## 

## What is Icarus?

Icarus is a collection of scripts and tools which form the CG pipeline at Gramercy Park Studios. It was conceived primarily as a tool for operators and artists, to ease the effort involved with working on large and complex VFX projects.

## What Icarus Does

* Job and shot management
* Asset management
* Task automation
* DCC app integration
* Workflow enhancements

## How Icarus Works

At its basic level, Icarus is simply a run-time environment where apps and tools can access a shared set of metadata.

Icarus is built around the concepts of **Jobs** and **Shots**. A job is another name for a project or a show, which can consist of many shots. Icarus can manage existing project folders or generate new ones. A job can have metadata such as its job number, title, client, brand, and which versions of applications to use. Shot metadata includes frame ranges, resolution formats, camera data, etc. These values are then used as default settings in supported apps launched via Icarus. File management and standardised naming conventions are also handled. Automating these often tedious and human-error-prone processes helps artists to focus on being artists.

Icarus integrates with various DCC applications and tools. The best supported are Maya and Nuke, as this is the primary workflow used at GPS. Also supported, but to a lesser degree, are RealFlow and Deadline. But the modular design means it is possible to add or improve integration with more tools in the future.

Icarus is designed to be a help to artists, although it’s an important design philosophy that it does not enforce any particular way of working. It also does not introduce dependencies that make it impossible to work on projects outside of the pipeline, should that be a necessity.

Icarus is written almost entirely in Python, and is distributed under the MIT license which allows complete freedom to distribute or modify at will.

## Icarus vs. Shotgun / ftrack

While there are certain similarities and overlap, Icarus is not a full production tracking/scheduling suite. Icarus is designed as an open, lightweight and portable pipeline, simple to setup with minimal dependencies.

## Who Developed Icarus?

Icarus was started shortly after GPS was founded, by Nuno Pereira, former CG Supervisor. In 2015 that role was taken over by Mike Bonnington, who has been the primary developer since then. Additional development from Ben Parry.

## Requirements

* Workstation running Windows / Mac OS / Linux
* Python (2.7 minimum, 3.x 64-bit recommended)
* Qt bindings for Python (Pyside or PyQt)

The following are optional, but there’s not a lot of point using Icarus without them:

* djv\_view frame viewer - <http://djv.sourceforge.net/djv_view.html>
* Maya (2014 or later)
* Nuke
* Shared network filesystem

## Installation

1. Icarus can be installed locally or on a network share. As it’s standard practice for VFX projects to be stored on a shared SAN, NAS or similar, we recommend installation to the same share where projects are hosted. Either unzip the Icarus distribution to the desired install location, or clone the GitHub repository. If installing to a shared location, this step only needs to be done once.

The following steps (2-5) need to be carried out on all client machines:

1. Download and install the latest version of Python from <https://www.python.org/> (we recommend Python 3.x, 64-bit)
2. Follow the installer, ensuring that you install for all users.
3. Install Qt bindings. For Python 2.7, we recommend PySide. Open a shell and type:

pip install PySide

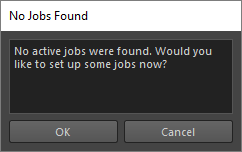
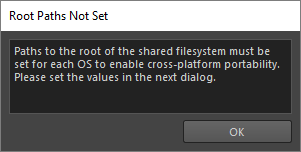
For Python 3.x, we recommend PyQt5. Open a shell and type:

pip3 install PyQt5

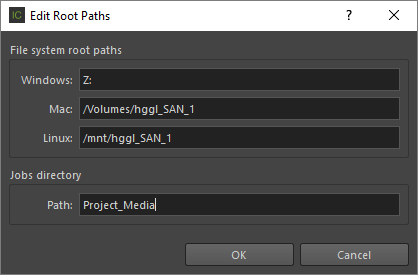
1. To run Icarus, navigate to the Icarus installation folder, then run the file icarus\_\_main\_\_.py located in the subfolder core/run. If desired, a shortcut to this file can be created.

## Getting Started

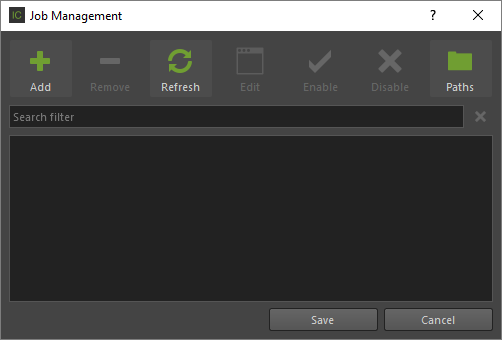
The first thing we need to do is set up some jobs.

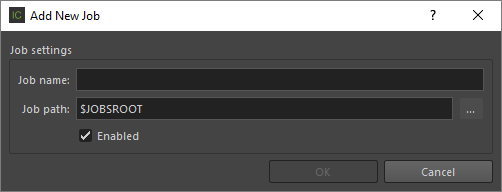
Before we can add a job, we need to tell Icarus the path to our shared filesystem, and where our jobs live. With support for path mapping it’s possible for Icarus projects to work seamlessly across client workstations running Windows, Mac OS or Linux. We can set this up in the next dialog:



GPS has an established folder structure for projects, and Icarus has been designed around this. However, it can be modified to support your pre-existing setup.



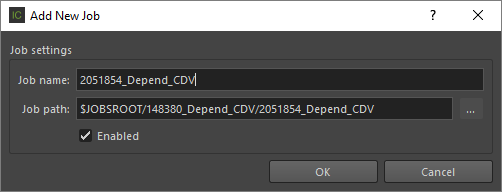
This is the job management dialog. To add a new job, click **Add**.

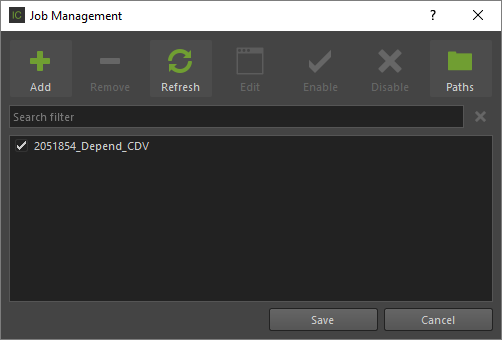


Here we can click the **Browse (…)** button and navigate to the root folder of our job. If the job folder already exists, this will be the folder that contains the folder named Vfx, in the GPS folder structure.

The job name field will be auto-filled based on the name of the folder, but it may also be renamed. This will be the name that Icarus uses to refer to this job from now on.

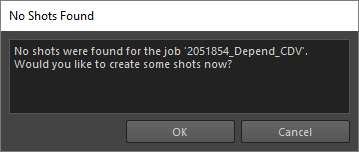
With this information filled in we can click OK to add the job to Icarus’ jobs database.



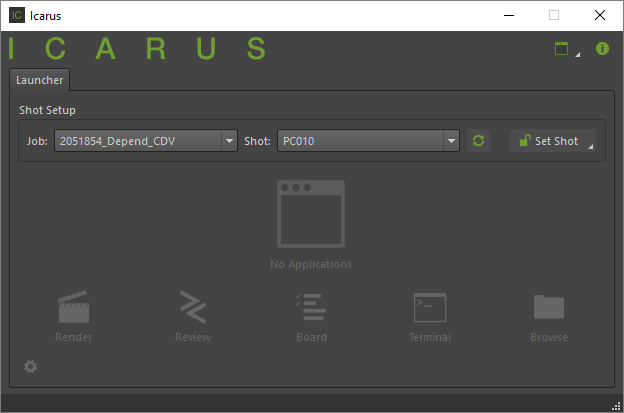


Now our job has been added, we can click **Save** to save the jobs database and return to the main Icarus UI.

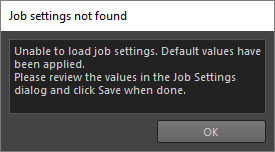
If this is a new job, we will need to create some shots:



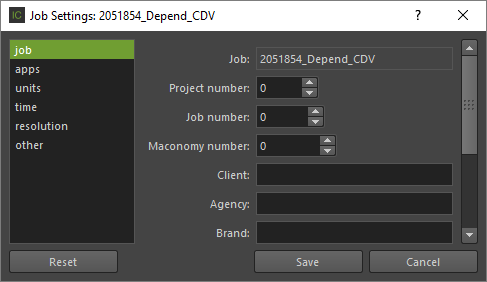
The Shot Creator dialog will appear where we can create some shots. The convention at GPS is that shots are named PC### (Proof-of-Concept) or SH### (Shot) and the editor allows you to create single shot or a sequence of shots. The folder structure for each shot will be placed under the Vfx folder within the job.



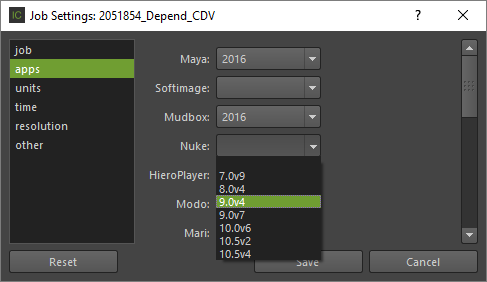
Once our job has at least one shot, we can select it in the main Icarus UI, and click the **Set Shot** button to enter the shot environment. If this is a new job, we will be asked to create some settings for the job:



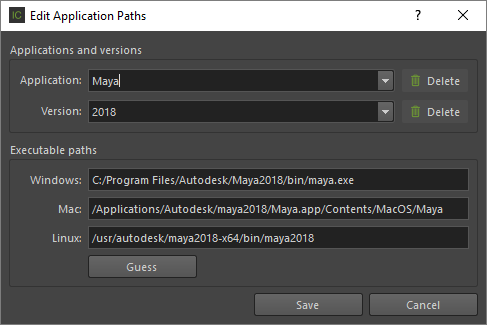
The Settings dialog will appear. Here you can fill in fields in the various categories to store information about the job.



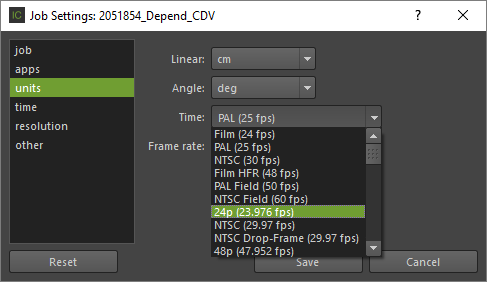
On the **job** panel we can enter miscellaneous metadata for the job.



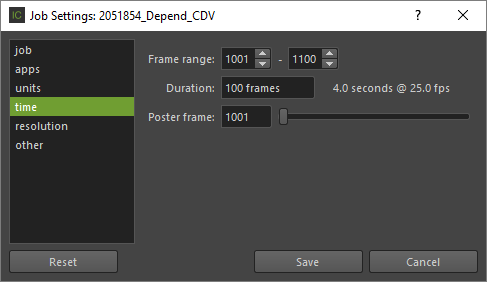
On the **applications** panel, we choose which versions of applications we want to use for this project. If the application or version is not listed, we can click the **Edit Versions…** button to open the Edit Application Paths dialog:



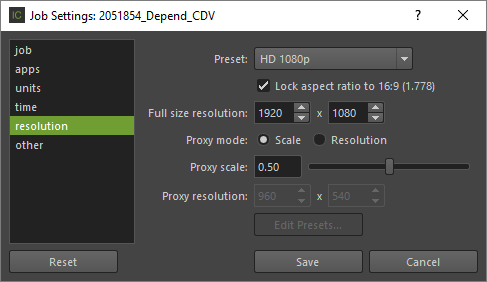
Here we can add, edit or delete applications, versions of those applications, and the executable paths for those applications for each OS that is supported. Click **Save** when done and the application list in the previous settings dialog will be updated.



On the **units** panel we can choose units for linear, angular and time values, including the frame rate. (Note that these values can be overridden on a shot basis)



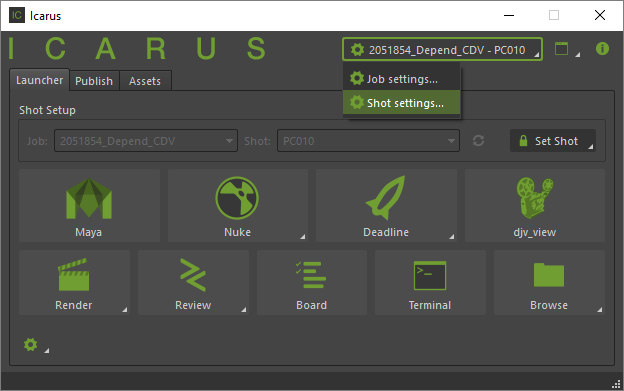
On the **time** panel we can specify the duration of the job. (Note that these values can be overridden on a shot basis)

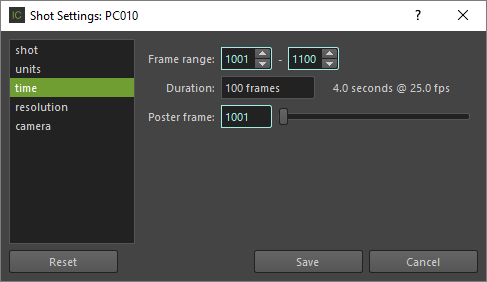


On the **resolution** panel, we can specify resolution settings. (Note that these values can be overridden on a shot basis)

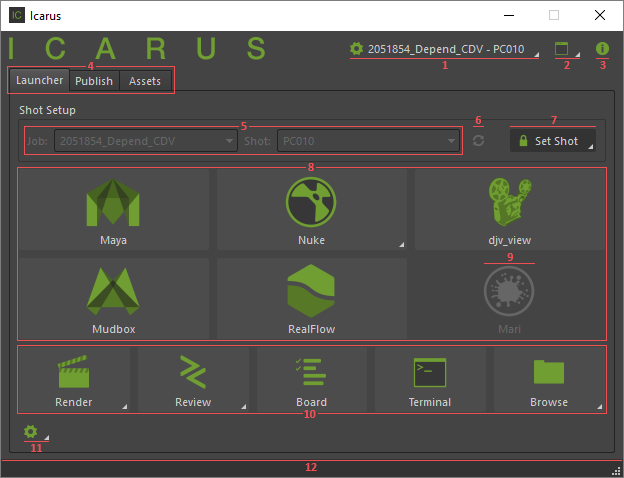
When we’re happy with the settings, click **Save** to return to the main Icarus UI. Now we can click the **Set Shot** button to enter the shot environment.

If desired, we can now edit the settings for the shot. If set, these will override the equivalent setting for the job. In particular we may want to set frame ranges and camera settings on a per-shot basis. The Shot Settings dialog is accessed by clicking on the drop-down menu under the job/shot name, and selecting **Shot Settings…**





Here we have much the same options as for the job settings. The turquoise highlight indicates that the value is not set for this shot, and is instead inheriting the value from the job settings. By editing the value, an override for the shot is automatically created and the highlight disappears. Clicking the **Reset** button removes the overrides and resets all the values on the current panel to the job settings value. Click **Save** to save the shot settings and return to the main Icarus UI.



This is the main Icarus UI, when the shot environment is set.

1. Name of current job/shot, popup menu to edit job or shot settings
2. Tools menu
3. About Icarus
4. Launcher, Publish, Assets/Gather Tabs
5. Shot selection – the user can select a job and shot environment to work in
6. Refresh jobs/shots
7. Set/Unset shot button, recent shots popup menu
8. Application launch icons – these are populated automatically depending on the applications and versions set in the job settings, and when clicked will launch the application within the shot environment
9. Disabled application launch icon – app not installed
10. Utility launcher icons
11. Launcher options menu
12. Status bar

Now we can launch an application by clicking on one of the icons in the launch area. The application project folders are created on demand.

## Launch Icarus within the DCC application

This currently is only supported within Maya and Nuke. Click the Icarus icon on the toolbar to open the Icarus UI. Here we do not have access to the project management sections.

## Areas of demo

* Icarus setup
* Existing project structure & environment
* Job setup
* Job/shot settings
* Folder creation
* App integration
* Asset publishing
* Rendering