List`1.Enumerator.MoveNextRare()
at System.Collections.Generic.List`1.Enumerator.MoveNext()
at GalleryServerPro.Business.Album.GetChildGalleryObjects(Boolean sortBySequence, Boolean excludePrivateObjects)
at GalleryServerPro.Business.Album.GetChildGalleryObjects(GalleryObjectType galleryObjectType, Boolean sortBySequence, Boolean excludePrivateObjects)
at GalleryServerPro.Web.Pages.GalleryPage.GetMediaObjectHtml(IGalleryObject mediaObject, DisplayObjectType displayType, Boolean isCallBack)
at GalleryServerPro.Web.Controls.mediaobjectview.RenderMediaObject()
at GalleryServerPro.Web.Controls.mediaobjectview.Page_Load(Object sender, EventArgs e)
at System.Web.Util.CalliHelper.EventArgFunctionCaller(IntPtr fp, Object o, Object t, EventArgs e)
at System.Web.Util.CalliEventHandlerDelegateProxy.Callback(Object sender, EventArgs e)
at System.Web.UI.Control.OnLoad(EventArgs e)
at System.Web.UI.Control.LoadRecursive()
at System.Web.UI.Control.LoadRecursive()
at System.Web.UI.Control.LoadRecursive()
at System.Web.UI.Control.LoadRecursive()
at System.Web.UI.Control.LoadRecursive()
at System.Web.UI.Control.LoadRecursive()
at System.Web.UI.Control.LoadRecursive()
at System.Web.UI.Control.LoadRecursive()
at System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
```

**Resolution:** Added a lock statement around the foreach in  
GalleryServerPro.Business.Album.GetChildGalleryObjects(bool sortBySequence, bool excludePrivateObjects).

## Defect: Error "Out of memory" when adding or syncing an image

**ID:** 493

**Created:** 25-Jun-11

**State:** Closed

**Resolved:** 25-Jun-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** I observed a synchronization where one of the images through an OutOfMemoryException exception during the call to BitmapDecoder.Create() in GalleryServerPro.Business.WpfMetadataExtractor.GetBitmapMetadata().

**Resolution:** Added a catch around the call to BitmapDecoder.Create() in GalleryServerPro.Business.WpfMetadataExtractor.GetBitmapMetadata for OutOfMemoryException exceptions. Inside the catch handler, the code waits for the garbage collector to finish and then tries again. If it fails again, it gives up. When this happens, the gallery will abandon the attempt to extract metadata from this file.  
Versions earlier than 2.5.0 had an exception handler around the reflection-based call to the WPF metadata extractor that caught exceptions like this (since any exception would be wrapped in a System.Reflection.TargetInvocationException that was handled).

---

## Defect: Error "Unexpected property type or value" when adding or syncing an image

**ID:** 495

**Created:** 28-Jun-11

**State:** Closed

**Resolved:** 28-Jun-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** The following error can occur when syncing or adding an image:  
System.IO.FileFormatException: Unexpected property type or value.  
Stack trace:  
at System.Windows.Media.Imaging.BitmapDecoder.SetupDecoderFromUriOrStream(Uri uri, Stream stream, BitmapCacheOption cacheOption, Guid& clsId, Boolean& isOriginalWritable, Stream& uriStream, UnmanagedMemoryStream& unmanagedMemoryStream, SafeFileHandle& safeFilehandle)  
at System.Windows.Media.Imaging.BitmapDecoder.CreateFromUriOrStream(Uri baseUri, Uri uri, Stream stream, BitmapCreateOptions createOptions, BitmapCacheOption cacheOption, RequestCachePolicy uriCachePolicy, Boolean insertInDecoderCache)  
at System.Windows.Media.Imaging.BitmapDecoder.Create(Stream bitmapStream, BitmapCreateOptions createOptions, BitmapCacheOption cacheOption)  
at GalleryServerPro.Business.WpfMetadataExtractor.GetBitmapMetadata(String imagePath)  
at GalleryServerPro.Business.WpfMetadataExtractor.AddWpfBitmapMetadata(String imagePath, IGalleryObjectMetadataItemCollection metadataItems)  
at  
GalleryServerPro.Business.Metadata.MediaObjectMetadataExtractor.GetGalleryObjectMetadataItemCollection()  
at GalleryServerPro.Business.Image..ctor(Int32 id, IAlbum parentAlbum, String title, String hashKey, String thumbnailFilename, Int32 thumbnailWidth, Int32 thumbnailHeight, Int32 thumbnailSizeKb, String optimizedFilename, Int32 optimizedWidth, Int32 optimizedHeight, Int32 optimizedSizeKb, String originalFilename, Int32 originalWidth, Int32 originalHeight, Int32 originalSizeKb, Int32 sequence, String createdByUsername, DateTime dateAdded, String lastModifiedByUsername, DateTime dateLastModified, Boolean isPrivate, Boolean isInflated, FileInfo imageFile)  
at GalleryServerPro.Business.Image..ctor(FileInfo imageFile, IAlbum parentAlbum)  
at GalleryServerPro.Business.Factory.CreateImageInstance(FileInfo imageFile, IAlbum parentAlbum)  
at GalleryServerPro.Business.Factory.CreateLocalMediaObjectInstance(FileInfo mediaObjectFile, IAlbum parentAlbum)  
at GalleryServerPro.Business.Factory.CreateMediaObjectInstance(FileInfo mediaObjectFile, IAlbum parentAlbum, String externalHtmlSource, MimeTypeCategory mimeTypeCategory)  
at GalleryServerPro.Business.SynchronizationManager.CreateNewMediaObject(IAlbum album, FileInfo file)  
at GalleryServerPro.Business.SynchronizationManager.SynchronizeMediaObjectFiles(DirectoryInfo directory, IAlbum album)  
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo parentDirectory, IAlbum parentAlbum)  
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo parentDirectory, IAlbum parentAlbum)  
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo parentDirectory, IAlbum parentAlbum)  
at GalleryServerPro.Business.SynchronizationManager.Synchronize(String synchId, IAlbum album, String userName)  
at Gsp.Gallery.Synchronize(Int32 albumId, String synchId, Boolean isRecursive, Boolean overwriteThumb, Boolean overwriteOpt, Boolean regenerateMetadata)

Inner Ex Type: System.Runtime.InteropServices.COMException  
Inner Ex Message: Exception from HRESULT: 0x88982F8E  
Inner Ex Source: <unknown>  
Inner Ex Target Site: <unknown>  
Inner Ex Stack Trace: <unknown>

**Resolution:** The problem is caused by the .NET Framework having trouble extracting metadata from an image file, most likely due to the metadata not conforming to the expected format or syntax. Added a catch block around the call to `BitmapDecoder.Create()` in `GalleryServerPro.Business.WpfMetadataExtractor.GetBitmapMetadata()` to silently handle `FormatException` exceptions.  
Versions earlier than 2.5.0 had an exception handler around the reflection-based call to the WPF metadata extractor that caught exceptions like this (since any exception would be wrapped in a `System.Reflection.TargetInvocationException` that was handled).

---

**Defect:** `ArgumentException` or `InvalidOperationException` when adding or syncing an image

**ID:** 496

**Created:** 28-Jun-11

**State:** Closed

**Resolved:** 28-Jun-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** An `ArgumentException` or `InvalidOperationException` can occur when adding or syncing an image. The exceptions are thrown when accessing one of the properties of the `BitmapMetadata` class in `GalleryServerPro.Business.WpfMetadataExtractor.AddWpfBitmapMetadata()` (such as `DateTaken`, `CameraModel`, etc).

**Resolution:** The problem is caused by the .NET Framework having trouble extracting metadata from an image file, most likely due to the metadata not conforming to the expected format or syntax. Added two catch blocks around the call to `GalleryServerPro.Business.WpfMetadataExtractor.AddWpfBitmapMetadata()` to silently handle these exceptions.  
Versions earlier than 2.5.0 had an exception handler around the reflection-based call to the WPF metadata extractor that caught exceptions like this (since any exception would be wrapped in a `System.Reflection.TargetInvocationException` that was handled).

---

## Defect: FileNotFoundException occurs during a synchronization

**ID:** 497

**Created:** 29-Jun-11

**State:** Closed

**Resolved:** 29-Jun-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** The following error can occur during a synchronization:

FileNotFoundException: Could not find file 'D:\Archive\Layalina\Layalina February10\By Photographers\Ousama\4\ليباليينا\جيهان---التلال في مطعم زبيب في عيد الحب للفنان أيمن زبيب في مطعم التلال (1).jpeg'.

Stack Trace:

```
at System.IO.__Error.WinIOError(Int32 errorCode, String maybeFullPath)
at System.IO.FileStream.Init(String path, FileMode mode, FileAccess access, Int32 rights, Boolean useRights, FileShare share, Int32 bufferSize, FileOptions options, SECURITY_ATTRIBUTES secAttrs, String msgPath, Boolean bFromProxy, Boolean useLongPath)
at System.IO.FileStream..ctor(String path, FileMode mode, FileAccess access, FileShare share, Int32 bufferSize)
at System.IO.File.OpenFile(String path, FileAccess access, SafeFileHandle& handle)
at System.IO.File.SetCreationTimeUtc(String path, DateTime creationTimeUtc)
at System.IO.FileSystemInfo.set_CreationTimeUtc(DateTime value)
at GalleryServerPro.Business.HelperFunctions.GetHashKeyUnique(FileInfo originalFile)
at GalleryServerPro.Business.SynchronizationManager.UpdateExistingMediaObject(IAlbum album, IGalleryObject mediaObject)
at GalleryServerPro.Business.SynchronizationManager.SynchronizeMediaObjectFiles(DirectoryInfo directory, IAlbum album)
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo parentDirectory, IAlbum parentAlbum)
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo parentDirectory, IAlbum parentAlbum)
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo parentDirectory, IAlbum parentAlbum)
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo parentDirectory, IAlbum parentAlbum)
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo parentDirectory, IAlbum parentAlbum)
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo parentDirectory, IAlbum parentAlbum)
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo parentDirectory, IAlbum parentAlbum)
at GalleryServerPro.Business.SynchronizationManager.Synchronize(String synchId, IAlbum album, String userName)
at Gsp.Gallery.Synchronize(Int32 albumId, String synchId, Boolean isRecursive, Boolean overwriteThumb, Boolean overwriteOpt, Boolean regenerateMetadata)
```

**Resolution:** The error is thrown when the GalleryServerPro.Business.HelperFunctions.GetHashKeyUnique() function attempts to update the creation timestamp and the .NET Framework generates a FileNotFoundException exception. This behavior was observed while syncing a media file where the path contained a mixture of English and Arabic characters, although it is unclear what the exact cause was, since the file obviously existed or else this function never would have been called. It is expected this is a very rare occurrence, as it has not previously been reported or observed.

While the root cause is unknown, an acceptable resolution is to catch the exception and return an empty

string. In this case the media object no longer has a valid hash key stored in the database, but the only consequence is that it won't be "found" by the sync process if a user moves it to another album with a tool like Windows Explorer. In this case, the original media record will be deleted and a new one inserted, causing the loss of any data associated with the original media record (title, sequence, etc). Since both the original exception and the resulting consequence of the workaround are rare occurrences, this is acceptable.

---

## Defect: OverflowException when adding or syncing an image

**ID:** 498

**Created:** 29-Jun-11

**State:** Closed

**Resolved:** 29-Jun-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** The following error can occur when syncing or adding an image:

System.OverflowException: The image data generated an overflow during processing.

Stack trace:

```
at System.Windows.Media.Imaging.BitmapDecoder.SetupDecoderFromUriOrStream(Uri uri, Stream stream,
BitmapCacheOption cacheOption, Guid& clsId, Boolean& isOriginalWritable, Stream& uriStream,
UnmanagedMemoryStream& unmanagedMemoryStream, SafeFileHandle& safeFilehandle)
at System.Windows.Media.Imaging.BitmapDecoder.CreateFromUriOrStream(Uri baseUri, Uri uri, Stream
stream, BitmapCreateOptions createOptions, BitmapCacheOption cacheOption, RequestCachePolicy
uriCachePolicy, Boolean insertInDecoderCache)
at System.Windows.Media.Imaging.BitmapDecoder.Create(Stream bitmapStream, BitmapCreateOptions
createOptions, BitmapCacheOption cacheOption)
at GalleryServerPro.Business.WpfMetadataExtractor.GetBitmapDecoder(String imagePath)
at GalleryServerPro.Business.WpfMetadataExtractor.GetBitmapMetadata(String imagePath)
at GalleryServerPro.Business.WpfMetadataExtractor.AddWpfBitmapMetadata(String imagePath,
IGalleryObjectMetadataItemCollection metadataItems)
at
GalleryServerPro.Business.Metadata.MediaObjectMetadataExtractor.GetGalleryObjectMetadataItemCollect
ion()
at GalleryServerPro.Business.Image..ctor(Int32 id, IAlbum parentAlbum, String title, String hashKey, String
thumbnailFilename, Int32 thumbnailWidth, Int32 thumbnailHeight, Int32 thumbnailSizeKb, String
optimizedFilename, Int32 optimizedWidth, Int32 optimizedHeight, Int32 optimizedSizeKb, String
originalFilename, Int32 originalWidth, Int32 originalHeight, Int32 originalSizeKb, Int32 sequence, String
createdByUsername, DateTime dateAdded, String lastModifiedByUsername, DateTime dateLastModified,
Boolean isPrivate, Boolean isInflated, FileInfo imageFile)
at GalleryServerPro.Business.Image..ctor(FileInfo imageFile, IAlbum parentAlbum)
at GalleryServerPro.Business.Factory.CreateImageInstance(FileInfo imageFile, IAlbum parentAlbum)
at GalleryServerPro.Business.Factory.CreateLocalMediaObjectInstance(FileInfo mediaObjectFile, IAlbum
parentAlbum)
at GalleryServerPro.Business.Factory.CreateMediaObjectInstance(FileInfo mediaObjectFile, IAlbum
parentAlbum, String externalHtmlSource, MimeTypeCategory mimeTypeCategory)
at GalleryServerPro.Business.SynchronizationManager.CreateNewMediaObject(IAlbum album, FileInfo file)
at GalleryServerPro.Business.SynchronizationManager.SynchronizeMediaObjectFiles(DirectoryInfo
directory, IAlbum album)
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo
parentDirectory, IAlbum parentAlbum)
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo
parentDirectory, IAlbum parentAlbum)
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo
parentDirectory, IAlbum parentAlbum)
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo
parentDirectory, IAlbum parentAlbum)
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo
parentDirectory, IAlbum parentAlbum)
```

parentDirectory, IAlbum parentAlbum)  
at GalleryServerPro.Business.SynchronizationManager.SynchronizeChildDirectories(DirectoryInfo  
parentDirectory, IAlbum parentAlbum)  
at GalleryServerPro.Business.SynchronizationManager.Synchronize(String synchId, IAlbum album, String  
userName)  
at Gsp.Gallery.Synchronize(Int32 albumId, String synchId, Boolean isRecursive, Boolean overwriteThumb,  
Boolean overwriteOpt, Boolean regenerateMetadata)

**Resolution:** The problem is caused by the .NET Framework having trouble extracting metadata from an image file, most likely due to the metadata not conforming to the expected format or syntax. Added a catch block around the call to `BitmapDecoder.Create()` in `GalleryServerPro.Business.WpfMetadataExtractor.GetBitmapMetadata()` to silently handle `OverflowException` exceptions. Versions earlier than 2.5.0 had an exception handler around the reflection-based call to the WPF metadata extractor that caught exceptions like this (since any exception would be wrapped in a `System.Reflection.TargetInvocationException` that was handled).

### Defect: User given incorrect confirmation message after rotating images

**ID:** 500

**Created:** 01-Jul-11

**State:** Closed

**Resolved:** 07-Jul-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** After a user rotates one or more images on the rotate images page, he or she is told "Media objects successfully rearranged... ". Instead, the message should be "Media objects successfully rotated... ".

**Resolution:** Updated the `OnBubbleEvent` function in `rotateimages.ascx.cs` to specify the correct message.

### Defect: Wrong MIME type may be sent to browser for some media objects

**ID:** 503

**Created:** 07-Jul-11

**State:** Closed

**Resolved:** 07-Jul-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** The HTTP handler `getmediaobject.ashx` always sends the MIME type of the original media object to the browser, even when the requested item is the thumbnail or compressed image. In most cases the browser is able to gracefully handle this and does not cause an issue, but for SVG files it is a problem: The thumbnail image, which is a JPEG, is sent to the browser as the MIME type 'image/svg+xml', resulting in a broken image icon.

**Resolution:** Modified `getmediaobject.ashx` so that the MIME type sent to the browser is the MIME type of the requested display type.

## Defect: Space character cannot be used for admin account during installation

**ID:** 504

**Created:** 13-Jul-11

**State:** Closed

**Resolved:** 21-Oct-11

**Area:** GSP-Main

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** If a space character is included in the password for the admin account in the installation wizard, the message "Password must be at least 3 characters" is shown and the user cannot continue.

**Resolution:** Made two changes: (1) Replaced RequiredFieldValidator with the HTML5 required attribute on the textbox. (2) Modified the regular expression to use '.' instead of '\S', which causes it to match on any character instead of just non-whitespace ones.

## Defect: Nothing happens when one clicks the Browse button on a file upload control

**ID:** 506

**Created:** 01-Aug-11

**State:** Closed

**Resolved:** 01-Aug-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** A file picker dialog should appear when the user clicks the Browse button on the file upload control. This control is used on the Add objects page and the Backup/restore page. When the gallery is used in a DotNetNuke 6 website and the browser is IE9 in standards mode (the default), nothing happens when one clicks the browse button. Ultimately, this is due to the javascript placing the hidden file input control in the wrong location. Notice that the behavior was not noticed in the stand-alone or WPI versions, nor does it occur in DotNetNuke 5.

**Resolution:** Had to modify a javascript function in the ComponentArt library to allow file picker dialog to appear when clicking the Browse button on the add objects page. Modified function \_q76 in file A573G988.js (which is only executed when the current browser is IE) to add an additional conditional statement. At column 11664 there is this if statement:  
if(document.compatMode=="CSS1Compat"&&o==document.body)  
Added "&&document.documentMode<9" so that the contents only run for IE browsers less than IE9 or IE9 in compatibility mode:  
if(document.compatMode=="CSS1Compat"&&document.documentMode<9&&o==document.body)  
Although the bug appeared only in DotNetNuke, this fix is being applied to all versions of GSP.

## Defect: A timeout error in FFmpeg interrupts synchronization

**ID:** 507

**Created:** 05-Aug-11

**State:** Closed

**Resolved:** 05-Aug-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** When ffmpeg.exe is installed in the bin directory and the app is running in Full Trust, it is used to create thumbnail images for videos. In some cases, ffmpeg.exe times out. When this happens, GSP detects it and throws a BusinessException, which halts the addition or syncing of media objects. However, in most cases the thumbnail is correctly created. It would be better to record the error in the error log but otherwise continue with the workflow.

**Resolution:** GalleryServerPro.Business.FFmpeg.ExecuteFFmpeg was modified so that the error was recorded but no exception was thrown.

---

## Defect: Cannot save gallery settings when invalid user is entered for error notifications

**ID:** 509

**Created:** 13-Aug-11

**State:** Closed

**Resolved:** 21-Oct-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** If you enter an invalid username in the error notification textbox on the gallery settings page and then attempt to save, the following error message correctly appears: "The user 'xxx@yyy.zzz' does not represent an existing user.' You can then delete the user and save. However, if instead you then uncheck the error notification checkbox and attempt to save, you continue to get the error, even if you cleared the textbox before unchecking the option.

You should be able to save your changes in this situation.

**Resolution:** Added code to BeforeUnbind\_ProcessDisplayBehaviorControls() so that when the notify option is not selected, any value in the users combo box is ignored.

---

## Defect: Browser shows cached image after rotation

**ID:** 514

**Created:** 29-Aug-11

**State:** Closed

**Resolved:** 21-Oct-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** The HTTP handler instructs browsers to cache items for 30 days. This works well except in cases where a user rotates an image. After this action, the browser still shows the cached image rather than the newly rotated ones. In this case, the browser should request a new version of the image.

**Resolution:** Changed the caching instructions that are sent to the browser. Now the browser is sent a Last-Modified date and an ETag. When the browser sends the server an If-Modified-Since request after a rotation, the server will respond with the updated timestamp, thus triggering the browser to show the updated image. In some cases a browser will use its cached image instead of sending an If-Modified-Since request, in which case the old image is still shown. But if the user clicks the refresh button, the updated image is shown. I don't have an easy solution for this scenario, but at least refreshing shows the right image, which is an improvement over the previous situation.

## Defect: UnauthorizedAccessException during upgrade

**ID:** 515

**Created:** 29-Aug-11

**State:** Closed

**Resolved:** 24-Oct-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** When a user attempts to upgrade GSP and the upgrade wizard does not have permission to modify web.config, a UnauthorizedAccessException exception can occur, even when no changes to web.config are required.

Two things should change: (1) If the exception occurs and an update to web.config is required, the error should be trapped and shown. Instead, a YSOD screen was shown in this case (from within DeleteSqliteReferences()). (2) When the first issue is handled, another UnauthorizedAccessException occurred in UpdateWebConfig().

Note: I noticed this issue while upgrading a site for a user. It continued to occur even after I manually deleted the < cachingConfiguration > section. The user was using SQL Server and no SQLite references were in web.config, although SQLite DLLs were in the bin directory.

**Resolution:** 1. Modified DeleteSqliteReferences() so that web.config is saved only when a change is required. 2. Modified WebConfigUpdate.Upgrade() so that web.config is saved only when UpgradeRequired == true or it contains a cachingConfiguration section.

**Defect:** Cannot navigate media objects when transition type is set to 'None'

**ID:** 521

**Created:** 17-Oct-11

**State:** Closed

**Resolved:** 17-Oct-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** When the transition type is set to None on the Media Objects - General page, jQuery is not sent to the browser, causing javascript dependent on it - such as the next/previous functions - to fail.

**Resolution:** This bug was unintentionally fixed in the 2.6 development process. Since 2.6 is designed to use jQuery on all pages, the faulty conditional logic was removed prior to closing this bug.

---

**Defect:** Sync status window may continue updating even after sync is complete

**ID:** 522

**Created:** 21-Oct-11

**State:** Closed

**Resolved:** 21-Oct-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** In certain circumstances (it is unclear what those are), the synchronization status popup window continues checking the server for the current status even after it is complete (status label shows 'Synchronization complete').

**Resolution:** Added a check in the javascript function checkProgressComplete to stop the timer if the status returned from the web service is 'Complete'.

---

## Defect: Admin pages may appear to be associated with a different gallery than expected

**ID:** 523

**Created:** 21-Oct-11

**State:** Closed

**Resolved:** 21-Oct-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** When the URL override setting is enabled, which is the default, the aid/moid parameters in the query string is used to set the current media object, album, and gallery for the current page. This can lead to confusing behavior when, for example, the user is on a page associated with another gallery but accesses it with with an aid or moid parameter belonging to another gallery.

Imagine this scenario: Two web pages - g1.aspx (assigned to gallery 1) and g2.aspx (assigned to gallery 2). Suppose the admin is looking at the Gallery Manager page on g1.aspx. For example, the URL may be localhost/g1.aspx?g=admin\_galleries&aid=12.

Now suppose the admin wants to quickly switch over to g2.aspx, so she updates the URL to point to the new page, while leaving the rest of the URL in place (that is, localhost/g2.aspx?g=admin\_galleries&aid=12). In this situation, the Gallery Manager will incorrectly indicate that g2.aspx is associated with gallery 1. This is because the aid parameter in the URL indicates album ID 12, which is detected by the page as belonging to gallery 1, and so the page temporarily switches to gallery 1 for the duration of the page load. Business logic allows a user to override a gallery setting in this way, but it is misleading to tell the user the current gallery is 2. All the admin pages behave this way.

**Resolution:** The fix was to add code to the page load event of the AdminPage base class to detect whether the inferred gallery for the page (based primarily on the 'aid' query string parameter) is the one assigned to it. If not, the aid parameter is removed from the query string and the page is reloaded, causing the inferred gallery to be the one the page is assigned to.

## Defect: Duplicate metadata items when 'discard original file' option is selected on add objects page (SQL CE only)

**ID:** 524

**Created:** 26-Oct-11

**State:** Closed

**Resolved:** 26-Oct-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** When the option 'Discard original file after creating web-optimized version' is selected on the add objects page, the metadata items are extracted twice. Applies only to SQL CE provider.

**Resolution:** Set IGalleryObjectMetadataItem.HasChanges to false after saving media object.

**Defect:** Copying a media object may result in an extra thumbnail image in the directory

**ID:** 525

**Created:** 26-Oct-11

**State:** Closed

**Resolved:** 26-Oct-11

**Area:** GSP

**Closed:** 01-Nov-11

**Iteration:** 2.6.0

**Issue Detail:** When copying a video, external media object, or generic media object (e.g. swf, doc, xls, pdf, etc), , the thumbnail from the source media object is copied, but a bug causes an additional thumbnail image to be created.

**Resolution:** Fixed by modifying the IsThumbnailImageFileMissing() function in VideoThumbnailCreator.cs so that it correctly returns false for a copied image.

---