Innhold

[Description 2](#_Toc159758059)

[Hardware Requirements 2](#_Toc159758060)

[Image organization on disk 2](#_Toc159758061)

[Usage 3](#_Toc159758062)

[Starting/opening 3](#_Toc159758063)

[Quick start 3](#_Toc159758064)

[Main Menu 3](#_Toc159758065)

[File 3](#_Toc159758066)

[Panels 4](#_Toc159758067)

[Help 5](#_Toc159758068)

[Reset Panel Layout 5](#_Toc159758069)

[Panel Descriptions 6](#_Toc159758070)

[Configuration 6](#_Toc159758071)

[Current Image 7](#_Toc159758072)

[Faces 8](#_Toc159758073)

[Files 10](#_Toc159758074)

[Filter 11](#_Toc159758075)

[Folders 13](#_Toc159758076)

[Log 17](#_Toc159758077)

[Map 17](#_Toc159758078)

[Objects 17](#_Toc159758079)

[Persons 18](#_Toc159758080)

[Playlist 19](#_Toc159758081)

[Properties 19](#_Toc159758082)

[Scenes 20](#_Toc159758083)

[Scripting 21](#_Toc159758084)

[Tags 21](#_Toc159758085)

[Thumbnails 22](#_Toc159758086)

[Background processes 23](#_Toc159758087)

[Scripting Manual 24](#_Toc159758088)

[New Script 24](#_Toc159758089)

[Script main components 24](#_Toc159758090)

[Automatically generated GUI 25](#_Toc159758091)

[Script execution 26](#_Toc159758092)

[Compiling 27](#_Toc159758093)

[Debugging 27](#_Toc159758094)

[User defined filters 27](#_Toc159758095)

[Filter main components 27](#_Toc159758096)

[Backup/restore 28](#_Toc159758097)

# Description

PhotoOrgz is a photo organizing windows application currently under construction. The plan is to release a first version by the summer of 2024.

The main purpose of the application is to ease finding of images. This is done by providing means for automatic (AI based) and manual tagging of images. The meta data is by filtering.

PhotoOrgz is **not** an application for photo editing. But it streamlines the process of identifying an image and communicating the image by drag/drop to a photo editing application.

The application works on locally stored images, there is no need for uploading to any cloud storage.

# Hardware Requirements

For performance reasons it is best to have the images stored on an SSD disk. The face, object and scene detections are done using the **CodeProject.AI** service. This service can take advantage of CUDA enabled GPU’s. For best performance on the AI part, ensure that your graphic card is CUDA enabled, check the **CodeProject.AI** web site for more information.

# Image organization on disk

PhotoOrgz handles the following image file extensions: ".JPG", ".JPE", ".JPEG", ".BMP", ".GIF", ".PNG".

Before starting to use the PhotoOrgz application, it is wise to have a look at how you have organized your image collection on disk. PhotoOrgz reads meta information from your images by using main entry points, i.e. source root folders. Meta information from images in and recursively below a root source folder is scanned into the system. It is recommended **not** to have to many root folders, organize your photo collection to be accessible from a few local disk root folders.

The images are displayed to the user upon a disk folder selection. If the folders contain to many images, it will be slow to select folders. By default, only the first 200 images are displayed upon a folder selection. The performance penalty is due to displaying a thumbnail image for each image.

To summarize, organize your images into a few source folders and ensure that each subfolder does not contain to many (< 200) images.

# Usage

## Starting/opening

PhotoOrgz is started by double clicking the exe file in the installation folder (the folder unpacked from the installation zip archive file). It is recommended to add the application to your start menu or make a shortcut to ease the startup.

## Quick start

After opening the app do the following:

1. Open the **PhotoOrgz** app and verify that the connection to the AI Server is established (Status bar in the lower left).
2. Click the **Add Source** button in the [Folders](#_Folders) panel to select a root source folder.
   1. Thumbnail images are created in the background. It is recommended to wait until this is finished before exploring the application further.
3. Select a folder in the **Folders** panel to display the images contained by this folder and all its subfolders.
4. Right click the selected folder and select [**Detect => Faces**](#_Detect)  
   A [background process](#_Background_processes) is started to detect faces in the images in and below the selected folder. When done, select the **Faces** panel to inspect the result.
5. Go to the [**Persons**](#_Persons) panel to create a person. Drag/drop the person to selected faces in the Face panel to assign a face to the person.
6. Enable the filter in the **Filter** panel. Double click the selected person in the Persons panel. Verify that only images containing the person is displayed.
7. Right click the selected folder and select **Detect => Objects**. When the background process is finished go to the Objects panel and double click an object. Verify that only images containing the object is displayed.
8. Right click the selected folder and select **Detect => Scenes**. When the background process is finished go to the Scene panel and double click a scene. Verify that only images representing the scene is displayed.

## Main Menu



### File

Et bilde som inneholder tekst, Font, skjermbilde, line

Automatisk generert beskrivelse

#### Report Database Consistency

It is recommended to run this process before doing a cleanup of the database to avoid unwanted loss of image metadata.

Reports the number of the following problems:

1. Images not contained by any valid source folder.
2. Images with a valid source but missing on disk.
3. Thumbnail image files not referenced from any meta data.
4. Face image files not referenced from any meta data.
5. Computed face similarities referencing none existing faces-

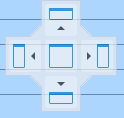
#### Clean Up Database

Fixes the problems listed in the previous chapter. If the report lists a lot of images not existing on the disk anymore, it might be because you have accidently removed or moved images. If you want to keep the metadata for these images, it is better to re-install the image files on disk than running this procedure.

### Panels

Et bilde som inneholder tekst, skjermbilde, display, programvare

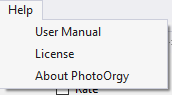
Automatisk generert beskrivelse

The application is build using a panel docking framework. This means that the end user can configure the panel layout. This is done by dragging the panel title to drop handles. The drop handles will appear after starting a drag operation. The panels can be dropped at the Left, Top, Right, Bottom or Center of a panel area. If it is dropped outside a drop handle, the panel will be floating. This applies to all panels but the [Configuration](#_Configuration), [Filter](#_Filter), and [Log](#_Log) panel. The **Configuration** panel is a floating modal panel, whilst the **Filter** panel is locked to the Top and the **Log** panel is locked to the bottom. If closing a panel, it can be re-opened using this menu. It is then re-opened in the same position as it was closed.

The menu provides the following options:

1. [Configuration](#_Configuration)
2. [Current Image](#_Current_Image)
3. [Faces](#_Faces)
4. [Files](#_Files)
5. [Filter](#_Filter)
6. [Folders](#_Folders)
7. [Log](#_Log)
8. [Map](#_Map)
9. [Objects](#_Objects)
10. [Persons](#_Persons)
11. [Playlist](#_Playlist)
12. [Properties](#_Properties)
13. [Scenes](#_Scenes)
14. [Scripting](#_Scripting)
15. [Tags](#_Tags)
16. [Thumbnails](#_Thumbnails)
17. [Reset Panel Layout](#_Reset_Panel_Layout)

### Help

The Help menu has the following options:

1. User Manual, opens a pdf file with the user manual.
2. License, shows the license agreement text.
3. Opens a dialog with version number and home page reference.

## Reset Panel Layout

Reset the window layout the default configuration as shown below.

Et bilde som inneholder tekst, skjermbilde, programvare, Dataikon

Automatisk generert beskrivelse

## Panel Descriptions

### Configuration

Et bilde som inneholder tekst, skjermbilde, display, programvare

Automatisk generert beskrivelse

The panel is a floating and modal (close it to access any other panels). The parameter values in bold face can be changed. The **AI parameters** are threshold values for detecting faces, objects, scenes, or it is used as a threshold when matching/recognizing faces, see [Match Faces](#_Match_Faces) and [Recognize Faces](#_Recognize_Faces). The **max listed images** parameter is used to limit the number of images listed in the [**Thumbnails**](#_Thumbnails) and [**Files**](#_Files) panel. This is done due to performance reasons. When a user selects a folder, all images complying to the current filtering in and below the folder level is listed. Each image is listed with a corresponding thumbnail. If a folder contains to many images, it would be inconvenient to wait for all to thumbnails to be displayed. If you are not able to set a proper filtering and really need to see all images in the folder simultaneously, the value can be raised, but it comes with a performance penalty.

The **Application storage** is the root folder for application files. In this folder you find a sub folder for:

1. DB  
   Contains a database file containing the image meta data. PhotoOrgz uses the [LiteDB](https://www.litedb.org/) database.
2. Faces  
   Contains extracted face image files.
3. Filters  
   User defined filters written in C# language.
4. Restart  
   User interface restart files.
5. Scripts  
   User defined C# script for image processing.

op a face image to the selected person

op a person to the selected faces

1. Thumbnails  
   Extracted thumbnail files.

### Et bilde som inneholder tekst, skjermbilde, Font Automatisk generert beskrivelseCurrent Image

Whenever an image is selected in the files, thumbnails or map panel, the corresponding image is displayed in the Current Image panel. The context menu can be used to save a copy of the image. The mouse scroll button is used to zoom the image, whilst the left mouse button is used for panning.

Et bilde som inneholder snø, vinter, utendørs, skjermbilde

Automatisk generert beskrivelse

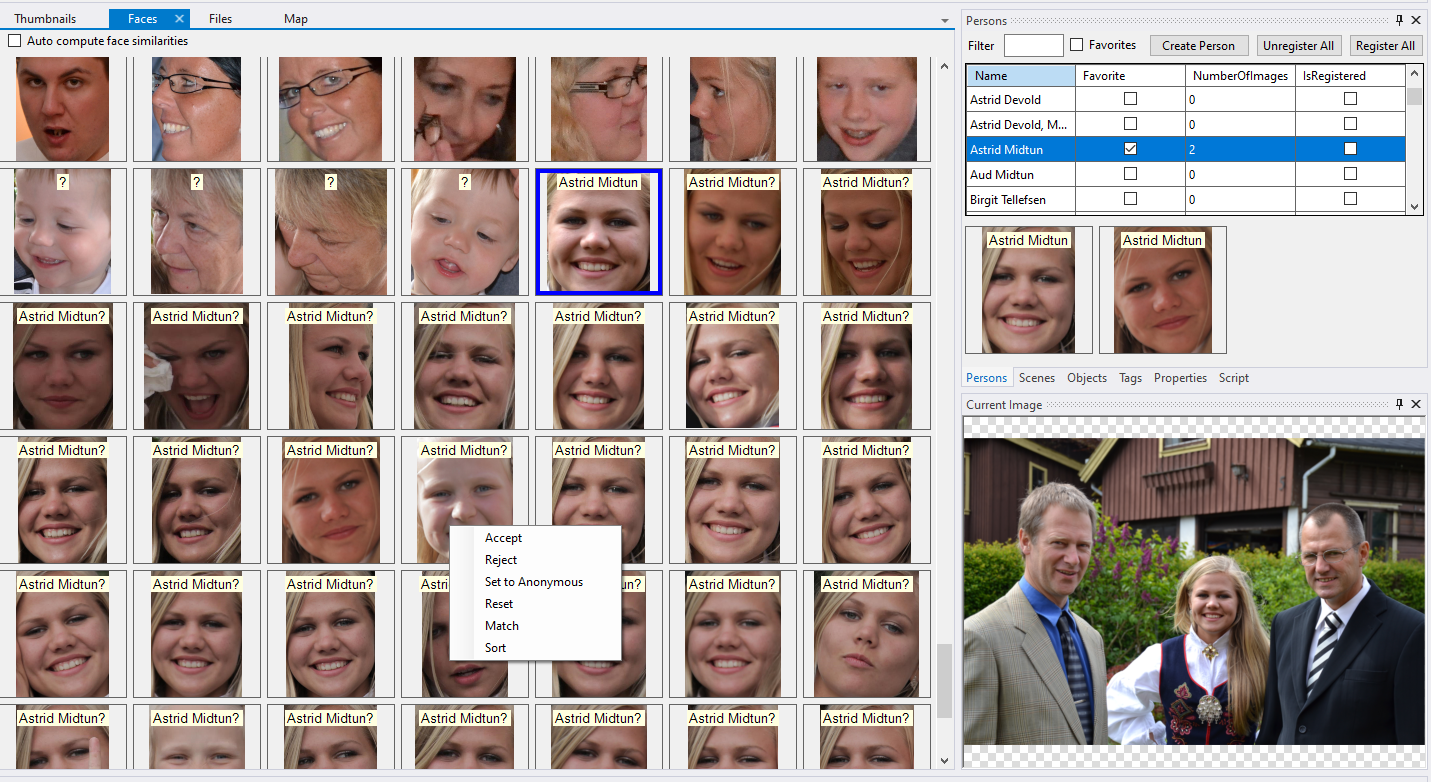
### Faces

The face panel lists all faces in all images complying to the current filtering in and below the current selected folder. The panel is often used together with the Persons panel. By dragging a face from the face panel to the person panel, the face image will be connected to that person. You can than register the face to be used to recognize that person.

By dragging a person to a face, the face will be assigned to that person. If the face used to drop the person was selected, then all selected faces will be assigned to that person. A face is selected by using the left mouse button. Holding the Ctrl button while selecting, adds to the selection, whilst holding the Shift button, selects a range. When selecting a face, the image from where it was extracted is automatically selected in the Current Image panel.

A face can have one of the following states:

1. **Assigned** => named equal to the person
2. **Suggested** => named as the suggested person name, but postfixed with a **?**
3. **Unknown** => named **?**
4. **Anonymous** => named Anonymous, anonymous faces are never used in matching/recognition processes



Drag/drop a face to selected person.

Drag/drop a person to selected faces.

The face context menu have the following options:

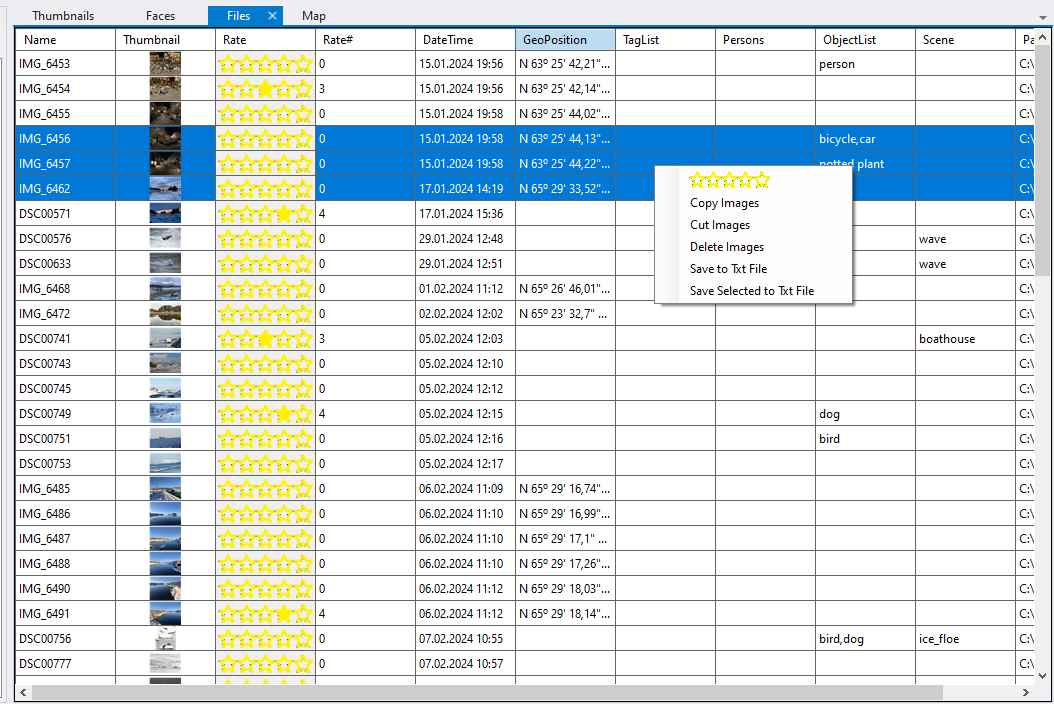
1. **Accept**  
   If a face is suggested to be a person (name postfixed with a ?), accepting means assigning the face to the named person (removing the ? mark)
2. **Reject**  
   Reject a suggestion, i.e. set to unknown (?)
3. **Set to Anonymous**  
   The face is assigned to the name Anonymous. Anonymous is not used in matching/recognizing processes.
4. **Reset**  
   The face is set to be unknown (?)
5. **Match**  
   All displayed images are compared to the selected images. This is done as a [background process](#_Background_processes). When finished, the selected image is displayed as the first image (top left), all subsequent faces are sorted according to similarity with the selected. All images having a similarity above the threshold configured in the [Configuration](#_Configuration) panel is marked as selected.
6. **Sort**  
   All faces are alphabetically sorted using the face name.

**Auto compute face similarities**

The toggle in the top left of the **Face** panel is used to turn auto computation of face similarities on/off. When **on**, a background process is started each time a folder is selected. This process computes face similarities between all assigned and all unknown/suggested faces in the selected folder. The result of these computations can be used in later matching/suggesting processes, see [Suggest Faces](#_Suggest_Faces) and [Match Faces](#_Match_Faces). This is done for performance reasons; the face matching algorithm is a resource demanding process.

### Files

Clicking the column header sorts the images according to the column.



Right clicking the table opens the context menu with the following option

1. Rate Images  
   Click a star to set the rate, left most means 1, rightmost 5. If all stars are unfilled the rating is undefined.
2. Copy Images  
   Copies the images into the paste buffer. A later [Paste](#_Paste) operation on a folder in the **Folders** panel will copy the images to the selected folder. In addition a copy of the meta data information is made.
3. Cut Images  
   Copies the images into the paste buffer. A later [Paste](#_Paste) operation on a folder in the **Foplders** panel will move the images to the selected folder. In addition the meta data information is copied.
4. Delete Images  
   Deletes the images and corresponding meta data from the disk/database.
5. Save to Txt File  
   Saves meta data for all displayed images to a column based TAB separated text file.
6. Save Selected to Txt File  
   Saves the meta data information for the selected images to a column based TAB separated text file.

If the context menu is activated on a selected image, the operation represented by option 1,2,3,4 is applied to all selected images, if activated on a unselected image, the operation is only applied to this image.

### Filter

Et bilde som inneholder tekst, programvare, Dataikon, Multimedieprogramvare

Automatisk generert beskrivelse

The filter mechanism is enabled by using the **Enable** checkbox at the top right. After enabling each sub component can be individually toggled on/off. The following sub components are provided:

1. FileName  
   A regular expression is applied on the full file name of the image.
2. Rate  
   Clicking on a star filters all images with using the selected rate. A rate range can be selected by first clicking a star as representing the start of the range and the holding the Shift key to select the end of the range.
3. Start Date  
   All images taken at or later then the given date.
4. End Date  
   All images taken at or before the given date.
5. Tags  
   A regular expression is used to check the tags of an image. The tag model is hierachical as shown above. If the selected tag text represents a tag group, the image tags are also checked against all children of the given tag, i.e searching for **Camera** will also return images tagged with **Sony**, **Nikon** and **Canon**. Double clicking a tag in the tag panel launches the tag text into the tag filter text.
6. People  
   A regular expression is applied to the name of the faces contained by the image. Double clicking a Person in the person panel, launches the person name into the Person filter text box. If persons are created with familyname, using the familyname will return images containing person with that family name; i.e. the search is done for text containment.
7. Objects  
   Images containg objects matching the given text is returned. The search mechanism is regular expressions. Double clicking an object in the object panel launches the object name into the Object filter text box.
8. Scene  
   Images containg representing a scene matching the given text is returned. The search mechanism is regular expressions. Double clicking a scene in the scene panel launches the scene name into the Scene filter text box.
9. Position  
   The folowing options are available:
   1. Undefined (The image does not have a GPS position)
   2. Defined (The image does have a GPS position)
   3. InsidePolygon (The image position is inside the current polygon digitized in the Map panel)
   4. OutsidePolygon (The image position is outside the current polygon digitized in the Map panel)
10. User Defined  
    User defined filters can be made using C# code. By placing a compileable C# file in the “Application Storage/Filter” folder, a filter with the same name as the code file will be available in the User Defined drop down menu. Filters can be implemented using the Script panel as shown below.

Et bilde som inneholder tekst, skjermbilde, programvare, Dataikon

Automatisk generert beskrivelse

The file name, tags, objects, and scene filters use a text as a search argument. The text matching process in these cases is [regular expression](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_expressions/Cheatsheet)s. The tags and objects filters are applied on the object/tag list, i.e. a comma separated list of text values. In most cases a regular expression is an overkill. If the search argument does not contain any regular expression special characters, the filter will work as a text containment filter. Below are a few useful examples:

1. \?  
   Search for a text containing a question mark, since the **?** is a special reg. ex. character, it has to be escaped using the \ character. Use it to search for unknown and suggested faces.
2. ^\?  
   Search for a text containing only a question mark. Use it to search for single unknown faces.
3. ^\?[,\?]\*$  
   Use it to search for images with only unknown faces (?,?,?)
4. ^(?:(?!UnwantedWord).)\*$  
   Searches for a text that do not contain the unwanted word, replace “UnwantedWord” with your own text. Useful to search all folders but not an unwanted.

### Folders

Et bilde som inneholder tekst, skjermbilde, display, programvare

Automatisk generert beskrivelseIn the **Folders** panel you can inspect all disk folders contained by the registered source root folders. The **Add Source** is used to add a source root folder to the system. When adding a new source folder, a [background process](#_Background_processes) to create thumbnail images is started. A progress meter showing the progress of the background process is shown in the lower left. To avoid performance lag when selecting folders, it is recommended to wait until the background process is finished before inspecting the images in the source folder.

To view only folders containing images satisfying the current filter setting, toggle the **Filter** checkbox at the top left of the panel.

#### Folder context menu

Et bilde som inneholder tekst, skjermbilde, programvare, display

Automatisk generert beskrivelse Right clicking a folder displays the menu to the left.

##### Remove Source

Only available for source root folders. If selected, all meta data for images contained by the source folder or any of its subfolders are removed from the PhotoOrgz database.

##### Move Source

Useful if you have made a backup of a source folder and want to switch to the backup.

##### Save To Txt File

The meta data information from all images contained by the selected folder and all its subfolders are written to a TAB separated text file as shown below.

Et bilde som inneholder tekst, nummer, Font, line

Automatisk generert beskrivelse

##### Generate Thumbnails

Useful if you have stopped the [background process](#_Background_processes) generating thumbnails before completion or if you have accidentally removed the thumbnails.

##### Detect

The following functions works on all images contained by the selected folder or any of its subfolders regardless of the filter setting.

1. Faces  
   The CodeProject.AI service is used to scan images for faces. After detection, the faces can be assigned to persons. This is done by drag/drop a person from the Person panel to selected images in the [Files](#_Files)/[Thumbnails](#_Thumbnails) panels, see [Faces](#_Faces).
2. Objects  
   The CodeProject.AI service is used to scan images for objects. Upon completion, the [Objects](#_Objects) panel contains a unique list of detected objects. Double click an item in the objects list to filter images on this object.
3. Scene  
   The **CodeProject.AI** service is used to scan images for scenes. Upon completion, the [Scene](#_Scenes) panel contains a unique list of detected scenes. Double click an item in the scene list to filter images on this scene.

##### Remove

The following functions works on all images contained by the selected folder or any of its subfolders regardless of the filter setting. It is only useful if you for some reason need to redo a process. The image meta data contains a flag for face, object and scene detection that needs to be reset. This is done for performance reasons, there is no need to redo a detection process if the flag tells that the process is already done. This means that you can add new images to a folder and redo the detection process on the folder without the performance penalty of redoing the detection for images already contained by the folder.

1. Thumbnail
2. Faces
3. Scenes
4. Objects
5. Tags (Removes the tag assignment to an image and not the tag itself.)
6. Face Similarities

##### Suggest Faces

When selecting a folder, the unknown faces on images in this folder and all its subfolders regardless of the filter setting is checked for similarities to known faces. Only pre-computed similarities are used. If the **Auto compute face similarities** is toggled in the [Faces](#_Faces) panel, similarities are automatically computed in the [background](#_Background_processes) when a folder is selected. When a person is assigned to a particular face, the similarity between this face and all unknown/suggested faces in the same folder is computed. An unknown face is suggested to be a person if it has a similarity above a certain threshold, see [Configuration](#_Configuration). The suggestion is flagged by assigning a name to the face equal to the person’s name but postfixed by a question mark.

Et bilde som inneholder Menneskeansikt, smil, tekst, person

Automatisk generert beskrivelse

##### Match Faces

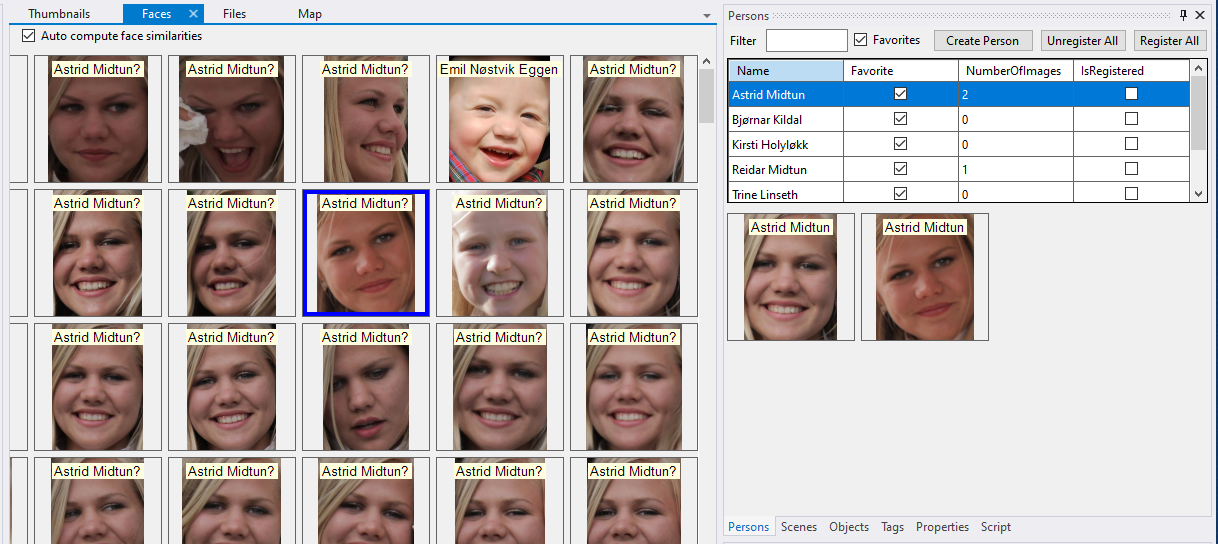
All faces in images in the selected folder and all its subfolder is divided into assigned and unknown faces, see [Faces](#_Faces). The similarity between all unknown faces and all assigned faces is computed. If an unknown face has a similarity to a known face above a certain threshold, the face is suggested to belong to the same person as the known face. The suggestion with the highest similarity is used. The suggestion is flagged by assigning a name to the face equal to the person but postfixed by a question mark. The threshold used in the matching process can be found and adjusted in the [Configuration](#_Configuration) panel opened from the Windows menu seen below.

Et bilde som inneholder tekst, elektronikk, skjermbilde, programvare

Automatisk generert beskrivelse

##### Recognize Faces

Face recognition depends on face registration. Preparing for face registration is done by selecting face images to be registered for that person. This is done by first selecting a person and then drag/dropping a face from the [Faces](#_Faces) panel to the [Persons](#_Persons) panel as seen below.



Assigning faces to a person as described above and hitting the **Register All** button, will register all persons with assigned face images. When this is done, face recognition against the registered persons can be done. The process is executed for all unknown/suggested faces in all images in the selected folder and all its subfolders. If a face is recognized with a higher confidence than given in the **Recognized face confidence** shown in the [Configuration](#_Configuration) panel, the face is assigned to that person.

##### Delete Folder

Deletes the images in the folder and all its subfolder from the PhotoOrgz database and deletes the folder from the disk.

##### New Folder

Creates a new disk folder below the selected folder.

##### Rename Folder

Renames the disk folder and updates the meta data information of the contained images accordingly.

##### Paste

Images copied/cut from the [Files](#_Files)/[Thumbnails](#_Thumbnails) panels are pasted to the selected folder. If copied, all meta information from the source images is copied to the new images.

### Log

Contains messages to the user.

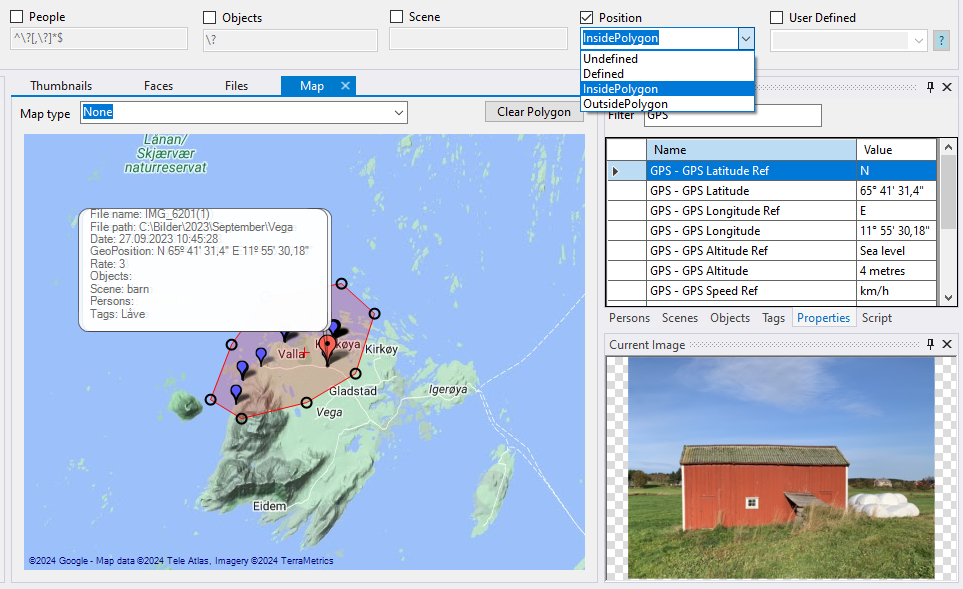
Et bilde som inneholder tekst, skjermbilde, Font, line

Automatisk generert beskrivelse

### Map

The map window is used to show the GPS location of images, but it can also be used to filter. Before creating a new filter polygon, hit the **Clear Polygon** to remove previous. A new polygon is digitized using the left mouse button and simultaneously holding the shift button. The points defining the polygon can be changed later by clicking inside the point and dragging it to a new position. By hovering the mouse over an image location, the meta data are displayed as a tool tip. Clicking an image position with the left mouse button, selects the image as the current.

The map is zoomed using the mouse wheel and panned using the right mouse button.



### Objects

After running an object detection process, see [Detect](#_Detect) for more details, the object panel contains a unique list of objected objects. The filter at the top can be used to narrow the list. Double clicking and object launches the object name to the object [filter](#_Filter) text box. Subsequent double clicks on the same object toggles the filter on/off.

Et bilde som inneholder tekst, skjermbilde, display, nummer

Automatisk generert beskrivelse

Drag an image from the thumbnail panel

to a specific position in the playlist.

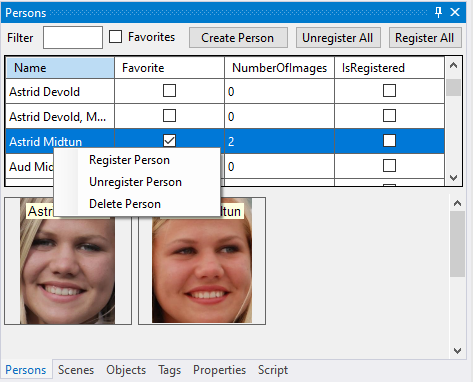
### Persons

Entering text in the filter text box filters the list of existing persons. A new person is created by writing a unique name in the filter box and hitting the **Create Persons** button. A person can be toggled on/off as a favorite. This provides a useful filtering mechanism when you want to work on a narrower list of persons. Dragging faces from the face panel attaches the face image to the person, see [Faces](#_Faces) for more information. After attaching face images to a person, the person can be registered to be used in a face recognition process, see [Recognize Faces](#_Recognize_Faces).

Clicking right mouse button on a person opens the context menu with the following options:

1. Register Person  
   Register the attached face images for person recognition.
2. Unregister Person  
   Removing the person from the list of registered persons.
3. Delete Person  
   Deletes the person from the database and unregisters images for that person. In addition, all faces assigned to the person is set to unknown.

Double clicking a person launches the person’s name into the person [filter](#_Filter) text box. Subsequent double clicks on the same object toggles the filter on/off.



### Playlist

A play list is a named ordered collection of images to be used in for instance a slide show. The playlist panel is used together with the thumbnail panel. To create a new playlist, enter the name and hit the create button. To delete a playlist hit the delete button. An image in the playlist can be displayed or removed from the playlist using the context menu.

Et bilde som inneholder tekst, skjermbilde, Nettsted, programvare

Automatisk generert beskrivelse

Drag an image from the thumbnail panel

to a specific position in the playlist.

Drag/drop thumbnails with in

the playlist panel to re-arrange the order.

### Properties

The image meta data properties contained by the image file is read for the currently selected image. The meta data are organized in folders. The property name consist of a folder part and a name part separated by a “-“. The panel provides a filtering mechanism to narrow the list.

Et bilde som inneholder tekst, skjermbilde, nummer, programvare

Automatisk generert beskrivelse

### Scenes

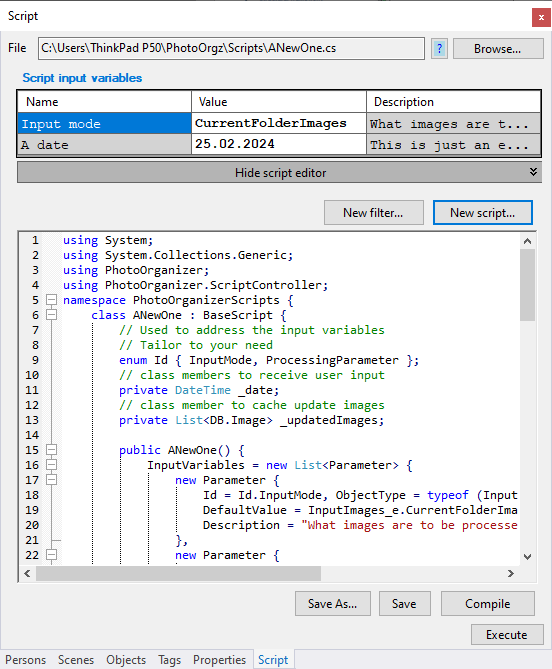
After running a scene [detection](#_Detect) process, the **Scene** panel contains an alphabetically sorted list of unique scenes being detected. Double clicking a scene launches the scene name into the scene [filter](#_Filter) text box. Subsequent double clicks on the same scene toggle on/off the filter.

Et bilde som inneholder tekst, skjermbilde, nummer, display

Automatisk generert beskrivelse

### Scripting

The scripting panel provides means to write scripts for image processing and user defined filters using the C# language. Saving a compilable C# file in the Filter sub folder under the application storage folder makes the filter immediately available in the user defined drop down menu. See [Scripting Manual](#_Scripting_Manual) for further details.



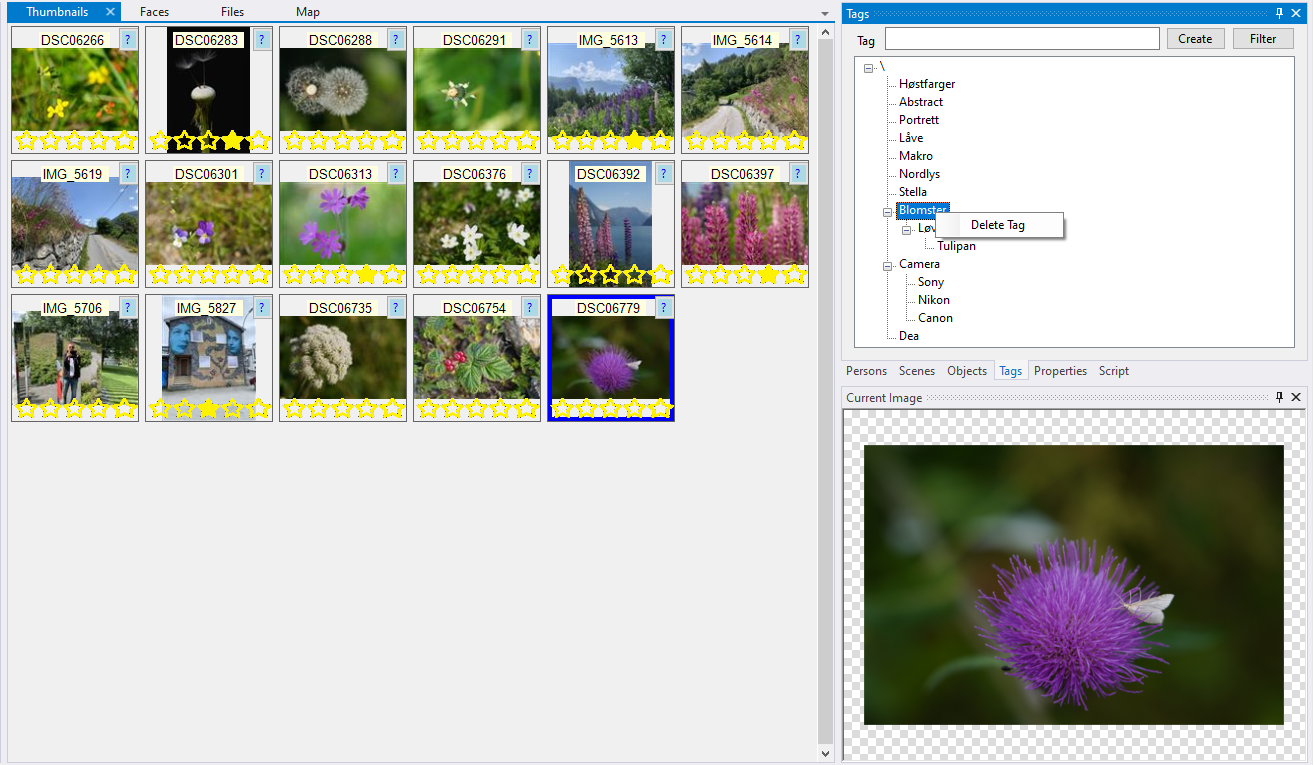
op a tag to selected images.

### Tags

The tag implementation is using a hierarchical tree model. This is done to be able to provide filtering using a hierarchy (By filtering on the tag Camera, all images tagged Camera, Sony and Nikon will satisfy the filter).

Enter a tag name and hit the **Create** button to create a new tag or the **Filter** button to filter the tag tree. When creating a tag, it will be created as a child of the selected node. Right click a node to delete it.

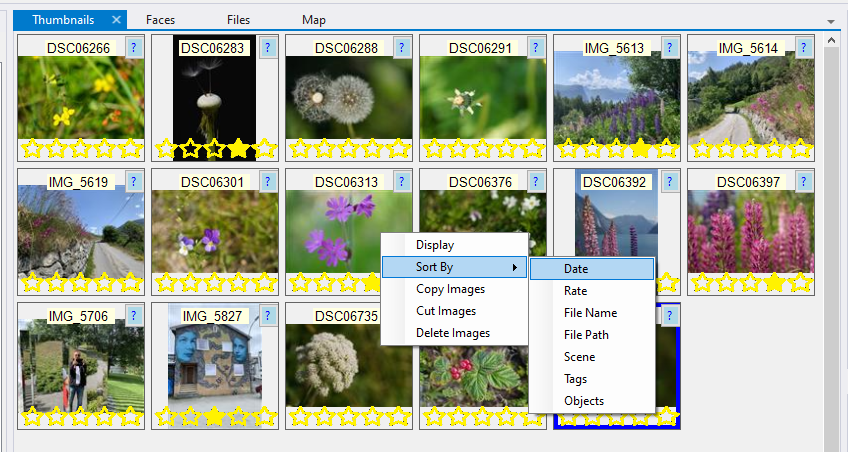
Images are tagged by dragging a tag from the tags panel to the thumbnails or files panel. If dragged to an unselected image, only that image is tagged. If dragged to a selected image, all selected images will be tagged. To remove a tag from an image, hold the Shift key while drag/dropping the tag.



Drag/drop a tag to selected thumbnails

### Thumbnails

The thumbnail panel contains thumbnails for all images in the currently selected folder and all sub folders below that complies to the current filtering.



Right clicking a thumbnail image opens the context menu with the following options:

1. Display  
   Displays selected images in a floating panel, can be used to compare images.
2. Sort By  
   The thumbnails can be sorted based on Date, Rate, File Name, File Path, Scene, Tags or Objects. By subsequently selecting the same filed you can toggle between ascending/descending sort order.
3. Copy  
   Copies the selected image into the paste buffer. Images can then be copied to a selected folder by using the [Paste](#_Paste) operation in the folder panel. Both image file and met data are copied in the operation.
4. Cut  
   Copies the selected image into the paste buffer. Images can then be moved to a selected folder by using the [Paste](#_Paste) operation in the folder panel. The image file is copied to a new folder location and the meta data are updated accordingly.
5. Delete  
   Selected images are deleted from disk and the meta data from the database.

If the context menu is opened on an unselected image, the selected operation is only applied to the selected image. If the context menu is opened on a selected image, the selected operation applied to all selected images.

# Background processes

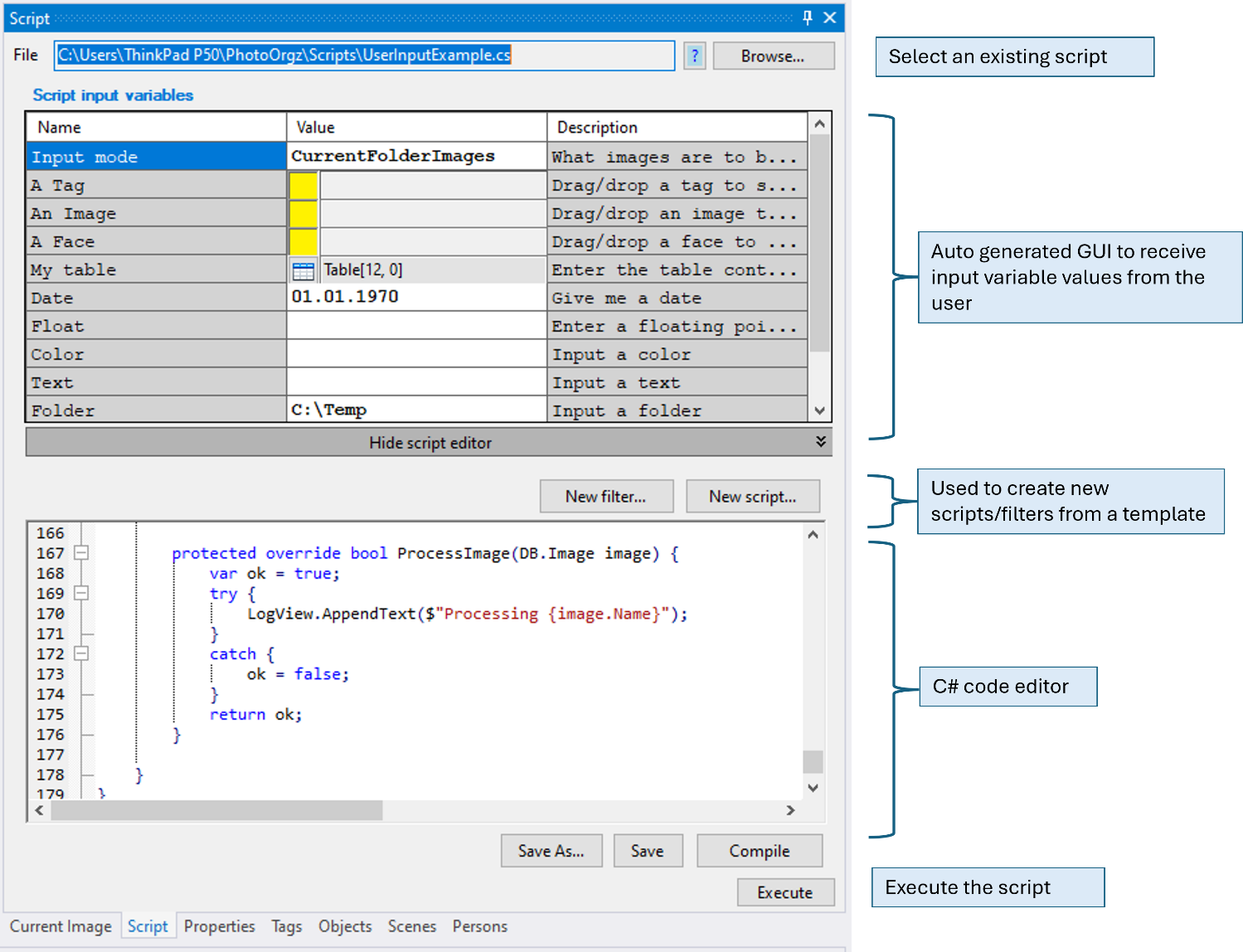
Thumbnail generation, face matching/recognition are done as a background process. This enables you to continue to work while the process is running. The completion percentage of the current background process can be seen in the progress bar  in the lower left. Hitting the cancel button  to the right of the progress bar stops the background process. Only one background process can be executed at a time.

# Scripting Manual

The scripting language is [C#](https://dotnet.microsoft.com/en-us/languages/csharp).

## New Script

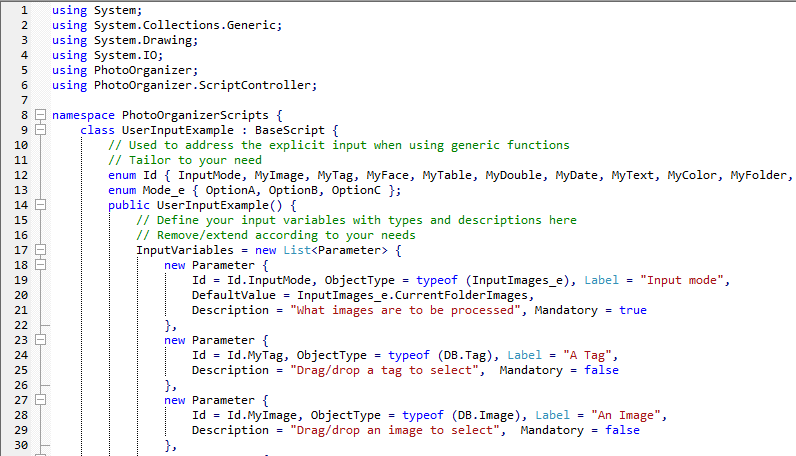
In the Script panel click the  to open the script editor as shown below. The **New Script…** creates a new script from a template. The script is sub typed from the BaseScript class, see the [api documentation](https://midtun.dyndns.org/photo_org/Documentation/ScriptApi/html/index.html) for further information. The most efficient way to understand the scripting possibilities is by studying the script examples found [here](https://midtun.dyndns.org/photo_org/Documentation/Scripts/index.html). The scripting panel provides a very basic C# code editor. A better way to write scripts is to use a professional code editor like [VisualStudio](https://visualstudio.microsoft.com/). To be able to compile the scripts outside the PhotoOrgz application, you need to add references to the dll’s in the PhotoOrgz system folder. If script files are edited outside PhotoOrgz, they are automatically re-loaded into the system when the files are saved.



### Script main components

A script consists of a few key elements.

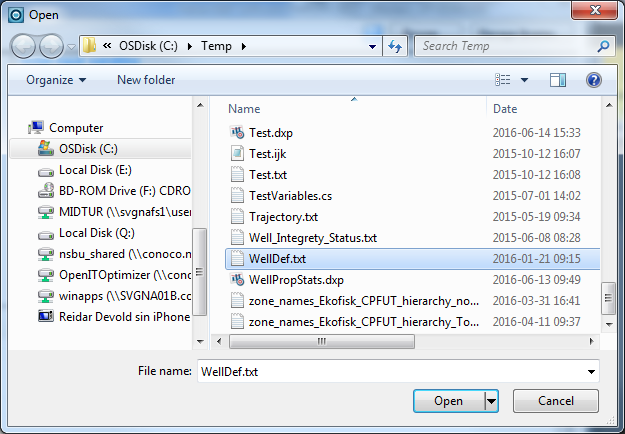
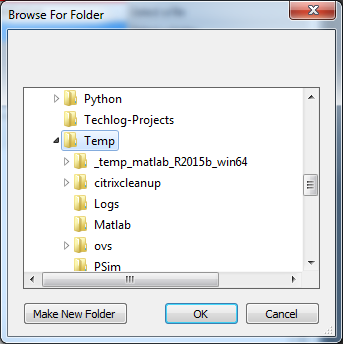
1. Enum Id  
   Used to address an input variable. The value provided by the user can be read by the script using a call to the GetObject<T>(Enum e) function.
2. InputVariables  
   All input variables needed by the script is defined with a name, type, default value, description and a flag stating if it is mandatory. Based on the type of the variable, a [GUI](#_Automatically_generated_GUI) component easing the selection of a value is automatically provided.
3. PrepareExecution  
   Override the PrepareExecution to read info provided by the user to be ready for image processing.
4. GetInputImages  
   Override the function to define the collection of input images, can be implemented using the BaseScript.GetInputImages functions, see this [example](https://midtun.dyndns.org/photo_org/Documentation/Scripts/Doc/ImagesToFolder.cs.html).
5. ProcessImage  
   Override the function to implement how an input image is to be processed.
6. EndExecution  
   Override the function if any wrap up of the process is needed.



### Automatically generated GUI

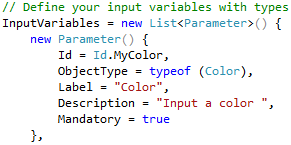
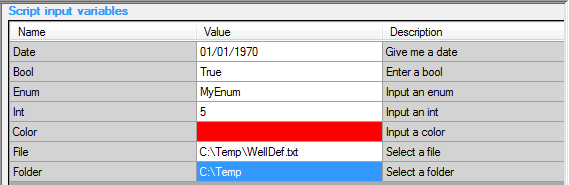
Depending on the type of the input variable the GUI selectors below are automatically created to receive input from the user.

1. FileInfo => File selector
2. DirectoryInfo => Folder selector
3. DateTime => Date picker
4. Bool => toggle
5. Enumeration => drop down menu
6. Color => color picker
7. DB.Objects => object container capable of receiving drag/drop of images, tags, persons etc.



File selector

Folder selector

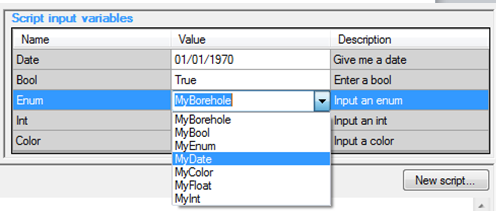
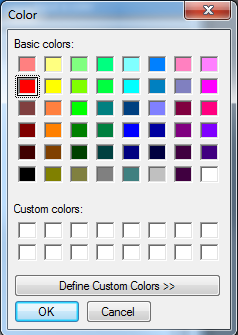
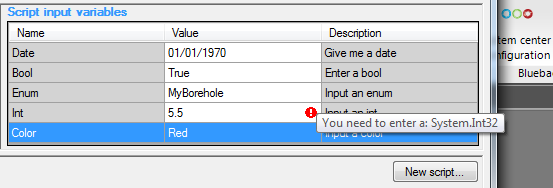
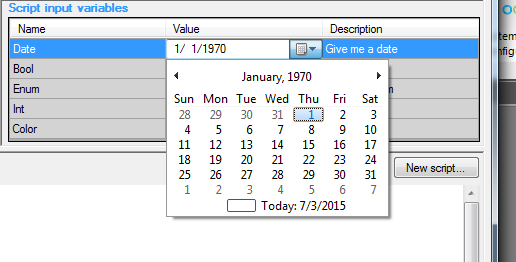


Date

Boolean

Enumeration

Color



### Script execution

Saving a script automatically compiles it to make it ready for execution. If the compilation is successful, the GUI components for selecting the input variables will be available. Hitting the Execute button will start the script execution given that the mandatory input variables are defined. The default implementation is that the script will be executed as a [background process](#_Background_processes).

### Compiling

A script is automatically compiled when the script is saved. If the compilation is unsuccessful, the compilation errors are written to the Log panel. If you need to add references to libraries not already in the default list of reference dlls, you need to add a reference directive to the code as seen below.

// css\_ref Library1.dll, Libray2.dl

The css\_ref directive is inside a // comment. The referenced libraries must be present in the PhotoOrgz system folder. If you need to use a library not present in the PhotoOrgz installation folder, you need to copy it and all its dependencies to the folder before referencing it from your script file.

Code shared between scripts can be separated into its own C# files and then include by the script file by using an include directive as seen below.

// css\_inc File1.cs, File2.cs

### Debugging

To debug your script you need to attach a debugger like [VisualStudio](https://visualstudio.microsoft.com/) to the PhotoOrgz process. By adding call to the BaseScript.DebugBreak() the debugger will halt the process at the code location of the call.

## User defined filters

Create a new user defined filter from a template by hitting the **New Filter…** button. The filter can be subtyped from the BaseScript class to be able to use the functionality implemented by this class, but it must implement the IFilter interface. When the script is saved to the Filters folder under the PhotoOrgz application folder it will be automatically added to the list of user defined filters given that it compiles Ok.

Et bilde som inneholder tekst, skjermbilde, line, Font

Automatisk generert beskrivelse

### Filter main components

The IFilter interface defines the following:

1. Description  
   Text property shown to the user describing the filter
2. Initialize()  
   Implement if any action is needed before applying the filter, see this [example](https://midtun.dyndns.org/photo_org/Documentation/Filters/Doc/InCurrentPlayList.cs.html).
3. IsValid(DB.Image image)  
   The meat of the filter validating if an image should pass the filter.

# Backup/restore

PhotoOrgz does not have any backup/restore procedures implemented. But it is recommended that you backup your metadata and image files using other means. The [FreeFileSync](https://freefilesync.org/) application can be recommended.

All meta data information is kept under the application storage folder, check the [Configuration](#_Configuration) panel for the location on the disk. In addition, you should also backup your image files, i.e. all source root folders. In case you need to restore the meta data information, either restore the content of the application storage folder or rename your backed up folder to the name of the application storage folder. In case you need to restore any of the image source root folders, you can either use the folder context menu [Move Source](#_Move_Source) to move the source folder to the backed-up location, or you can restore the content of the folder.