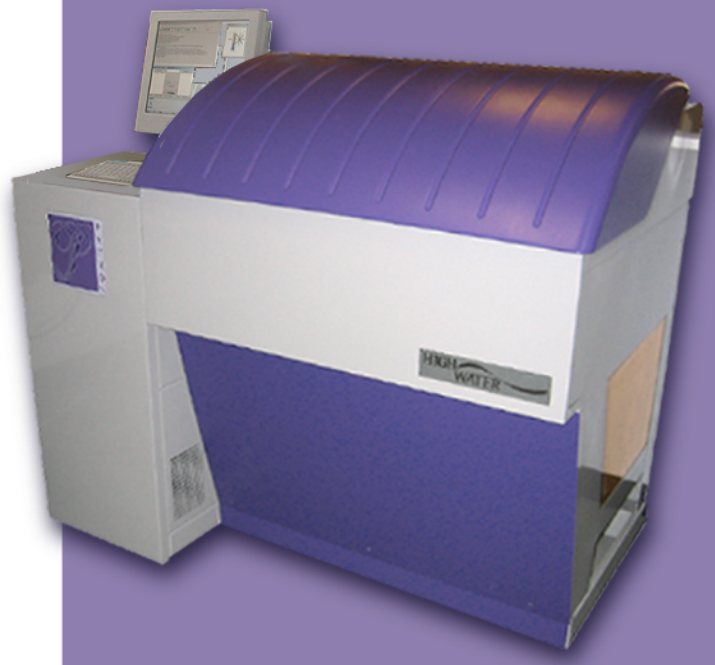
Python User Guide



Preface

Version 4.0	October 2006	(for software version 5.0.0 and including Python Console v1.0.1)
Version 3.0	July 2005	(for software version 4.0.0 SP1/2)
Version 2.1	December 2004	(for software version 4.0.0 SP1)
Version 2.0	September 2004	(for software version 4.0.0)
Version 1.7	March 2004	(for software version 3.0.0)

HighWater's contact information

HighWater Designs Limited
1-6 St. George's Business Park
Alstone Lane
Cheltenham
Gloucestershire
UK
GL51 8HF

Telephone: +44 (0) 1242 542100
Fax: +44 (0) 1242 251600
Email: sales@highwater.co.uk
Web: www.highwater.co.uk

HighWater's support department

Telephone: +44 (0) 1242 542102
Email: support@highwater.co.uk
Web: www.4hws.co.uk

Contents

1. Using this guide

This chapter includes the following sections:

- Navigating around this guide in Acrobat Reader.
- Warnings and notes.
- Printing this guide.

Note: The instructions in this chapter refer to Acrobat Reader v6.0.

1.1 Navigating around this guide in Acrobat Reader

If you are viewing this guide using Acrobat Reader you can move around the guide using Acrobat's navigation buttons:


Go to the first page of the guide


Go back one page


Go forward one page


Go to the last page of the guide

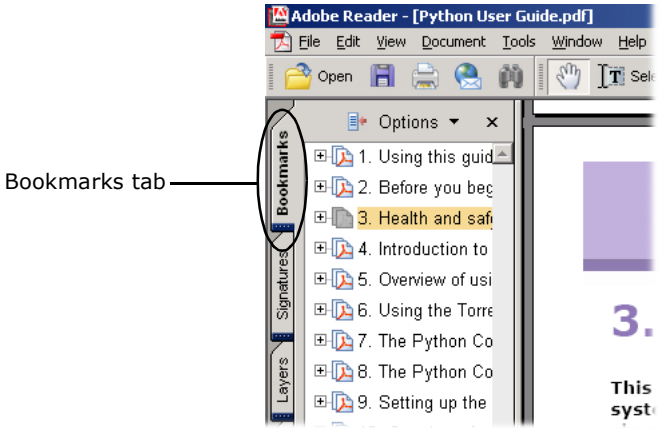

Go back to previous view


Go forward to next view


Fit whole page in window

1.1.1 Bookmarks

You can use Acrobat's Bookmarks window to navigate to the required chapter/section of this manual. If the bookmarks are not visible, click on the **Bookmarks** tab (see below) to display them:



Click on a bookmark link to move to the relevant chapter. Click on a ⊞ link to view the sub-sections within a chapter.

1.1.2 Using hot links

If you are viewing this guide in Acrobat Reader you can use 'hot links' to move quickly to the section you want (when you place the mouse cursor over a hot link it turns into a hand icon). There are three types of hot link:

- On the Contents page: click on the entry to take you to the relevant section.
- Cross references: you will find these throughout the manual, for example, "Refer to section 5.3 for instructions". Click on the section reference to take you to the relevant page.
- On the Index page: click on the page number for the index entry to take you to the relevant page.

1.2 Warnings and notes

1.2.1 Warnings

The warnings in this manual are mainly intended to protect you from injury and the Python platesetter and your plates from damage. Please read all warnings carefully and follow any instructions. Warnings are shown like this:

WARNING: When working with the Python system you must protect yourself from injury and protect the Python platesetter from damage.

1.2.2 Notes

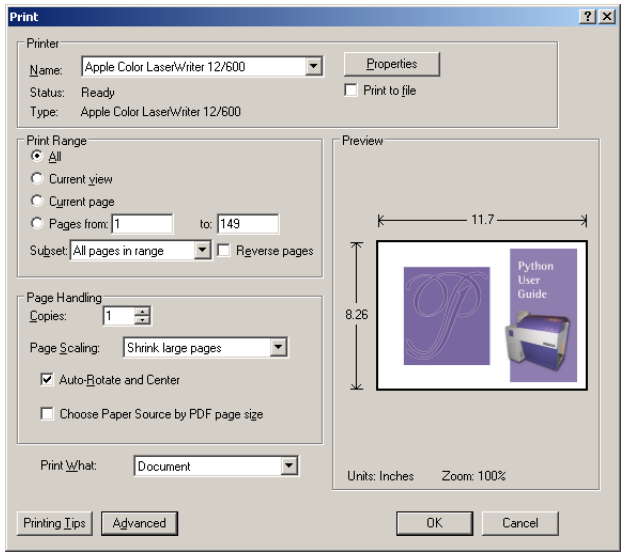
The notes in this manual give additional information on using the Python system. They are shown like this:

> **Note:** For most of the above operations, you can select more than one file at a time using the **Shift** and/or **Ctrl** keys on the keyboard.

1.3 Printing this guide

To print this guide:

1. In Acrobat Reader select **Print...** from the **File** menu to display a dialog similar to the following:



2. Select the required printer from the **Name** pull-down menu.
3. Click on the **Properties** button and select the **Landscape** orientation on the **Layout** tab.

   Note: Select the **Portrait** orientation if you are printing the manual 2-up.
4. Click on **OK** to return to the 'Print' dialog.
5. Set the other options, as required, then click on **OK** to print this guide (or the specified pages).

2. Before you begin

This chapter includes the following sections:

2.1 Version information

The information in this manual refers to the following software versions:

- Python Console v1.0.1.
- Torrent RIP v7.1.
- Queue Configuration v4.4.
- Python Layout Tool v5.0.0.
- Python Test Tool v5.0.0.

Note: Throughout this manual, vx.x or vx.x.x is used to indicate the software version number.

2.2 Basic requirements for using the Python system

Before you start working through this manual, please make sure that:

- You have carefully read the safety notices and information in chapter 3. This covers how to work safely with the Python system and how to stop the Python platesetter in an emergency.
- You are familiar with Windows 2000 and the platform(s) you will be printing to Torrent from (for example, Macintosh OSX).
- You know how to print a job from a page make-up application, such as QuarkXPress.
- You have a basic knowledge of how the Torrent RIP works and how to use it (refer to chapter 6 for details on using the Torrent RIP).
- This guide assumes you will be using AppleTalk to print jobs from your page make-up workstation to the Torrent RIP. Refer to the **Torrent User's Guide** for details about other methods of printing jobs to Torrent.

2.3 Other manuals supplied with the Python system

The following manuals may also be supplied with the Python system:

Torrent User's Guide (optional)

The **Torrent User's Guide** gives full details on setting up and using the RIP. The **Torrent User's Guide** is available:

- As a hard copy supplied with the Python system.
- In PDF format on the machine that Torrent is running on. To view the guide, select **Start > Programs > HighWater Designs > Torrent vx.x > Torrent User Guide vx.x**.

InkMonitor User's Manual (optional)

InkMonitor is an optional software application that generates ink duct settings and/or ink duct control files suitable for automatic ink duct control on many different presses. This manual gives basic instructions for setting up the Python system to use InkMonitor. However, you should also refer to the separate **InkMonitor User's Manual** for full details on using the software. You can find this manual in PDF format on Pre-Press Solutions CD 1.

3. Health and safety

This chapter contains important information about the Python system with regard to your health and safety. Please read this chapter carefully and also refer to the Python Safety Information manual for additional information.

This chapter contains the following sections:

3.1 Safety notices

WARNING: This equipment must be earthed.

Python requires a single power outlet, accessible AT ALL TIMES and located as close as possible to the Python system.

This equipment complies with BS EN 60825-1:1994 +A1 and A2, and with 21 CFR 1040.10 and 1040.11.

Python is a CLASS 1 EMBEDDED LASER product. Once the laser carriage cover is removed, Python becomes a CLASS 3B LASER DEVICE and appropriate safety precautions must be taken.

CAUTION: The use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous laser exposure.

WARNING: When the laser cover is removed, the laser beam used in the Python platesetter is harmful to the eyes. Servicing should be carried out by HighWater qualified personnel only.

3.1.1 Laser properties

Wavelength: 405 nm
Emission duration: continuous
Laser output: maximum 60 mW (30 mW or 5 mW on early models)

3.1.2 Labels

The labels shown in Appendix D are present on all Python machines. Please ensure that you follow all the necessary safety precautions.

3.2 Working safely

WARNING: You must work safely in order to protect yourself from injury and the Python platesetter from damage. Please follow the instructions in this section carefully.

Under normal operating conditions it is not possible for you to come into contact with the laser beam when Python's lid is open. However, you the user must ensure that:

- All access doors to the Python room display the BSI-approved warning symbol:

- When engineers are working on the laser, no-one enters the room unless they are wearing safety goggles that meet approved standards.
- You have received training on safety procedures, as well as instructions on how to operate Python.
- Python is never left in a condition where the operator can be exposed to the laser beam.

3.2.1 Protecting yourself from injury

- Make sure that nothing, **especially your clothing**, gets trapped in Python's lid.
- The edges of a plate are sharp so take great care when handling plates and, in particular, keep the edges of the plate away from your face.

3.2.2 Protecting the Python drum and carriage from damage

- Do not put any object, other than a plate, onto the tilt-table. When the imaging process starts, anything left on the tilt-table will fall down into Python's drum. This could damage Python's drum/carriage and the plate.

3.2.3 Protecting the plate from being fogged by light

- In the brightroom, use the appropriate safety light (as recommended by the plate manufacturer) to prevent fogging of your plates.
- While the brightroom is in use you need to prevent its door being opened. To do this, you could:
  - Use an external light on the brightroom to show when it is in use.
  - Put a message on the door to indicate that the brightroom is in use.

  WARNING: For safety reasons, we recommend that you DO NOT lock the brightroom door.

- If your monitor does not have a yellow filter you can use the Blackscreen utility to darken the monitor when there is a danger that your plates could be exposed to its light. Refer to section 5.4 for details about using Blackscreen.
- After output, protect the plate from unsafe light sources until it has been processed.

3.2.4 Protecting the plate from contamination

- When you load or unload a plate from Python, avoid touching the part of it that will be exposed. You can wear special lint-free gloves to avoid getting fingerprints on the plate.

- Regularly inspect the tilt-table for dust and other particles and remove these, as described in section 16.1. Any particles left on the tilt-table may fall down into Python's drum, and these could spoil your plates.

3.2.5 Protecting your system from data loss or corruption

- When you have finished using Python you must always shut down the system properly to avoid data loss — section 5.3 shows you how to shut down the Python system.

3.3 Stopping the Python platesetter in an emergency

If you need to stop the Python platesetter **immediately** because of potential injury or damage to a person or the Python platesetter, then do any of the following:

- Lift the Python platesetter's lid. This will immediately stop the tilt-table, crossbar and carriage.

  Note: If Python was in its imaging cycle when the lid was opened, refer to section 9.4 for details of how the Python system will respond and how to restore the system to normal operation.

- Turn the Python platesetter off using the power switch on the back of the machine.
- Disconnect the system from the power source by removing the mains plug from the wall socket.

WARNING: Switching off Python using the power switch or removing the mains plug without shutting the system down properly first, is NOT recommended and should only be used in an emergency situation.

---

4. Introduction to the Python system

This chapter gives an overview of the whole Python system. It includes the following sections:

4.1 The Python system

HighWater's Python platemaker is designed to meet the needs of a wide range of the pre-press and printing industry. It produces high quality, aluminium plates, which have the cleanness, accuracy and repeatability only achievable with digital imaging. The Python system has a modular design making it easy to install, set up and maintain.

The Python system's two main components are:

- **The Python platesetter**, which produces the imaged plates.
- **The Python workstation**, which runs all the software required to process your jobs and output them to the Python platesetter.

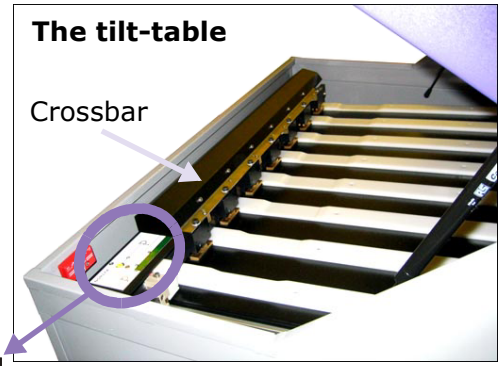
The next four sections describe the Python system in more detail.

4.2 The Python platesetter

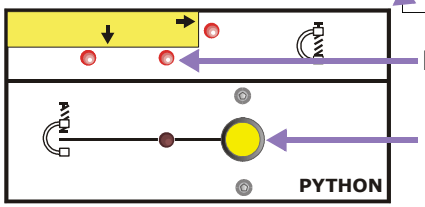
The Python platesetter images your plates. It has three main parts: the tilt-table, the drum and the carriage. Each of these is described below.

4.2.1 The tilt-table

The **tilt-table** is where you load the plate. On or near to the tilt-table you will find the following:



- The **registration contacts** — these are located in the crossbar and help you position the plate correctly on the table.
- The **Registration LEDs** and **clamp button:**



Registration LED

Clamp button

The three registration LEDs help you position the plate correctly. You use the clamp button to unclamp the plate after imaging.

- The **clamp** — this automatically clamps the plate once it is in the correct position on the tilt-table.
- The **crossbar** — this sits at the end of the tilt-table. When the imaging cycle begins, the crossbar moves down the tilt-table to insert the plate into the drum.

4.2.2 The drum

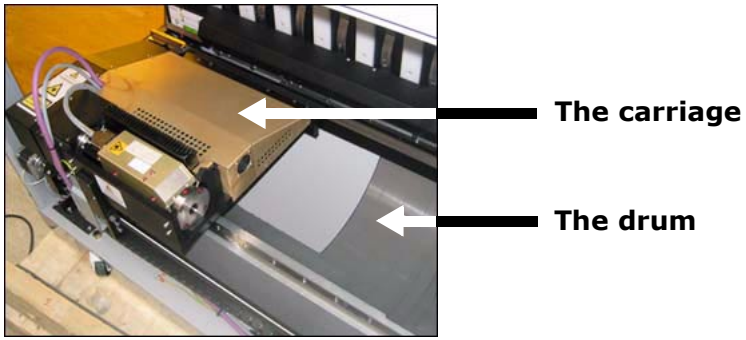
The metal **drum** is located underneath the tilt-table and is where the plate is imaged. The drum contains:

- The **vacuum** — this holds the plate securely in the drum during imaging.
- The **lead screw**, also called the **ball screw** — the carriage (see below) moves along this as the plate is imaged.
- The **plate width system** (only present on early models) — this measures the width of the plate to help position the pressure roller (see below).

4.2.3 The carriage

The **carriage** sits over the drum and contains:

- The **pressure roller** — this rolls over the plate, prior to imaging, to ensure the plate sits tightly in the drum.
- The **laser** — this images the plate.
- The **spinner** — this directs the laser beam around the drum.



4.3 Plate auto-unload

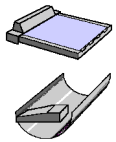
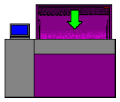
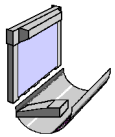
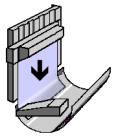
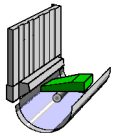
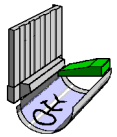
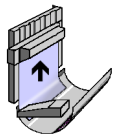
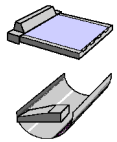
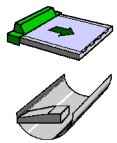
A plate auto-unload option is available with the Python system and means that you don't need to manually remove each imaged plate from Python's tilt-table: instead, the plate is automatically pushed out of the back of the Python platesetter onto a bridge to a processor.

The table overleaf shows the difference in operation when unloading plates manually and when using the plate auto-unload feature.

Python operation with manual plate unload	Python operation with plate auto-unload
The user loads a plate into the clamp.	The user loads a plate into the clamp.
↓	↓
The table is tilted to its vertical position and the crossbar pushes the plate into the drum.	The table is tilted to its vertical position and the crossbar pushes the plate into the drum.
↓	↓
The vacuum is enabled and the plate is squeezed against the drum to ensure correct vacuum hold down.	The vacuum is enabled and the plate is squeezed against the drum to ensure correct vacuum hold down.
↓	↓
The image is exposed onto the plate.	The image is exposed onto the plate.
↓	↓
The vacuum is disabled and the crossbar pulls the (still clamped) plate out of the drum.	The vacuum is disabled and the crossbar pulls the (still clamped) plate out of the drum.
↓	↓
The table is tilted back to the horizontal position.	The table is tilted back to the horizontal position.
↓	↓
The user unclamps and removes the plate.	The plate is automatically unclamped.
↓	↓
The user loads another plate into the clamp.	The crossbar pushes the plate across the table towards the rear plate exit slot in the Python to a bridge leading to a processor.
	↓
	After a delay, the crossbar moves back to the front of the table in preparation for the loading of a new plate.
	↓
	The user loads another plate into the clamp.

4.4 The Python platesetter in operation

Imaging a plate on the Python platesetter involves the following steps:

	You set up a job for output using the Python Console.
	You load a plate onto Python's tilt-table. When the plate is in the correct position it is automatically clamped.
	Close Python's lid.
	The horizontal tilt-table moves downwards to a vertical position.
	The crossbar moves down the tilt-table, inserting the plate into the drum.
	Once the plate is fully in the drum, the pressure roller and vacuum are applied to the plate to ensure that it fits tightly against the drum.
	The carriage moves across the drum and the laser images the job onto the plate.
	Once the job is imaged, the vacuum is released and the crossbar moves back up the tilt-table, removing the plate from the drum.
	The tilt-table moves back to its horizontal position.
	When requested to do so by the Python Console, open the lid and remove the imaged plate. (Alternatively, the plate will be automatically unloaded through the rear exit slot to a bridge leading to a processor, if this feature is available with your system.)

4.5 The Python workstation

The Python workstation is a high-performance computer, which is supplied with all the software you need to process your jobs and output them to the Python platesetter.

The workstation runs Windows 2000 and the following HighWater software applications:

- **Python Console**
  The Python Console gives you full control of job output to the Python platesetter, allowing you to view current, pending, completed and errored jobs, delete and requeue jobs, as well as configure various Python software applications.

  See chapters 7 and 8 for more information about using the Python Console.

- **Torrent RIP**
  HighWater's Torrent RIP (based on the Harlequin RIP) accepts incoming PostScript/PDF files from the network and interprets them into bitmap TIFF files suitable for output to Python.

  See chapters 6, 11 and 12, and the **Torrent User's Guide** for more information about the Torrent RIP.

  Note: If you purchased a two-computer Python system then the Torrent RIP runs on the second computer, which can be connected anywhere on the network.

- **Queue Configuration**
  The Queue Configuration application lets you create the Python Console queues, which output TIFF files from the Torrent RIP to the Python platesetter.

  See chapter 14, Appendix A and Appendix B for more information about creating queues using the Queue Configuration application.

- **Python Layout Tool**
  The Python Layout Tool lets you create layout (ICF) files for use with the Python Console. Layout files contain the plate and job position information that the Python Console needs when outputting your jobs to the platesetter (and also information required for using InkMonitor if this application is installed with your system).

  See chapter 13 for more information about the Python Layout Tool.

- **Barcode Plate Requeue (BPR)**
  The Barcode Plate Requeue (BPR) option adds a barcode to your plates to allow you to quickly locate jobs for re-output.

  See section 7.5 and Appendix C for more information about BPR.

- **Low Res Generator (LRG)**
  The Low Res Generator creates low-res 'view' files for use with the Python Console.

  See section 5.4 for more information about the Low Res Generator.

- **Blackscreen**
  You can use Blackscreen to darken the computer monitor when you are handling plates.

  See section 5.4 for more information about Blackscreen.

- **Python Test Tool**
  The Python Test Tool helps to diagnose problems with the Python system. It also lets you check the laser power setting for your plates, and save system settings.

  See chapters 15, 16 and 18 for more information about the Python Test Tool.

- **InkMonitor (optional)**
  This optional software calculates the amount of ink required on the press. Where relevant, this manual gives instructions for setting up the Python system for use with InkMonitor.

  You should also refer to the **InkMonitor User's Manual** for more information about using InkMonitor.

4.6 Plate information

4.6.1 Plate types

The Python system supports the following plate types:

Plate type	Pos/neg	Filter fitted	Plate type
Agfa Lithostar Plus LAP-V / Ultra-V	Positive	Yes	Silver
Agfa Lithostar N91V	Negative	Yes*	Photopolymer
Mitsubishi Silver Alpha V	Positive	Yes	Silver
Fuji Film LP-NV	Negative	No	Photopolymer
Lastra P-LV-2	Negative	No	Photopolymer

*This is a 0.4D filter, which is different to the filter used for the Agfa LAP-V and Mitsubishi Alpha V plates.

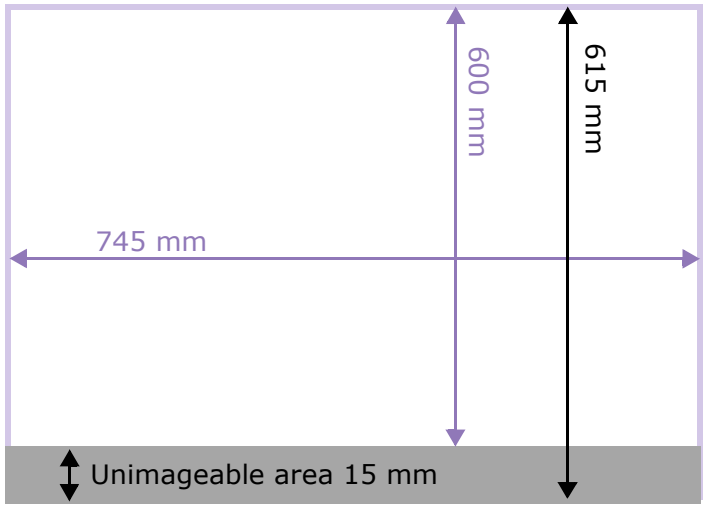
Note: If you are considering changing the type of plate you use on the Python, the filter may need to be changed or removed. Please contact HighWater Designs or your support provider for more information about this.

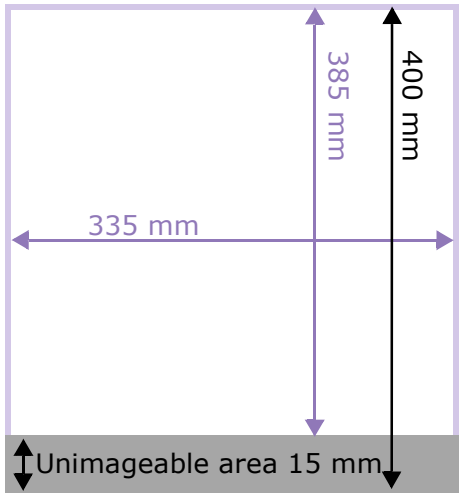
4.6.2 Plate sizes

The Python platesetter accepts plates between the following sizes:

Maximum size	745 x 615 mm 0.3 mm thick	Maximum expose area 745 x 600 mm
Minimum size	335 x 400 mm 0.15 mm thick	Maximum expose area 335 x 385 mm

Maximum size plate





Minimum size plate

Python supports all plate sizes between the minimum/maximum sizes, including (but not limited to) the following plates:

Press size	A4 pages on plate*	Paper size
Hamada B452	2-up	A3
Heidelberg GTO 46	2-up	SRA3
Heidelberg GTO 52	2-up	A3
Heidelberg MO	4-up	SRA2
Heidelberg SM52	2-up	A2
Heidelberg SM72	4-up	B2
Heidelberg SM74	4-up	B2+
Komori 20	2-up	A3
Komori 26	4-up	B2
Komori 28	4-up	B2
Ryobi 34	2-up	SRA3
Ryobi 52	2-up	A3
Ryobi 75	4-up	B2+
Sakurai Oliver 52	2-up	A3
Shinohara 52	2-up	A3

* You need imposition software for this

4.6.3 More information

Plate thickness	0.15–0.3 mm
Grip area	15 mm at front edge of plate (please note that this grip area does not get exposed)
Imaging resolution	2540 dpi / 100 dpmm at 150-200 lpi
Spot size	10 microns

4.7 Plate handling

Plates must be handled with care. You should be aware of the following guidelines:

- Plate edges are sharp, particularly the corners. Very carefully remove plates from their packaging and keep the edges of the plate away from your face.
- Handle plates at their edges.
- Contact with human skin can damage the emulsion surface. Lint free gloves are available for handling plates.
- Keep the interleave paper on the emulsion side of the plate during handling, but remove it before placing the plate onto the tilt-table.
- Always put the plate onto the tilt-table emulsion-side up.

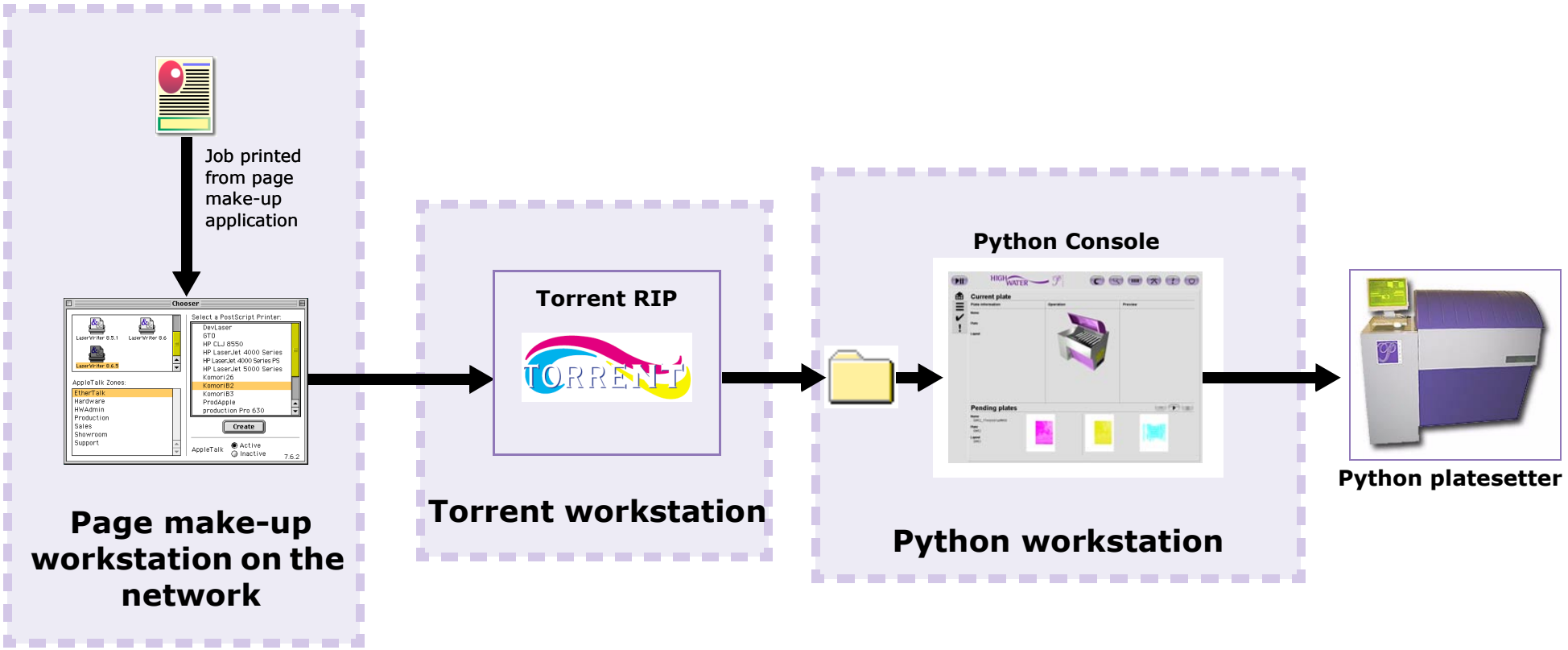
5. Overview of using the Python system

This chapter gives an overview of using the whole Python system. It includes the following sections:

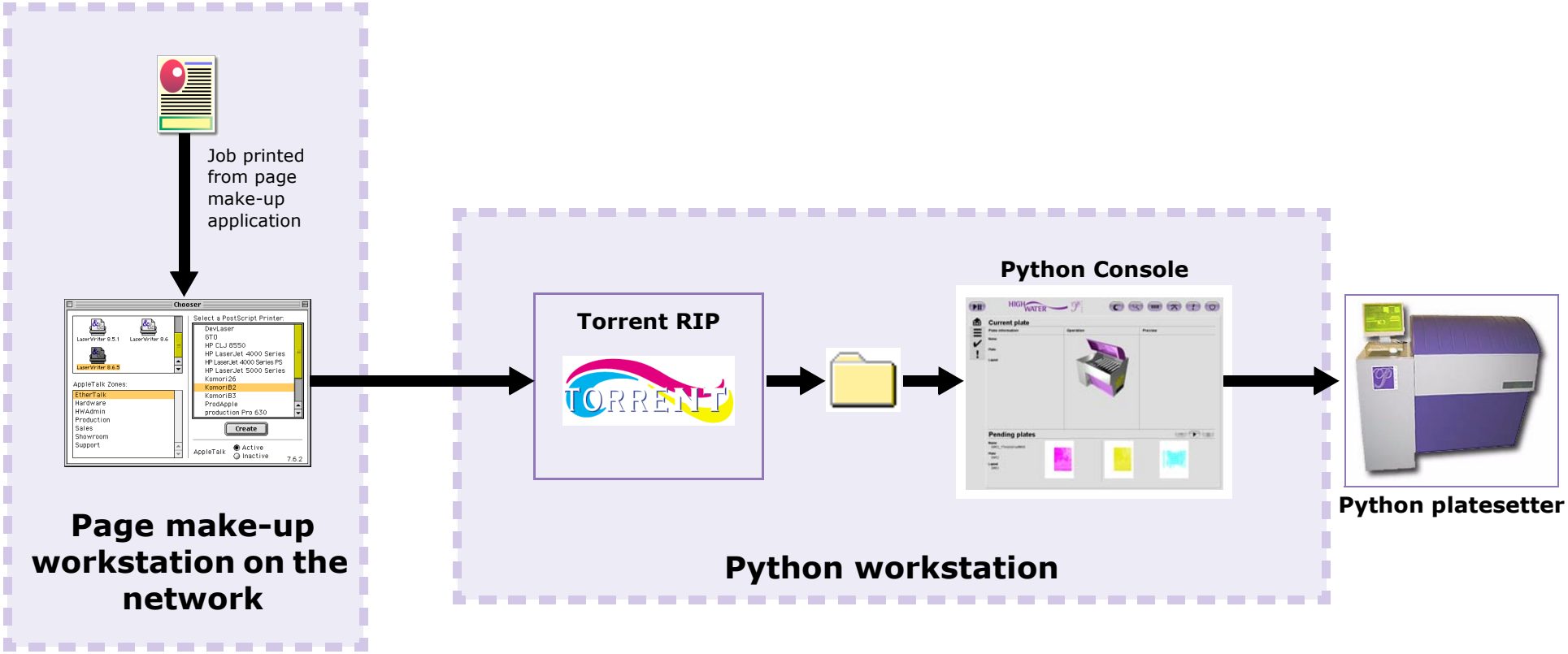
5.1 Outputting a job to the Python platesetter

The two diagrams on the following pages show how you output a job to the Python platesetter. First, you print a job from a page make-up workstation to Torrent (which runs on either a separate workstation or on the Python workstation, depending on your particular system configuration). Torrent RIPs the job and produces TIFF bitmap files that you output to Python using the Python Console.

Python system workflow (two computer Python configuration)



Python system workflow (one computer Python configuration)



5.2 Starting the Python system

To start the Python system:

1. Press the power switch on the back of the Python platesetter.
2. The Python platesetter and workstation start up.
3. Log on to the Python workstation, using the username **python** with no password.
4. The Python Console (and associated software) should start up automatically.

   Note: If you have a two-computer configuration, the Torrent RIP resides on the second computer. Start the Torrent RIP.

The Python system is now ready for use.

5.2.1 Re-launching the Python Console

If you need to re-launch the Python Console at any point, select **Start > Programs > HighWater Designs > Python Console v1.x.x > Python Console**.

Note: If you have any problems running the Python Console after re-launching it, close down all running applications, log off then log back on again using the username **python**. The Python Console will automatically start up.

5.3 Shutting down the Python system

WARNING: When you have finished using the Python system, quit out of all the running applications properly (as described in this section) then shut down the workstation before you switch off the platesetter. If you do not shut down the Python system properly, you may corrupt files and lose data.

5.3.1 Shutting down the Python workstation:

To shut down the Python workstation:

1. In the Python Console, suspend processing (using the **Pause/resume** button in the top left-hand corner) and deal with any requested Python plate operation.
2. Click on the **Quit** button in the top right-hand corner of the 'Python Console' window  .

   Any jobs that are currently being processed or are being output to the Python platesetter are completed before the Python Console quits.
3. Close down all other running applications.
4. From the Windows **Start** menu, select **Shut Down...**.
5. Select **Shut down** then click on the **OK** button.
6. Switch off the monitor, if necessary.

5.3.2 Switching off the Python platesetter

To switch off the Python platesetter:

1. Make sure the workstation has been shut down, as described above.
2. Press the power switch on the back of the Python platesetter.

WARNING: To disconnect power completely from the Python platesetter, the AC power cord must be removed from the wall outlet.

5.4 The Low Res Generator (LRG) and Blackscreen

The Python system includes two useful software utilities, the **Low Res Generator** (LRG) and **Blackscreen**, as described below.

5.4.1 The Low Res Generator (LRG)

The Low Res Generator (LRG) launches automatically during the workstation's startup. When a TIFF file has been output by Torrent and arrives in the bitmaps directory (usually in K:\PLOTDATA\BITMAPS*sub-dir*), LRG produces a low-resolution view file of it for use in the Python Console.

Note: LRG saves the low-res view file at K:\PLOTDATA\VIEW*sub-dir*.

5.4.2 Blackscreen

In the brightroom you need to protect your plates from being exposed to light. The Python workstation monitor has a yellow filter attached to protect your plates from light. However, if you are also using LAP-O type plates or film in the same room as the Python platesetter, you may find the Blackscreen utility useful for blacking the screen.

To use Blackscreen:

1. Select **Start > Programs > HighWater Designs > Python Engine vx.x.x > Blackscreen**.
2. The workstation's monitor will go dark.
3. When you have finished loading/unloading the plate/film, either press the **Esc** key or double-click the mouse to exit Blackscreen.

6. Using the Torrent RIP

The first step in using the Python system is to send your jobs from a page make-up application to the Torrent RIP, which will output them as TIFF bitmap files. Later, you will output these bitmap files to the Python platesetter via the Python Console.

This chapter shows you how to use the Torrent RIP and includes the following sections:

6.1 Before you begin

Before you start working through this chapter, please be aware of the following:

- **Before you can print to Torrent you need to create the Python print queue(s) on every computer that you will be printing from. This is covered in chapter 19.**
- You can use your usual page make-up and imposition software with the Torrent RIP if it can produce PostScript or PDF files.
- The Python installation engineer created one or more Torrent page setups and input queues specifically for use with the Python system. If there is more than one page setup or input queue, make sure you know the name, purpose and settings of each one.
- You should know how to print a job from your page make-up application. If not, please refer to your page make-up application's manual for full details on printing jobs.
- If you are not printing your jobs to Torrent via an AppleTalk network, please refer to the **Torrent User's Guide** which gives details on other methods of printing your jobs to Torrent.
- Quit out of Torrent properly when you have finished using it.

6.2 Overview of using the Torrent RIP

You use Torrent to RIP your jobs and produce TIFF bitmap files, which you later output to the Python platesetter via the Python Console. This chapter gives a basic overview of the Torrent RIP and using it as part of the Python system. However, setting up and using Torrent is potentially a complex task so, if you require more information on Torrent and its many capabilities, you should refer to the **Torrent User's Guide**.

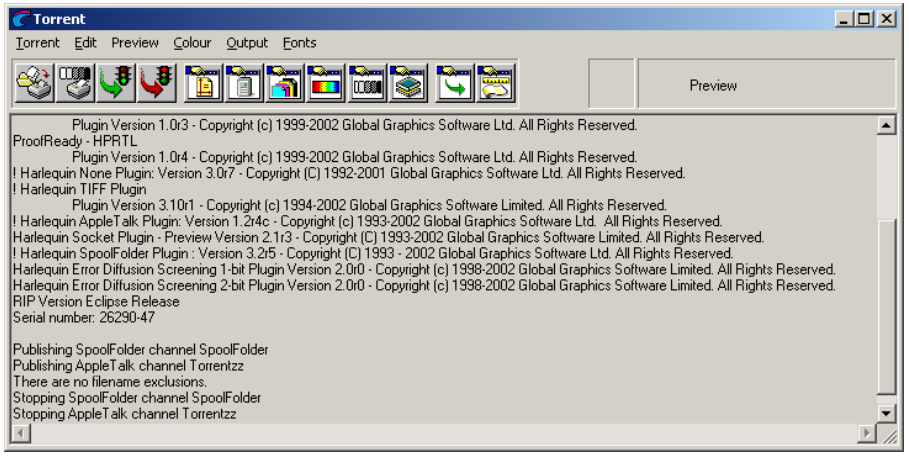
Using Torrent as part of the Python system involves the following steps:

- Start the Torrent RIP, if it is not already running.
- If necessary, start Torrent's input queues, which let you send jobs across your network to Torrent.
- On the page make-up workstation, open or create a job.
- Print the job to Torrent from your page make-up application.
- Optionally, use Torrent's Output Controller to preview jobs and carry out job management.
- Once the jobs have been output as TIFF files by Torrent, they will appear in the relevant queue in the Python Console and will be output to the Python platesetter.

6.3 Torrent's main window

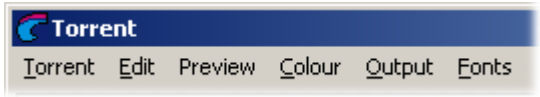
If Torrent is not already running, launch it now by clicking on the Torrent RIP button in the Python Console. (If the Python Console is not running, select **Start > Programs > HighWater Designs > Torrent vx.x > Torrent vx.x**.)

Torrent's main window appears:



Torrent's main window contains:

- The menus, which contain all the commands needed to run Torrent:



- The toolbar containing 12 buttons, which let you quickly access Torrent's most commonly used menu options and dialog boxes (section 6.9 describes each of these buttons):

- The status area, which shows what Torrent is doing (refer to the **Torrent User's Guide** for more details about this):

- The RIP Monitor — this large text area below the toolbar shows the progress of jobs through Torrent, and displays information about timing, errors, job completion, fonts, etc. If you have problems with the Torrent RIP, check for error messages here.

6.4 Page setups and input queues

In Torrent, a page setup specifies the settings to be applied to a job (for example, the resolution, page size, orientation, scaling, screening, dot shape, and the 'device' a job is to be printed to). Each different combination of settings requires a separate page setup.

One or more page setups have already been created for you by the Python installation engineer. Each of these page setups is associated with its own input queue, which lets you send jobs across the network to the Torrent RIP. Once your input queues have been 'broadcast' (as described below), they appear as virtual printers on your page make-up workstations on the network, and you can print your jobs directly to them.

6.4.1 Broadcasting the input queues

You need to broadcast the input queues so that you can print jobs to Torrent. To broadcast them, do any of the following:

- Select **Torrent > Start Inputs...** (so that there is a tick next to it).
- Press **Ctrl-I** on the keyboard.
- Click the **Start Inputs** button on the toolbar.

While the input queues are being broadcast, you will see the 'Input Manager Dial'. When it has finished, Torrent is ready to receive and process incoming PostScript/PDF jobs from across the network (assuming that the print queues have been set up on those machines, as described in chapter 19).

6.4.2 Stopping the input queues

There are times when you need to stop the Torrent RIP accepting incoming jobs (for example, when you are creating or editing page setups). To do this, you need to stop the input queues by doing any of the following:

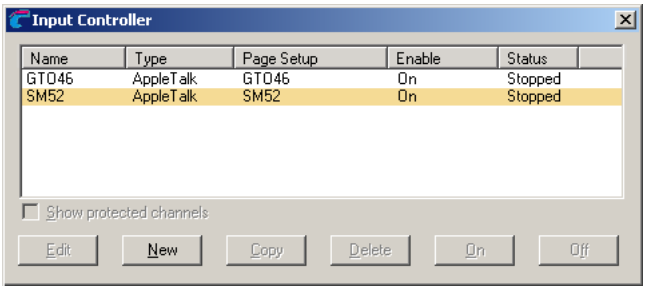
- Select **Torrent > Start inputs...** (so that there is no tick by it).
- Press **Ctrl-I** on the keyboard.
- Click the **Stop Inputs** button on the toolbar.
- Select **Stop Input Queue** from the **Input Queue** menu.

When the input queues have stopped you will no longer be able to print jobs to Torrent from across the network.

6.4.3 Viewing the input queues

In Torrent, to check which input queues have been set up and are available:

1. Select **Torrent > Input Controller...** to display the Input Controller:



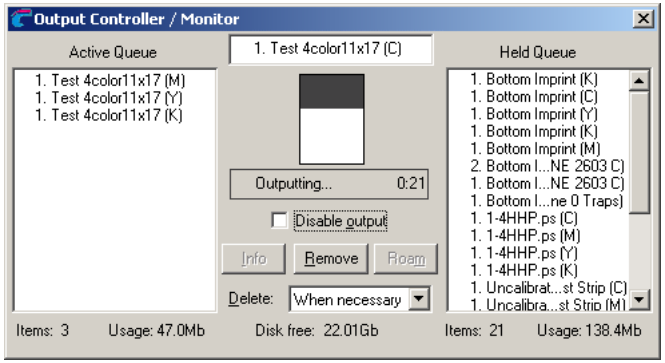
2. This window lists each input queue along with other information, including the associated page setup.
3. Make sure that each required input queue is listed as **On** in the **Enable** column, otherwise you will not be able to print to it when you have broadcast the inputs. (To enable an input queue, highlight it and click the **On** button.)
4. When you have finished, close the Input Controller.

6.5 The Output Controller

When you print jobs to Torrent you can use the Output Controller to monitor and control the progress of those jobs through the RIP. This section gives a brief overview of the Output Controller.

Note: Use of the Output Controller is optional.

To display the Output Controller, select **Output > Output Controller**. The following window is displayed:



The Output Controller's **Active Queue** contains pages that are waiting to be output. Pages listed at the top of the Active Queue will be printed first. The **Held Queue** contains pages that have been output (or you can move them there from the Active Queue to suspend them).

When a page is being output, it is shown in the box at the top of the Output Controller between the Active Queue and the Held Queue.

The number of pages in each queue, together with the amount of disk space they use, is displayed below each queue.

The Output Controller allows you to:

- Suspend job output.
- Preview jobs on screen.
- Re-print, re-order, stop and remove jobs.

These functions are described in the following sub-sections.

Note: Refer to the **Torrent User's Guide** for full details on the Output Controller.

6.5.1 Suspending job output

To stop jobs being output by the Torrent RIP, check the **Disable output** box. All job processing will stop once the current job has been processed.

Note: Disabling output does not stop jobs being printed to Torrent from across the network.

6.5.2 Previewing jobs on screen

Before your jobs are output as TIFF files, you may want to preview them on screen, for example, to make sure that jobs have separated correctly.

Note: The Python Console has a more powerful program for previewing your jobs, which you may prefer to use instead. Please refer to section 7.9 for more details.

To preview your jobs in the Torrent RIP:

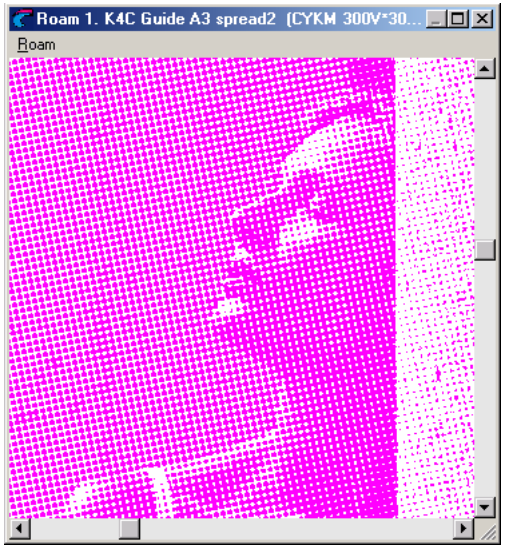
1. Before you send jobs to Torrent from your page make-up application, check the **Disable output** box on Torrent's Output Controller to stop Torrent outputting the jobs as TIFF files.

   Note: You need to disable output because you cannot preview a job while Torrent is outputting jobs.

2. Print your job to Torrent (see section 6.6) then highlight one or more of the job's separations in the Active Queue and click on the **Roam** button.

   Note: You can select more than one job separation using the **Shift** and/or **Ctrl** keys on the keyboard.

3. A preview of the job appears on-screen, for example:



4. The mouse cursor turns into a hand icon when it is placed over the image. You can use this to move the image around to view specific parts of it.
5. To view the whole job, select **Reduced Roam** from the **Roam** menu.

   Note: The 'Options' dialog, available in the **Roam** menu, gives you more viewing options.
6. When you have finished previewing the job, close all open 'Roam' windows.
7. Uncheck the **Disable output** box when you want Torrent to start outputting your jobs again.

6.5.3 Re-printing, re-ordering, stopping and removing jobs

You can also use the Output Controller to:

- **Re-print a job**: drag the job(s) from the Held Queue back to the Active Queue. To check or edit the settings before the job is re-printed, highlight the required job and click on the **Info** button (refer to the **Torrent User's Guide** for more information about changing job settings).
- **Change the order that jobs are printed**: drag the job(s) to the required place in the Active Queue (jobs at the top of the queue are printed first).
- **Stop a job that is currently being printed**: drag the job from the top box to either the Held Queue or the Active Queue.
- **Stop a job that has not yet been printed**: drag the job from the Active Queue to the Held Queue.
- **Permanently remove a job from either queue**: highlight the job and click on the **Remove** button. You will be asked to confirm the removal of the job. Click on **Yes**.

Note: For most of the above operations, you can select more than one job at a time using the **Shift** and/or **Ctrl** keys on the keyboard.

6.6 Printing a job to Torrent

This section gives you basic instructions for printing a job from your page make-up workstation to Torrent.

Note: You should already know how to print a job in your page make-up application. Please consult the relevant application's manual for more details about printing jobs.

WARNING: Before you can print to Torrent you need to set up the Python print queue(s) on your page make-up workstation(s). This is covered in chapter 19 and you should do this now if it has not been done already.

6.6.1 Making the Torrent input queues available

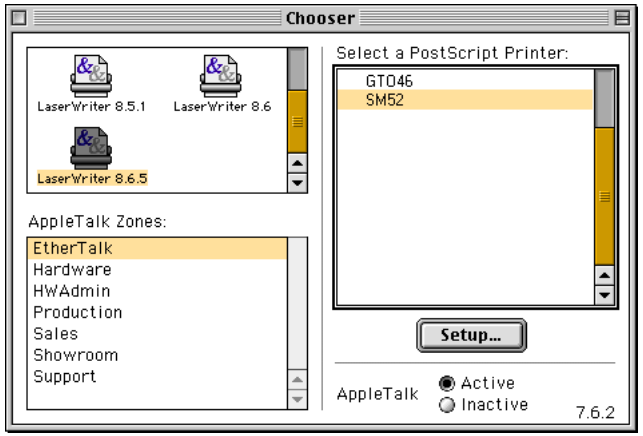
Before printing to Torrent, you need to make Torrent's input queues available over the network. To do this:

1. In Torrent, select **Start Inputs** (so there is a tick by it) from the **Torrent** menu. While the input queues are being broadcast you will see the 'Input Manager Dial'.
2. After a short time, the Torrent RIP is ready to receive jobs printed from across the network.

6.6.2 Selecting the printer (Mac Classic users only)

If you are printing to Torrent from Mac Classic you need to select the printer first:

1. On the Macintosh, open the **Chooser**. Click once on the LaserWriter icon in the top left-hand window, and then on **EtherTalk** in the 'AppleTalk Zones' window (the bottom-left window). The names of the available Torrent input queues appear in the right-hand window along with any other printers, for example:



2. Highlight the required input queue then close the Chooser.

6.6.3 Setting the print settings

To set the print settings:

1. Launch your page make-up application and open or create a job within it.
2. In the 'Print' or 'Page Setup' dialog, set the following options:
   - **Separations** should be **On**.
   - For the **Printer Description**, select **Python 2-up**.
   - Select **Custom** for the **Paper Size** and specify the **Paper Width** and **Paper Height** values, taking into account the additional measurement for registration marks (for example, Quark jobs typically add 11 mm all round).

WARNING: The paper width and height must include any extra measurement for registration marks, etc., otherwise the job will not output correctly.

- For the **Printer**, choose the name of the Torrent input queue.

3. Set all other options, as required.

6.6.4 Printing the job

When you have finished setting all the options on the 'Page Setup'/'Print' dialog:

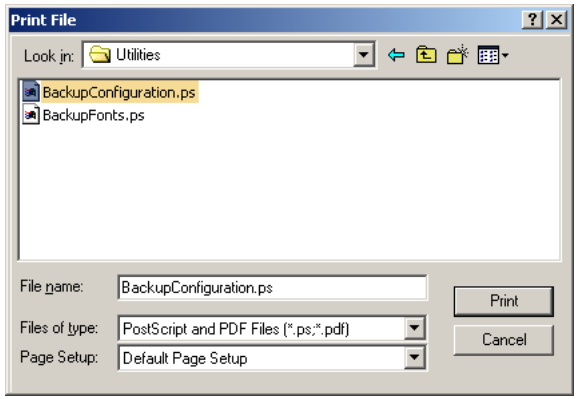
1. On the 'Print' dialog, click the **Print** or **OK** button to print the file to Torrent.
2. In Torrent, you will see a progress dialog as the job is being input.
3. Once the job has been input, each separation appears in the Output Controller's Active Queue. As each separation is output as a TIFF file, it is moved to the Held Queue.

   Note: Section 6.5 shows you how to use the Output Controller to preview and control the processing of your jobs in Torrent.

6.7 Creating a Torrent settings backup file

You can create a backup file of your Torrent configuration settings (page setups, input queues, and so on). To do this:

1. In Torrent, select **Print File...** from the **Torrent** menu to display the following dialog:

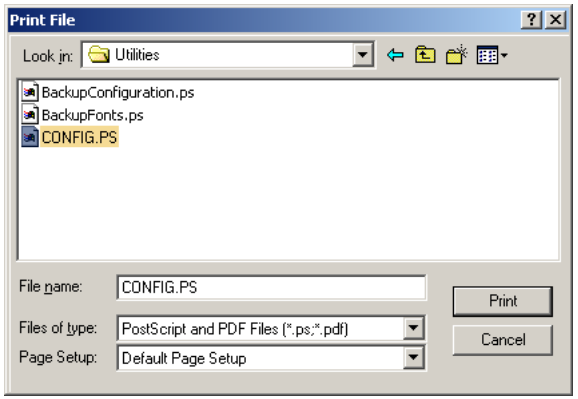


2. Navigate to the Torrent RIP's **Utilities** folder (the default location for this is D:\TORR7.1r0\SW\Utilities), select the **BackupConfiguration.ps** file and click on the **Print** button.
3. In the RIP Monitor you will see the backup file, CONFIG.PS, being created.

6.7.1 Restoring the backup file

To restore your Torrent RIP settings (page setups, input queues, and so on) from a previously created Torrent backup file:

1. In Torrent, select **Print File...** from the **Torrent** menu to display the following dialog:



2. Navigate to the Torrent RIP's **Utilities** folder (the default location for this is D:\TORR7.1r0\SW\Utilities), select the **CONFIG.PS** file and click on the **Print** button.
3. In the RIP Monitor you will see the RIP's configuration settings being restored.
4. Re-start the RIP to apply the restored settings.

6.8 When you have finished

When your jobs have been output as TIFF files by Torrent they are ready for output to the Python platesetter using the Python Console software. The job separations will automatically appear in the relevant Python Console queue(s).

- Chapters 7 and 8 show you how to use the Python Console to output jobs to the Python platesetter and manage your jobs.
- Chapter 9 shows you how to set up the Python platesetter for imaging the plate.

6.9 Torrent's toolbar

Torrent's toolbar gives you quick access to the most frequently used menu options and dialog boxes. The toolbar buttons and their meanings are shown below.

Note: Not all of these buttons are required for use with the Python system.

	**Print File** This displays the 'Print File' dialog, which lets you print a file. This is equivalent to the **Torrent > Print File** command.
	**Print Calibration** This displays the 'Print Calibration' dialog, which lets you print calibration targets for use in calibrating output devices. This is equivalent to the **Output > Print Calibration** command.
	**Start Inputs** This starts any enabled inputs. This is equivalent to the **Torrent > Start Inputs** command when the inputs are stopped.
	**Stop Inputs** This stops any enabled inputs. This is equivalent to the **Torrent > Start Inputs** command when the inputs are started.

Page Setup Manager
This displays the 'Page Setup Manager' dialog, which lets you create and edit page setups. This is equivalent to the **Torrent > Page Setup Manager** command.

Device Manager
This displays the 'Device Manager' dialog, which lets you create and configure devices driven from a multiple device driver. This is equivalent to the **Torrent > Device Manager** command.

Separations Manager
This displays the 'Separations Manager' dialog, which lets you create and edit separation styles for the selected device. This is equivalent to the **Colour > Separations Manager** command.

Colour Setup Manager
This displays the 'Colour Setup Manager' dialog box, which allows you to create and edit colour setups with or without colour management (depending on whether the password-enabled ColourPro option is activated).This is equivalent to the **Colour > Colour Setup Manager** command.

Calibration Manager
This displays the 'Calibration (Dot Gain) Manager' dialog, which lets you create and edit calibration sets. This is equivalent to the **Output > Calibration Manager** command.

Cassette Manager
This displays the 'Cassette Manager' dialog, which lets you create cassettes and edit their information. This is equivalent to the **Output > Cassette Manager** command.

Input Controller
This displays the 'Input Controller' dialog (or it hides the controller if it is already displayed). This is equivalent to the **Torrent > Input Controller** command.

Media Manager
This displays the 'Media Manager' dialog, which lets you set up automatic media management. This is equivalent to the **Output > Media Manager** command.

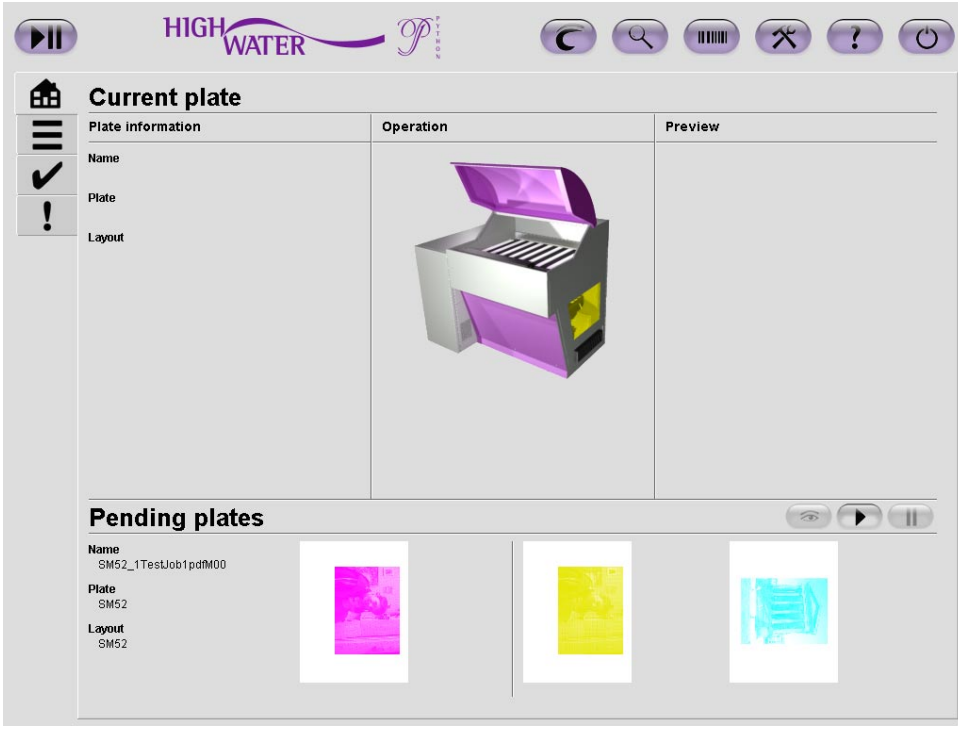
7. The Python Console — Overview

You use the Python Console to output plates to the Python platesetter, manage plate queues, view errored plates, and so on. This chapter gives an overview of the Python Console and it includes the following sections:

Note: Using the Python Console for processing jobs and outputting them to the Python platesetter is covered in chapter 8.

7.1 The Python Console

When you log in to the Python workstation the Python Console launches:



Note: If you need to re-launch the Python Console at any point, select **Start > Programs > HighWater Designs > Python Console v1.x.x > Python Console**. If the Python Console does not launch, log off then log back into Windows using the username **python** with no password.

The 'Home' tab (shown in the previous dialog) is displayed when the Python Console first launches. The 'Current plate' panel shows the plate (if there is one) that is currently being processed/output to Python. The 'Pending plates' panel shows the next plates (up to three) that are waiting for output. These two panels are discussed fully in the next chapter.

The following sections describe the 'Python Console' window and the buttons that are common to all tabs.

7.2 The pause/resume button

The **pause/resume** button in the top left-hand corner of the window controls the overall processing of the whole Python Console. It indicates the state of the Python Console:

- Flashing grey — the system is starting up.
- Flashing green/black — the Python Console is suspended. No processing is occurring (apart from any jobs that were being processed or were being output to the Python platesetter when the system was suspended).
- Solid green — the Python Console is active and is processing jobs.

Click on the **pause/resume** button to toggle the Python Console's processing status between suspended and active.

Notes: It may take a short time for the system to change between the active and suspended states.

As well as pausing/resuming processing of the whole Python Console, you can also suspend individual plates and queues. This is described further in chapter 8.

7.3 The toolbar buttons

The Python Console has six toolbar buttons along the top right of the window:

The **Torrent** button displays the Torrent RIP application (or it will launch the RIP if it is not already running). Using the Torrent RIP is covered in more detail in chapters 6, 11 and 12.

Note: If the Torrent RIP is running on a separate workstation to the Python workstation, then this button will not do anything. You can remove this button from the Python Console by following the instructions in section 7.10.

The **Search for plate** button lets you search for a plate (in any queue). This is covered further in section 7.4.

The **barcode** button lets you requeue a completed plate using its barcode (if this feature has been implemented). This is covered further in section 7.5.

The **Tools** button gives you access to:

- **The Python Test Tool**, which can:
  - Help diagnose problems with the Python system.
  - Give you access to the platesetter's drum for cleaning.
  - Let you check the laser power setting for your plates.
  - Save system settings.

  See chapters 15, 16 and 18 for further details.
- **Configuration** options for setting up the various Python software applications and specifying their locations. See section 7.6 for further details.
- **Preferences**, which let you specify email addresses for sending log files. See section 7.8 for further details.

Note: The **Tools** button is not available when the Python Console is processing jobs.

The **Help** button opens this manual, the **Python User Guide**, in Adobe Acrobat.

The **Quit** button quits out of the Python Console software.

Note: The **Quit** button is not available if the Python Console is processing jobs.

WARNING: Before quitting out of the Python Console, you MUST first suspend processing (using the Pause/resume **button in the top left-hand corner) and deal with any requested plate operation.**

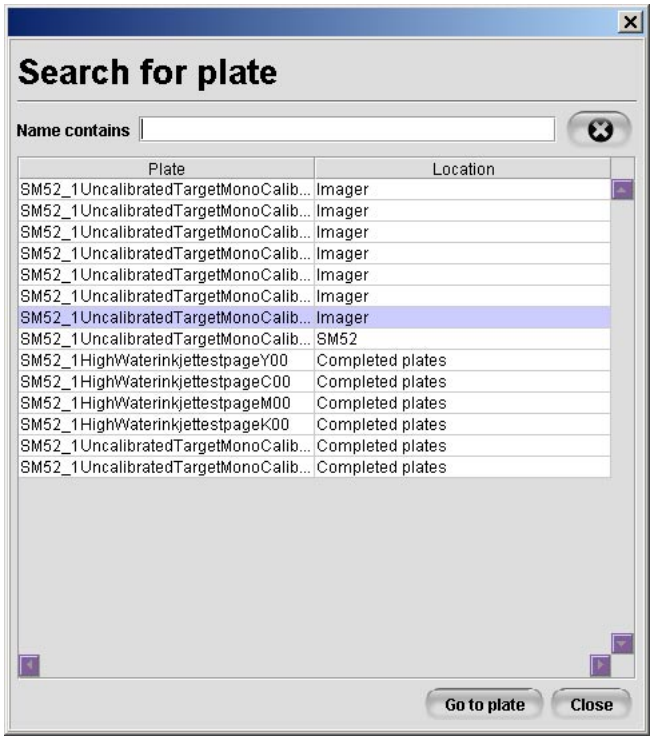
Any jobs that are currently being processed or are being output to the Python platesetter are completed before the Python Console quits.

7.4 Plate search

The plate search button  lets you search for a plate in any queue. To do this:

1. Click on the **Search for plate** button in the top right-hand corner of the Python Console.

2. The 'Search for plate' dialog is displayed listing all plates that are currently in the system (that is, processing, pending, completed and errored plates), for example:

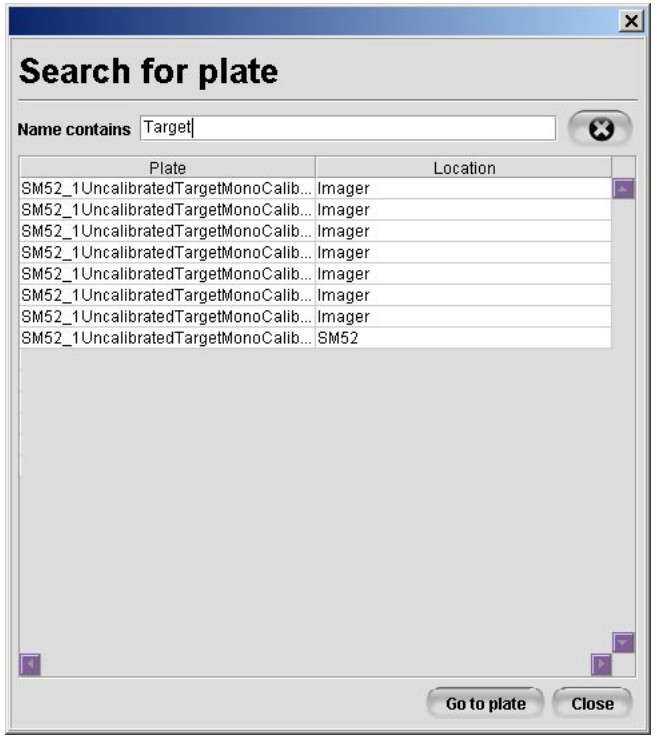


3. To quickly locate the required plate, simply type the plate name, or part of it, into the **Name contains** text box.

   Notes: The search is not case-sensitive and is also carried out on the queue name (which is prepended to the plate name).

   To clear the **Name contains** text box, click on the clear button  .

4. All the plate names containing the text you entered are displayed, along with their location, for example:



5. Select the plate you want to locate then click on the **Go to plate** button. You will be taken to the plate in its relevant queue.

7.5 Requeueing a plate using its barcode

Any plate that has already been output to the Python platesetter can be requeued for output. If a barcode was printed on the plate, you can locate the plate and requeue it using its barcode.

Notes: Refer to Appendix C for more information on setting up BPR to add a barcode to your plates.

The record of a plate's barcode is held for a certain amount of time since the plate was last output (this time limit is specified in the BPR Preferences, as described in section C.2). When the time limit is exceeded, a 'Cannot find the plate...' message is displayed when you attempt to requeue a plate.

To requeue a plate for output using its barcode:

1. In the Python Console, click on the Barcode icon in the top right-hand corner of the window. The following dialog is displayed:

2. Either scan the barcode off the plate, or manually type in the 12-digit barcode number (with or without hyphens).

   Notes: The correct format for the barcode is 12 digits (for example, 000000000523, or three groups of 4 digits separated by hyphens, for example, 0000-0000-0523).

   If the barcode was not typed in the correct format, as specified above, you will see an 'Invalid identifier' message when you click on **OK**.

3. Click on the **OK** button.

4. If found, the plate will be automatically moved back to its original queue, where it will be output to the Python platesetter again, when that queue is active. You will see a message similar to the following:

   If the plate cannot be found, you will see the following message:

   Click on **OK** to close the dialog.

7.6 The Tools button

The Tools button in the top right-hand corner of the Python Console gives you access to:

- The **Python Test Tool**, which can help diagnose problems with the Python system, give you access to the platesetter's drum for cleaning, check the laser power setting for your plates and save system settings. See chapters 15, 16 and 18 for more details about using the Python Test Tool.
- **Configuration** options. These let you configure the various software applications (for example, Queue Configuration and Torrent) needed to run the Python system. The configuration options also let you specify the paths to these applications. This is covered in section 7.7.

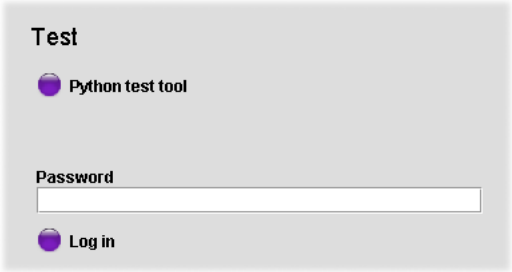
- The **Preferences** let you specify email address(es) for sending log files and let you change the password for accessing the Configuration options. This is covered in section 7.8.

To access these tools:

1. Set the Python Console to its suspended state (if necessary, click on the **Pause/resume** button in the top left-hand corner of the window).

   Note: The current plate output must also be completed and the plate unloaded before you can access the **Tools** button.

2. Click on the **Tools** button in the top right-hand corner of the window.
3. The following is displayed:



   - To launch the Python Test Tool, click on the **Python test tool** button. (See chapters 15, 16 and 18 for more details about the Python Test Tool.) You will see a 'Suspending processing' dialog before the Python Test Tool launches.

     Note: If the button is red, the location of the Python test tool on disk has not been defined. You can set this by following the instructions in section 7.7.2.

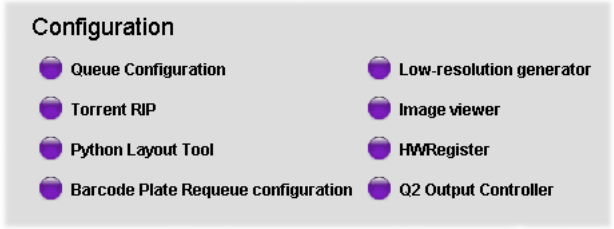
   - To access the Configuration options or Preferences, type in the **Password** (by default, this is **administrator**) then click on the **Log in** button. The 'Configuration' and 'Preferences' tabs at the top of the window become active. These are described in the next two sections.

7.7 Configuration options

The 'Configuration' tab lists a number of software applications that are used as part of the Python system.

To access the Configuration options:

1. Click on the **Configuration** tab. You will see a list of the Python-related software applications:



   Note: If an application has a red button alongside it, the location of that application on disk has not been defined. You can set this by following the instructions in section 7.7.2.

You can use the 'Configuration' window to:

- Launch these applications in order to configure the Python system, as described in section 7.7.1.
- Set the paths to these applications, as described in section 7.7.2. You need to do this so that the Python Console knows where to find the required applications.

The available applications are as follows:

- **Queue Configuration**
  You use this to create the Python Console queues. See chapter 14, Appendix A and Appendix B for more information about creating queues using the Queue Configuration application.

- **Torrent RIP**
  HighWater's Torrent RIP (based on the Harlequin RIP) accepts incoming PostScript/PDF files from the network and interprets them into bitmap TIFF files suitable for output to the Python platesetter.

  See chapters 6, 11 and 12, and the **Torrent User's Guide** for more information about using the Torrent RIP.

  Note: In some Python configurations, the Torrent RIP resides on a separate workstation.

- **Python Layout Tool**
  The Python Layout Tool lets you create layout (ICF) files for use with the Python Console. Layout files contain the plate and job position information that the Python Console needs when outputting your jobs to the Python platesetter (and also information required for setting up InkMonitor if this application is installed with your Python system).

  See chapter 13 for more information about the Python Layout Tool.

- **Barcode Plate Requeue configuration**
  The Barcode Plate Requeue (BPR) option adds a barcode to your plates to allow you to quickly locate jobs for re-output.

  See section 7.5 and Appendix C for more information about using BPR.

- **Low-resolution generator**
  This creates low-res 'view' files for use with the Python Console.

  See section 5.4 for more information about the Low Res Generator.

- **Image viewer**
  The image viewer (HighWater's HWRoam application) allows you to preview a plate. Refer to section 7.9 for more information about using the image viewer.

- **HWRegister**
  This is used to enable the Python software applications. (The installation engineer should already have done this.)

- **Q2 Output Controller**
  This underlying application processes your jobs and outputs plates to the Python platesetter.

7.7.1 Launching the software applications

To launch the software applications listed on the Python Console's 'Configuration' tab (except the Q2 Output Controller and Low-res generator, which you cannot launch):

1. Click on the relevant button to launch that particular software application.

   Notes: If the application is already running in the background, it will be moved into the foreground.

   If an application has a red button alongside it, that application's location hasn't been defined so the application will not launch. Set the location, as described next.

7.7.2 Specifying an application's location on disk

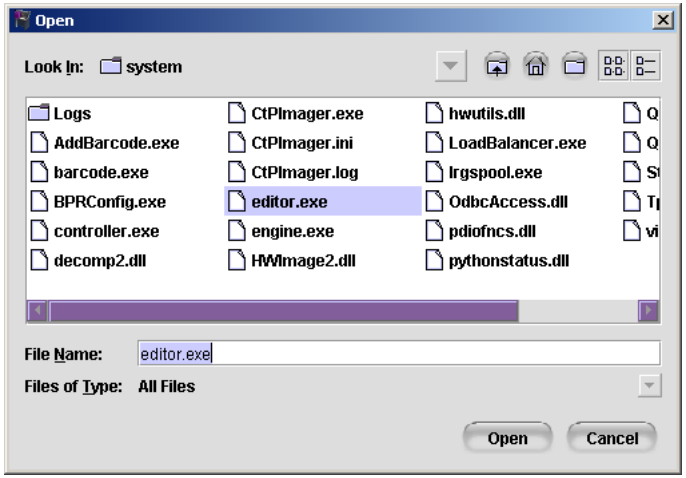
The 'Configuration' tab allows you to specify an application's location on disk. You may need to do this if you have installed a new version of the software and you need to tell the system where it can be found or if an application has a red button alongside it indicating that its application path is not set.

To specify an application's location on disk:

1. Hold the **shift** key down and click on the application's button.

   Note: If the application's button is red, you do not need to hold the **shift** key down.

2. An 'Open' dialog is displayed:



3. Locate the required application then click on the **Open** button.

   The default application locations are:

   - **Python Test Tool**
     D:\Python\Engine v5.0.0\testtool.exe
   - **Queue Configuration**
     D:\Python\Q2v4.4\system\Editor.exe
   - **Torrent RIP**
     D:\TORR7.1r0\HIGHWI.exe

     Note: If the Torrent RIP resides on a separate workstation, you will not be able to launch it from the Python workstation or set its application path.
   - **Python Layout Tool**
     D:\Python\Engine v5.x.x\plotter.exe
   - **Barcode Plate Requeue**
     D:\Python\Q2v4.4\system\BPRConfig.exe
   - **Low-res generator**
     D:\Python\Q2v4.4\System\LRGD.exe
   - **Image viewer**
     D:\HWRoam v1_3\HWRoam.exe
   - **HWRegister**
     D:\Python\Q2v4.4\security\HWRegister.exe
   - **Q2 Output Controller**
     D:\Python\Q2v4.4\system\controller.exe

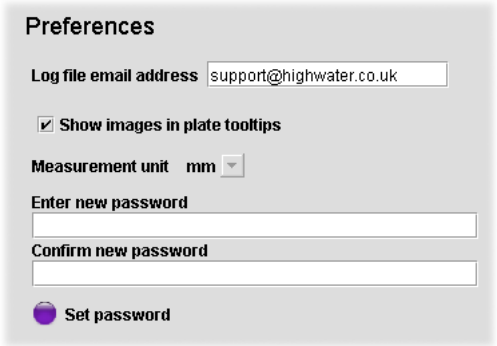
The next section describes how to set the Preferences. Otherwise, if you have finished using the Python Console tools, click on the **Return to user mode** button  .

7.8 Python Console Preferences

The Python Console Preferences let you specify email addresses for sending log files and let you change the password for accessing the Configuration options.

To set the Preferences:

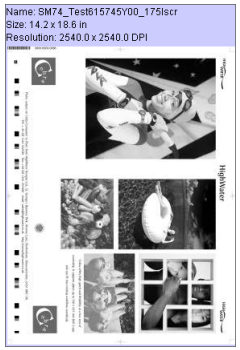
1. If you haven't done so already, open the tools by following the instructions in section 7.6 and type in the password (the default is **administrator**).
2. Click on the **Preferences** tab. The following window is displayed:



3. In the **Log file email address** field, type the email address(es) where you would like error log files to be sent. If there is more than one email address, separate them with a semi-colon, for example:

   support@highwater.co.uk;support@colorquest.co.uk

4. Check the **Show images in plate tooltips** box if you wish to view a thumbnail of the image and image details when you hover your mouse over the preview, for example:



5. Select the **Measurement units** you wish to work in (millimetres or inches).
6. If you wish to change the password for accessing the Python Console tools, then type the new password into the **Enter new password** field and again into the **Confirm new password** field.

   Click on the **Set password** button.

WARNING: Any password you set is not secure — that is, it is held in an unencrypted file in a human-readable format.

If you have finished using the Python Console tools, click on the **Return to user mode** button  .

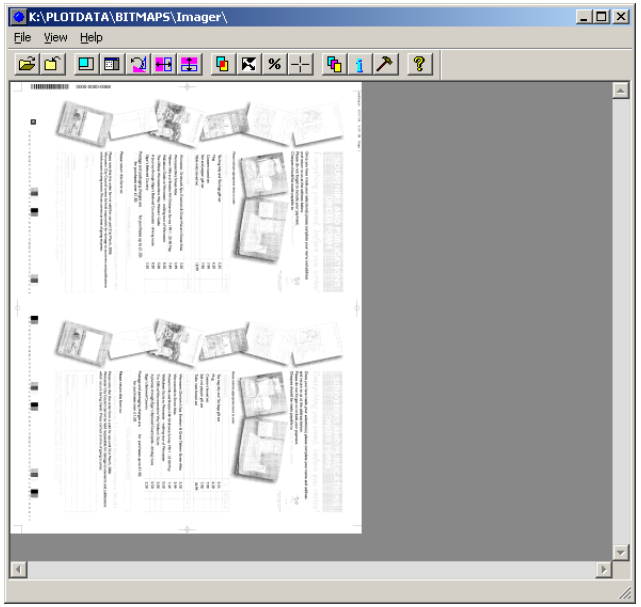
7.9 Previewing a plate

All tabs on the Python Console let you preview plates using HighWater's HWRoam application. To preview a plate with HWRoam:

1. Highlight a plate (either its preview or in a list):

Plate	Status
SM52_1Frontcoverpdf00	Suspended
SM52_1TestJob1pdfM00	Suspended
SM52_1CALENDARcolpdf00	Suspended
SM52_3CALENDARcolpdf00	Suspended
SM52_2CALENDARcolpdf00	Suspended
SM52_5CALENDARcolpdf00	Suspended
SM52_4CALENDARcolpdf00	Suspended
SM52_1TestJob1pdfC00	Suspended

2. Click on the **Viewer** button  . The plate is opened in the HWRoam application, for example:



Note: Please refer to the HWRoam documentation for full details on using HWRoam. This can be found from the Windows' **Start** menu in **Programs > HighWater Designs > HWRoam 1.4 > HWRoam manual**.

3. When you have finished, close the 'HWRoam' window.

7.10 Removing the Torrent button from the Python Console

The Torrent button  in the top right of the 'Python Console' window switches the Torrent RIP application into the foreground, if Torrent is installed on the Python workstation. However, if the Torrent RIP is running on a separate workstation, this button has no use in which case you can remove it from the Python Console, if desired. To do this:

1. On the Python workstation, open the Windows Explorer.

   Note: If the Python Console is running, you can do this by pressing the **Alt-tab** keys to display the **Start** menu, then right-click in the **Start** menu and select **Explore**.

2. In Windows Explorer, navigate to D:\Python\Console v1.x.x.
3. Open the **CCGController.ini** file in Notepad.
4. Scroll down to the section starting:

   # The RIP.
   application.rip.path=D:\TORR7.1r0\HIGHWI.exe

   Change this to:

   # The RIP.
   application.rip.path=

5. Save the file and close the Notepad window.

When you next log in to the Python workstation and run the Python Console, there will be no Torrent button.

7.10.1 Restoring the Torrent RIP button in the Python Console

If you wish to display the removed Torrent button in the Python Console then you need to edit the CGGController.ini file as follows:

1. In Windows Explorer, navigate to D:\Python\Console v1.x.x.
2. Open the **CCGController.ini** file in Notepad.
3. Scroll down to the section starting:

   # The RIP.
   application.rip.path=

   Change this to:

   # The RIP.
   application.rip.path=D:\TORR7.1r0\HIGHWI.exe

   Note: If **D:\TORR7.1r0\HIGHWI.exe** is not the path to the Torrent RIP executable file, then replace it with the correct location.

4. Save the file and close the Notepad window.

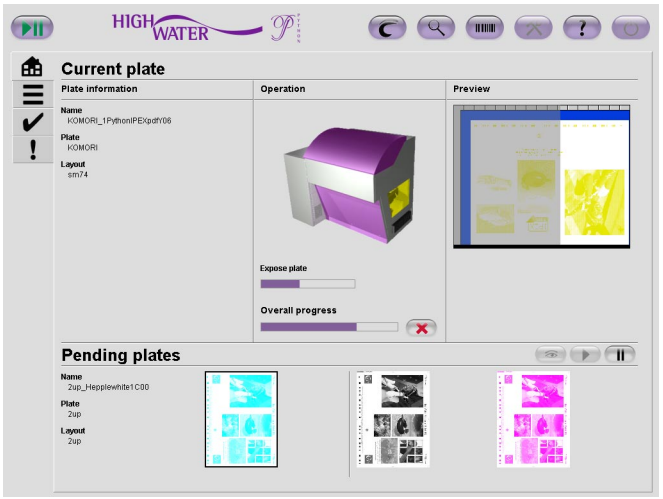
When you next log in to the Python workstation and run the Python Console, the Torrent button will be present.

8. The Python Console tabs

This chapter describes the tabs available in the Python Console:

8.1 The 'Home' tab

The Python Console's 'Home' tab is displayed when the Python Console launches. The 'Home' tab shows the plate (if any) that is currently being processed/output to the Python platesetter and the next plates (up to three) that are queued for output, for example:



The 'Home' tab has a 'Current plate' and 'Pending plates' panel, as described in the next two sections.

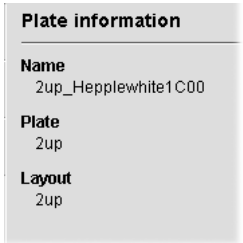
8.1.1 The 'Current plate' panel

The 'Current plate' section of the window contains the following sub-panels:

- **Plate information**: This displays information for the plate that is currently being processed.
- **Operation**: The displays the operation and status of the Python platesetter (or any other processing that is occurring).
- **Preview**: This shows a preview of the current plate.

Each of these is described in the following sub-sections.

'Plate information' panel



The 'Plate information' panel displays the following information:

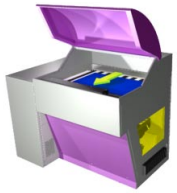
- **Name** — The plate image name (with the queue name prepended to it).
- **Plate** — This is the plate name as created on the Python Layout Tool's 'Plate' tab (see section 13.5 for more information about this).
- **Layout** — This is the layout name as created on the Python Layout Tool's 'Layout' tab (see section 13.4 for more information about this).

'Operation' panel

The 'Operation' panel shows the current status of the platesetter:

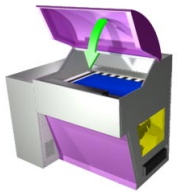
Ready

Ready
The Python engine is idle and ready to accept a new job, or the Console's processing status has been set to suspended (indicated by a flashing green/black icon in the top left-hand corner of the window).



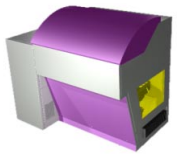
Load plate...

Load plate...
Python is waiting for the user to load a plate onto the tilt-table (as described in section 9.2).



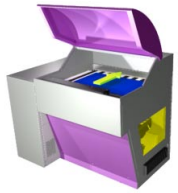
Close cover...

Close cover...
A plate has been successfully loaded and Python is waiting for its cover to be closed so that job output can start.



Busy...

Busy...
The plate is currently being imaged.

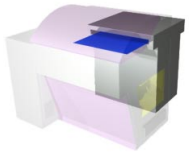


Unload plate...

Unload plate...
The plate has been imaged (or the plate was aborted). Python is waiting for the plate to be unloaded.

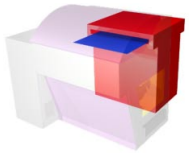
Plate badly registered

Plate badly registered
This error occurs if the plate becomes mis-registered (after you have correctly registered it). Refer to section 9.2 for details on re-loading the plate.



(Status message)

Bridge initialising... / Bridge ready / Plate unloading
These messages appear when the bridge (if you are using one) is operational after plate imaging has finished.



Check bridge

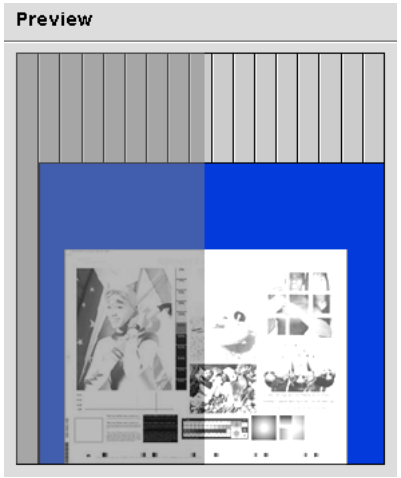
Check bridge
This message occurs if there is a problem with the bridge and it is unable to unload the current plate. Check that the plate is not stuck, the serial cable is connected and that the bridge is connected.

‘Preview’ panel

The ‘Preview’ panel shows one of the following two images:

Before imaging has started

Before imaging has started, the preview panel shows the plate that will be output to the platesetter next.



During imaging

Once imaging has started, the ‘Preview’ panel shows a preview of the job being imaged on the plate in the drum. The moving grey shaded area indicates how much of the plate has not yet been imaged (the grey area clears from right to left).

8.1.2 The ‘Pending plates’ panel

The ‘Pending plates’ panel in the bottom half of the window shows the next plates (up to three) that are waiting for output to the Python platesetter, for example:

The plate Name (with the queue name prepended to it), Plate and Layout information (as defined in the Python Layout Tool) are shown for the first pending plate in the queue.

The following buttons are available:

Open plate in viewer
This displays the highlighted plate in HWRoam. See section 7.9 for more details about using HWRoam.

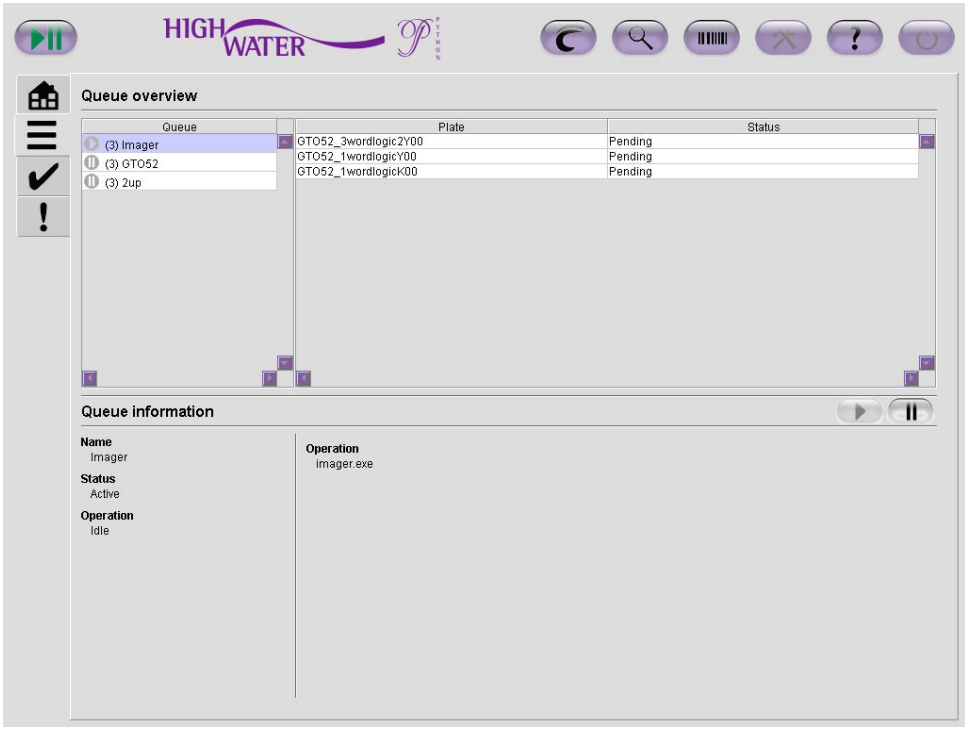
Activate queue
If the queue is suspended, click on the **Activate** button to re-start output to the Python platesetter.

Suspend queue
If the queue is active, click on the **Suspend** button to suspend plate output to the Python platesetter (this will not suspend a plate that is currently being output).

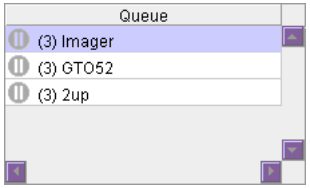
8.2 The 'Queue' tab

The 'Queue' tab lets you view all the jobs that are waiting in the queues for processing. You can delete jobs, and suspend and activate individual jobs and queues.

Click on the **Queue** tab ☰ to display the 'Queue overview' window:



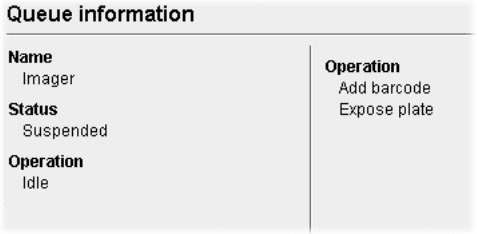
The 'Queue' panel lists all the queues that have been created, for example:



Notes: These queues were created using the Queue Configuration application, as described in chapter 14.

Jobs are moved from the TicketMaker queues to the Imager queue which outputs them to the platesetter.

In the bottom half of the window, in the 'Queue information' panel, you will see more information relating to the queue for example:



The 'Queue information' panel shows the following information:

- **Name** — The queue name (as created in the Queue Configuration application).
- **Status** — The queue's status—either suspended or active.
- **Operation** — The current operation that the queue is performing (or if it is idle).

The 'Operation' panel, to the right, lists all processing tasks that the queue will perform on all jobs in the queue.

The following buttons are available for each queue listed on the right-hand side of the window:

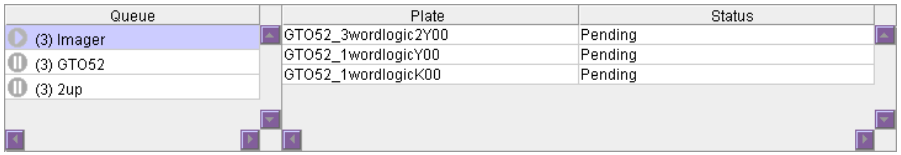
Note: These options are also available from a pop-up menu when you right-click on a plate name.

Activate queue
This activates a suspended queue (any jobs that have been individually suspended will stay suspended).

Note: Only one TicketMaker queue can be active at a time. Therefore, when you activate a TicketMaker queue, all other TicketMaker queues are automatically suspended. The Imager queue can be activated or suspended regardless of the other queues' states.

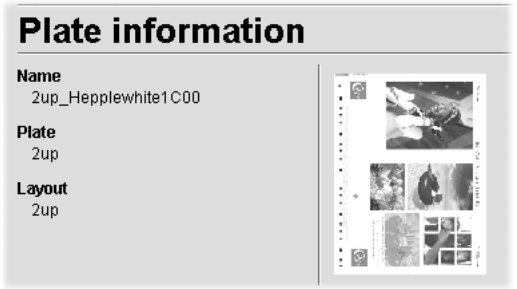
Suspend queue
This suspends an active queue.

Click on a queue name to view the plates in that queue, for example:



- The 'Plate' column shows the plate name (with the queue name prepended).
- The 'Status' column shows the status of the plate (pending, completed or suspended).

Click on a plate name to view plate information in the bottom half of the window, for example:



The 'Plate information' panel shows the following information:

- **Name** — The plate name (with the queue name prepended to it).
- **Plate** — This is the plate name as created on the Python Layout Tool's 'Plate' tab (see section 13.5 for more information about this).
- **Layout** — This is the layout name as created on the Python Layout Tool's 'Layout' tab (see section 13.4 for more information about this).
- A preview of the job (spot colours are shown in black).

The following buttons are available for the highlighted plate(s):

Note: These options are also available from a pop-up menu when you right-click on a plate name.

Open plate in viewer
This displays the plate image in HWRoam. (See section 7.9 for more details about using HWRoam.)

Note: This button is not available if more than one plate is selected.

Abort
Use this to abort a plate that is currently being processed. The plate will be moved to the 'Errored plates' tab. (See section 8.4 for more details about errored plates.)

Activate
Use this button to activate a suspended plate.

Note: If the queue itself is suspended, the status of the plate changes to Pending and the plate will not be processed until the queue itself is made active.

Suspend
Use this button to suspend an active plate.

Move down
This moves the selected plate(s) down one place in the queue (plates at the top of the queue get processed first).

Move up
This moves the selected plate(s) up one place in the queue (plates at the top of the queue get processed first).

Move to top of queue
This moves the selected plate(s) to the top of the queue (plates at the top of the queue get processed first).

Move to queue...
This moves the plate to the specified queue. (Select the required queue from the 'Move plate(s) to queue' window that appears.)

Delete
This deletes the selected plate(s). You will be asked to confirm the delete.

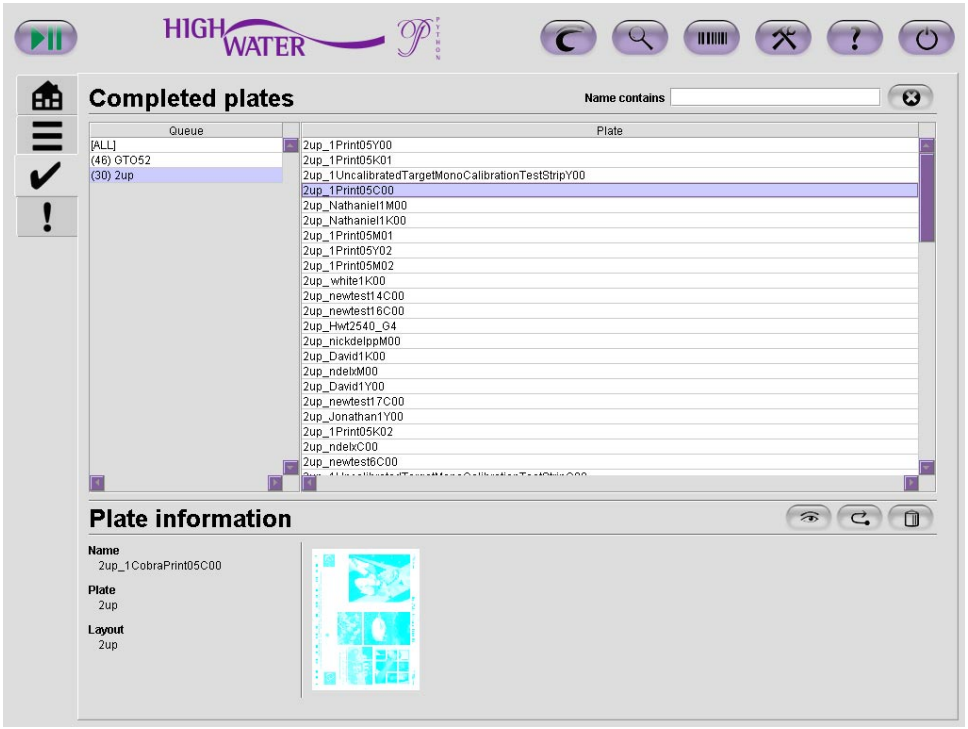
WARNING: A deleted plate is not retrievable in any way.

Note: For most actions, you can select more than one plate using the **Shift** and/or **Ctrl** keys.

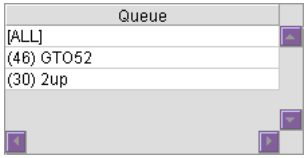
8.3 The 'Completed plates' tab

The Python Console's 'Completed plates' tab lets you view all the plates that have been successfully processed. You can view, requeue and delete plates, as described in this section.

Click on the **Completed plates** tab ✔ to display the following window:

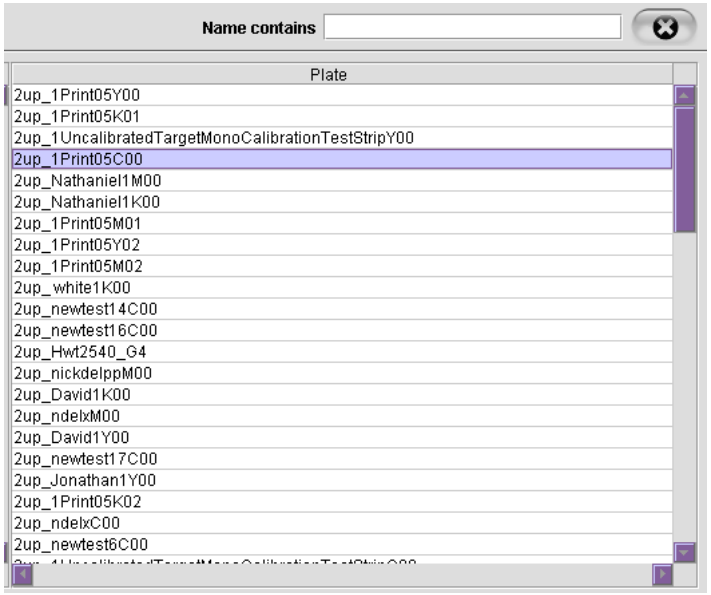


The 'Queue' panel lists all of the queues that have been created (except the Imager queue), for example:



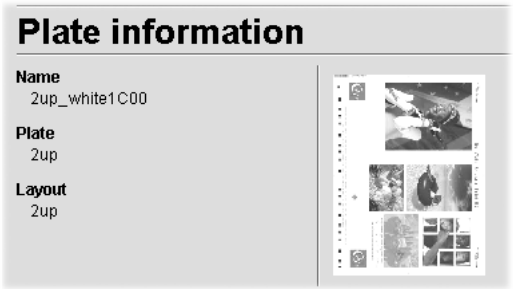
The number in brackets before the queue name shows how many completed jobs are in that queue.

Click on the **ALL** entry to view the completed plates in all queues, or click on a specific queue name to view completed plates for that queue only, for example:



To search for a specific plate/plates, use the **Name contains** field and either type all or part of the name into the text box. (Use the clear button  to clear the text box.)

Click on a plate name to view its plate information, for example:



The 'Plate information' panel displays the following information:

- **Name** — The plate image name (with the queue name prepended to it).
- **Plate** — This is the plate name as created on the Python Layout Tool's 'Plate' tab (see section 13.5 for more information about this).
- **Layout** — This is the layout name as created on the Python Layout Tool's 'Layout' tab (see section 13.4 for more information about this).
- A preview of the plate (spot colours are shown in black).

The following buttons are available for the highlighted plate(s):

Note: These options are also available from a pop-up menu when you right-click on a plate name.

Open plate in viewer
This displays the plate image in HWRoam. (See section 7.9 for more details about using HWRoam.)

Note: This button is not available if more than one plate is selected.

Requeue
This requeues the selected plates back to their original queue(s) for reprocessing.

Delete
This deletes the selected plate(s). You will be asked to confirm the delete.

WARNING: A deleted plate is not retrievable in any way.

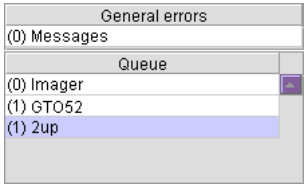
Note: For the **Requeue** and **Delete** actions you can select more than one plate using the **Shift** and/or **Ctrl** keys.

8.4 The 'Errored plates' tab

The Python Console's 'Errored plates' tab lets you view all the plates that have terminated with an error. When an error has occurred, the 'Errored plates' tab will display a flashing red icon:

Click on the **Errored plates** tab to display the following window:

The 'Queue' panel lists all of the queues that are available, for example:

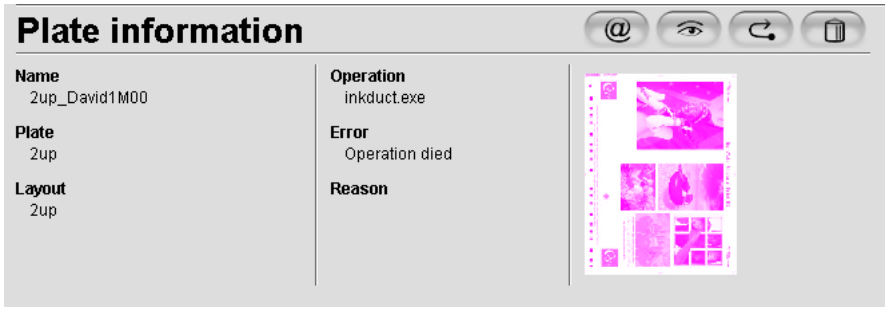


The number in brackets before the queue name shows how many errored plates are in that queue.

Click on the queue name to view the errored plates for that queue.

To search for a specific plate/plates, use the **Name contains** field and either type all or part of the name into the text box. (Use the clear button  to clear the text box.)

Click on a plate name to view the plate's information, for example:



The 'Plate information' panel displays the following information:

- **Name** — the plate image name (with the queue name prepended to it).

- **Plate** — This is the plate name as created in the Python Layout Tool's 'Plate' tab (see section 13.5 for more information about this).
- **Layout** — This is the layout name as created in the Python Layout Tool's 'Layout' tab (see section 13.4 for more information about this).
- **Operation** — This shows what processing stage the plate was at when the error occurred.
- **Error** — This shows the error that occurred (for example, user aborted plate).
- **Reason** — This shows the reason why the error occurred.
- A preview of the plate (spot colours are shown in black).

The following buttons are available for the highlighted plate(s):

Note: These options are also available from a pop-up menu when you right-click on a plate name.

Email log files
This emails log files to the email address(es) specified in the Python Console's Preferences (see section 7.8 for more details about this).

Open plate in viewer
This displays the plate image in HWRoam. (See section 7.9 for more details about HWRoam.)

Note: This button is not available if more than one plate is selected.

Requeue
This requeues the selected plates back to their original queue(s) for reprocessing.

Delete
This deletes the selected plate(s). You will be asked to confirm the delete.

WARNING: A deleted plate is not retrievable in any way.

Note: For the **Requeue** and **Delete** actions you can select more than one plate using the **Shift** and/or **Ctrl** keys.

9. Setting up the Python platesetter for output

This chapter shows you how to set up the Python platesetter so that it is ready to image a job. It includes the following sections:

Note: The Python platesetter can optionally use sound alerts during operation. Please refer to section 15.3 for details on changing these sound alerts.

9.1 Initial checks of the Python platesetter

Before loading a plate onto Python's tilt-table you should make the following checks:

- Ensure that there are no 'foreign' objects on the Python tilt-table (pens, cups, paper, etc.).
- Check that the Python tilt-table and surrounding areas are free from dust and particles. Refer to section 16.1 for details on cleaning the tilt-table.

WARNING: Anything left on the Python tilt-table will fall into the drum once imaging starts. This could damage the Python platesetter and/or the plate, or it could leave particles in the drum which may contaminate plates.

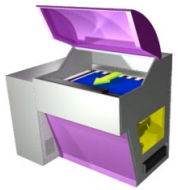
- Make sure the plate is the correct size for the job being output (the plate size should be between 335 x 400 mm minimum, and 745 x 615 mm maximum). See section 4.6 for more plate information.
- If the plate auto-unload option is set up on your Python platesetter then also:
  - Make sure the bridge and/or processor are powered on and ready to accept plates **before** imaging a plate on the Python.
  - Check that the previous plate is completely unloaded from the Python via the bridge/processor. If it is not, **do not** load another plate until the previous plate has cleared the Python platesetter completely.

9.2 Loading the plate

WARNING: When handling plates, make sure that you follow the safe working procedures covered in chapter 3 and the plate handling procedures in section 4.7.

Remove the interleave sheet from the plate before clamping, and do not punch the plate until after it has been imaged.

Before loading the plate on the tilt-table, make sure the Python Console is ready to output a job to the platesetter and is prompting you to load a plate:



Load plate...

To load a plate onto Python's tilt-table:

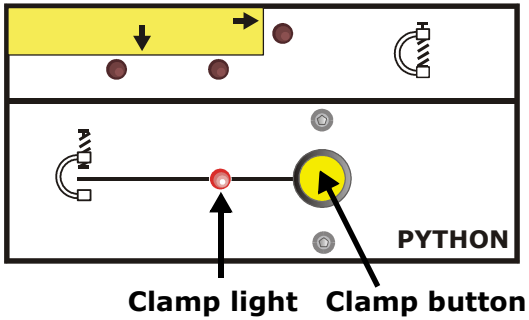
1. Switch on the brightroom's yellow light or other safety lighting and close the door.

   Note: You should ensure that no-one enters the room while you are loading/unloading plates.

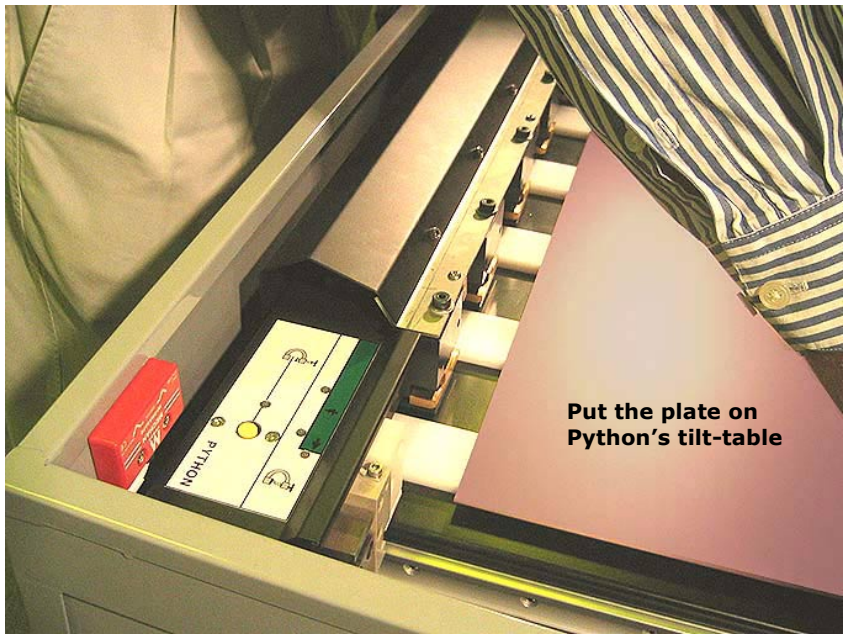
2. In the Python Console, make sure that the next job is ready for output.
3. When prompted to load a plate, open Python's lid and load the plate, as described in the following sub-section.

9.2.1 To load the plate

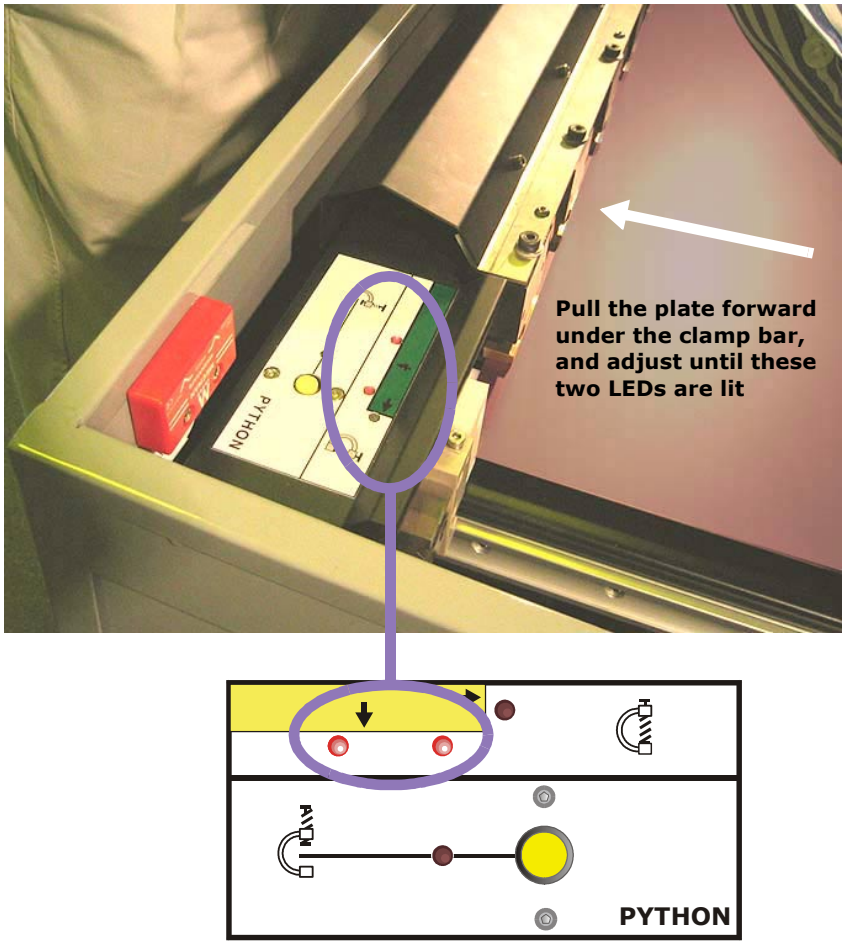
Note: Initially, if the clamp light is on (as shown below) and it does not go off within a few seconds, press the clamp button to switch off the clamp light. The clamp is now ready to accept a plate.



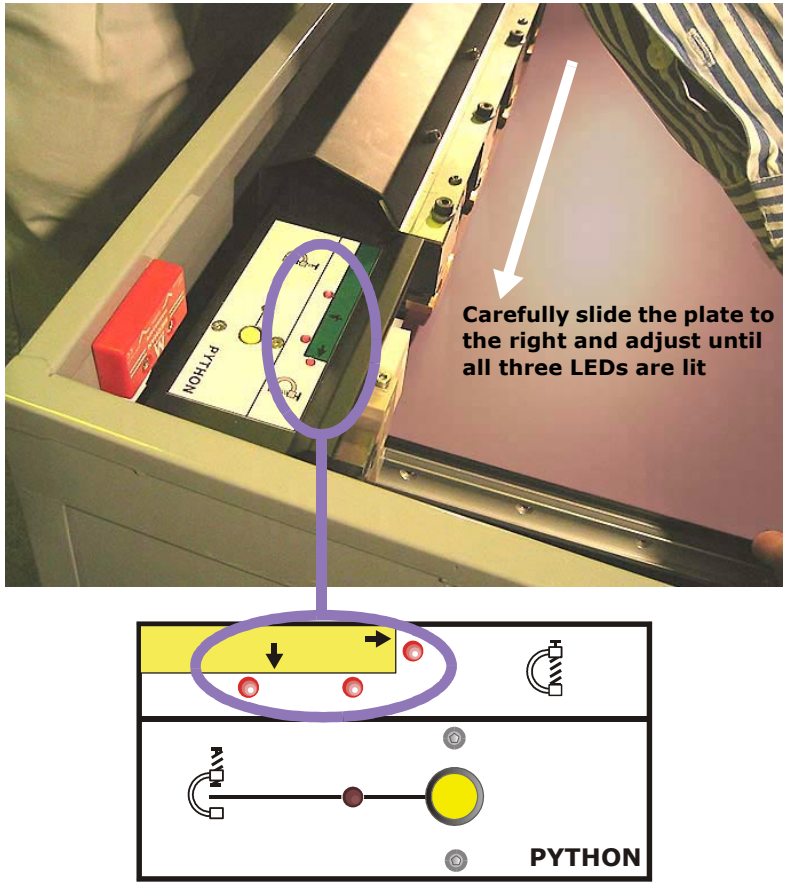
1. Put the plate, emulsion-side up, onto the tilt-table:



2. Pull the plate towards the front of the tilt-table until it contacts the two register pins under the clamp (the plate is in the correct position when the two left-most registration LEDs light up):



3. Carefully slide the plate sideways to the right until the plate makes contact with the third register pin.



Note: Try not to move the plate backwards as you slide it to the right.

4. When the plate is in the correct position the third LED will light. When all three LEDs are lit, the plate will automatically be clamped. Hold the plate securely in place until the clamp engages.

   Note: If any of the three LEDs go out while you are positioning the plate, carefully reposition the plate until all three LEDs are lit, then the clamp will automatically engage.

   Once clamped, check that all three LEDs are still lit. If not, follow the instructions in section 9.2.2 opposite.

5. When the plate is correctly clamped, close Python's lid. Now, Python will start its imaging cycle. The Python Console shows Python's progress as it images the plate.

6. When Python has finished imaging the plate, the Python Console will ask you to unload the plate.

   Unloading the plate manually is described in the next section.

   If you have the auto-unload feature, the plate will automatically be unloaded through the back of the Python onto a bridge to the processor.

9.2.2 Re-registering a plate

It takes a fraction of a second for the clamp to engage on the plate. During this short time, it is possible for the plate to move out of position slightly. In this case, one or more of the registration LEDs will not be lit after the clamp has engaged and you will see a 'Plate badly registered' error message in the Python Console:

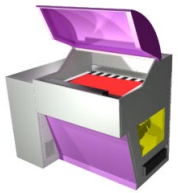


Plate badly registered

If this happens, the Python software will not image the plate until you have re-clamped the plate correctly. To do this:

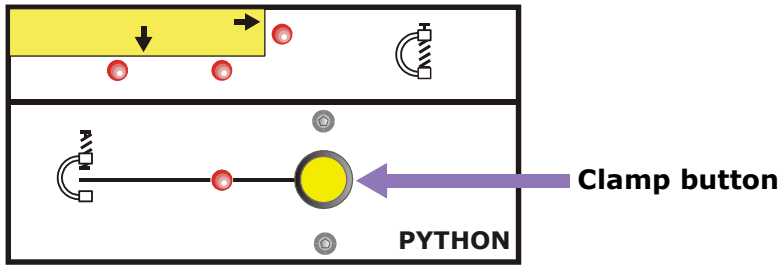
1. Press the clamp button to disengage the clamp from the plate.
2. Pull the plate out from under the clamp.
3. If the red LED next to the clamp button is on (and does not go off within a few seconds) press the clamp button again so that the light goes out.
4. Re-clamp the plate following the previous instructions.

9.3 Manually unloading the plate

Note: This section is only relevant to users whose Python platesetter does not automatically unload the plate onto a bridge.

Once the plate has been imaged, unload it as follows:

1. Make sure the plate will not be exposed to any light source that might fog it.
2. Open Python's lid.
3. Press the clamp button to disengage the clamp:



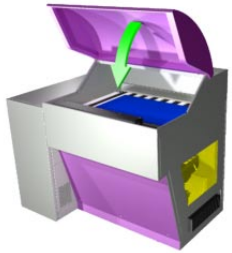
4. Carefully remove the plate from the tilt-table.
5. Process the plate.

9.4 If Python's lid is opened during the imaging cycle

WARNING: Unless there is an emergency, you should never open Python's lid to stop job output. If you wish to abort the currently outputting job, then click the button next to the Progress bar on the Python Console.

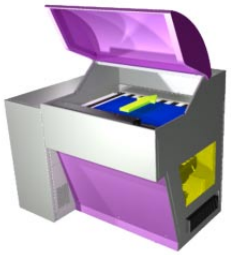
If Python's lid is opened during the imaging cycle:

1. Python's carriage will stop moving immediately. If the laser is operating, it will switch off immediately so that you cannot be exposed to it.
2. On the Python Console you will be asked to close the Python platesetter's lid:



Close cover...

3. When you have closed the lid, Python's carriage will return to its default position (to the left-hand side of the drum) and you will see a message on the Python Console asking you to unload the plate:



Unload plate...

4. Remove the plate and discard it.

   Note: The plate will be moved to the 'Errored plates' tab in the Python Console, from where it can be requeued for re-output, if required.

5. You can now continue with normal plate output.

10. Overview of creating a new workflow

When the Python system was installed, the engineer set it up to work with your plates and other job settings. However, if you want to introduce a new plate size or type, or change other settings, you need to create a new 'workflow' through the system. This workflow gives the system all the information needed to correctly route, process and output jobs to the Python platesetter.

This chapter includes the following sections:

Note: Chapters 11–14 show you how to create a new workflow.

10.1 When you need to create a new workflow

You need to create a new workflow in your system when you introduce either of the following:

- A new plate size or type.
- A new job position on the plate.

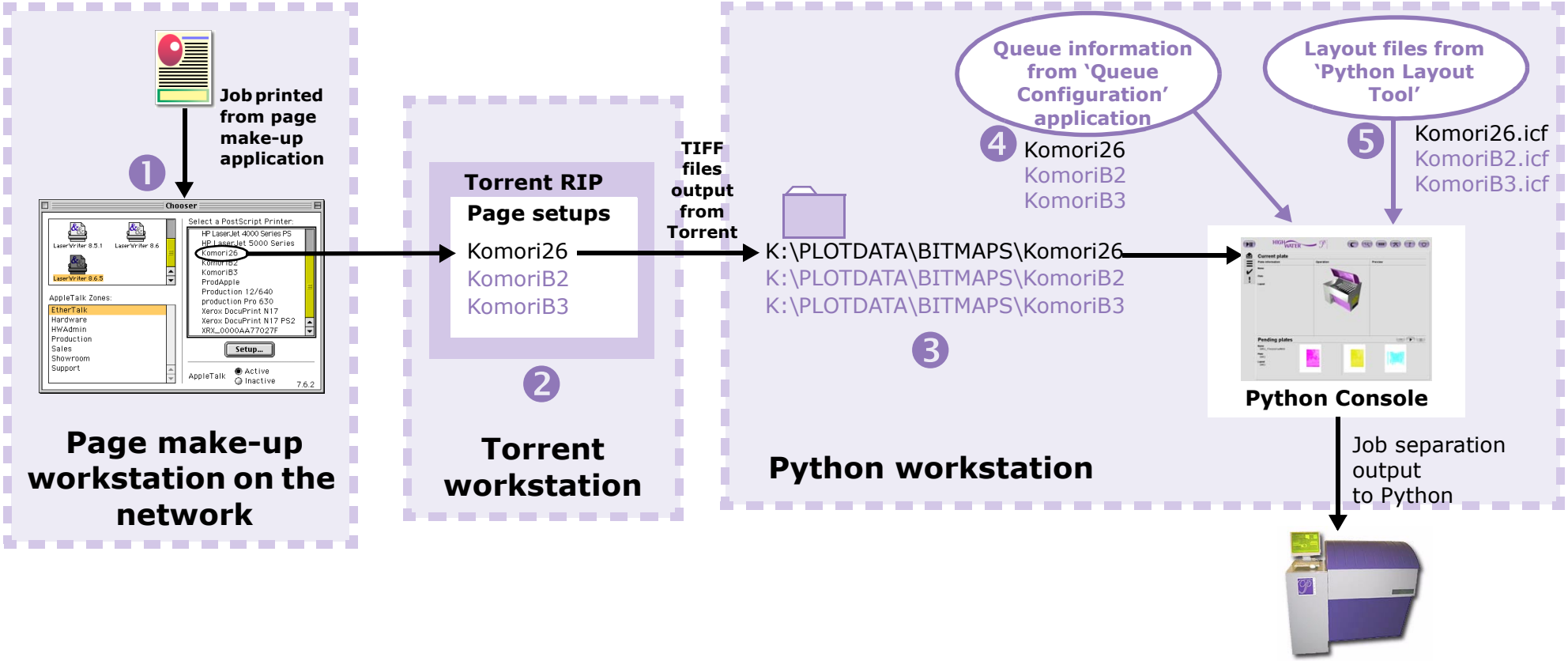
10.2 The elements that make up a workflow

The whole process of job output, from page make-up right through to final output to the Python platesetter, consists of processing 'elements' that include software applications and parameter files. The diagram overleaf shows how each of these elements makes up the workflow:

- First, you print the job to a Torrent **input queue** from a workstation on the network.
- Torrent RIPs the job and applies the settings specified in the Torrent **page setup** (resolution, page size, job rotation, etc.).
- Torrent outputs the job (as bitmap TIFF file separations) to the **directory** specified in the Torrent page setup.
- The Python Console looks in this directory for jobs to process, and places any job it finds there into the appropriate **queue**.

  The Python Console processes jobs in the queues and then moves them to the Imager queue which outputs them to the Python platesetter, using the parameters stored in the associated **Layout file** (created in the Python Layout Tool) and **queue information** (from the Queue Configuration application).

Note: The items shown in bold above are the 'elements' that make up the workflow. You need to create these elements, as described in the following chapters.



Workflow elements

1	2	3	4	5
Torrent input queue	**Torrent page setup**	**Bitmaps directory**	**Queue information**	**Layout file**
You print a job to the Torrent input queue from a page make-up workstation on the network	Jobs are processed by Torrent using the settings in the associated page setup	The resulting TIFF bitmap files are placed in a sub-directory of K:\PLOTDATA\BITMAPS on the Python workstation	The Python Console gets the queue names and other processing details from the Queue Configuration application	The Python Console gets the plate and job information from the appropriate Layout file (created in the Python Layout Tool)

10.3 Naming workflow elements

Before you create the new workflow you should choose a common name for all the separate workflow elements. Giving all workflow elements the same name makes it easier to:

- Create, edit and delete workflows.
- Track jobs.
- Track errors.
- Maintain your system (for file deletion, etc.).

Using the press name is the recommended way of naming your workflows, for example, **SM74**. However, the workflow name may need to be more specific: for example, if you have more than one workflow for the same press. To avoid confusion, make sure the name you choose clearly identifies the workflow. For example, the names **SM74_centred** and **SM74_default** are more meaningful than **SM74_1** and **SM74_2**.

11. Creating a Torrent page setup and input queue

The first stage in creating a workflow is to create a page setup and input queue in the Torrent RIP. The page setup defines the settings (resolution, page size, etc.) to be applied to the jobs sent to Torrent. The input queue lets you print jobs to Torrent from across the network.

This chapter shows you how to create a Torrent page setup and input queue, and it includes the following sections:

11.1 Before you begin

Before you start to work through this chapter please bear the following points in mind:

- Choose one unique name for the workflow, as discussed in section 10.3. You will use this name for both the page setup and input queue (and also for the Layout file and queue that you will create in the following chapters). In this example, we will call the new workflow **SM74**.
- Before you create the new page setup, create the directory where the TIFF files are to be saved. This should be a sub-directory of K:\PLOTDATA\BITMAPS\ on the Python workstation. The sub-directory should have the same name as the workflow you are creating, for example, K:\PLOTDATA\BITMAPS\SM74.

WARNING: If you have a two-computer Python system, this directory MUST be created on the Python workstation, not on the Torrent RIP workstation.

- You should be familiar with using the Torrent RIP. Chapter 6 gives an overview of the Torrent RIP and the **Torrent User's Guide** gives full details on setting up and using the RIP.
- This chapter shows you how to set the Python-specific settings in the Torrent page setup. For more information about other settings, such as trapping and colour management, please refer to the **Torrent User's Guide** (and/or other documentation that was supplied with any Torrent options/plugins you are using).

11.2 Launching the Torrent RIP

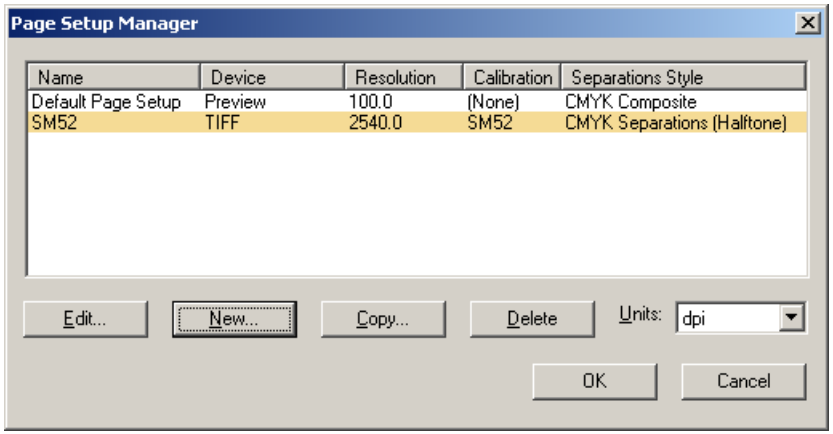
Launch the Torrent RIP by clicking on the Torrent RIP icon  in the top right-hand corner of the Python Console.

Note: If the Torrent RIP resides on a separate workstation, launch it by selecting **Start > Programs > HighWater Designs > Torrent vx.x > Torrent vx.x**.

11.3 The Page Setup Manager

In Torrent, you create a page setup using the Page Setup Manager. To display the Page Setup Manager:

1. If necessary, stop the input queues by selecting **Start Inputs** in the **Torrent** menu (so that it does not have a tick by it).
2. Either choose **Page Setup Manager...** from the **Torrent** menu or click on the **Page Setup Manager** button on the toolbar. The following window is displayed:



3. The Page Setup Manager lists all the page setups that have been created. For each one, it shows the page setup's name, output device, resolution, calibration and separations style.

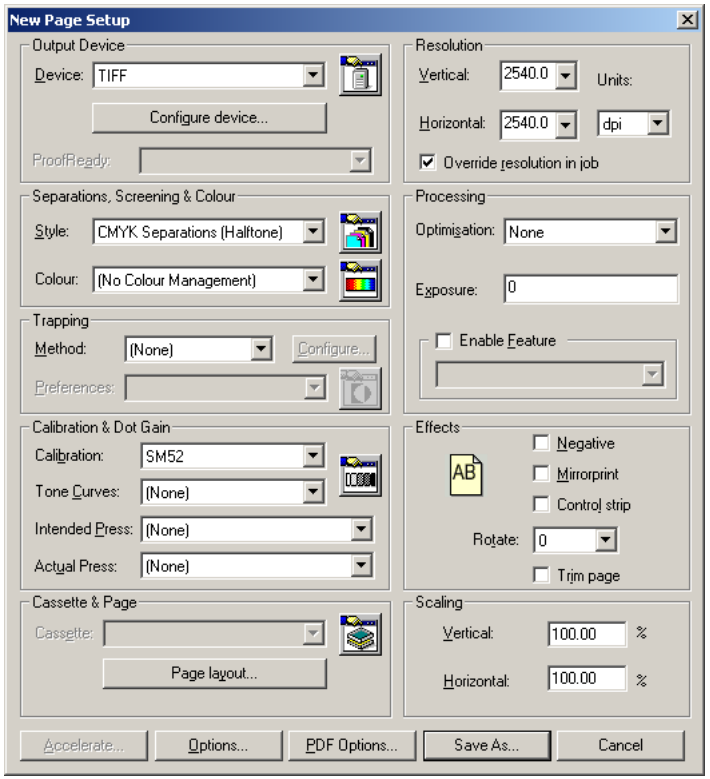
The Page Setup Manager lets you edit, create, copy and delete page setups. Creating a new page setup is described in the following section.

11.4 Creating a new page setup

The quickest way to create a new page setup is to copy an existing one and edit the details. To do this, in the Page Setup Manager highlight an existing page setup and click on the **Copy...** button.

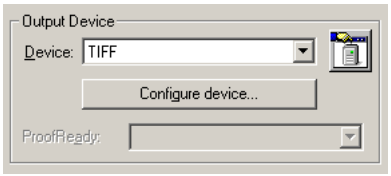
Note: If possible, copy a page setup that is similar to the one you will be creating.

The 'New Page Setup' dialog is displayed:



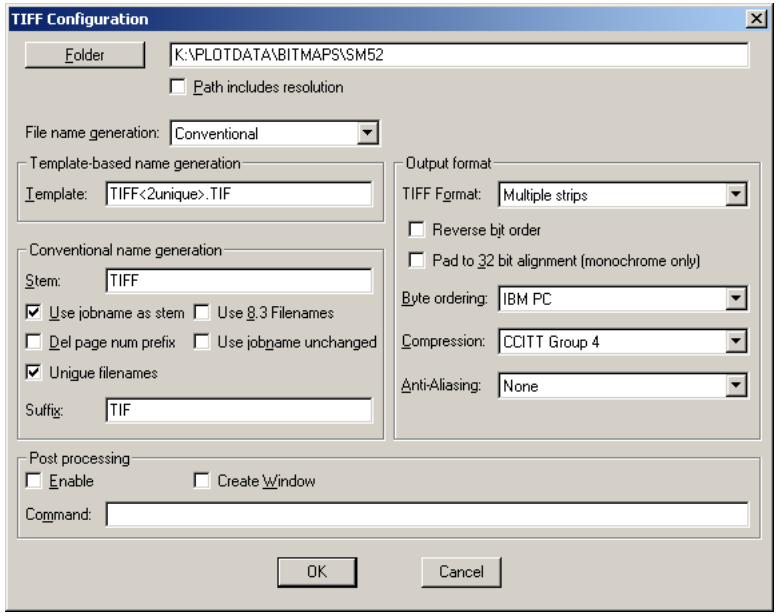
This dialog contains all the details of the page setup you copied. The dialog is divided into a number of panels that allow you to specify the settings to be applied to your jobs. Each of the panels is described in the following sections.

Output Device



The 'Output Device' panel specifies where your output will be sent. For use with the Python system, the output will be to TIFF files, which will later be output to the Python platesetter. To set this:

1. From the **Device** pull-down list, select **TIFF**.
2. Click on the **Configure device...** button to display the 'TIFF Configuration' dialog:



3. To tell Torrent where to output the TIFF files, click on the **Folder** button. The 'Select Folder' dialog is displayed.

   Locate and highlight the **K:\PLOTDATA\BITMAPS*sub-dir*** folder that you created on the Python workstation (as described in section 11.1) then click on **OK**. The location of the TIFF files is now shown alongside the Folder button.

   Note: *sub-dir* is the new workflow directory you created (in this example, **SM74**).

4. Make sure that all other fields are set as shown in the 'TIFF Configuration' dialog shown previously:
   - The **Path includes resolution** box is unchecked.
   - **File name generation** is set to **Conventional**.
   - The **Template** field is not required.

   In the 'Conventional name generation' panel:
   - The **Stem** is set to **TIFF**.
   - The **Use jobname as stem** box is checked.
   - The **Use 8.3 Filenames** box is unchecked.
   - The **Del page num prefix** box is unchecked.
   - The **Use jobname unchanged** box is unchecked.
   - The **Unique filenames** box is checked.
   - **Suffix** is set to **TIF**.

   In the 'Output format' panel:
   - **TIFF Format** is set to **Multiple strips**.
   - The **Reverse bit order** box is unchecked.
   - The **Pad to 32 bit alignment (monochrome only)** box is unchecked.
   - **Byte ordering** is set to **IBM PC**.
   - **Compression** is set to **CCITT Group 4**.
   - **Anti-Aliasing** is set to **None**.
   - Do not select any options in the 'Post processing' panel.
5. Click on **OK** to close the 'TIFF Configuration' dialog.

You have now finished setting the TIFF options.

Separations, Screening & Colour

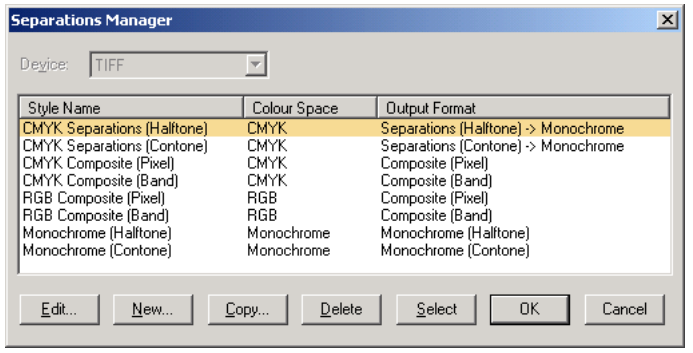
You use this panel to set the separations and screening options.

Notes: HighWater recommends the following settings but if you have different requirements then change the settings, as appropriate.

You only need to create a new separations style for Python once (unless you need to create additional separations styles) — it will then be available for all page setups.

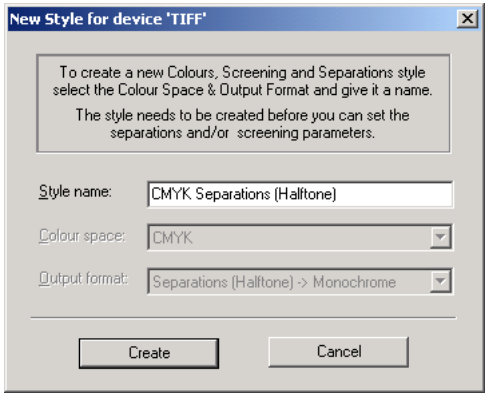
To set the separations and screening options:

1. Click on the **Separations Manager** button ( ) to display the following dialog:



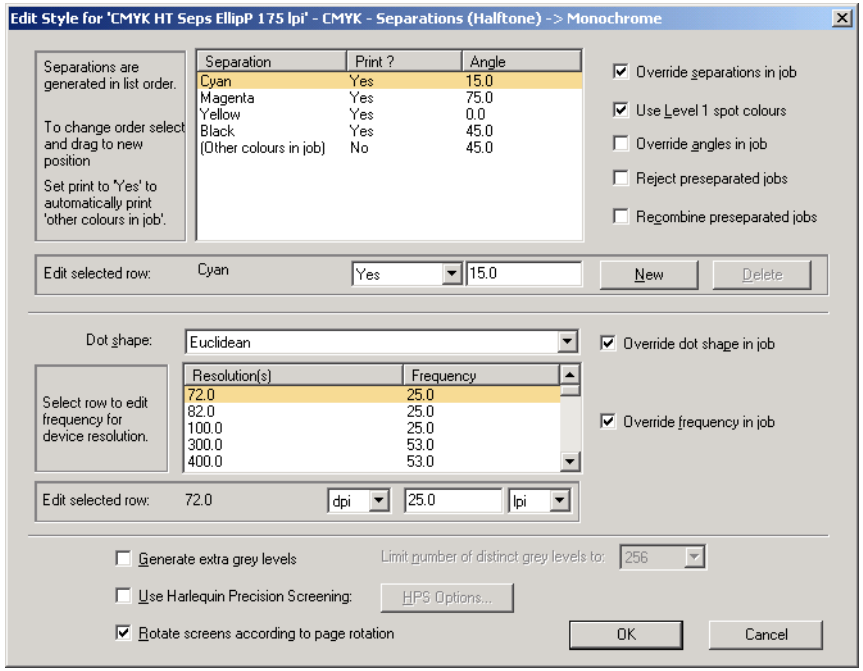
2. If there is already a separations style for use with Python then highlight it and click on the **Select** button. Otherwise, to create a new separations style for use with Python, continue with the instructions in this section.

3. Highlight the **CMYK Separations (Halftone)** entry then click on the **Copy...** button. The following dialog is displayed:



4. Type in a new name for the separations style, for example, **CMYK HT Seps EllipP 175 lpi**.

5. Click on the **Create** button. The following dialog is displayed:



6. From the **Dot shape** pull-down menu, select **EllipticalP**.

7. In the top window, highlight the **Black** separation.

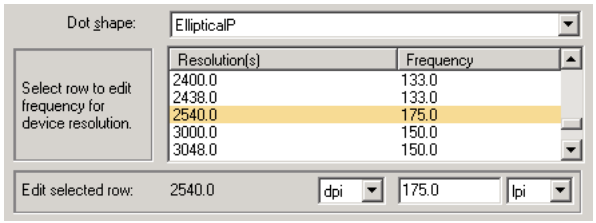
8. In the **Edit selected row** box type **135** for the new angle:

9. If there are other colours in the job to be printed (other than cyan, magenta, yellow and black), highlight **(Other Colours in job)** in

the top window. In the 'Edit selected row' panel, select **Yes** from the first pull-down menu and change the angle to **135**:

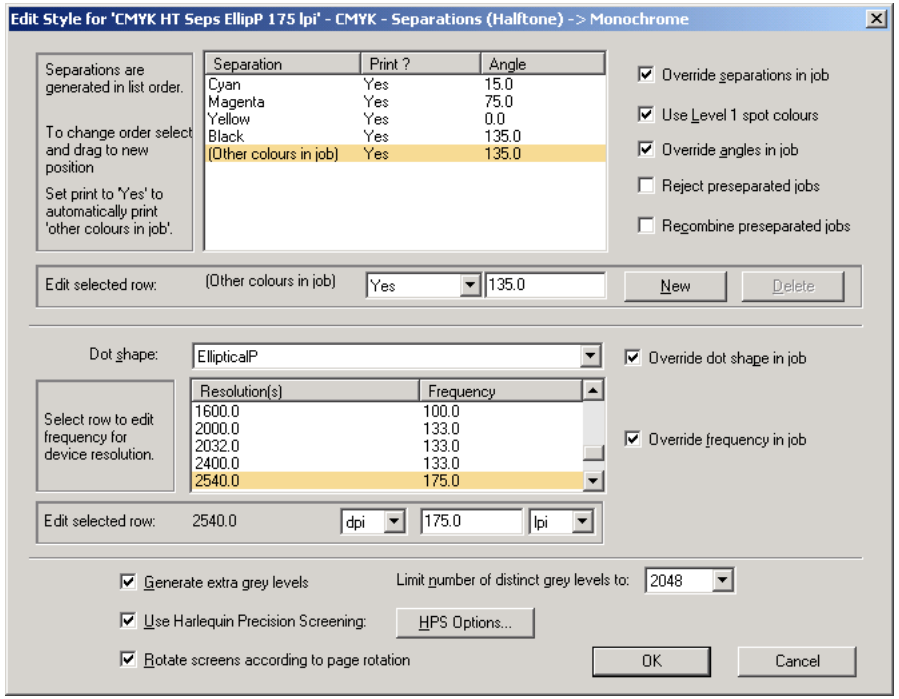
Note: Leave the **Cyan**, **Magenta** and **Yellow** angles at 15°, 75° and 0°, respectively.

10. Select the **Override separations in job**, **Use Level 1 spot colours**, **Override angles in job**, **Reject preseparated jobs** and **Recombine preseparated jobs** options, as required.
11. Check the **Override dot shape in job** box.
12. Check the **Override frequency in job** box.
13. In the **Resolution(s)** and **Frequency** list, scroll down to **2540** resolution and change the frequency to the required value (between **175** and **200** lpi) in the 'Edit selected row' panel:

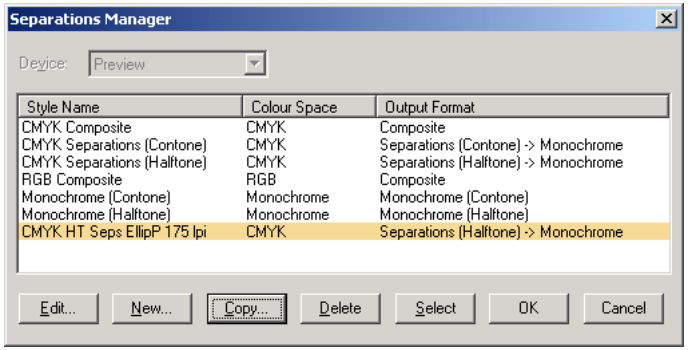


Note: We recommend that you do **not** set a frequency of over 200 lpi.

14. Check the **Generate extra grey levels** box near the bottom of the dialog, then select **2048** from the **Limit distinct number of grey levels to** list.
15. Check the **Use Harlequin Precision Screening** box.
16. You have now finished creating the new separations style. The finished dialog should look similar to this:



17. Click on **OK** to close the dialog. You will now see the new separations style in the Separations Manager:

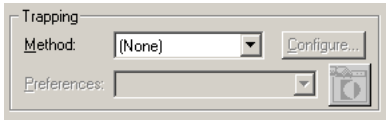


18. Make sure the new separations style is highlighted, then click on the **Select** button to choose this style and return to the 'New Page Setup' dialog.

WARNING: You must recalibrate the page setup after changing the separations style settings. Please refer to chapter 12 for full details on calibration.

Note: Refer to the **Torrent User's Guide** for more details about setting the colour management options, if required.

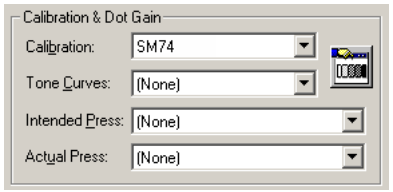
Trapping



Select the required trapping option, if any.

Note: Please refer to the **Torrent User's Guide** or other supplied documentation for more information about trapping.

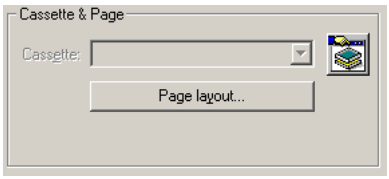
Calibration & Dot Gain



This panel lets you set the calibration and dot gain settings. Continue with the remaining instructions in this chapter first then go to chapter 12 to calibrate the page setup after you have finished creating it.

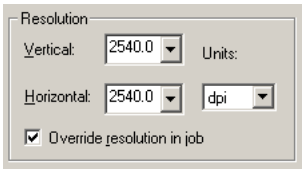
WARNING: You MUST calibrate your Torrent page setups to ensure quality of output.

Cassette & Page



The 'Cassette and Page' panel is not required for use with the Python system.

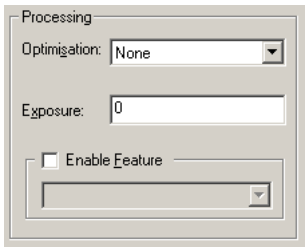
Resolution



To specify the resolution:

1. Select **dpi** from the **Units** pull-down list, then set the **Vertical** and **Horizontal** resolution values to **2540**.
2. Make sure the **Override resolution in job** box is checked.

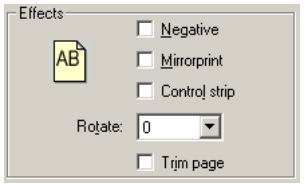
Processing



To set any required processing options:

1. The **Optimisation** and **Exposure** fields are not required.
2. To specify a feature, check the **Enable Feature** box then select a feature from the pull-down menu. Refer to the **Torrent User's Guide** for more information about using features.

Effects



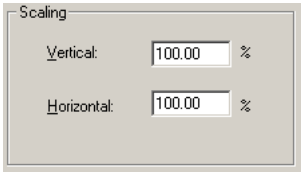
The following effects are available:

- **Negative** outputs the job in negative.
- **Mirrorprint** produces a mirror image (reflected in the vertical axis).
- The **Control strip** option is **not** suitable for use with the Python system so do not select it.

- You can **Rotate** the job by **0**, **90**, **180** or **270** degrees.
- **Trim page** removes white space at the top and bottom of the job.

  Note: You can see the effects of selecting the **Rotation**, **Negative** and **Mirrorprint** options in the small page icon  .

Scaling



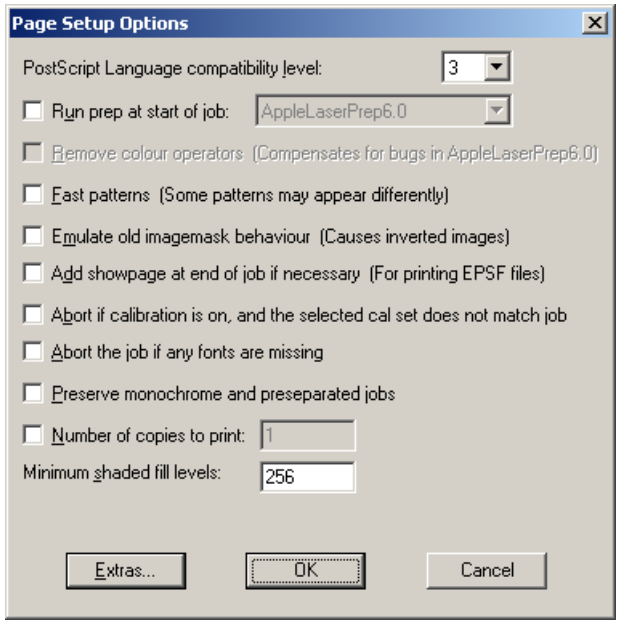
To set the scaling:

1. Set the **Horizontal** and **Vertical** scaling to **100**%.

Options...

To set the page setup options:

1. Click on the **Options...** button at the bottom of the 'New Page Setup' dialog to display the following dialog:

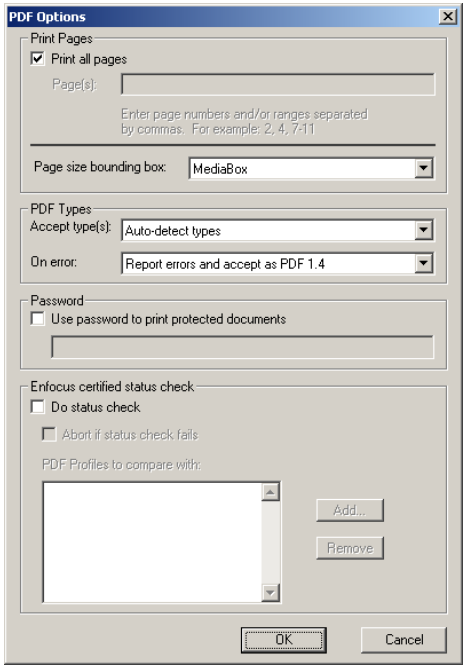


2. Set the fields, as required (refer to the **Torrent User's Guide** for more information about the available options).
3. When you have finished, click on **OK** to close the 'Page Setup Options' dialog.

PDF Options...

To set the PDF options:

1. Click on the **PDF Options...** button to display the following dialog:

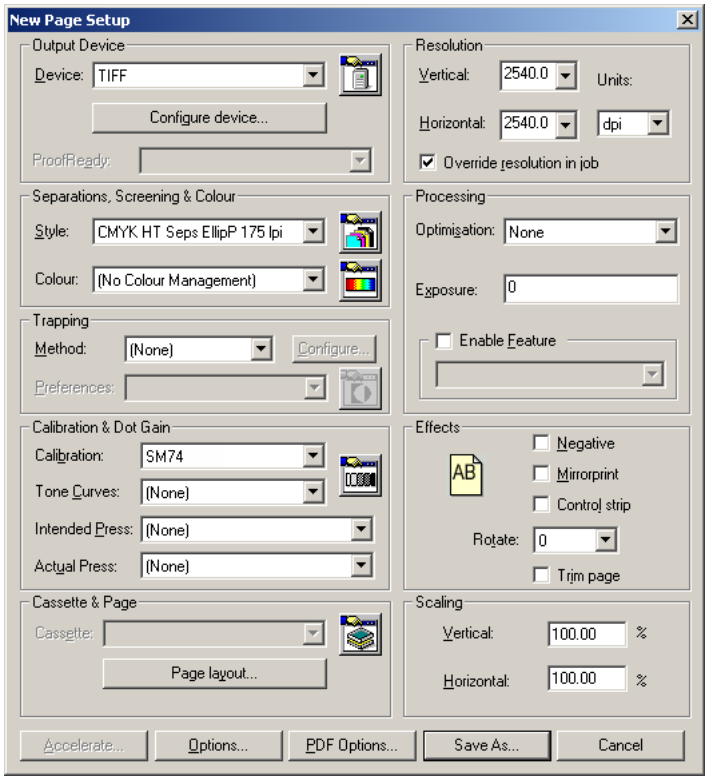


2. Set the fields, as required (refer to the **Torrent User's Guide** for more information about the available options).
3. When you have finished, click on **OK** to return to the 'New Page Setup' dialog.

You have now finished creating the page setup. Saving it is covered in the next section.

11.5 Saving the page setup

When you have finished creating the new page setup, the dialog will look like this:

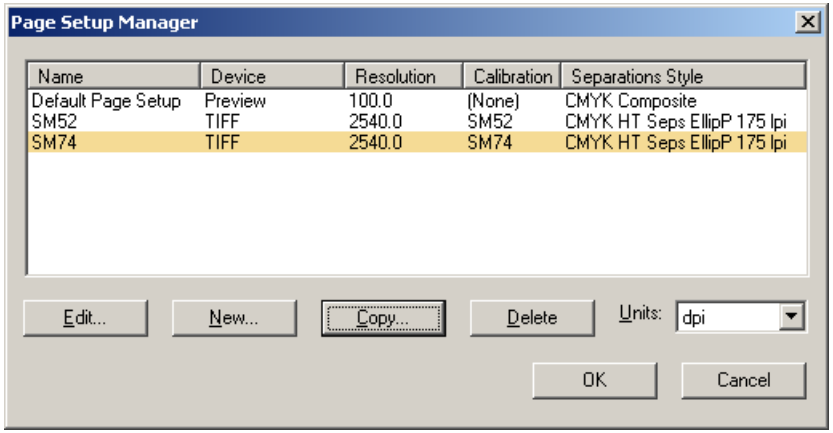


To save it:

1. Click on the **Save As...** button, type a name into the **Save As** box, then click on the **Save** button.

   Note: You should give the page setup the same name as the workflow, for example, **SM74**.

2. The page setup you have just created now appears in the Page Setup Manager, for example:



3. Click on **OK** to close this window.

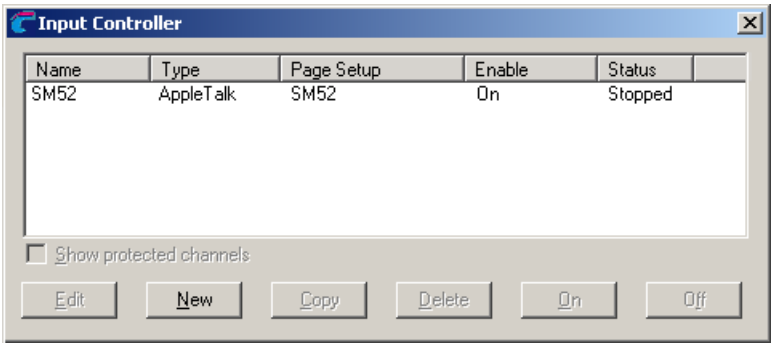
WARNING: You MUST close the Page Setup Manager by clicking the OK button, otherwise any changes you have made to the page setups will be lost.

Now that you have created a page setup, you need to associate it with an input queue which will allow you to send jobs from your page make-up workstation to Torrent. Creating an input queue is covered next.

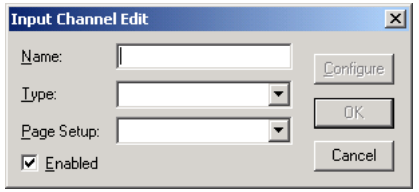
11.6 Creating a new input queue

To create an input queue for the page setup you have just created:

1. Select **Input Controller...** from the **Torrent** menu to display the 'Input Controller' window:



2. Click on the **New** button. The 'Input Channel Edit' dialog is displayed:



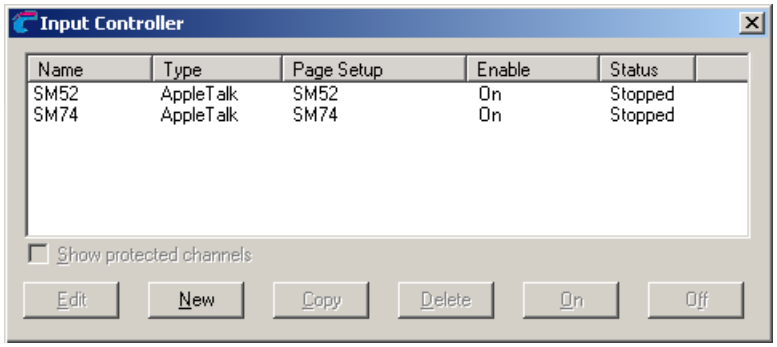
3. Select the **Page Setup** you have just created from the pull-down list.
4. Type in a **Name** for the new input queue (this should be the workflow name, for example, **SM74**).
5. Select **AppleTalk** from the **Type** pull-down list.

   Note: It is possible to send jobs to Torrent other than via AppleTalk. Please refer to the **Torrent User's Guide** for further details about this.

6. Make sure the **Enabled** box is checked:

Note: If you do not set **Enabled**, the input queue will be marked **Off** in the Input Controller, and you will not see it on any workstations after you have broadcast your input queues.

7. Click on **OK** to close the 'Input Channel Edit' dialog. In the Input Controller you will see the new input queue you have created:



8. Close the 'Input Controller' window.

11.7 When you have finished

To check that the new Torrent page setup and input queue you have created are working properly:

1. If you haven't done so already, copy the appropriate Python PPD to a workstation you are printing from and create a Python print queue (chapter 19 shows you how to do this).
2. Test the input queue and page setup by printing a job to Torrent from the page make-up workstation (see section 6.6 for more information about printing a job to Torrent).
3. Use the Torrent RIP's Roam feature to check that the job has output correctly (see section 6.5.2 for more information about this).
4. Check that the TIFF bitmap separations are in the correct output directory on the Python workstation (K:\PLOTDATA\BITMAPS*sub-dir*).

When you have checked that the new input queue and page setup are working properly you should calibrate the page setup, as described next in chapter 12.

WARNING: You must go through the calibration procedure before printing jobs to Torrent for anything other than test purposes.

Creating a Layout file to associate with a Python Console queue is covered in chapter 13.

12. Calibrating Torrent page setups

Every Torrent page setup needs to be calibrated to ensure that the Python platesetter produces the correct dot densities for the jobs it outputs. This chapter describes how to calibrate a Torrent page setup and includes the following sections:

Note: Also refer to the **Torrent User's Guide** for more information about calibrating page setups.

WARNING: You must go through the calibration procedure described in this chapter before you produce output for anything other than test purposes.

12.1 Before you begin

Before you start working through this chapter, make sure that:

- You have created the required page setup.
- You have a suitable plate densitometer.
- You have suitable plates and a processor.

12.2 Introduction to calibration

Calibration settings need to be created for each Torrent page setup that you use. The calibration process is summarized below and step-by-step instructions are included in the following sections.

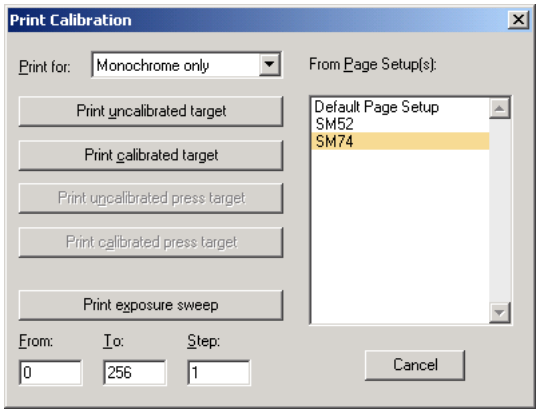
1. Select the required Torrent page setup and print out an uncalibrated 'target' on the Python platesetter.
2. Use a densitometer to read the densities off the plate.
3. Use Torrent's Calibration Manager to create new calibration settings for the page setup. Enter the values from the densitometer and save them.
4. Select the page setup again, attach the new calibration settings to it and save it.
5. Print out the calibrated target to check the new settings and refine the calibration, if necessary.

Note: Although the engineer who installed your Python system calibrated the Torrent page setups that were initially created, you should check the calibration from time to time. You will also need to calibrate any new page setups you create.

12.3 Printing an uncalibrated target

The first step in calibrating a page setup is to print an uncalibrated target. To do this:

1. In Torrent, deselect **Start Inputs...** on the **Torrent** menu (so that there is no tick by it).
2. Select **Print Calibration** from the **Output** menu to display the following dialog:

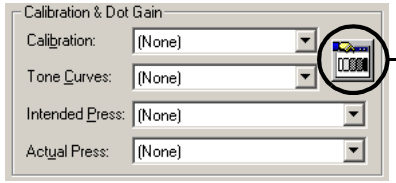


3. From the **Print for** pull-down menu, select **Monochrome only**.
4. Select the page setup you wish to calibrate from the list in the **From Page Setup(s)** column.
5. Click on the **Print uncalibrated target** button. This produces a test strip which you should output to the Python platesetter.
6. When you have output the job to plate and processed it, use an appropriate densitometer to measure the values on the test strip. Make a note of the values you measure.

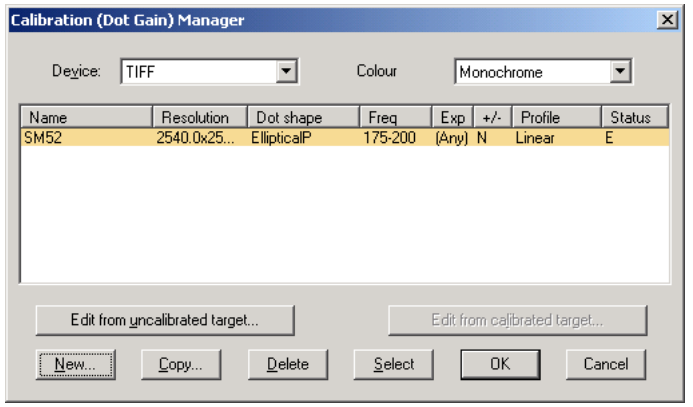
12.4 Using the Calibration Manager

Once you have output the test strip to plate and read the values, you need to create new calibration settings in the Calibration Manager. To do this:

1. Select **Page Setup Manager...** from the **Torrent** menu to open the Page Setup Manager.
2. Select the page setup you are calibrating and click on the **Edit...** button to open the 'Edit Page Setup [name]' dialog.
3. In the 'Calibration & Dot Gain' panel, click on the **Calibration (Dot Gain) Manager** button:

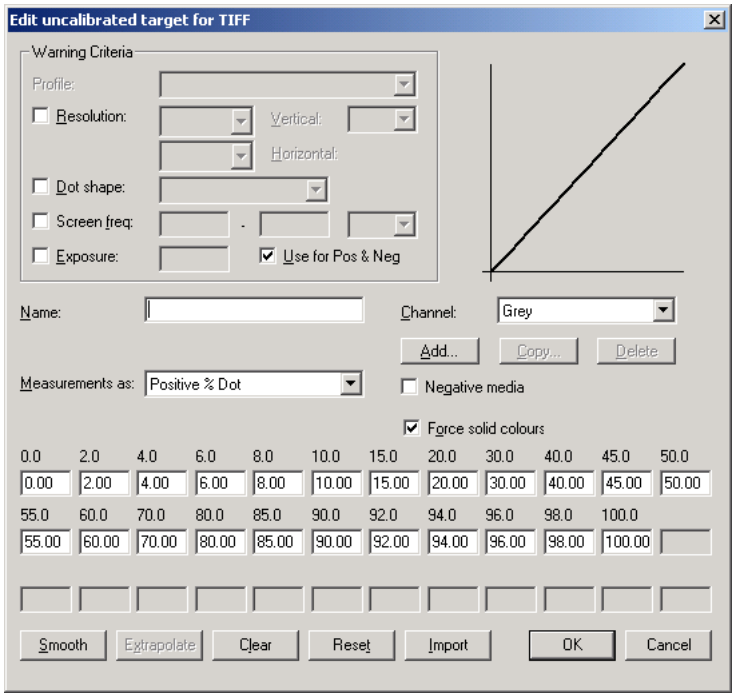


4. The 'Calibration (Dot Gain) Manager' dialog is displayed:



This window lists any calibration sets that have already been created.

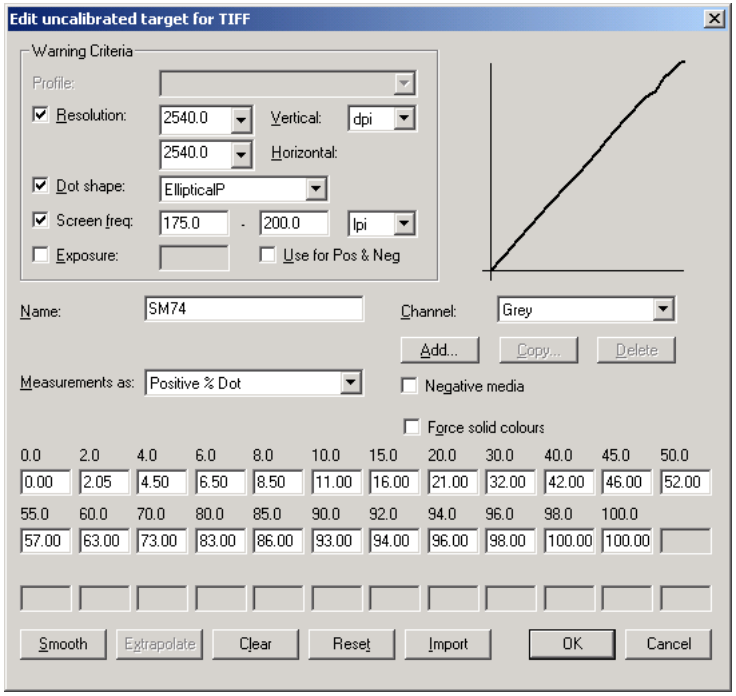
5. From the **Device** pull-down menu at the top of the dialog, select **TIFF**.
6. From the **Colour** pull-down menu, select **Monochrome**.
7. Click on **New...** to display the 'Edit uncalibrated target for TIFF' dialog:



8. In the **Name** field, give the calibration set a name based on the one for the page setup/workflow (for example, **SM74**).
9. In the 'Warning Criteria' panel, enter values for **Resolution**, **Dot shape** and **Screen Freq** that match those of the page setup you are calibrating (these should be **2540** dpi, **EllipticalP** and **175** to **200** lpi, respectively).

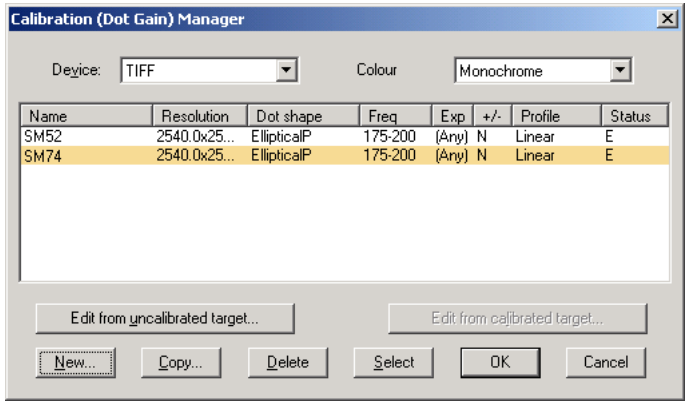
   Note: The calibration is normally valid for all screen frequencies, but you can enter a range for the **Screen freq.**, if required, and select the appropriate units of measurement.
10. Leave the **Exposure** box unchecked.
11. Make sure that **Use for Pos & Neg** is unchecked. This is handled by the Python software.
12. Select the appropriate value from the **Measurements as** pull-down menu, depending on the type of densitometer that you are using.
13. The **Force solid colors** box should be left unchecked.
14. In the boxes marked **0.0** to **100.0** type in the values that you measured from your test strip.

15. When you have finished, the dialog will look similar to this:

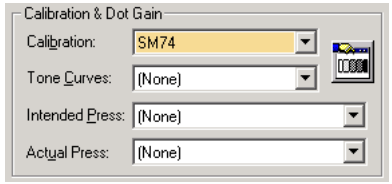


16. Click on **OK** to save the calibration settings.

17. In the Calibration (Dot Gain) Manager, make sure the new calibration set is highlighted and click on **Select**.



18. You are returned to the 'Edit Page Setup [name]' dialog. The new calibration set is now displayed in the **Calibration** field:



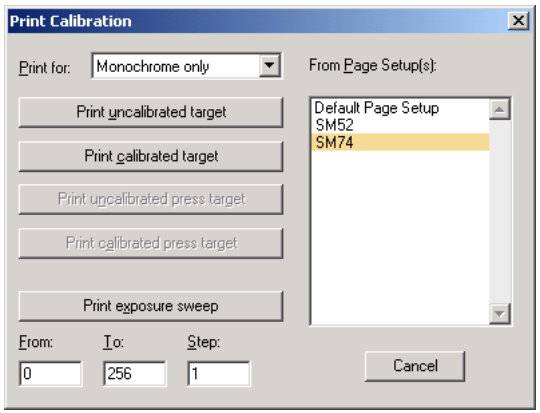
19. Click on **OK** to return to the Page Setup Manager, and on **OK** to exit it.

WARNING: Make sure that you close the Page Setup Manager by clicking the OK button, otherwise any changes you have made to the page setups will be lost.

12.5 Refining the calibration

We recommend that you repeat the calibration process at least one more time in order to verify that the calibration is correct and to provide further refinement to it. To do this:

1. In Torrent, select **Print Calibration...** from the **Output** menu to display the following dialog:



2. From the **Print for** pull-down menu, select **Monochrome only**.
3. Select the required page setup from the **From Page Setup(s)** column.

   Note: This is the same page setup that you used in the previous section.
4. Click on the **Print calibrated target** button. This produces a test strip which you should output to Python.
5. When you have output the job to plate and processed it, use an appropriate densitometer to measure the values on the test strip. Make a note of the values you measure.
6. If the measured values compare acceptably with the test strip values then no further action is required. However, if the measured values are significantly different, continue with the instructions in the rest of this section.
7. Select **Page Setup Manager...** from the **Torrent** menu to open the 'Page Setup Manager'.
8. Select the page setup you are calibrating and click on the **Edit...** button to open the 'Edit Page Setup [name]' dialog.
9. In the 'Calibration & Dot Gain' panel, click on the **Calibration (Dot Gain) Manager** button. The 'Calibration (Dot Gain) Manager' is displayed.
10. From the **Device** pull-down menu at the top of the dialog, select **TIFF**.
11. From the **Colour** pull-down menu, select **Monochrome**.
12. Highlight the required calibration and click on the **Edit from calibrated target...** button.
13. In the boxes marked **0.0** to **100.0** type in the values that you measured from your test strip.
14. When you have finished, click on **OK** to save the calibration settings.
15. In the 'Calibration (Dot Gain) Manager', make sure that the calibration set is highlighted, and click on **Select**.
16. You are returned to the 'Edit Page Setup [name]' dialog.
17. Click on **OK** to return to the Page Setup Manager and on **OK** to exit it.

WARNING: Make sure that you close the 'Page Setup Manager' by clicking the OK button, otherwise any changes you have made to the page setups will be lost.

During calibration, the most important rule is to match the edit command to the type of target, so:

- When entering measurements using uncalibrated test strips, always do so with **Edit from uncalibrated target** (or **New**).
- When using calibrated test strips, always do so using **Edit from calibrated target**.

WARNING: Calibration settings will be incorrect if you do not follow this rule.

12.6 Re-calibrating a page setup

You may need to re-calibrate a page setup when, for example, you change inks, processing chemicals or toner, or you change the paper stock. If the page setup has not been changed and you are using its existing calibration set, the process is straightforward:

1. In Torrent, select **Print Calibration...** from the **Output** menu.
2. From the **Print for** pull-down menu, select **Monochrome only**.
3. Select the required page setup from the **From Page Setup(s)** column.
4. Click on the **Print calibrated target** button. This produces a test strip which you should output to Python.

   Note: If preferred, you can print out the original, uncalibrated job. Click on the **Print uncalibrated target** button.
5. When you have output the job to plate and processed it, use an appropriate densitometer to measure the values on the test strip. Make a note of the values you measure.
6. If the measured values are significantly different, select **Calibration Manager...** from the **Output** menu to display the 'Calibration (Dot Gain) Manager'.
7. From the **Device** pull-down menu at the top of the dialog, select **TIFF**.
8. From the **Colour** pull-down menu, select **Monochrome**.
9. Highlight the required calibration and click on the **Edit from calibrated target...** button.

   Note: If you printed the uncalibrated target, click on the **Edit from uncalibrated target** button.
10. In the boxes marked **0.0** to **100.0** type in the values that you measured from your test strip.
11. When you have finished, click on **OK** to save the calibration settings.
12. In the 'Calibration (Dot Gain) Manager', make sure that the calibration set is highlighted and click on **Select**.
13. You are returned to the 'Edit Page Setup [name]' dialog.
14. Click on **OK** to return to the Page Setup Manager and on **OK** to exit it.

    WARNING: Make sure that you close the Page Setup Manager by clicking the OK button, otherwise any changes you have made to the page setups will be lost.
15. Select **Print calibration...** from the **Output** menu.
16. Select the required page setup from the **Page Setup(s)** column.
17. From the **Print for** pull-down menu, select **Monochrome only**.
18. Click on the **Print calibrated target** button. This produces a test strip, which you should output to Python.
19. When you have output the job to plate and processed it, use an appropriate densitometer to measure the values on the test strip. Make a note of the values you measure.

20. If you are satisfied with the values, then the re-calibration process is complete.

    If you are **not** satisfied with the values, then refine this calibration further by following the instructions in section 12.5 again.

13. Creating a Layout file

After creating the Torrent page setup and input queue you need to create a Layout file (also known as an Imager Configuration File, or ICF file) using the Python Layout Tool. Layout files contain the plate and job position information that the Python Console needs when outputting your jobs to the Python platesetter (and also information required for InkMonitor, if you are using this application with your Python system).

This chapter includes the following sections:

13.1 Before you begin

Before you create a new Layout file, you need to determine the correct laser power for the plate that will be used. This is covered in chapter 18 and you should do this now if you have not done so already.

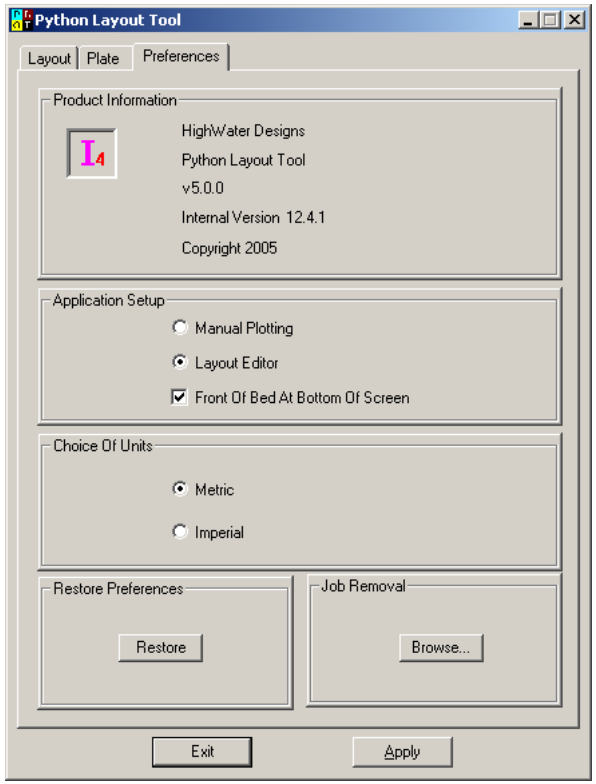
13.2 Launching the Python Layout Tool

Launch the Python Layout Tool, as described in section 7.7.

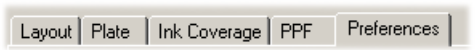
13.3 Setting the preferences

First, you need to tell the Python Layout Tool that you are creating a Layout file. To do this:

1. Click on the **Preferences** tab to display the following dialog:



Note: If the InkMonitor software is installed on your system, you will see more tabs on this dialog:

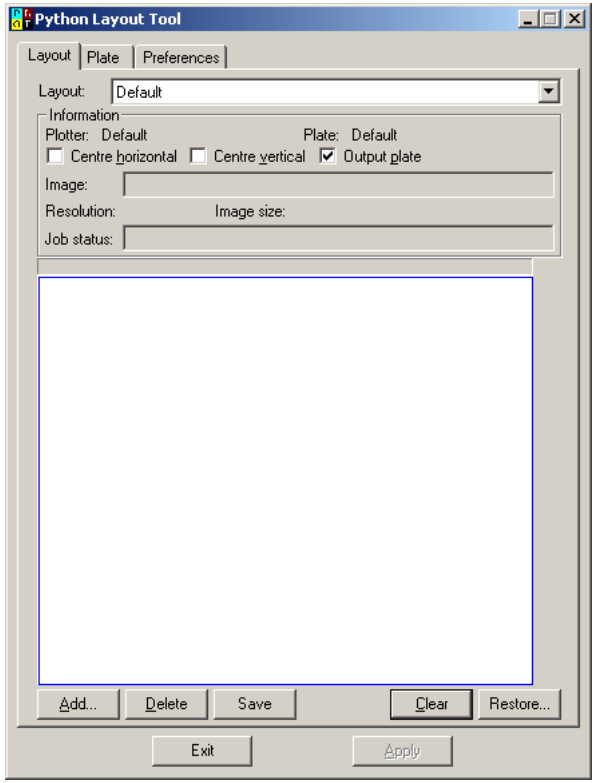


2. In the 'Application Setup' panel, click on the **Layout Editor** radio button.
3. Check the **Front Of Bed At Bottom Of Screen** box.
4. In the 'Choice of Units' panel, select either **Metric** to display measurements in millimetres, or **Imperial** to display measurements in inches.
5. Click on the **Apply** button to apply any changes.

13.4 Creating a new layout

Next, you will use the 'Layout' pane to create a new layout, which specifies the position of the job on the plate and other settings. To do this:

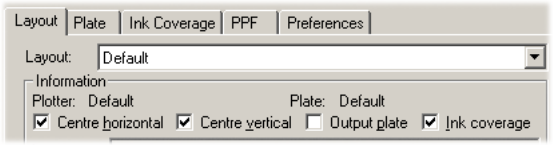
1. Click on the **Layout** tab to display the following pane:



Note: If InkMonitor is installed on your system, this dialog will have extra options, as described in step 4 below.

2. Click on the **Add...** button and type in a new name for the layout (this should be the same as the workflow name, for example, **SM74**) then click on **OK**.
3. In the 'Information' panel, click on the **Centre horizontal** and/or **Centre vertical** boxes, if you wish to centre the image on the plate.
4. **InkMonitor users only:**

   If you are using InkMonitor, the information panel will look like this:



   - Check the **Ink coverage** box if you want to generate ink duct settings.
   - Do not check the **Output plate** box.

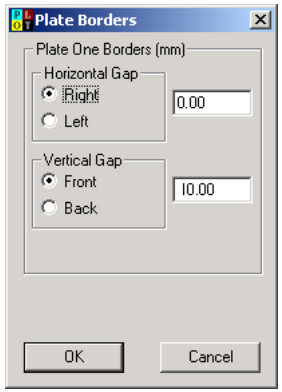
13.4.1 Setting the plate borders

Note: If your system is set up to use image output clipping (as configured by the installation engineer) then go straight to section 13.4.2 now for details on setting the plate borders.

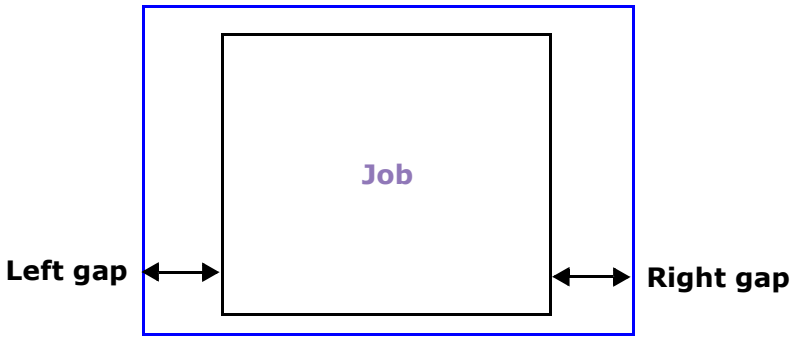
The plate borders settings are applied in addition to the **Centre horizontal** and **Centre vertical** settings to give you more flexibility when positioning the image on the plate.

To set the plate borders:

1. Right-click the mouse in the white plate layout area and select **Plate Borders...** from the pop-up menu. The following dialog appears:

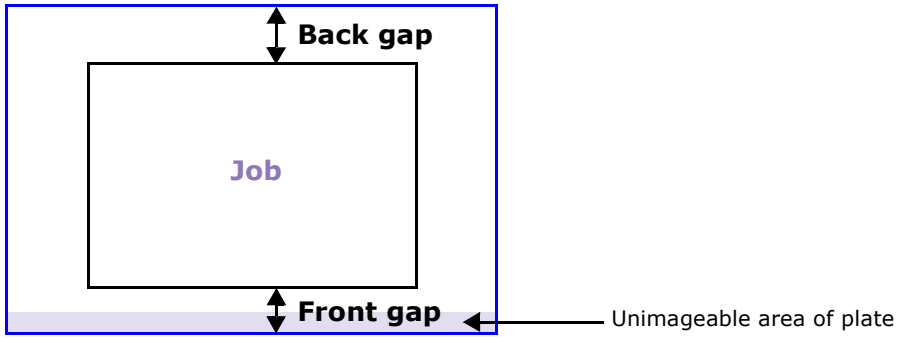


2. The 'Horizontal Gap' setting specifies the amount of space between the right or left edge of the plate and the image. Set the **Right** or **Left** gap (you cannot set both), as required.



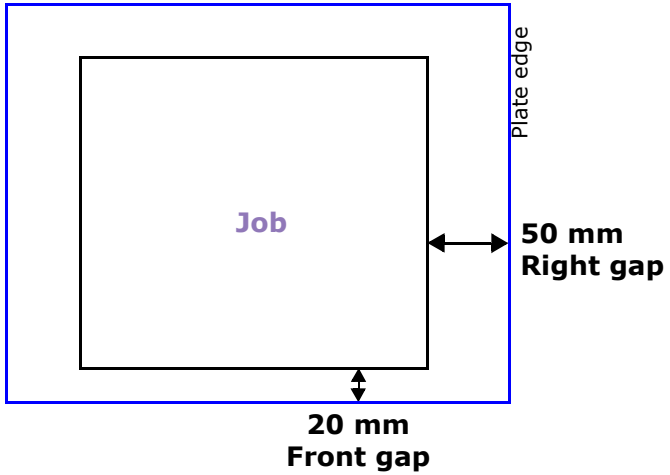
Note: Python can image right up to the edges of the plate in the horizontal direction so it is not absolutely necessary to set a horizontal gap.

3. The 'Vertical Gap' setting specifies the amount of space between the top or bottom edge of the plate and the image. Set the **Front** or **Back** gap (you cannot set both), as required.



Note: Whether you set the **Front** or **Back** gap, remember that part of the plate is unimageable (typically, 10-15 mm at the bottom of the plate) so the image must not overlap this area.

Example: If you set a **Right** gap of **50 mm** and a **Front** gap of **20 mm**, the image will be positioned thus on the plate:

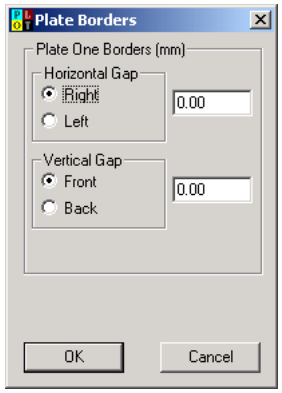


4. When you have finished, click on **OK** to close the 'Plate Borders' dialog.
5. On the 'Layout' pane, click on the **Apply** button.
6. Now, go to section 13.5.

13.4.2 Setting the plate borders (with image output clipping enabled)

If your system is set up to use image output clipping (as configured by the installation engineer) and your jobs are plate sized, as they would normally be from your imposition package, then you must set the plate borders, as follows:

1. Right-click the mouse in the white plate layout area and select **Plate Borders...** from the pop-up menu. The following dialog appears:



2. Make sure the horizontal and vertical gaps are both set to **0**.

WARNING: Ensure that the job image is plate sized, otherwise the position of the image on the plate may not be as expected.

If you set a horizontal or vertical gap greater than 0, you will see an error message in the Python Console that the job will not fit on the plate.

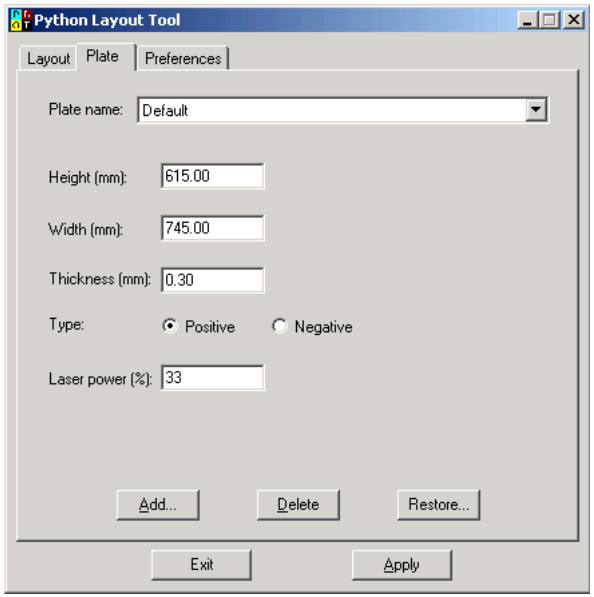
Note: With the horizontal and vertical gaps set to 0, the position of the image on the plate is determined by your desktop publishing or imposition software, rather than the Python Layout Tool.

3. When you have finished, click on **OK** to close the 'Plate Borders' dialog.
4. On the 'Layout' pane, click on the **Apply** button.

13.5 Selecting a plate

To select a plate for this layout:

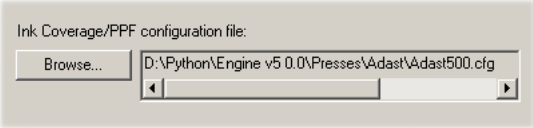
1. Click on the **Plate** tab to display the 'Plate' pane:



2. Select the required **Plate name** from the pull-down list.

Note: If the plate you require isn't displayed in the **Plate name** list then you need to create a new one. See section 13.6 for details on how to do this.

3. **InkMonitor users only:** On the 'Plate' pane, you also need to select an ink coverage or PPF configuration file. Click on the **Browse...** button to locate and select the required file:



Note: Please refer to the **InkMonitor User's Guide** for more information about creating and using configuration files.

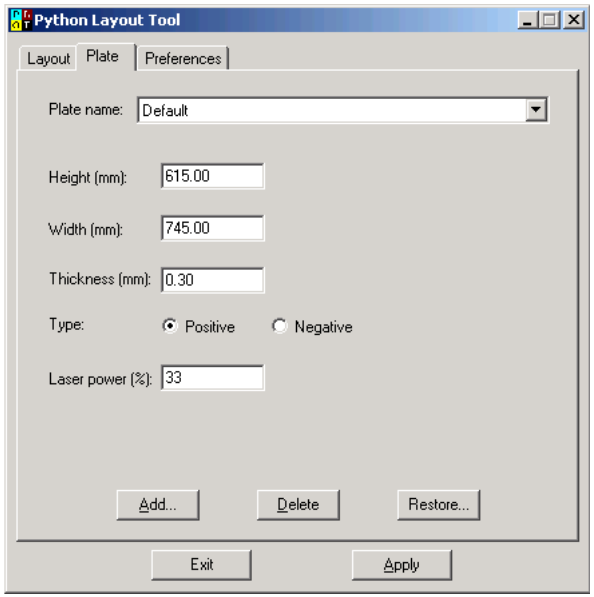
4. **All users:** Click on the **Apply** button.

You have now finished creating a new layout. Next, you need to save it as a Layout file, as described in section 13.7.

13.6 Creating a new plate definition

To create a new plate definition:

1. Click on the **Plate** tab to display the following dialog:



Note: If InkMonitor is installed with your Python system, this dialog will have extra options.

2. Click on the **Add...** button, type a new name for the plate then click on **OK**.
3. Type in the **Height** and **Width** of the plate.
4. Type in the **Thickness** of the plate (usually **0.3** or **0.15** mm).
5. Select **Positive** or **Negative** for the plate **Type**.

6. Type in a value for the **Laser power**.

   Note: Instructions for determining the laser power can be found in chapter 18.

7. When you have finished, click on the **Apply** button to apply the changes.

   Note: The plate you select here will be applied to the currently selected layout on the 'Layout' tab. Therefore, make sure that you select the correct plate for the currently selected layout, then click on the **Apply** button.

You have now finished creating a new plate definition.

13.7 Saving the Layout file

When you have selected the required plate and layout you need to save the information as a Layout file. To do this:

1. Click on the **Layout** tab and check that all settings are correct.
2. Click on the **Save** button to display the 'Save As' dialog.
3. If necessary, navigate to the default ICF directory (usually D:\Python\Engine v5.x.x), then type a name for the Layout file (this should be based on the workflow name, for example, **SM74.icf**).
4. Click on the **Save** button. The Layout file is saved.
5. When you have finished using the Python Layout Tool, quit out of it by clicking the **Exit** button.

When you create the new Python Console queue (using the Queue Configuration application), you will associate it with this Layout file. This is covered in the next chapter.

Note: If you have created a new layout, plate definition or Layout file, it is a good idea to save this new information (as described in section 16.2) when you have finished creating the whole workflow.

14. Creating a TicketMaker queue

The final stage of creating the new workflow is to create a TicketMaker queue. You do this using the Queue Configuration application, as described in this chapter.

This chapter includes the following sections:

Note: This chapter also includes instructions for setting up an InkMonitor queue.

14.1 Before you begin

Before you follow the instructions in this chapter:

- Create the required Layout file(s) using the Python Layout Tool (refer to chapter 13 for instructions).
- Make sure that the Imager and Completed queues have already been created, as described in appendices A and B.
- If you wish to add a barcode to the plates output in this queue then refer to section C.3 for instructions on how to do this once you have finished creating the new queue.

14.2 An overview of creating a new queue

You use the Queue Configuration application to create a new Python Console queue. For each queue you will specify the following information:

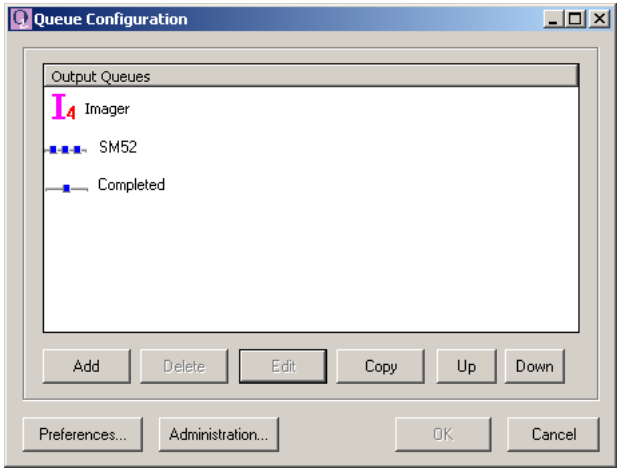
Name	The queue's name (the new queue will have the workflow name).
Inputs	Where the Python Console looks for jobs to process. This is the directory where the Torrent RIP saves the bitmap TIFF files.
Thumbnail images	This specifies where the job's thumbnail images can be found.
Job settings	These specify settings such as the job's priority.
Processes to run	This specifies which application(s) should process your jobs.
Process instructions	This specifies the Layout file (created in the previous chapter), which contains information about the plate size, job position, etc.
Clean-up	These options specify what happens to the job after it has been processed.
Resource management	These settings ensure that only one job can be output to the Python platesetter at a time.

Each different combination of these settings requires a new queue to be created.

14.3 Launching the Queue Configuration application

To launch the Queue Configuration application:

1. Follow the instructions in section 7.7. The 'Queue Configuration' window opens:



This window lists all the queues that you see in the Python Console (as well as the 'Completed' queue, which forms the 'Completed' tab in the Python Console).

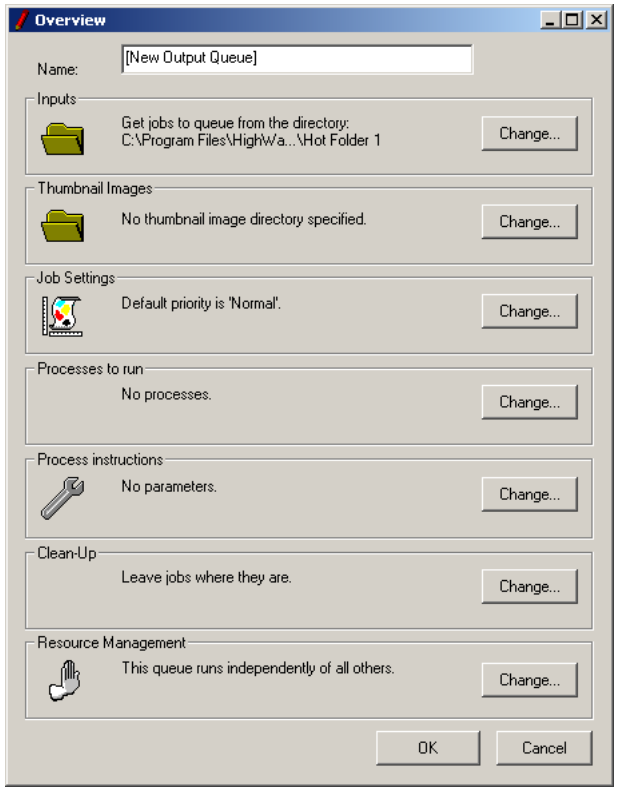
WARNING: If there are no 'Imager' and 'Completed' queues then you must create these now before you create any TicketMaker queues. Appendices A and B show you how to do this.

Do not delete the 'Imager' or 'Completed' queues.

14.4 Creating a new queue

To create a new queue:

1. Click on the **Add** button to create a new queue.
2. The 'Overview' dialog appears:

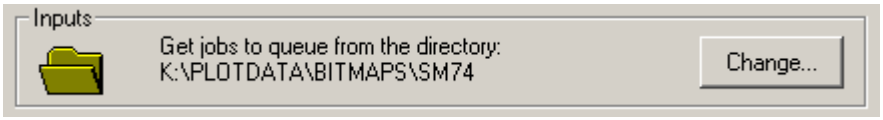


The 'Overview' dialog lets you view and edit the queue's settings. The following sub-sections show you how to set each option in the 'Overview' dialog.

Name

In the **Name** field, type in the workflow name for the new queue (for example, **SM74**).

Inputs

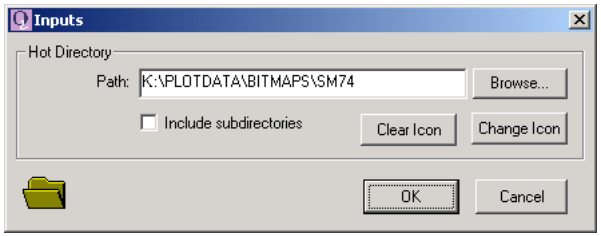


The 'Inputs' panel specifies the directory where the Python Console should look for files to be processed. This is the directory that holds the TIFF files coming from the Torrent RIP.

Note: Each queue must have a unique input directory.

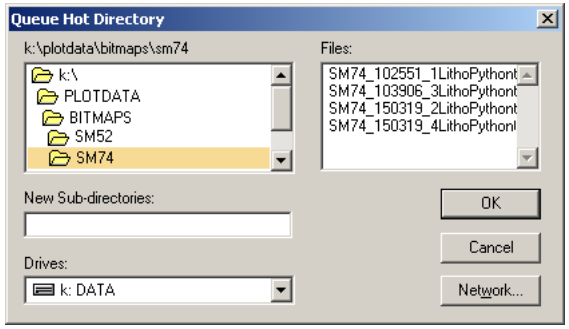
To specify the directory:

1. Click on the **Change...** button to display the following dialog:



2. The required directory is K:\PLOTDATA\BITMAPS*sub-dir*, where *sub-dir* is the name of the sub-directory where the high-res TIFF files from the Torrent RIP are saved. (If you have followed the recommended naming convention, then the sub-directory should be the same as the workflow name, for example, **SM74**.)

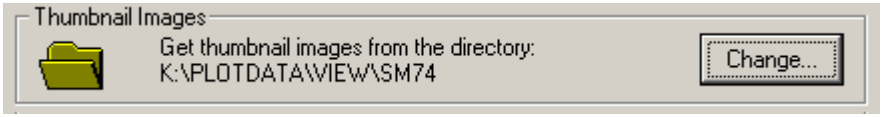
   Either type the full pathname of this directory into the **Path** field, or click on the **Browse...** button to display the 'Queue Hot Directory' dialog:



   Locate and highlight the required directory then click on **OK**.

3. In the 'Inputs' dialog, check that the correct directory name is shown.
4. When you have finished, click on **OK** to return to the 'Overview' dialog.

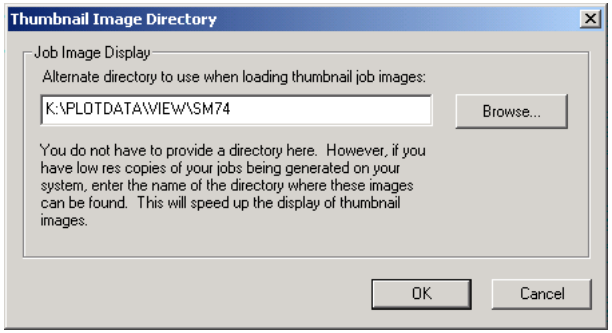
Thumbnail Images



The LRG application creates job thumbnail images for display in the Python Console (see section 5.4.1 for more details about the LRG application).

To specify the location of the thumbnail images:

1. Click on the **Change...** button. The following dialog appears:

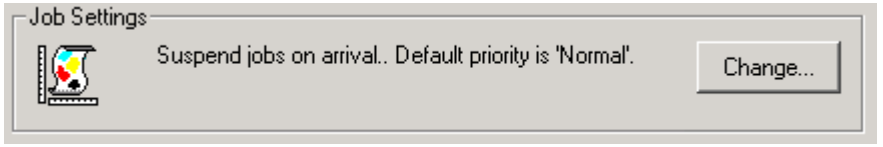


2. Click on the **Browse...** button, locate the required directory, then click on **OK**.

   Note: The Thumbnail Images directory is located at K:\PLOTDATA\VIEW*sub-dir*, where *sub-dir* is the name of the sub-directory where the high-res TIFF files from the Torrent RIP are saved. (The directory should be the workflow name, for example, **SM74**.)

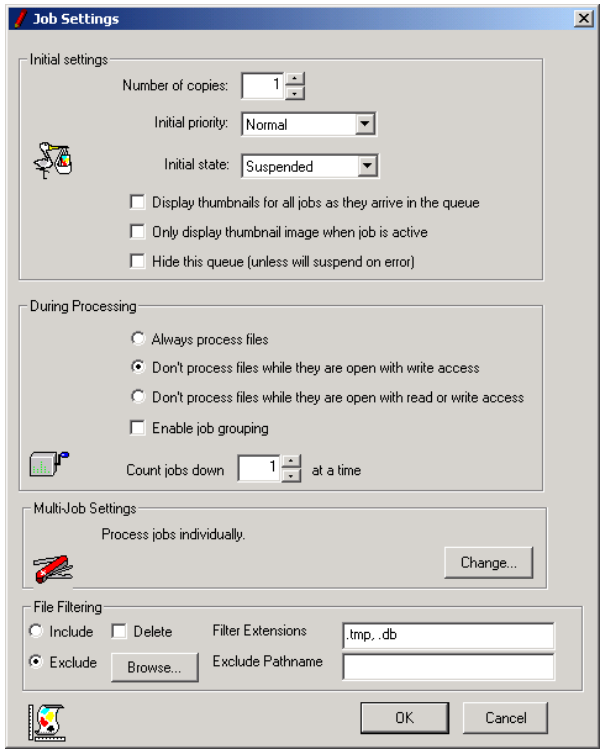
3. When you have finished, click on **OK** to close the 'Thumbnail Image Directory' dialog.

Job Settings



The 'Job Settings' let you specify options such as the job's priority. These settings will be applied to all jobs arriving in the queue. To specify the job settings:

1. Click on the **Change...** button. The following dialog is displayed:

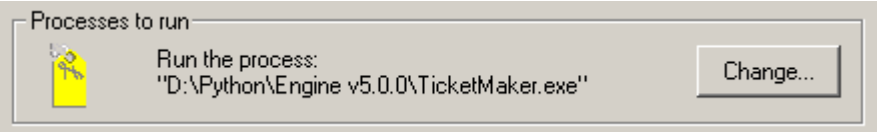


2. In the 'Initial settings' panel, set the following options:
   - Set the **Number of copies** to **1**.
   - Set the **Initial priority** to **Normal**.
   - Set the **Initial state** of all jobs arriving in the queue to **Suspended**.
   - Do not select the following options: **Display thumbnails for all jobs as they arrive in the queue**, **Only display thumbnail image when job is active** and **Hide this queue...**.
3. In the 'During Processing' panel, select the **Don't process files while they are open with write access** option.
4. Make sure the 'Multi-Job Settings' option is set to **Process jobs individually**.
5. In the 'File Filtering' panel, select the **Exclude** option and type **.tmp, .db** into the **Filter Extensions** box:

6. When you have finished, click on **OK** to return to the 'Overview' dialog.

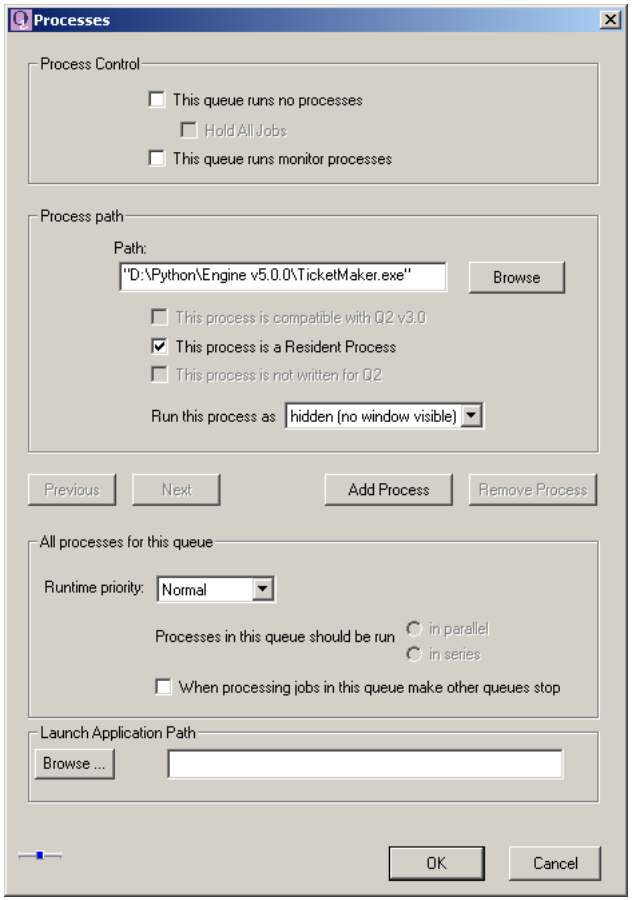
Processes to run

Note: If you are an InkMonitor user, go to the next sub-section "Processes to run (InkMonitor users only)" for instructions on how to set the options in this panel.



The 'Processes to run' panel specifies which application(s) will process jobs (in this case, the TicketMaker application). To set this:

1. In the 'Processes to run' panel, click on the **Change...** button to display the following dialog:



2. Do not select any options in the 'Process Control' panel.
3. The **Path** field should specify the **TicketMaker.exe** application (this file can be found at D:\Python\Engine v5.x.x\). Use the **Browse** button to locate this file.
4. Check the **This process is a Resident Process** box.

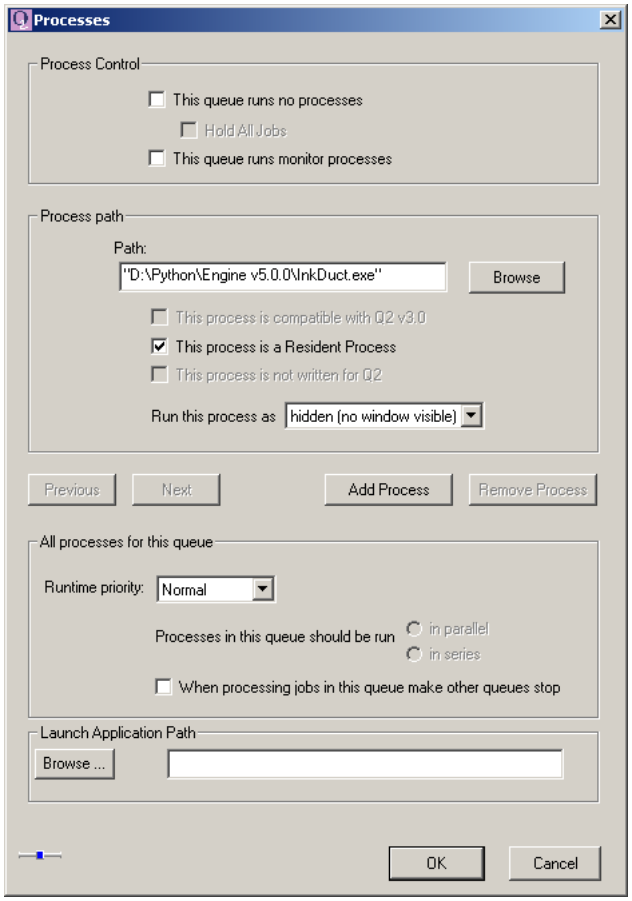
   Note: If the **This process is compatible with Q2 v3.0** box is checked, you need to deselect it before you can select the required option.
5. Set the **Run this process as** option to **Hidden (no window visible)**.
6. Set the **Runtime priority** to **Normal**.
7. Set all other options as shown in the previous 'Processes' dialog.
8. When you have finished, click on **OK** to return to the 'Overview' dialog.

Processes to run (InkMonitor users only)

The 'Processes to run' panel specifies which application(s) will process your jobs. If you wish to run the InkMonitor application for this queue, jobs need to be passed to both the InkMonitor and TicketMaker

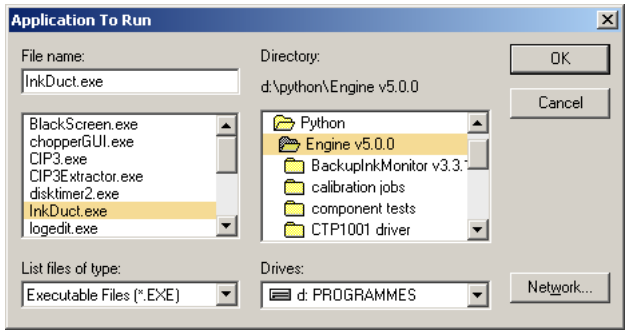
applications. To set this up:

1. In the 'Processes to run' panel, click on the **Change...** button to display the following dialog:



2. The **Path** field should specify the required InkMonitor application file (**InkDuct.exe** if you want to output to a printer or file, or **CIP3.exe** if you want to output CIP3 files).

- **Either** type the full path name of the executable InkMonitor file into the **Path** box (for example, D:\Python\Engine v5.x.x\InkDuct.exe)
- **Or** click on the **Browse** button to display the 'Application To Run' dialog:



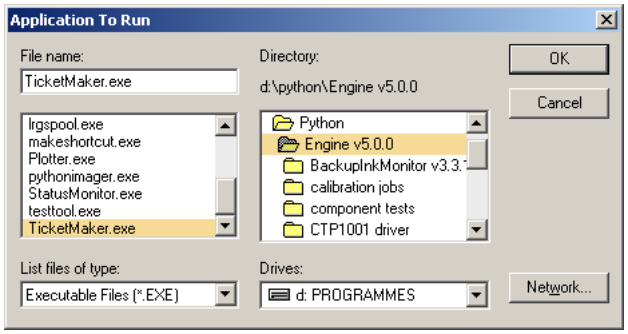
Navigate to the D:\Python\Engine v5.x.x directory, select the required InkMonitor executable file (**InkDuct.exe** or **CIP3.exe**) then click on **OK**.

3. Check the **This process is a Resident Process** box.

   Note: If the **This process is compatible with Q2 v3.0** box is checked, you need to deselect it before you can select the required option.

4. Set the **Run this process as** option to **Hidden (no window visible)**.
5. Set the **Runtime priority** to **Normal**.
6. Now click on the **Add Process** button. The **Path** field will clear.
7. In the 'Process path' panel, the **Path** should specify the TicketMaker application file.

- **Either** type the full path name of the executable TicketMaker file into the **Path** box (for example, **D:\Python\Engine v5.x.x\TicketMaker.exe**)
- **or** click on the **Browse** button to display the 'Application To Run' dialog:



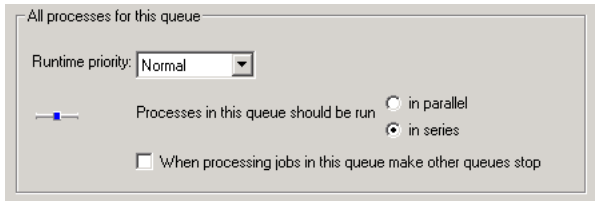
Navigate to the **D:\Python\Engine v5.x.x** directory, select the **TicketMaker.exe** file then click on **OK**.

8. Select the **This process is a Resident Process** option.

   Note: If the **This process is compatible with Q2 v3.0** box is checked, you need to deselect it before you can select the required option.

9. Set the **Run this process as** option to **Hidden (no window visible)**.

10. In the 'All processes for this queue' panel, set the **Runtime priority** to **Normal** and the **Processes in this queue should be run** to **in series**:

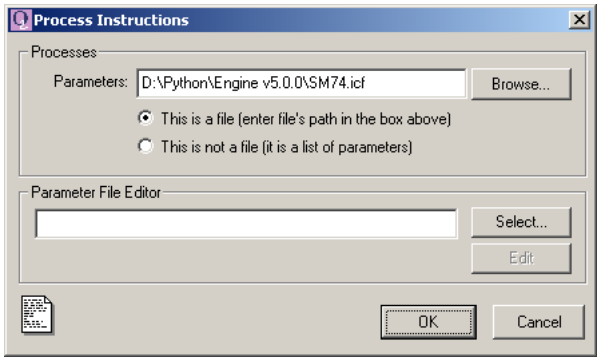


11. When you have finished, click on **OK** to return to the 'Overview' dialog.

Process instructions

The 'Process instructions' panel lets you specify the 'parameters' that the TicketMaker application should run with. In this case, the parameters are the plate and job position details that you saved in the Layout file created in chapter 13. To set the process instructions:

1. Click on the **Change...** button to display the following dialog:



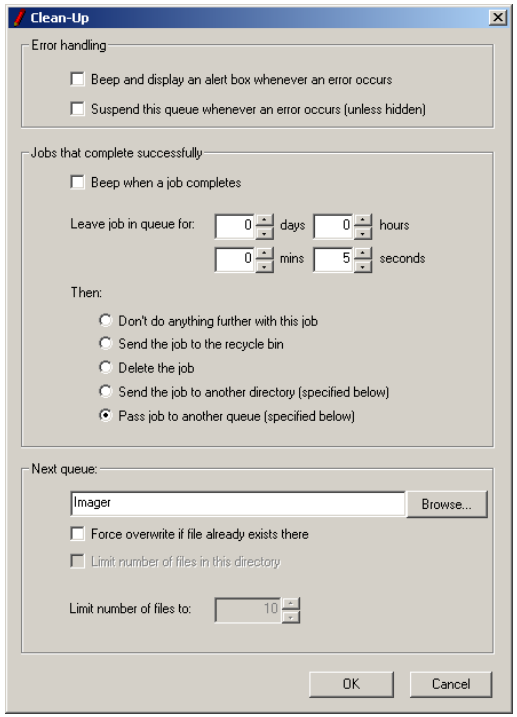
2. Click on the **This is a file (enter file's path in the box above)** radio button.
3. For the **Parameters** you need to specify the Layout file that you created in chapter 13 (for example, **SM74.icf**). Click on the **Browse...** button, locate the required Layout file then click on **OK**.

   Note: The default directory for the Layout files is D:\Python\Engine v5.x.x.
4. Click on **OK** to return to the 'Overview' dialog.

Clean-Up

Once a job has been processed, you need to specify what happens to it next (jobs are passed to the Imager queue, which outputs them to the platesetter). You specify this in the 'Clean-Up' dialog:

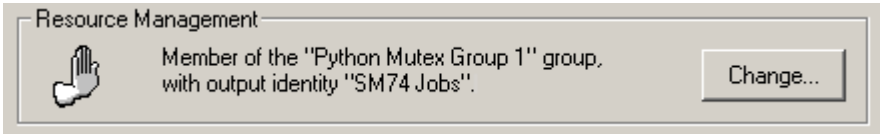
1. Click on the **Change...** button to display the 'Clean-Up' dialog:



2. **Do not** select the options in the 'Error handling' panel.

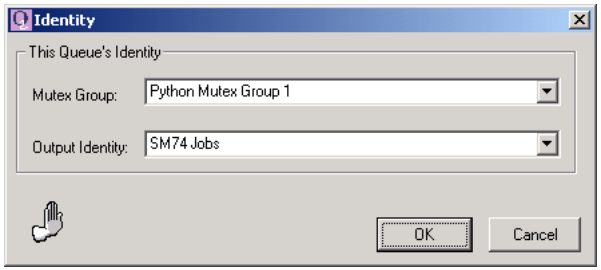
3. In the 'Jobs that complete successfully' panel, select **Pass job to another queue (specified below)**.
4. In the 'Next queue' panel, type in **Imager** or click on the **Browse...** button to select it from the list of available queues.
5. Do not set any other options in this dialog (the previous dialog shows the correct settings).
6. When you have finished, click on **OK** to return to the 'Overview' dialog.

Resource Management



You need to set the 'Resource Management' options so that only one TicketMaker queue is active at a time. All other TicketMaker queues are automatically suspended. To set the 'Resource Management' options:

1. Click on the **Change...** button to display the 'Identity' dialog:



2. Select the required **Mutex Group** from the pull-down menu.

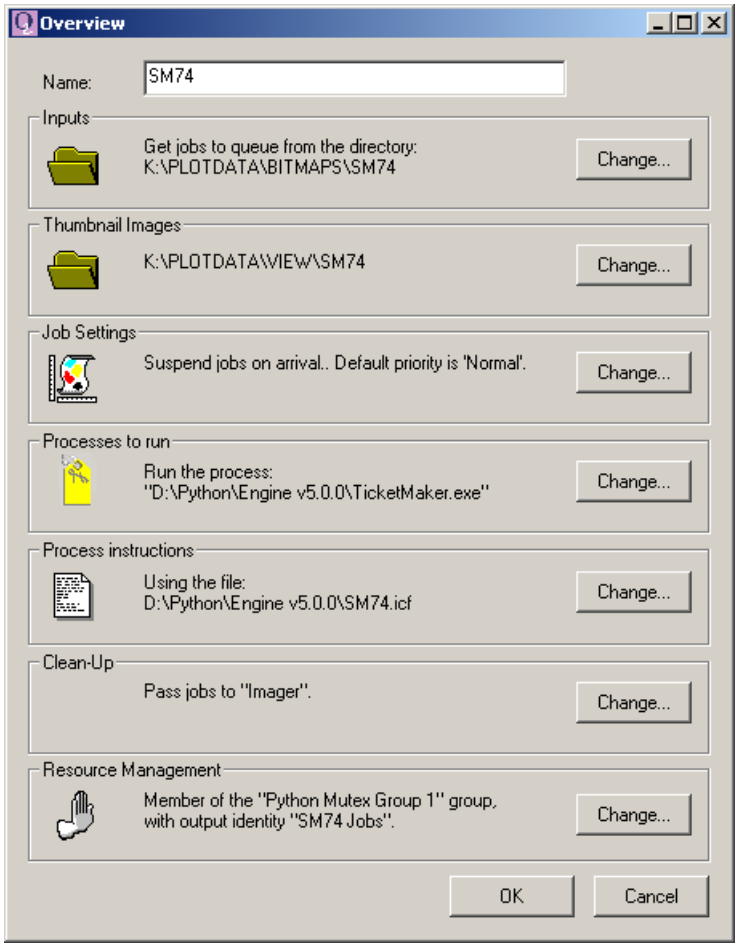
   Note: If there is no mutex group available then type in a new name (for example, Python Mutex Group 1).

WARNING: All TicketMaker queues (that is, queues with jobs destined for the platesetter except for the Imager and Completed queues) must belong to the same mutex group.

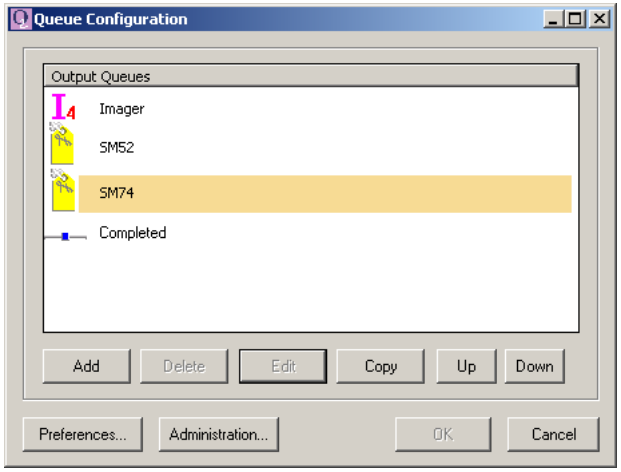
3. In the **Output Identity** field, type a unique name to identify the queue (use the workflow name, for example, **SM74**).
4. When you have finished, click on **OK** to return to the 'Overview' dialog.

14.5 The finished queue

You have now finished setting up the new queue. The completed dialog should look similar to this:



Click on **OK** to return to the 'Queue Configuration' dialog. You will now see the new queue you have just created, for example:



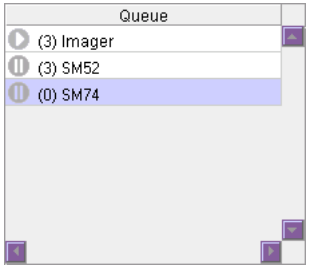
Now, create any other required TicketMaker queues by following the instructions in this chapter again.

The following section shows you how to check the new queue(s) in the Python Console. Section 14.7 describes more Queue Configuration options and section 14.8 shows you how to set the Queue Configuration preferences.

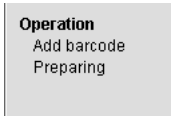
14.6 Checking the new queue

When you have finished creating a queue, you can quit out of the Queue Configuration application by clicking on the **OK** button. Now, you should make sure that the new queue is set up and operating correctly in the Python Console. To do this:

1. Close the Queue Configuration application, if you have not done so already.
2. Return to the main 'Python Console' window by clicking on the  button.
3. Click on the **Queue** tab. You should now see the new queue(s) you have just created, for example:



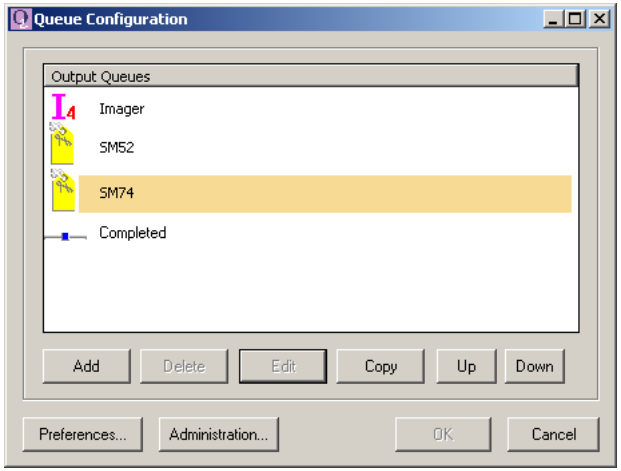
4. You can check which processes are configured to run by looking in the 'Operation' list:



5. The new queue is now ready to output your jobs to the Python platesetter.

   Note: When you have finished creating all new queues, we recommend that you save the new information. Section 16.2 shows you how to do this.

14.7 Other Queue Configuration options



The 'Queue Configuration' dialog also contains the following buttons:

Up/Down

Use these buttons to move a queue's position up or down in the list (in the Python Console, TicketMaker queues are displayed in the same order that they appear here).

Note: We recommend that you keep the queues ordered as follows: **Imager**, **[TicketMaker queues]**, **Completed**.

Preferences...

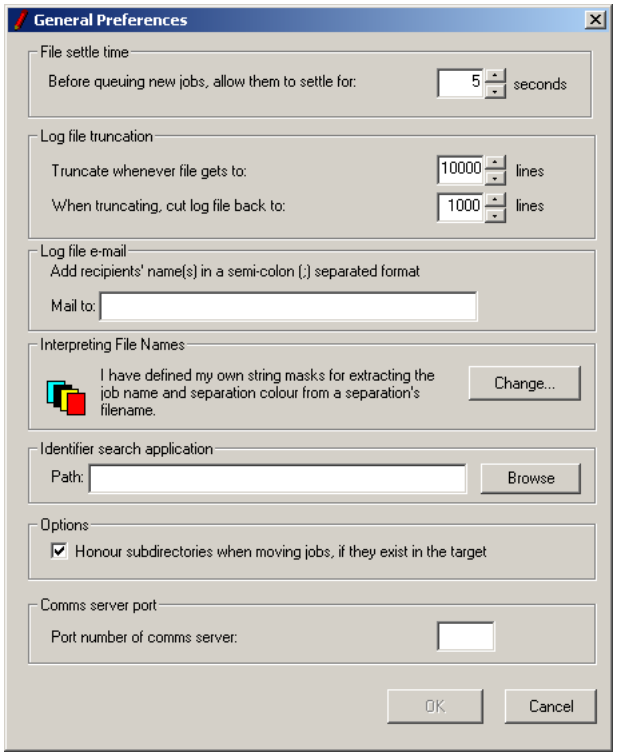
Use this to set the preferences, as described in the next section.

14.8 Setting the preferences

The 'General Preferences' dialog sets a number of parameters although, for use with the Python system, the only settings you need to change are those for the log file size.

To change the log file size:

1. Click on the **Preferences...** button to open the 'General Preferences' dialog:



2. In the 'Log file truncation' panel, use the **Truncate whenever file gets to** option to set the maximum size of the log file (in lines).
3. Use the **When truncating, cut log file back to** option to set the minimum size of the log file (in lines).

   When the log file exceeds the maximum size, it will automatically be truncated to the minimum size you have specified (oldest records are deleted first).
4. There is no need to set any other options on this dialog.
5. When you have finished, click on **OK** to close the 'General Preferences' dialog.

15. The Python Test Tool

The Python Test Tool software is used to change the sound alerts, obtain Python platesetter status information, and test and help diagnose problems with the platesetter. This chapter describes the procedures that you may need to use.

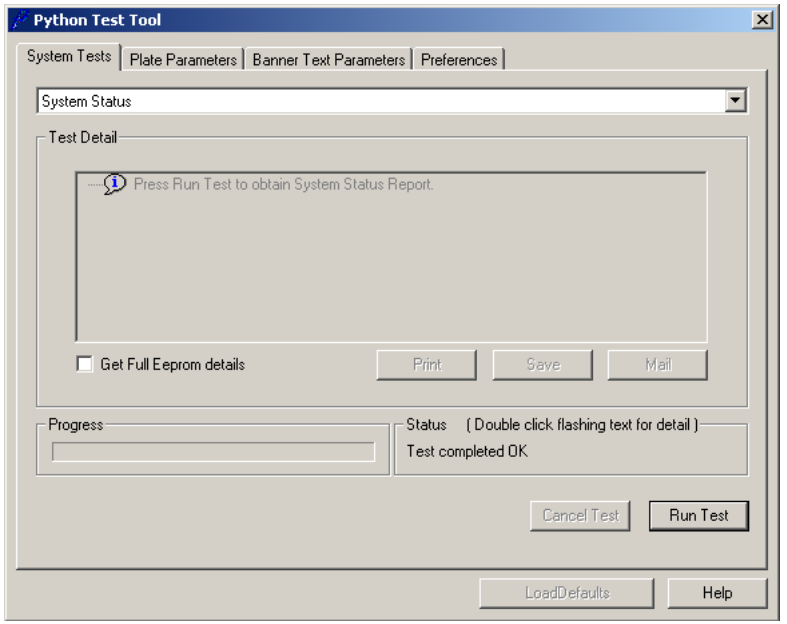
Note: Apart from changing the sound alerts you will, typically, only carry out the procedures in this chapter if asked to do so by a support engineer.

This chapter includes the following sections:

Note: The Python Test Tool's 'Calibration Strips' procedure (used to determine the correct laser power setting for your plates) is described in chapter 18.

15.1 Launching the Python Test Tool

Launch the Python Test Tool, as described in section 7.6. The main 'Python Test Tool' window is displayed:

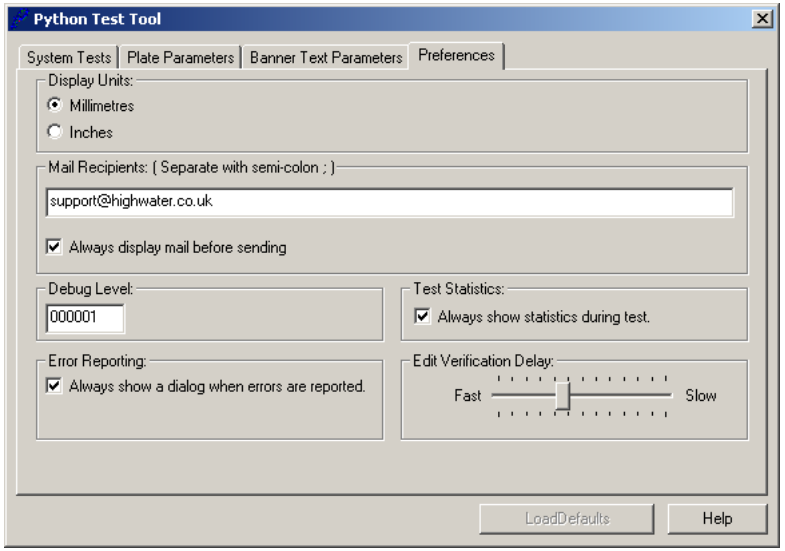


15.2 Setting the preferences

The 'Preferences' dialog lets you specify units of measurement, the addresses of email recipients and the debug level.

To set the preferences:

1. Click on the **Preferences** tab to display the following pane:



2. Select **Millimetres** or **Inches** for the **Display Units**.
3. In the **Mail Recipients** box type in the email address(es) where results from the Python Test Tool are to be sent.

   If you are specifying more than one email address, separate them with a semicolon (but no spaces), for example:

   support@highwater.co.uk;dev@colorprint.co.uk

   Note: Contact your system administrator if the Python workstation has not been set up to send and receive emails.
4. If you want to view the email before it is sent, check the **Always display mail before sending** box.
5. If required, change the **Debug Level**. By default, the debug level is **1**. Leave it at this value unless an engineer has asked you to change it.

   Note: When you launch the Python Test Tool the debug level is always reset to **1**. Higher debug levels give more information.
6. In the 'Test Statistics' panel, check the **Always show statistics during test** box to display test statistics, including elapsed time, iteration count, and an indication of how the test is proceeding (that is, pass, fail or active).
7. In the 'Error Reporting' panel, check the **Always show a dialog when errors are reported** option to show an error summary pop-up as soon as the Python Test Tool generates an error.

   If this option is not checked, the status text in the Python Test Tool will flash red and you can see the error summary by double clicking on the flashing text.
8. The value you type into some fields in the Python Test Tool (for example, the 'Strip Count' on the 'Plate Parameters' tab) is automatically edited or corrected by the software, if necessary, to ensure that it does not contain illegal characters and that it falls within the allowable range.

   The 'Edit Verification Delay' panel lets you set the delay between when you type a value and when the Python Test Tool edits/corrects it if it contains illegal characters or is out of range.

The delay for checking and amending user-typed values defaults to 1 second (i.e. in the middle of the slider). Adjust the rate to be slower or faster, as required.

Note: Illegal characters are, in general, corrected immediately so if, for example, you type **G234** into a field it would be corrected to **234**.

15.3 Changing the sound alerts

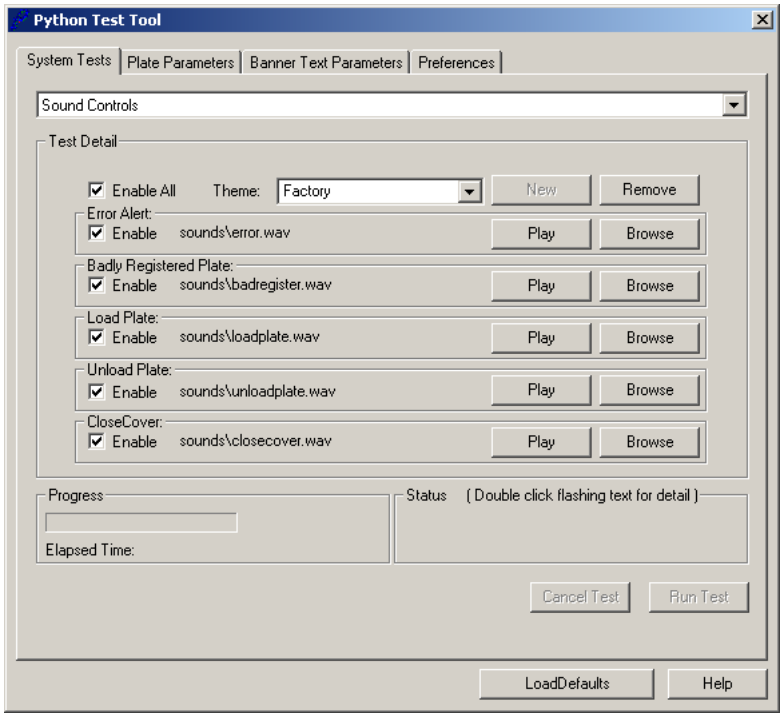
The Python Test Tool lets you change the sound alerts that play during the Python platesetter's operation. Sounds can be played for the following alerts:

Error Alert	The sound played when the platesetter detects an error.
Badly Registered Plate	The sound played when the plate is not correctly positioned in the clamp.
Load Plate	The sound played when the platesetter requires the user to load a plate.
Unload Plate	The sound played when the platesetter requires the user to unload a plate.
Close Cover	The sound played when the platesetter requires the user to close the cover.

Note: You will also see status images in the Python Console for most of the above alerts. Refer to the section " 'Operation' panel" on page 46 for more details about this.

To display the sound controls dialog:

1. In the Python Test Tool, click on the **System Tests** tab.
2. Select **Sound Controls** from the pull-down menu at the top of the pane to display the following dialog:



15.3.1 Themes

The sound alerts are grouped as 'themes' to simplify the process of switching between different sound sets. Two standard themes are supplied with the Python system:

- **Silent** — this is the default setting, which plays no sounds.
- **Factory** — this is the sound set supplied with the Python system.

  Note: These themes cannot be edited.

15.3.2 Selecting a theme

To select a theme:

- **Either** select the required **Theme** from the pull-down list,
- **or** start typing the required theme name into the **Theme** field. After a short delay, the theme name matching your typing is shown.

  Note: If the theme name you type does not exist, the **New** button is enabled and you can click on it to define a new theme, as described below.

15.3.3 Creating a new theme

To create a new theme:

1. Type the new theme's name into the **Theme** field.
2. Click on the **New** button.

   Note: Initially, the new theme contains the sound files that were displayed when you clicked the **New** button.
3. To change the sound for a particular alert, click on the **Browse** button alongside it to display the 'Open' dialog.
4. Navigate to the appropriate directory (the default location is D:\Python\beep\sounds), highlight the required sound file and click on the **Open** button.

   Note: Currently, only .wav files can be played.
5. To hear the sound sample, click on the **Play** button alongside the alert.

   Note: If the file cannot be found, a default sound is played.
6. Click on the **Cancel Test** button to stop the sound playing.

15.3.4 Removing a theme

To remove a theme:

1. Select the required **Theme** from the pull-down list.
2. Click on the **Remove** button to remove the selected theme.
3. You will be asked to confirm the deletion.

Note: You cannot remove the standard themes (**Silent** and **Factory**).

15.3.5 Disabling sound alerts

To disable sound alerts for the current theme:

1. **To disable all sounds**, uncheck the **Enable All** box.
2. **To disable specific sounds in the theme**, make sure the **Enable All** box is checked, then uncheck the individual box(es) for the particular sound alert(s) you want to disable.

15.4 The System Status test

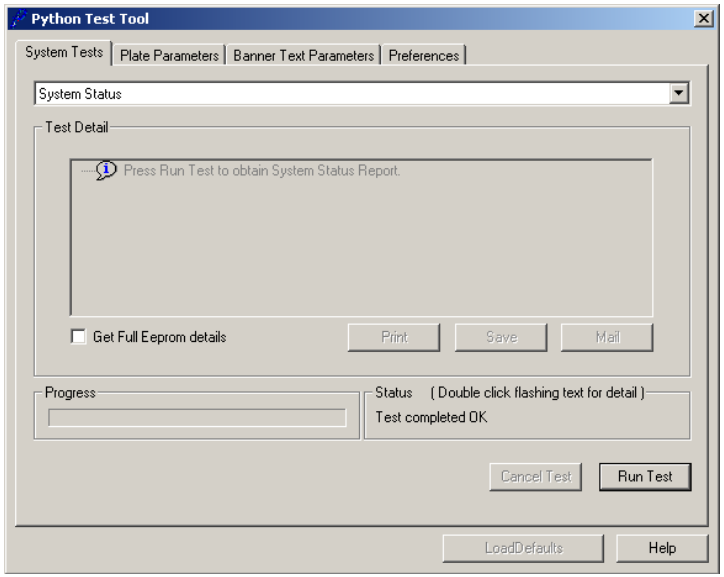
The 'System Status' test performs the following functions:

- Reports software and firmware version numbers for the Python platesetter.
- Reports component settings for the Python platesetter hardware.
- It can indicate problems with the Python platesetter.

Note: You are only likely to run this test if requested to do so by your support provider.

To run the 'System Status' test:

1. Click on the **System Tests** tab then select **System Status** from the pull-down menu at the top of the pane. The following dialog is displayed:

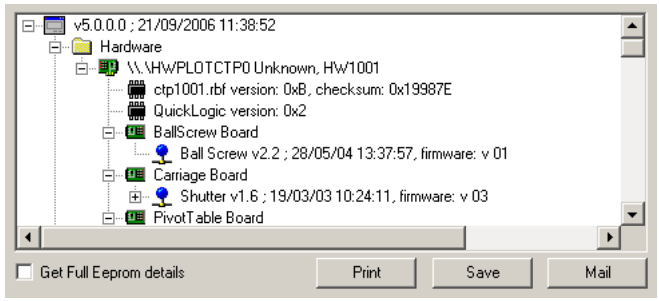


2. If you also want the test to report the contents of the Eeproms associated with each board, check the **Get full Eeprom details** box.

   Note: This will slow the test down considerably, so do not select this option unless requested to do so by your support provider.

3. Click the **Run Test** button. You will see the message **Scanning system, please wait...** in the 'Status' panel.

4. When the test is complete, the results appear in the 'Test Detail' window, for example:



   Each of the machine's sub-systems is shown with its corresponding icon. If a machine sub-system's icon is crossed through in red ( ) then it is not responding and needs further investigation from your support provider.

5. You can now print the test report or mail it to your support provider:

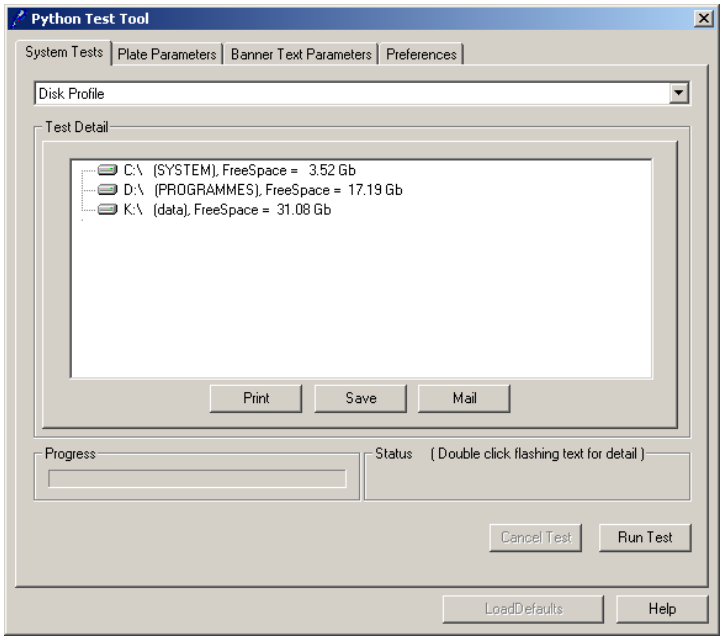
   - **To print the report**: first make sure that a printer is available on your network then click on the **Print** button. Once printed, you can send or fax the report to your support provider.
   - **To save the report**: click on the **Save** button, select a directory then type in a file name. Click on the **Save** button.

- **To email the report**: click on the **Mail** button (make sure you have set up one or more email addresses in the 'Preferences' dialog — see section 15.2). Your email package will launch and send the email. If you have set up the preferences to do so, the report will be displayed before it is sent.

15.5 The Disk Profile test

The 'Disk Profile' test checks the speed/data rate of the selected disk(s) and/or file(s). You can run the disk profile tests on all available drives or on individual drives. To access the 'Disk Profile' test:

1. Click on the **System Tests** tab, then select **Disk Profile** from the pull-down menu to display the following:

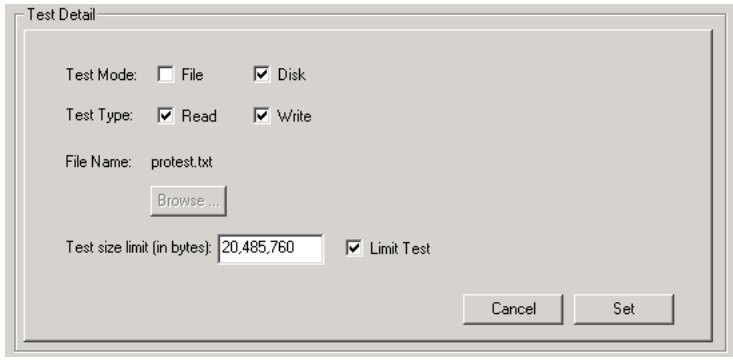


The 'Test Detail' window lists each drive on the Python workstation.

15.5.1 Setting the test options

First you need to set the options for the test(s). To do this:

1. In the 'Test Detail' window, right-click on the required drive and select **Set Test Options** from the pop-up menu. The following panel appears:



2. For the **Test Mode** you can choose to test a **File** and/or a **Disk**:
   - **If testing a file**, to get more accurate test results, choose a large file (greater than 10Mb). It is not possible to write-test a file so only select the **Read** option.

     Click on the **Browse...** button and select the file from the dialog.
   - **If testing a disk**, profiling may use all the available free space on the drive. You MUST select both the **Read** and **Write** tests otherwise the 'Read' results will be inaccurate.

3. To limit the test to a specified size (even if the file size/disk free space is greater than this value), check the **Limit Test** box and type a value into the **Test size limit (in bytes)** field. If you do not limit the test, the entire file or free disk space may be used, depending on the test mode.
4. When you have finished setting the options, click on the **Set** button.

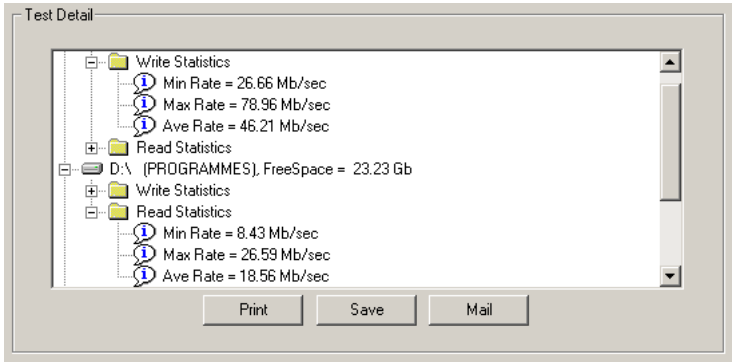
15.5.2 Running the Disk Profile tests

You can either run the disk profile tests on all listed drives, or on individual drives, as described in the following sub-sections.

Running the tests on all drives

To run the tests on all drives:

1. Set the **Test Options** for all the drives, as described in the previous sub-section.
2. Click on the **Run Test** button. After a while, the test results are displayed in the 'Test Detail' window, for example:

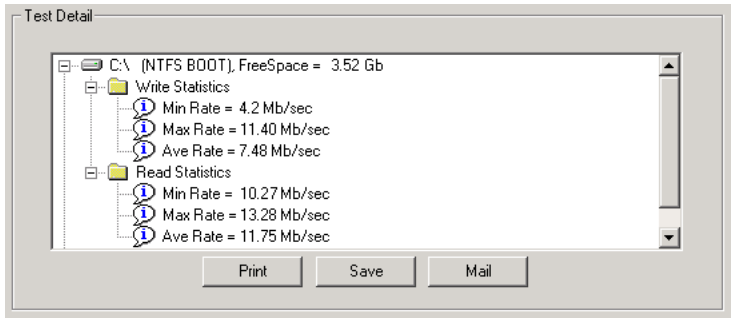


See section 15.5.3 for a description of these results.

Running the tests on an individual drive

To run the tests on an individual drive:

1. Set the **Test Options** for the required drive, as described in section 15.5.1.
2. Right click on the drive, and select **Profile** from the pop-up menu.
3. The test results for the drive are displayed in the 'Test Detail' window, for example:



15.5.3 The test results

The test results consist of write and read statistics as follows:

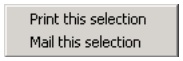
- **Write Statistics:** Expanding the item will display the minimum, average and maximum data rates. An average data rate of 6–8 Mb/s is acceptable.
- **Read Statistics:** Expanding the item will display the minimum, average and maximum data rates. An average data rate of 8–12 Mb/s is acceptable.

15.5.4 Printing or mailing the test results

Once the tests are complete you can print, save or mail them:

- **Print** — Prints the results to the selected printer.
- **Save** — Saves the results to the specified file.
- **Mail** — Mails the results to the email recipient(s) listed in the 'Preferences' dialog.

  It is possible to print and/or mail a subset of the results by right clicking on the **Write statistics** or **Read statistics** items. A sub-menu appears with the following options:

  Select one of the following options:

  - **Print this selection** — Print the expanded selection to the selected printer.
  - **Mail this selection** — Mail the expanded selection to the recipient(s) specified in the 'Preferences' panel.

16. Maintenance

This chapter shows you how to carry out routine maintenance on your Python system. It includes the following sections:

- 16.1, Cleaning the Python platesetter (p110).
- 16.2, Saving Python system settings (p111).

16.1 Cleaning the Python platesetter

WARNING: All maintenance to the Python platesetter, other than that described here, should only be carried out by an engineer who has been approved by HighWater Designs. Any damage to the Python platesetter resulting from carelessness or non-approved maintenance is not covered by HighWater's warranty.

The laser beam used in the Python platesetter is harmful to the eyes, so do not attempt to remove the laser carriage cover.

The tilt-table and inside of the Python platesetter need to be kept as clean as possible to minimize the risk of dust, dirt and other particles contaminating your plates and the Python drum.

You should inspect the Python tilt-table and the surrounding case areas every time you load a plate, and you should periodically clean the tilt-table and drum, as described next.

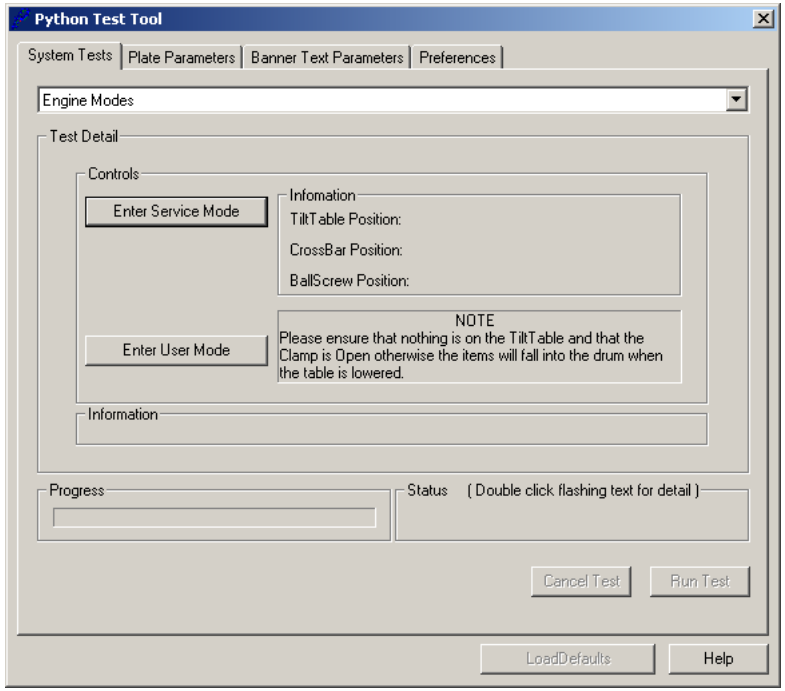
16.1.1 Cleaning the tilt-table

We recommend that you clean the tilt-table with a vacuum cleaner hose.

16.1.2 Cleaning the drum

To clean the drum or to remove anything that has fallen into it:

1. Launch the Python Test Tool (refer to section 7.6).
2. Click on the **System Tests** tab and select **Engine Modes** from the pull-down menu. The following dialog is displayed:



3. Remove anything that is on the Python tilt-table, including any plate.
4. Close Python's lid.

5. In the Python Test Tool, click on the **Enter Service Mode** button. The Python platesetter will respond as follows:
   - The laser carriage moves to the far left-hand side of the drum.
   - The tilt-table drops down to a vertical position.
   - The crossbar moves halfway down the tilt-table.
6. When you see the message 'Test Complete' you can lift Python's lid. You now have access to Python's drum.

WARNING: When you have access to the inside of the Python platesetter, take great care not to knock its carriage or other sensitive parts as this could result in damage.

7. If anything has fallen into the drum, carefully remove it.
8. Carefully clean the inside of the Python drum using a vacuum cleaner hose.
9. When you have finished, close the lid.
10. In the Python Test Tool, click on the **Enter User Mode** button. This will reset the Python to its normal operating mode.

    Note: If you forget to do this, the next time you try to output a plate, the system will automatically reset to user mode.
11. You can now exit out of the Python Test Tool and continue to use the Python Console to output your jobs to the Python engine.

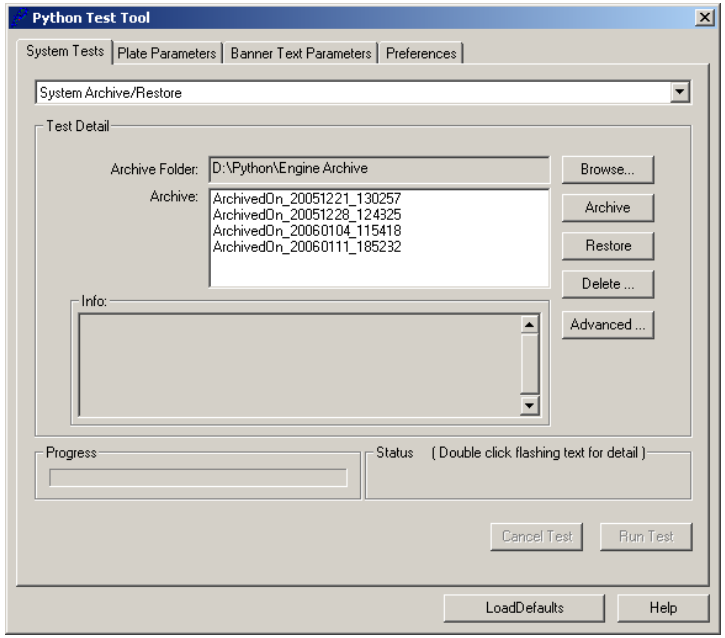
16.2 Saving Python system settings

The Python Test Tool lets you save Python software settings (the Layout (ICF) files created in the Python Layout Tool, the Python Console queue information and the Torrent RIP Configuration file) so you can restore previously used settings, if required (for example, if your current settings get deleted or corrupted).

16.2.1 Saving Python system settings

To save the current Python system settings:

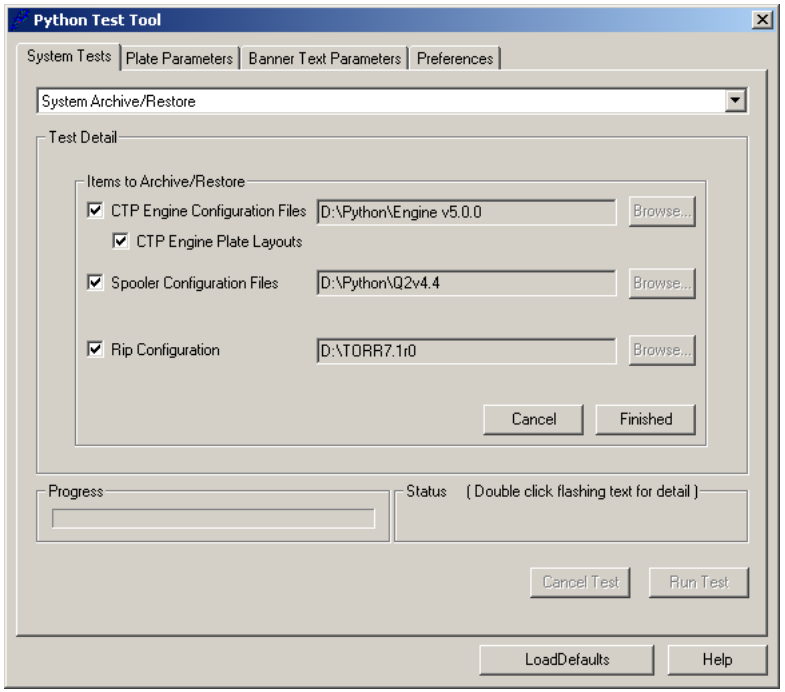
1. Launch the Python Test Tool (refer to section 7.6), if it is not already running.
2. Click on the **System Tests** tab, then select **System Archive/Restore** to display the following dialog:



Previously archived files are shown in the Archive list (click on the **Browse** button to change the current folder).

The 'Info' panel shows a summary of the most recently performed operation (when a new operation is performed the previous operation's details are removed from the 'Info' panel).

3. Click on the **Advanced...** button to display the following dialog:



Notes: There may be a delay (sometimes long) before this dialog is displayed because the program first tries to locate the relevant software applications on the computer.

If the Q2 software cannot be found, you will not see the Spooler Configuration Files field.

4. Select the file information to save by checking the appropriate boxes:
   - **CTP Engine Configuration Files**: These are the Layout (ICF) files created in the Python Layout Tool.
   - **CTP Engine Plate Layouts**: These contain the plate and layout information used by the Python Layout Tool.
   - **Spooler Configuration Files**: These contain the Python Console queue information (created in the Queue Configuration application).
   - **RIP configuration:** This contains the Torrent RIP Configuration file (if you have created one. Refer to section 6.7 for instructions).

   Note: Typically, you should select all four file types.

5. If necessary, select the location of the **CTP Engine Configuration Files**, **Spooler Configuration Files** and **RIP Configuration** file using the **Browse...** buttons. The default locations of these files are usually:
   - D:\Python\Engine v5.x.x for the CTP Engine Configuration files
   - D:\Python\Q2v4.4 for the Spooler Configuration files.
   - D:\Torrent7.1r0 for the RIP Configuration file.

   Notes: This step should only be required if more than one version of the Q2, Python or Torrent software is installed on the machine or if you have not saved the Torrent configuration file in its default location.

   You can click on the **Load Defaults** button to restore the default locations for the configuration files.

6. When you have finished setting the Advanced options, click on the **Finished** button.

7. In the 'Test Detail' panel select an **Archive Folder** by clicking on the **Browse...** button. Locate the required folder then click on the **OK** button.

   Note: The default folder is D:\Python\Engine Archive. We recommend that you always use this folder.

8. Click on the **Archive** button to save the information to file.

   The archive file name has the following structure:

   ArchivedOn_[date:year/mm/dd]_[time:hours/mins/secs]

   For example, **ArchivedOn_20060117_143309**

9. If no errors occur during the archiving process, you will see an 'Archive completed OK' message at the bottom of the 'Info' window.

   If an error occurs during the archiving process, you will see an error message (either at the bottom of the 'Info' window or in a separate dialog). Scroll through the 'Info' panel to check what errors have occurred.

16.2.2 Restoring an archive

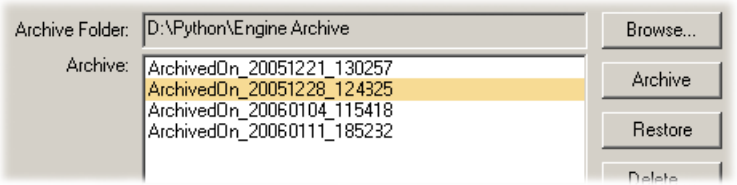
WARNING: Restoring an old archive will overwrite your current settings, so first save/archive your current settings in case you should later need them.

To restore previously archived settings:

1. Click on the **Advanced...** button and select the file types to be restored.
2. Click on the **Finished** button.
3. In the 'Test Detail' panel, select the required **Archive Folder** using the **Browse...** button.

   Note: The default folder is D:\Python\Engine Archive.

4. Select the file you wish to restore from the list of **Archive** files:



5. Click on the **Restore** button. The system settings will now be restored. (You should also follow the instructions in section 6.7.1 for restoring the Torrent RIP settings back into the RIP.)
6. If no errors occur during the restore process, you will see an 'Restore completed OK' message at the bottom of the 'Info' window.

   If an error occurs during the restore process, you will see an error message (either at the bottom of the 'Info' window or in a separate dialog). Scroll through the 'Info' panel to check what errors have occurred.

16.2.3 Deleting an archive

To delete old archives that are no longer required:

1. If necessary, select the required **Archive Folder** using the **Browse...** button.

   Note: The default folder is D:\Python\Engine Archive.

2. Highlight the archive in the **Archives** list, then click on the **Delete...** button.

3. You will be asked to confirm that you wish to delete the archive:

4. Click on **Yes** to delete the selected archive.

17. Troubleshooting

This chapter provides troubleshooting information for some common problems. If you cannot solve your particular problem or find the help you need, please contact your authorised Python support provider.

This chapter covers the following problems:

17.1 Torrent RIP problems

Below are some problems you may encounter when using the Torrent RIP. (Please refer to the **Torrent User's Guide** for a comprehensive list of errors that can occur within the RIP.) Torrent's main window is also useful for troubleshooting as it provides information about what the RIP is doing.

17.1.1 On the Macintosh Classic, an input queue does not appear in the Chooser

There are several reasons why a Torrent input queue does not appear in the Chooser on the Macintosh:

- Torrent may not be running on the Python workstation, or the workstation may be switched off.
- You may not have broadcast your input queues (refer to section 6.4.1).
- The input queue may be switched off in the Input Controller (refer to section 6.4.3).
- You have not selected the correct options in the Chooser (refer to section 6.6.2).
- Torrent has crashed. If so, re-start the system and re-RIP any jobs that were being processed when the crash occurred.
- A network failure has occurred, possibly due to a cable fault or because a cable has become disconnected from a machine. Check all cabling.

17.1.2 'Insufficient disk space' error

If your disk runs out of space while a job is being RIPed, Torrent will display a warning message, for example:

> **System warning - Insufficient disk workspace**
>
> **System warning: Free disk space has gone below requested limit**

Make some more free space available on your disk and the job will be re-processed.

17.2 Python Console problems

Below are some problems you may encounter when using the Python Console.

17.2.1 No jobs are being processed

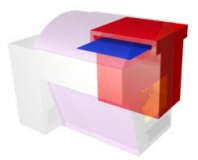
Check that the overall Python Console processing status is on and that the Imager queue status is active. Also check the status of individual queues and jobs as these may have been suspended.

17.2.2 The Python Console will not re-launch

If the Python Console will not re-launch from the Windows **Start** menu or you experience problems running it after a re-launch, then log off the Python workstation and log back on again using the username **python** (with no password).

17.2.3 'Check bridge' error message

A 'Check bridge' error message occurs in the Python Console if there is a problem with the bridge and it is unable to unload the current plate:



Check bridge

Check that the plate is not stuck, check the serial cable connection to the bridge, and check that the bridge is connected.

17.2.4 'Plate badly registered' error message

A 'Plate badly registered' error message occurs in the Python Console if the plate becomes mis-registered after you have correctly registered it:

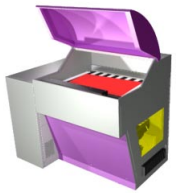


Plate badly registered

Refer to section 9.2.2 for details on re-registering the plate.

17.3 Python Layout Tool problems

Below are some problems you may encounter when using the Python Layout Tool.

17.3.1 There is no Save button on the 'Layout' window

If the 'Layout' window has a **Start** button instead of a **Save** button:

1. Click on the **Preferences** tab.
2. In the 'Application Setup' panel, select the **Layout editor** radio button, then click the **Apply** button.
3. On the 'Layout' window there should now be a **Start** button.

17.4 Python engine problems

Below are some problems you may encounter when using the Python engine.

17.4.1 The register lights don't come on when you try to position the plate

This is not an error, but will occur if the workstation is switched off or if you are not running the Python Console software.

Check if the clamp button light is on. If it is, press the clamp button to switch it off. The register lights should now work.

17.5 Plate problems

Below are some plate problems you may encounter.

17.5.1 Fogged plates

Your plates will fog if they are exposed to light sources other than the appropriate safety lights for the plate type. Please see section 3.2.3 for information about protecting your plates from light exposure.

17.5.2 Dirty plates

Plates will get dirty from fingerprints and from dust/dirt particles from the tilt-table or the Python drum. Therefore, it is essential that you handle plates carefully and keep the Python engine clean — section 16.1 shows you how to do this.

17.5.3 Over/under-exposed plates

If you find that plates are consistently over- or under-exposed over the whole plate area, you should check the laser power setting for the plate — chapter 18 shows you how to do this.

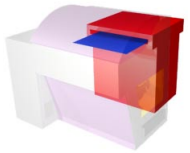
Note: You should also re-calibrate your Torrent page setups if you are having expose problems or if you change the laser power setting for a plate.

17.6 Auto-unload problems

If the plate auto-unload feature is set up on your Python machine, then call your support provider:

- If it stops working.
- If you wish to revert to manually removing the imaged plate from the Python engine.

A ‘Check bridge’ error message occurs if there is a problem with the bridge and it is unable to unload the current plate:



Check bridge

Check that the plate is not stuck, check the serial cable connection to the bridge, and check that the bridge is connected.

18. Checking the laser power setting for your plates

You need to check the laser power setting for your plates to ensure output quality. This chapter describes how to determine and change the laser power settings, and it includes the following sections:

18.1 When to check the laser power setting

You need to check the laser power settings for your plates when any of the following occur:

- You start a new box of plates.
- You use a new type of plate (for example, a different size or thickness).
- You change your processor chemistry.
- If plates are consistently over- or under-exposed over the whole plate area.

18.2 Overview of determining the laser power setting for your plates

Determining and setting the correct laser power setting for your plates involves the following steps:

1. Launch the Python Test Tool.
2. Output the Calibration Strips job to plate.
3. Measure the 50% density patches on the plate, and find the patch that gives the correct reading.
4. Enter this value into the Python Layout Tool's plate information section.

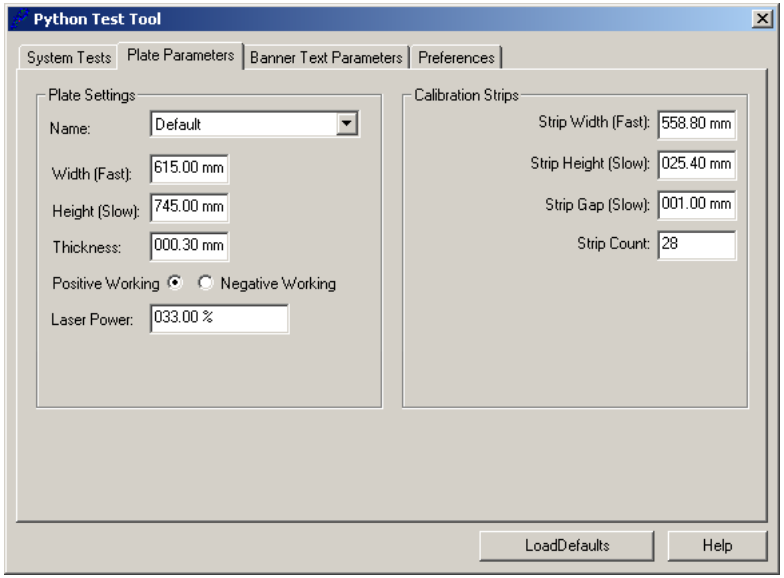
Each of these steps is covered in the following sections.

Note: You need to check and set the laser power setting for every plate size/type that you use.

18.3 Outputting the Calibration Strips job

The first stage in determining the correct laser power setting for the plate is to output the Calibration Strips job. To do this:

1. Launch the Python Test Tool, as described in section 7.6.
2. To set up the plate details, click on the **Plate Parameters** tab to display the following:

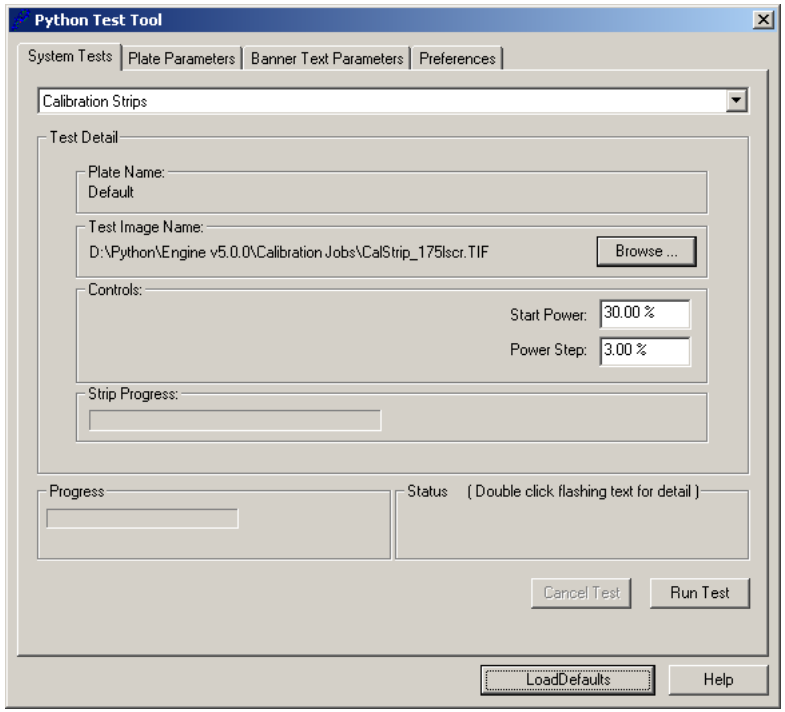


3. Select the required plate from the **Name** menu and check that the plate settings (**Width**, **Height, Thickness**, etc.) are correct.

   Notes: The plates available from the **Name** menu are those that were created in the Python Layout Tool. If the required plate is not available in the list, you need to create it using the Python Layout Tool (this is covered in section 13.6).

   Generally, it is not necessary to change the **Calibration Strips** parameters (these set the size of, and distance between, the strips/patches on the plate).

4. Now, click on the **System Tests** tab and select **Calibration Strips** from the pull-down menu at the top of the pane. The following dialog is displayed:



5. Check that the Plate Name and details are correct.

   Note: If you place the mouse cursor over the plate name, the plate's details appear, for example:

6. Click on the **Browse...** button and select the **CalStrip_175lscr.TIF** or **CalStrip_200lscr.TIF** file, depending on the screen ruling required, then click on **Open**.

   Note: The default directory for these files is D:\Python\Engine v5.x.x\Calibration Jobs.

7. Enter **30.00 %** for the **Start Power**.
8. Enter **3.00 %** for the **Power Step**.
9. Now load the plate onto the Python tilt-table (refer to chapter 9 for full details on doing this).

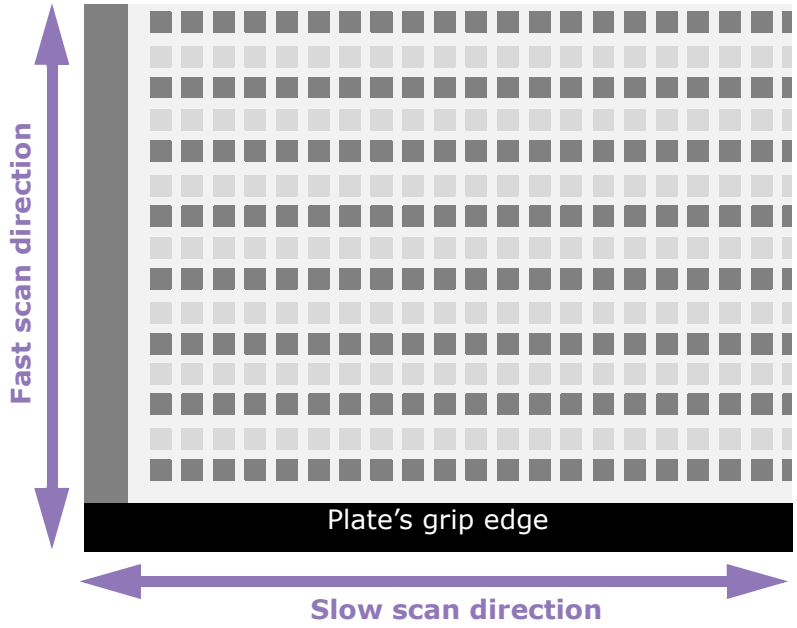
   Note: Make sure you load the plate for which you are determining the laser power setting.

10. When the plate is loaded, close Python's lid and click on the **Run Test** button. The Python Test Tool will image the plate with the Calibration Strips job that you specified.
11. When the plate has been imaged, remove it from the Python tilt-table and process it.

Measuring the plate densities is covered in the next section.

18.4 Measuring the plate densities

The job that is output to your plate consists of 50% and 2% patches laid out in rows and columns similar to this:

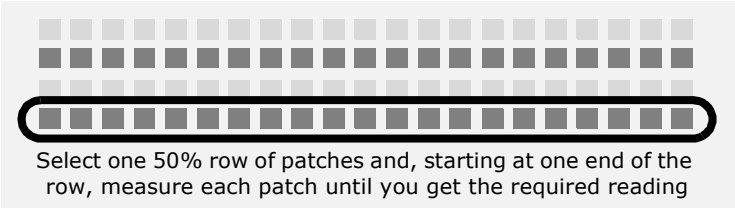


To find the correct laser power setting for this plate:

1. Select any one row of 50% patches to measure (it does not matter which row you choose).

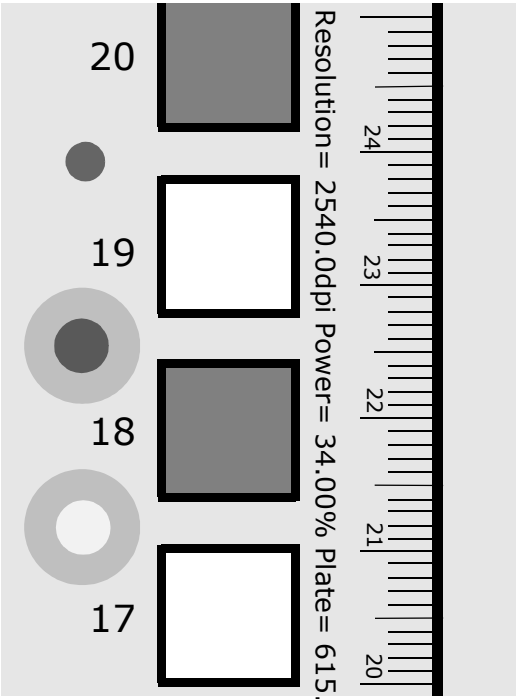
2. Using a plate densitometer and starting from either the left or right-hand side of the plate (and with the grip edge at the bottom), measure each 50% patch in the row until you get a reading of approximately 52% for a positive working plate, or 58% for a polymer, negative working plate.

   Note: For correct readings for other types of plate, please contact your dealer.



Select one 50% row of patches and, starting at one end of the row, measure each patch until you get the required reading

3. Running down the right-hand edge of the selected patch is a vertical line of text, which includes the date, plate size, job name and resolution, for example:



4. This text also contains a laser power value (34.00% in the example above), which is the correct setting for your plate. **Make a note of this laser power value**.

   Note: If necessary, keep reading up/down the line of text until you reach the Power information.

18.5 Setting the laser power in the Python Layout Tool

Once you have determined the correct laser power for your plate, you need to specify it in the Python Layout Tool. Please refer to sections 13.6 and 13.7 for details on specifying the laser power and saving the information as a Layout file for use with the Python Console.

19. Creating a print queue on the page make-up workstation

Before printing to the Torrent RIP, you need to copy the appropriate Python PPD onto each workstation you are printing from, and create a Python 'printer' for each Torrent input queue you will be printing to. This printer will let you send jobs from the page make-up workstation to the Torrent RIP.

This chapter includes the following sections:

19.1 Copying the PPDs to the page make-up workstation

First, you need to copy the Python PPDs to every page make-up workstation you will be printing to Torrent from.

You can find the Python PPDs at the following locations:

- On the Python workstation at D:\Python\Engine v5.x.x\PPD
- On the Python CD supplied with your system.

In the PPD folder, you will find the following sub-folders:

- **Mac** — this folder contains the PPD file for the Macintosh (Classic and OSX).
- **Win2k_XP** — this folder contains the PPD file for workstations running Windows XP, Windows 2000 and Windows Server 2003.
- **WinNT** — this folder contains the PPD file for workstations running Windows NT.
- **Win9x** — this folder contains the PPD file for workstations running Windows 95 and 98.

19.1.1 Copying Python PPD for Mac Classic

To copy the Python PPDs to a Mac Classic workstation:

1. Locate and open the *Printer Descriptions* folder on your Macintosh's hard drive. This is usually located at:

   Macintosh HD:System Folder:Extensions:Printer Descriptions

2. Copy the Python PPD (from the **Mac** folder) into this folder.

19.1.2 Copying Python PPD onto Mac OSX and all Windows workstations

Note: If you are using Windows XP and are updating/re-installing the Python PPD files then please follow the instructions in section 19.1.3 below first.

To copy the Python PPD to a Mac OSX or Windows workstation:

1. Copy the relevant PPD **folder** (Mac, Win2k_XP, WinNT or Win9x) onto the workstation. Make a note of the location as you will need to find the folder later when you create the Python print queue.

19.1.3 Updating/re-installing PPDs for Windows XP

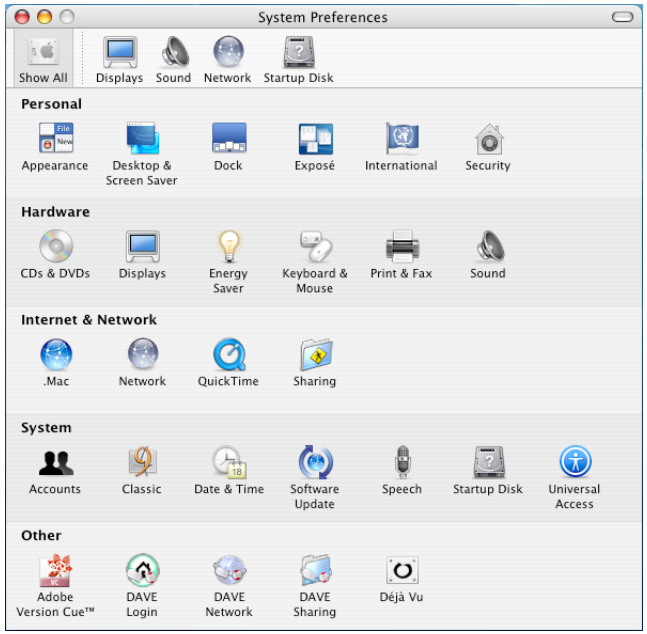
On Windows XP, if you are updating/re-installing a Python PPD file and it is intended to replace an existing PPD with the same name, you first need to remove the existing PPD. To do this:

1. Check that the PPD you wish to delete is not being used by currently installed printers. If it is, you should delete the printers then recreate them after you have re-installed the PPD file.
2. In Windows Explorer, navigate to the Windows\System 32\Spool\Drivers directory.
3. Locate and delete the .ppd file you are intending to re-install.

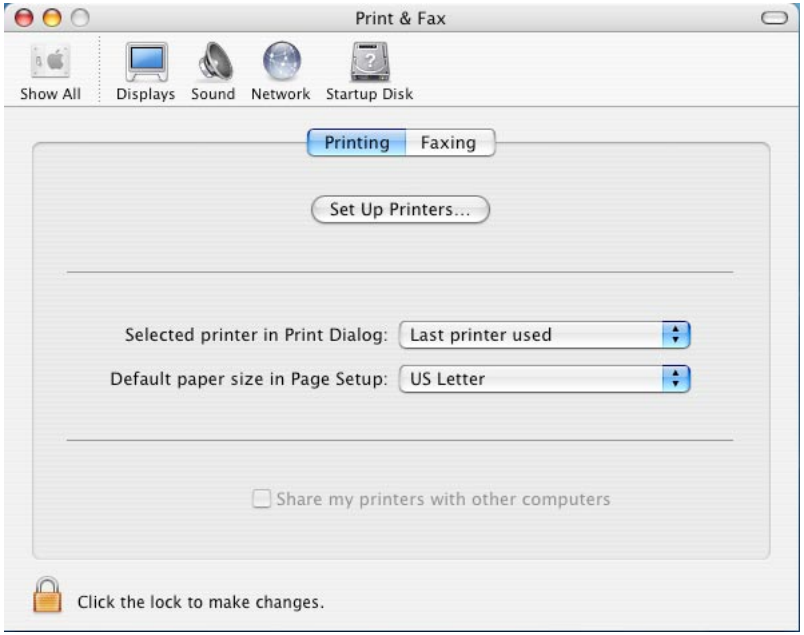
19.2 Creating a print queue on a Mac OSX workstation

To create a Python print queue on a Mac OSX workstation:

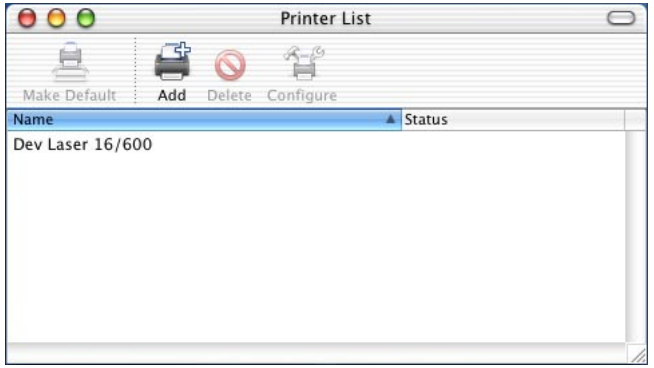
1. If you haven't done so already, copy the relevant PPD file to the Mac OSX workstation (see section 19.1.2).
2. **In Torrent:** broadcast the input queues by selecting **Start Inputs...** from the **Torrent** menu (so there is a tick by it).
3. **On the Mac OSX page make-up workstation:** select **System Preferences** from the Apple menu or the dock at the bottom of the screen to display the following dialog:



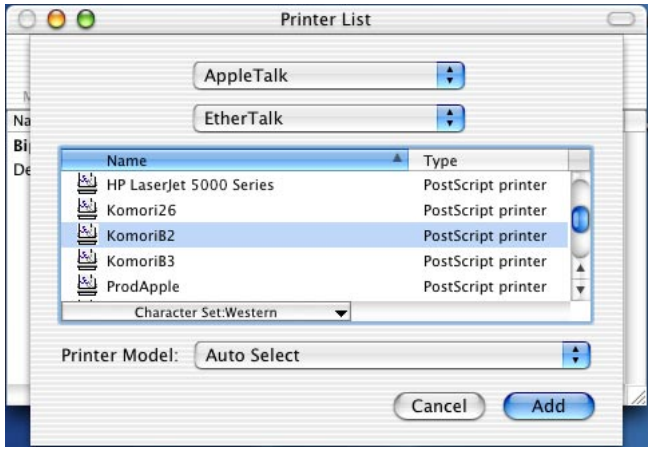
4. Double-click on the **Print & Fax** option to display the following dialog:



5. Click on the **Printing** button then on the **Set Up Printers...** button to display the 'Printer List' dialog:



6. Click on the **Add** icon to display the following window:



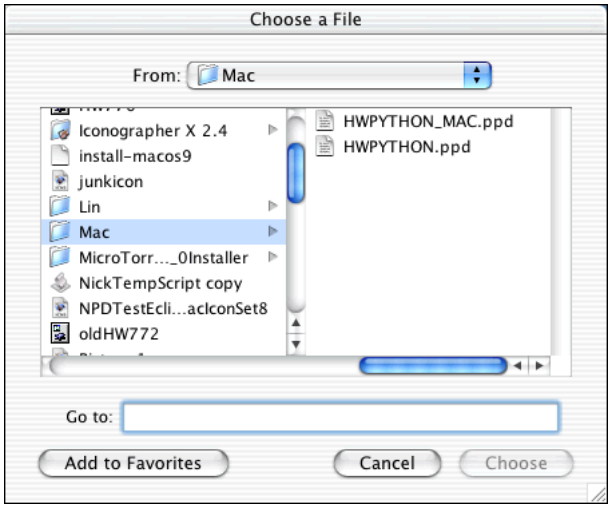
7. Select **AppleTalk** from the top pull-down list.

8. Select the required zone, if applicable, from the second pull-down list.

   Note: This is the zone where the Torrent input queues are broadcast.

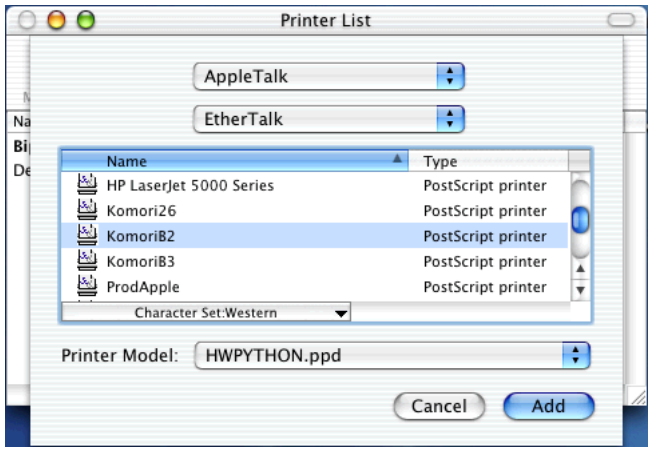
9. Highlight the required Torrent input queue from the list of available printers.

10. From the **Printer Model** menu select **Other...** to display the 'Choose a File' dialog:



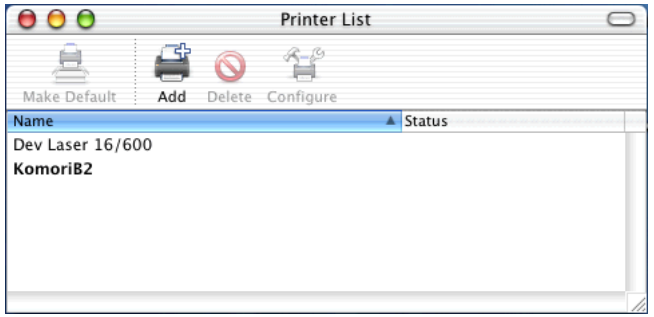
11. Locate the Python PPD file (usually called **HWPYTHON.ppd**) that you copied to the machine earlier, then click on the **Choose** button.

12. You should now see the Python PPD file listed for the Printer Model:



13. Click on the **Add** button.

14. You will see the new printer in the Printer List:



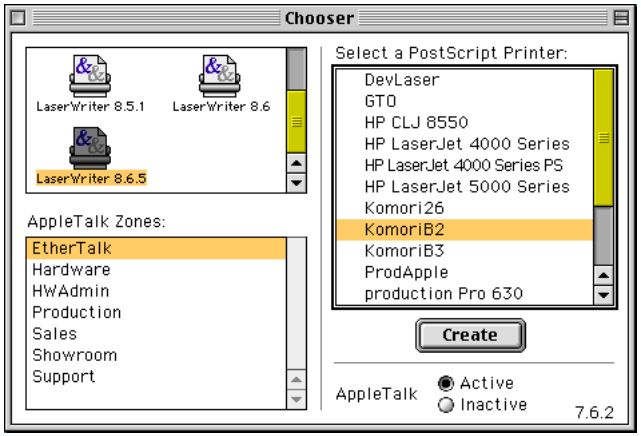
15. You can now print to this printer, as described in section 6.6.

16. Now, create a printer for any other Torrent input queues you wish to print to by following the instructions in this section again.

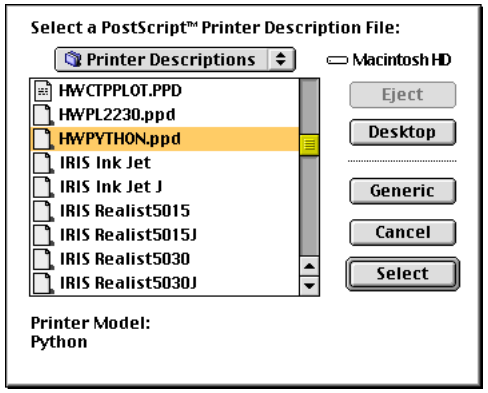
19.3 Creating a print queue on a Mac Classic workstation

To create a print queue on a Mac Classic workstation:

1. If you haven't done so already, copy the relevant Python PPD file to the Mac Classic workstation (see section 19.1.1).
2. **In Torrent:** broadcast the input queues by selecting **Start Inputs...** from the **Torrent** menu (so there is a tick by it).
3. **On the Mac Classic page make-up workstation:** launch the Chooser.
4. Click once on the **LaserWriter** icon in the top left-hand window, and then on the required **AppleTalk Zone**, if applicable. The names of the available Torrent input queues appear in the right-hand window along with any other printers:



5. Highlight the required Torrent input queue in the right-hand window, then click on the **Create** button to display the following dialog:

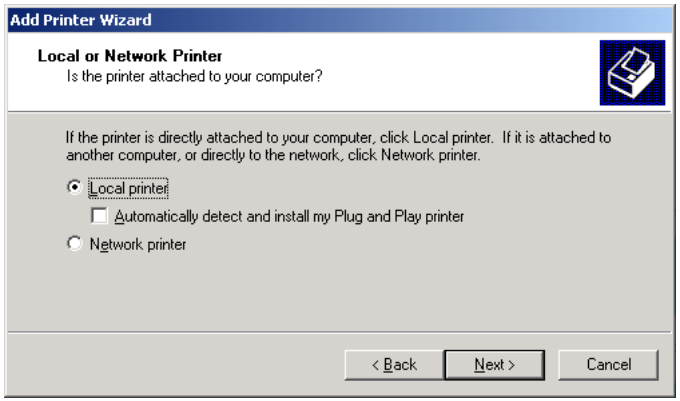


6. Select the Python PPD (usually called **HWPYTHON.ppd**) from the list then click on the **Select** button. The printer is now built on the desktop.
7. Close the Chooser.
8. You can now print to this printer, as described in section 6.6.
9. Now, create a printer for any other Torrent input queues you wish to print to, by following the instructions in this section again.

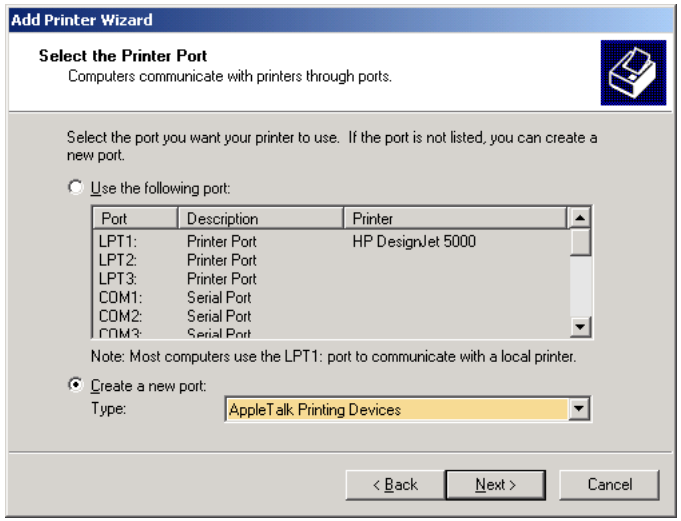
19.4 Creating a print queue on a Windows 2000/Server 2003 workstation

To create a print queue on your Windows 2000/Server 2003 workstation:

1. If you haven't done so already, copy the relevant Python PPD folder to the Windows 2000/Server 2003 workstation (see section 19.1.2).
2. **In Torrent:** broadcast the input queues by selecting **Start Inputs...** from the **Torrent** menu (so there is a tick by it).
3. Depending on your platform do either of the following:
   - **On the Windows 2000 page make-up workstation:** from the **Start** menu, choose **Printers** from the **Settings** sub-menu to display the 'Printers' dialog.
   - **On the Windows Server 2003 page make-up workstation:** from the **Start** menu, choose **Printers and faxes** to display the 'Printers' dialog.
4. Double-click on **Add Printer** to display the Add Printer Wizard. Click on **Next** to display the following dialog:

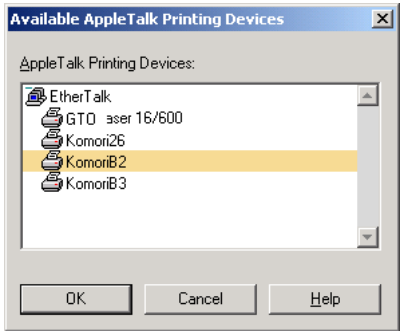


5. Click on the **Local printer** radio button and make sure that the **Automatically detect and install my Plug and Play printer** option is **not** selected. Click on **Next** to display the following dialog:

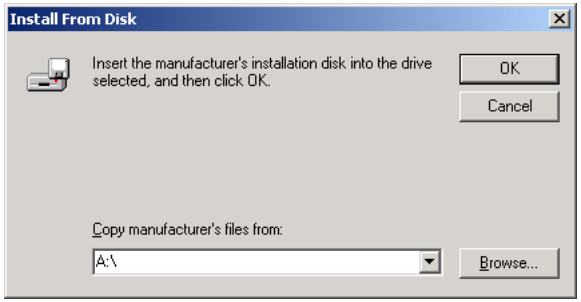


6. Click on the **Create a new port** radio button and select **AppleTalk Printing Devices** from the **Type** pull-down menu. Click on **Next** to display the following dialog:

7. Double-click on the required AppleTalk zone, if applicable, to open the list of printers:



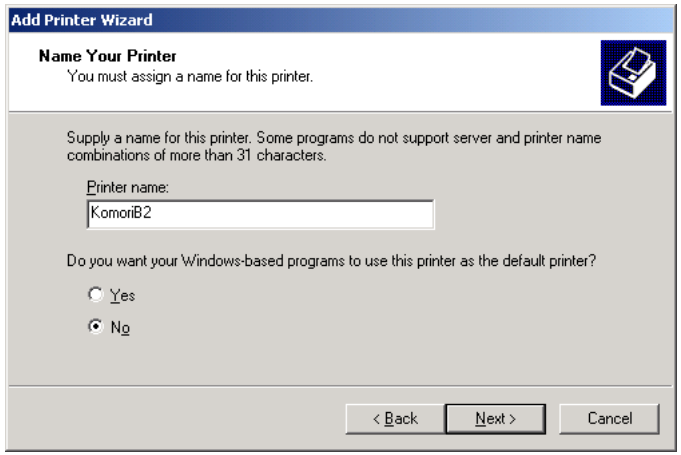
8. Select the required Torrent input queue then click on **OK**.
9. When you are asked if you want to capture the device, click on **No**.
10. Next, you are asked for the printer manufacturer and model. Click on **Have Disk...** to display the 'Install From Disk' dialog:



11. Click on the **Browse...** button to display the 'Locate File' dialog.

    Note: If you get an error 'The device is not ready', click on the **Cancel** button. The 'Locate File' dialog is then displayed.

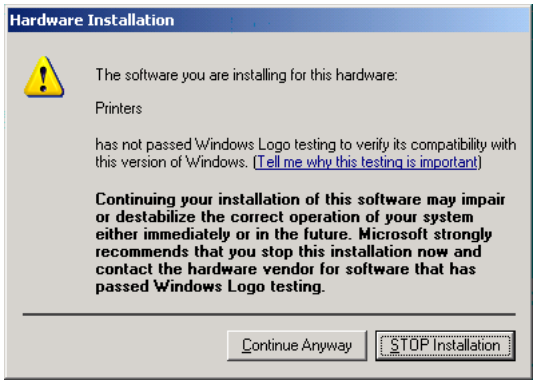
12. In the 'Locate File' dialog, navigate to the **Win2k_XP** PPD folder that you copied to your workstation earlier.
13. Open the folder and select the **setup.inf** file. Click on the **Open** button.
14. Click on **OK** on the 'Install From Disk' dialog.
15. In the Add Printer Wizard, Python should now be listed. Click on the **Next** button to display the following dialog:



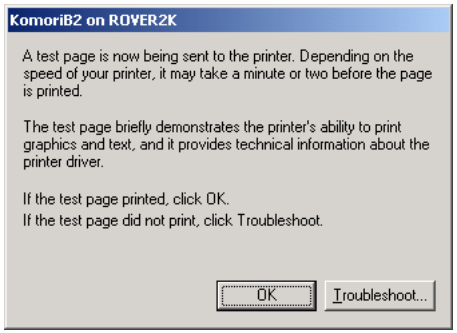
16. Edit the **Printer name** (give it the same name as the Torrent input queue), then select whether you want this to be the default printer. Click on **Next**.
17. Choose whether you want the printer to be shared, then click on **Next**.
18. Choose whether to print a test page (we recommend that you do), then click on **Next**.
19. The Add Printer Wizard now displays a summary of your choices. Check that the information is correct then click on **Finish**.

    Note: If any of the information is incorrect, click on the **Back** button until you get to the required screen, then edit the information.

20. **Windows 2000 setup only:** If the message 'Digital Signature Not Found' appears, click on **Yes** to continue with the installation.
21. **Windows Server 2003 setup only:** If the following warning dialog appears, select the **Continue Anyway** option, otherwise the PPD will not be installed:

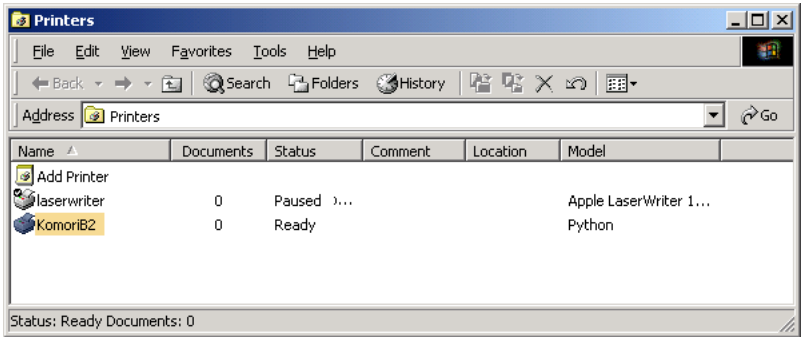


22. If you chose to print a test page, you are asked if the test page printed out correctly:



Check that the test page is in the Torrent RIP's Output Controller, then click on the **OK** or **Troubleshoot...** button, depending on whether the page printed out correctly or not.

23. When you have finished, you will see the printer you have just created in the 'Printers' dialog:



24. You can now print to this printer, as described in section 6.6.
25. Now, create a printer for any other Torrent input queues you wish to print to, by following the instructions in this section again.

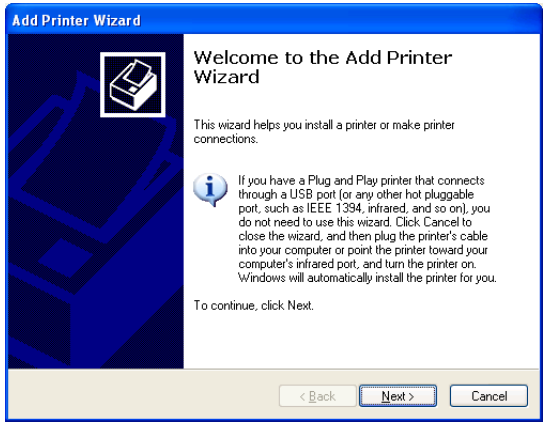
19.5 Creating a print queue on a Windows XP workstation

Notes: On Windows XP you cannot print to the AppleTalk Torrent input queue(s) until you have installed the Miramar AppleTalk Protocol from Miramar Systems Inc.

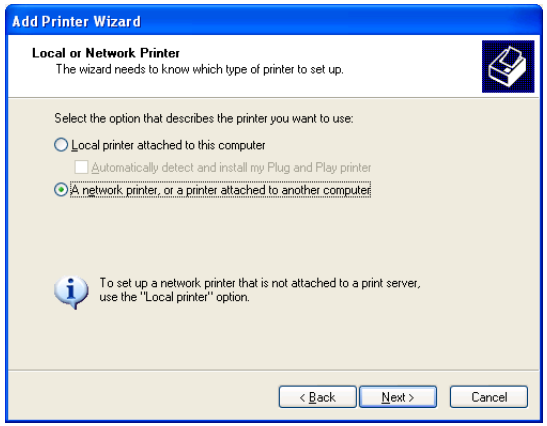
You may need to ask your system administrator for help with installing and setting up the Miramar AppleTalk protocol, or refer to the **Torrent User's Guide** for more information about printing to Torrent.

Once you have installed the Miramar AppleTalk Protocol, you can create the Torrent print queues on your Windows XP workstation, as described below:

1. If you haven't done so already, copy the relevant Python PPD folder to the Windows XP workstation (see section 19.1.2).
2. **In Torrent:** broadcast the input queues by selecting **Start Inputs...** from the **Torrent** menu (so there is a tick by it).
3. **On your Windows XP page make-up workstation:** from the **Start** menu select **Settings > Printers and Faxes**. The 'Printers and Faxes' window is displayed.
4. Select **Add Printer** from the **File** menu. The Add Printer Wizard is displayed:

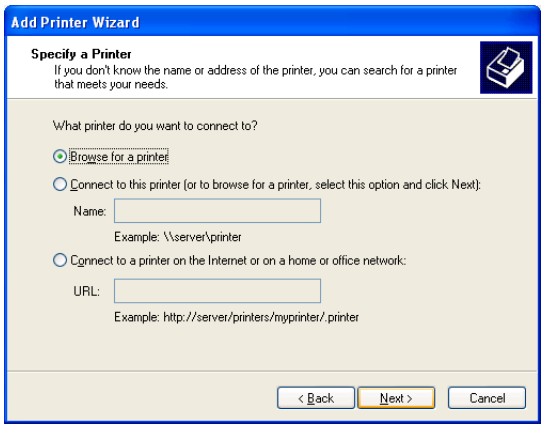


5. Click on the **Next** button to display the following dialog:

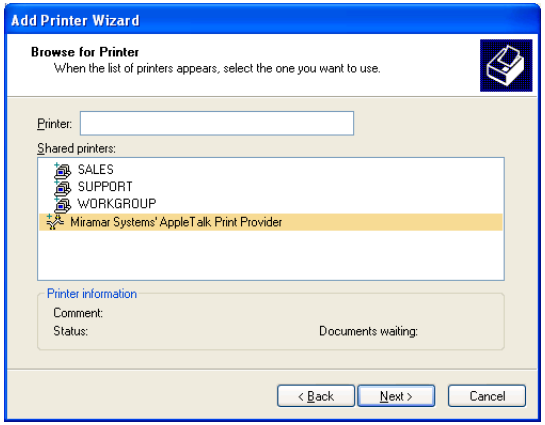


6. Select **A network printer, or a printer attached to another computer** then click on **Next**.

7. The following dialog is displayed:



8. Select the **Browse for a printer** option then click on the **Next** button. The system searches for printers on the network and displays a list of what it finds, for example:



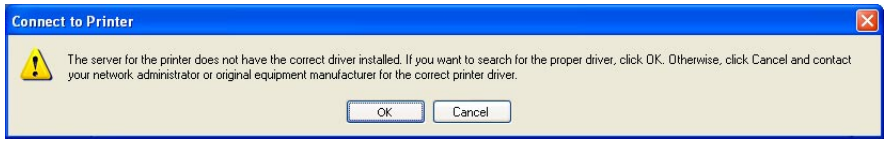
9. Double-click on the **Miramar Systems' AppleTalk Print Provider** option to display the list of available AppleTalk zones.

   Note: Please see your system administrator if the Miramar option is not there.

10. If necessary, double-click on the required AppleTalk zone to display the list of available printers. The published Torrent input queue(s) should be listed here.

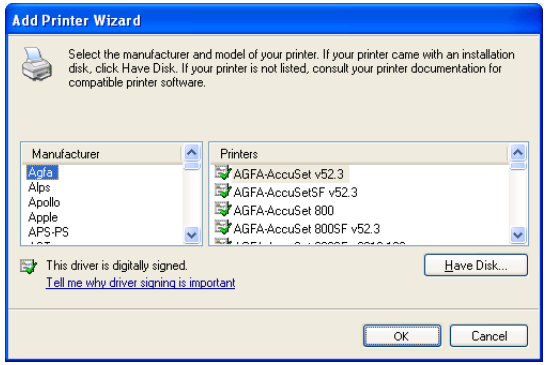
11. Highlight the required Torrent input queue and click on **Next**.

12. You will see the following message asking you to select a printer driver file:



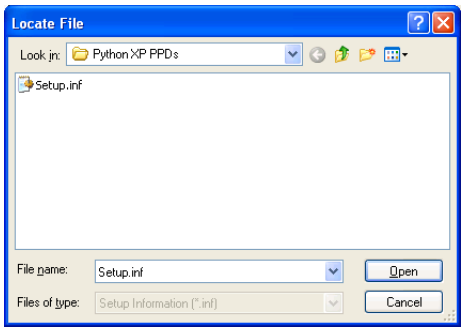
13. Click on **OK**.

14. Next, you are asked to locate the printer driver file. Click on **OK** to search for the file. The following dialog is displayed:

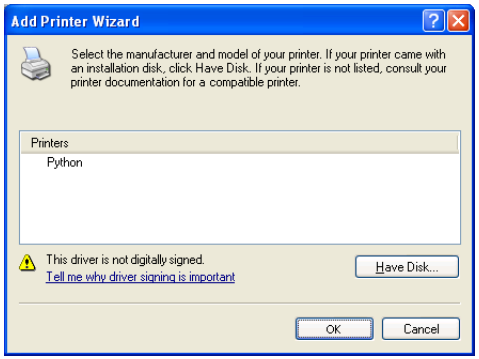


15. Click on **Have Disk...** to display the 'Install From Disk' dialog.

16. Click on the **Browse...** button to display the 'Locate File' dialog:



17. On the 'Locate File' dialog, navigate to the Python PPD folder that you copied to your workstation earlier, open the folder and select the **setup.inf** file. Click on the **Open** button.
18. On the 'Install From Disk' dialog you will now see the Python print queue. Click on **OK**.
19. You should now see the Python print queue listed in the Add Printer Wizard, for example:



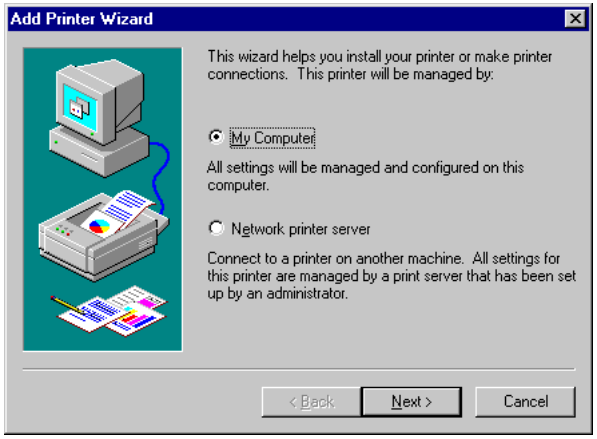
20. Click on **OK**.
21. Choose whether the printer is to be the default printer: select **Yes** or **No** then click on **Next**.
22. Next, a dialog is displayed summarizing the choices you have made. Click on **Finish**.
23. In the 'Printers and Faxes' dialog you should now see the printer you have just created.
24. You can now print to this printer, as described in section 6.6.
25. Now, create a printer for any other Torrent input queues you wish to print to by following the instructions in this section again.

19.6 Creating a print queue on a Windows NT workstation

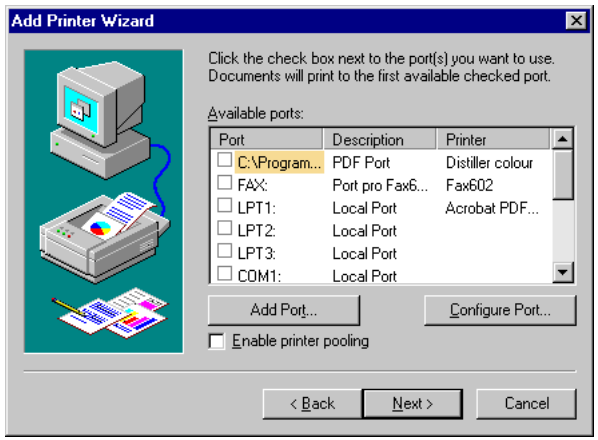
Note: You may need the Windows NT Installation disk to successfully complete this installation.

To create a print queue on a Windows NT workstation:

1. If you haven't done so already, copy the relevant Python PPD folder to the Windows NT workstation (see section 19.1.2).
2. **In Torrent:** broadcast the input queues by selecting **Start Inputs...** from the **Torrent** menu (so there is a tick by it).
3. **On your Windows NT page make-up workstation:** from the Windows **Start** menu, choose **Printers** from the **Settings** sub-menu to display the 'Printers' dialog.
4. Double-click on **Add Printer**. The Add Printer Wizard is displayed:

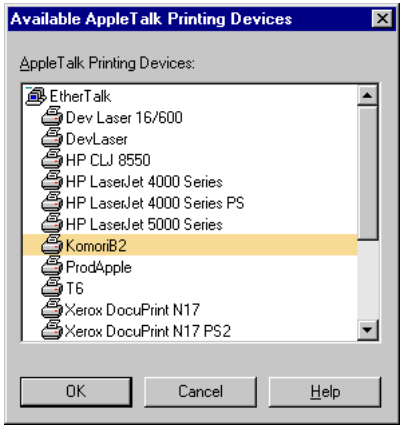


5. Select **My Computer** then click on **Next**.
6. The currently available ports are displayed:

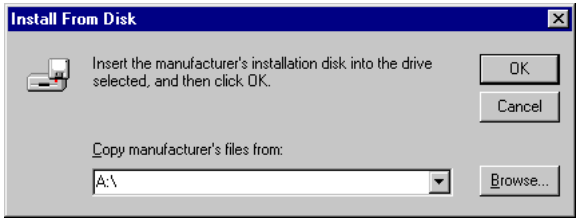


7. Click on the **Add Port...** button.
8. Double-click on **AppleTalk Printing Devices** to display the list of available zones, for example:

9. Double-click on the AppleTalk zone, if applicable, where the Torrent input queues are broadcast to display the following dialog:



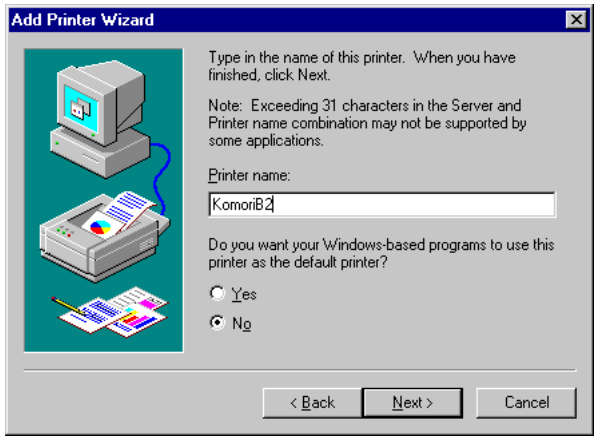
10. Highlight the required Torrent input queue and click on **OK**.
11. Next, you are asked if you want to capture the device. Click on **No**.
12. On the 'Printer Ports' dialog, click on **Close**.
13. On the 'Add Printer Wizard' dialog, click on **Next**.
14. You are now asked to select the printer manufacturer and model. Click on **Have Disk...** to display the 'Install From Disk' dialog:



15. Click on the **Browse...** button.

    Note: If you get an error 'The device is not ready', click on the **Cancel** button. The 'Locate File' dialog is then displayed.

16. In the 'Locate File' dialog, navigate to the Python PPD folder that you copied to your workstation earlier, open the folder and select the **setup.inf** file. Click on the **Open** button.
17. Click on **OK** on the 'Install From Disk' dialog.
18. You should now see Python listed in the 'Add Printer Wizard'. Click on **Next** to display the following dialog:



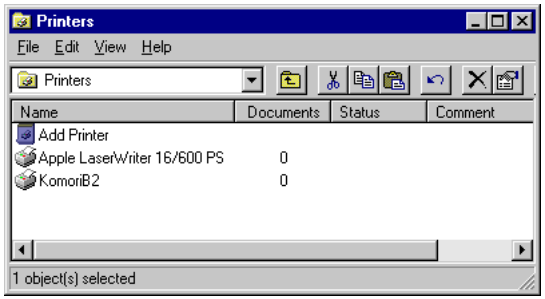
19. Type in a new printer name (use the same name as the Torrent input queue) then select whether you want this printer to be your default printer. Click on **Next**.
20. When you are asked if the printer will be shared or not, select **Not shared** then click on **Next**.
21. Choose whether to print a test page (we recommend that you do), then click on **Finish**.

22. While the printer is being configured, you may be asked to insert the Windows NT Installation CD in order to complete the installation.

    Note: You may need to locate the required files on the CD.

23. If you chose to print a test page, you are asked if the test page printed correctly. Check this in the Torrent RIP's Output Controller, then click on the **Yes** or **No** button, depending on whether the page printed correctly or not.

24. You will now see the printer you have just created in the 'Printers' dialog:



25. You can now print to this printer, as described in section 6.6.

26. Now, create a printer for any other Torrent input queues you wish to print to by following the instructions in this section again.

19.7 Creating a print queue on a Windows 95/98 workstation

If you are printing from a workstation running Windows 95 or 98 you need to carry out the following steps:

1. Set up an input queue in the Torrent RIP using NTprint.
2. Broadcast the input queue then generate a shared printer on a Windows NT or Windows 2000 machine using the ScriptWorks protocol.
3. Set up the printer on your Windows 95/98 machine.

Please see your system administrator for more help and refer to the **Torrent User's Guide** for more information about using NTprint.

Alternatively, you can use a software application such as PC MACLAN to print directly to the RIP, via AppleTalk, without having to set up an NTprint queue. Refer to the **Torrent User's Guide** for more information about this.

Appendix A. Creating the Imager queue

This appendix shows you how to create the Imager queue (in the Python Console, this queue outputs jobs to the Python platesetter).

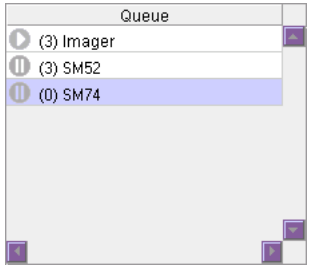
There must be only **one** Imager queue and, typically, it has already been created by the installation engineer. However, if your system data gets lost, becomes corrupted or you inadvertently delete or edit the Imager queue, you will need to re-create it. You can do this either by copying configuration files (which contain definitions for the Imager queue) to a specific location, or by manually creating the Imager queue using the Queue Configuration application.

This chapter includes the following sections:

A.1 Checking for the Imager queue

To check if the Imager queue exists:

1. In the Python Console, click on the **Queues** tab.
2. In the 'Queue' list, you should see the Imager queue plus any TicketMaker queues that have been created, for example:



3. If the Imager queue is not in this list then you need to create it, as described in this chapter. Alternatively, if there is an Imager queue but you think its settings may be wrong you can check them from the instructions given in section A.3.

A.2 Creating the Imager queue by copying configuration files

You can create the Imager queue by copying specific configuration files into the required directory. However, in doing this, all existing TicketMaker queues will be deleted, which means you will have to re-create them. If you have a lot of TicketMaker queues or do not wish to re-create them, then create the Imager queue manually, as described in the next section. Otherwise, to create the Imager queue by copying configuration files:

1. In Windows Explorer, navigate to D:\Python\Engine v5.x.x\preconfigured settings\Q2 v4.4 Config Files.

2. Copy the following files to D:\Python\Q2v4.4:

   Applications.cfg
   Preferences.cfg
   Queues.cfg

   Note: If you are asked to overwrite existing files, click on **Yes**.

   WARNING: Make sure you COPY rather than MOVE the required files to D:\Python\Q2v4.4 otherwise you will lose the backup copies.

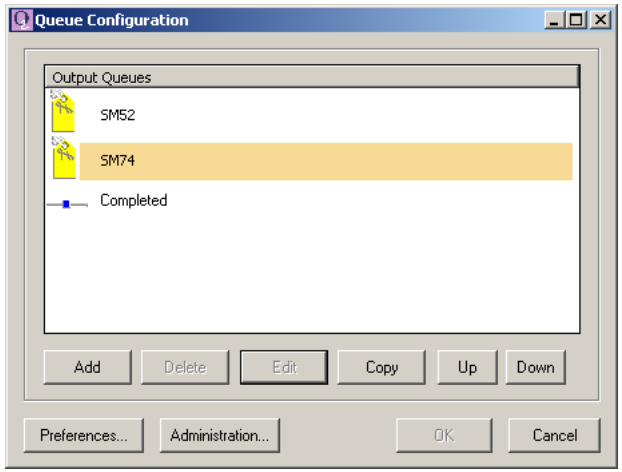
3. If necessary, re-create any TicketMaker queues, as described in chapter 14.

A.3 Creating an Imager queue using the Queue Configuration application

Note: Before you start, make sure that the Completed queue exists, as described in Appendix B.

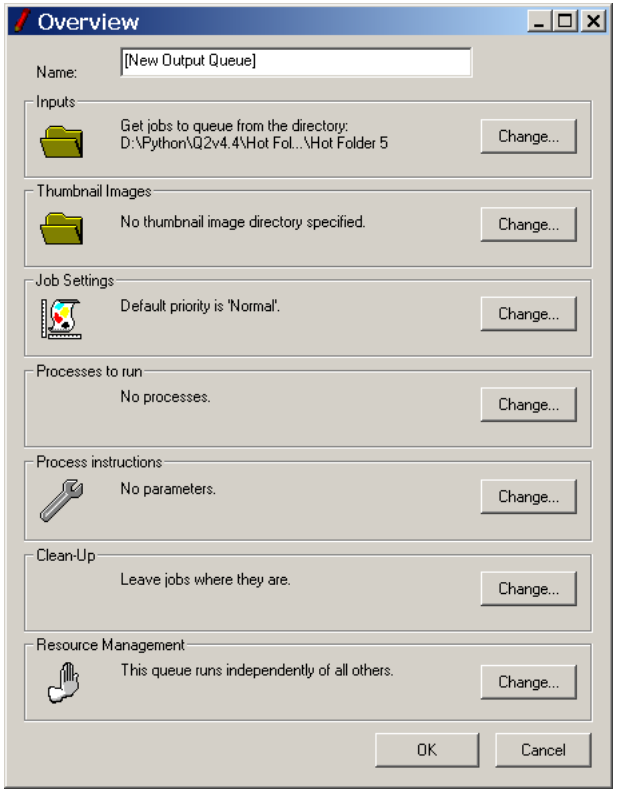
To create a new Imager queue using the Queue Configuration application:

1. Launch the Queue Configuration application, as described in section 7.7. The 'Queue Configuration' window opens:



   Note: The 'Queue Configuration' window shows any queues that have already been created. In this example, two TicketMaker queues and the Completed queue are shown.

2. Click on the **Add** button to open the 'Overview' dialog:



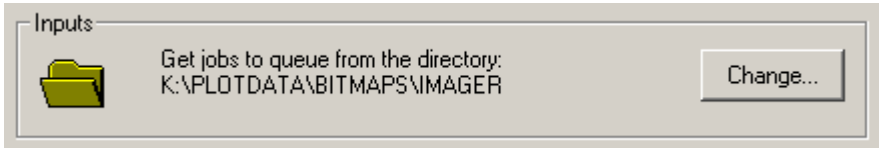
This dialog contains a number of panels, which are described in the following sub-sections.

Name

In the **Name** field, type **Imager** for the queue's name.

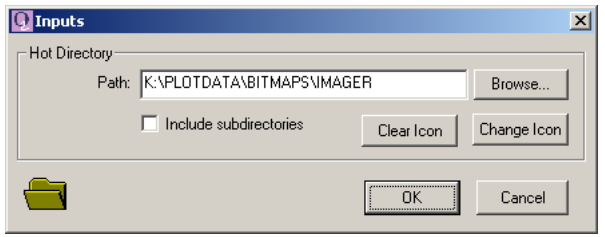
WARNING: This queue must be called Imager**.**

Inputs



The 'Inputs' panel specifies the directory where the Python Console should look for files to be processed. To specify the directory:

1. Click on the **Change...** button to display the following dialog:



2. Here, you need to specify the K:\PLOTDATA\BITMAPS\IMAGER directory. Either type the full pathname of this directory into the **Path** field, or click on the **Browse...** button to display the 'Queue Hot Directory' dialog, then locate and highlight the directory and click on **OK**.

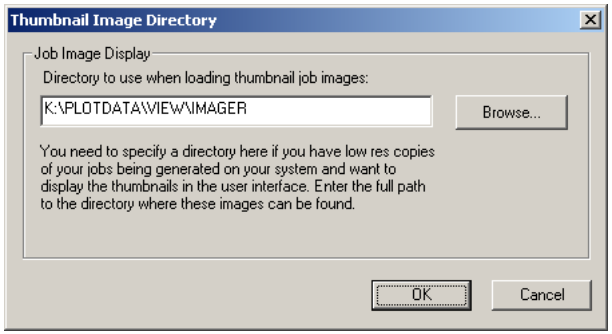
3. In the 'Inputs' dialog, check that the correct directory name is now shown.
4. Click on **OK** when you have finished. You will now see the directory listed in the 'Overview' dialog.

Thumbnail Images

The LRG application creates job thumbnail images, which are displayed in the Python Console (see section 5.4.1 for more details about the LRG application).

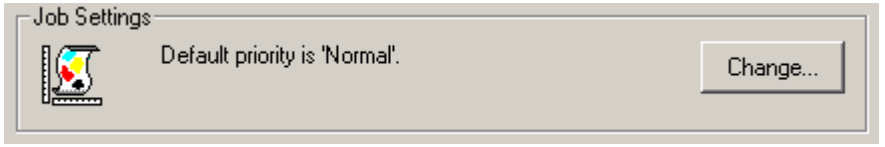
To specify the location of the thumbnail images:

1. Click on the **Change...** button to display the following dialog:



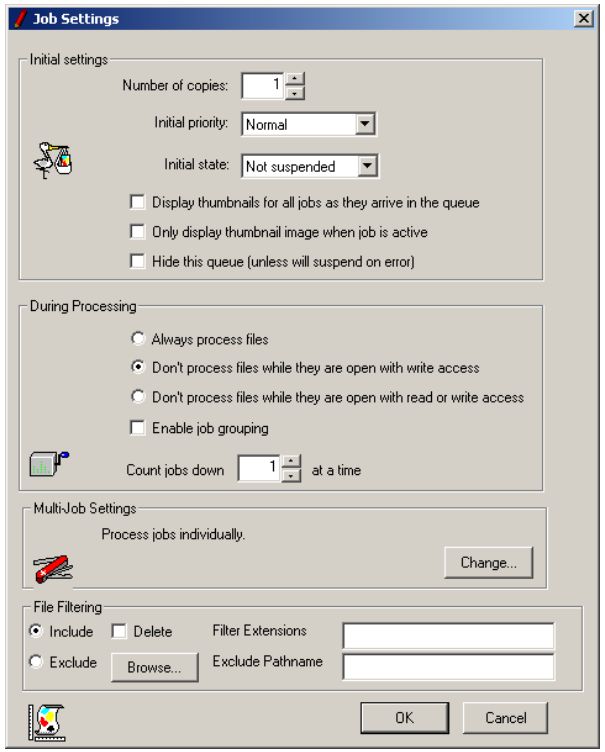
2. Click on the **Browse...** button and locate the thumbnails directory, K:\PLOTDATA\VIEW\IMAGER.
3. Click on **OK** when you have finished. You will now see the thumbnail images directory listed in the 'Overview' dialog.

Job Settings



The 'Job Settings' let you specify options, such as a job's priority. These settings will be applied to all jobs arriving in the queue. To specify the job settings:

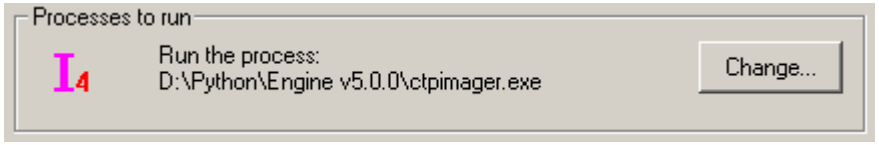
1. Click on the **Change...** button. The following dialog is displayed:



2. In the 'Initial settings' panel, set the following options:
   - Set the **Number of copies** to **1**.
   - Set the **Initial priority** to **Normal**.
   - Set the **Initial state** to **Not Suspended**.
   - Do not set the **Display thumbnails for all jobs as they arrive in the queue** option.
   - Do not set the **Only display thumbnail image when job is active** option.
   - Do not set the **Hide this queue (unless will suspend on error)** option.
3. In the 'During Processing' panel, select the **Don't process files while they are open with write access** option.

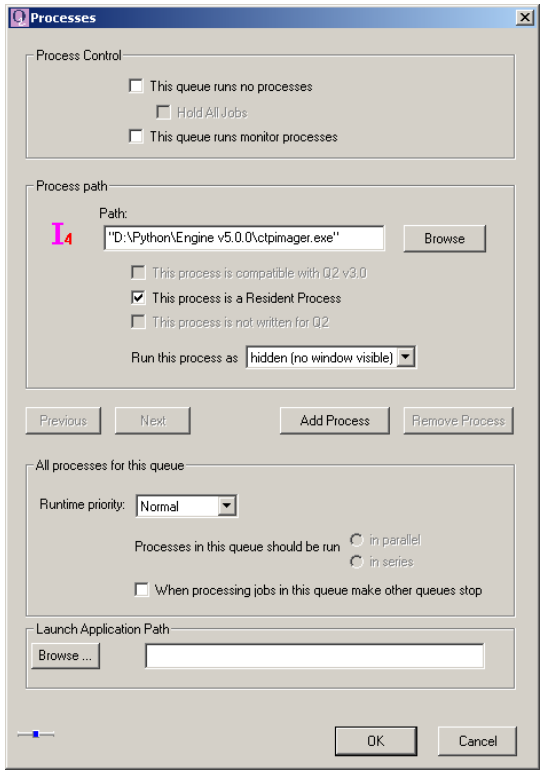
   Make sure that no other options are set in this panel.
4. Do not change the 'Multi-Job Settings' option — make sure it is set to **Process jobs individually**.
5. Do not set any 'File Filtering' options.
6. When you have finished, click on **OK** to return to the 'Overview' dialog.

Processes to run



This panel specifies the application that processes your jobs:

1. In the 'Processes to run' panel, click on the **Change...** button to display the following dialog:



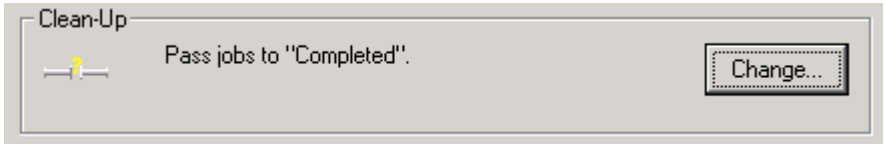
2. Do not set any of the options in the 'Process Control' panel.
3. In the 'Process Path' panel, click on the **Browse** button and locate the **ctpimager.exe** file (this can be found at D:\Python\Engine v5.x.x\).
4. The box **This process is a Resident Process** MUST be checked.

   Note: If the **This process is compatible with Q2 v3.0** box is checked, you need to deselect it before you can select the required option.
5. Set the **Run this process as** option to **Hidden (no window visible)**.
6. Set the **Runtime priority** to **Normal**.
7. Set all other options as shown in the 'Processes' dialog shown previously.
8. When you have finished, click on **OK**. The 'ctpimager' application is now shown in the 'Overview' dialog.

Process instructions

The 'Process instructions' panel lets you specify the parameters that the Imager application should run with. As the parameters are always passed to Imager from the TicketMaker queues, there is no need to set any process instructions.

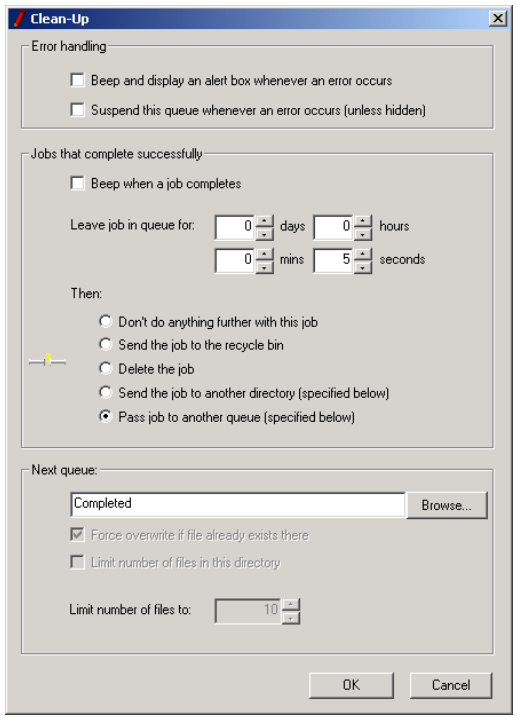
Clean-Up



The 'Clean-Up' panel lets you specify that after a job has been output to the Python platesetter, it will be moved to the Completed queue. (Make sure the 'Completed' queue already exists. See Appendix B.)

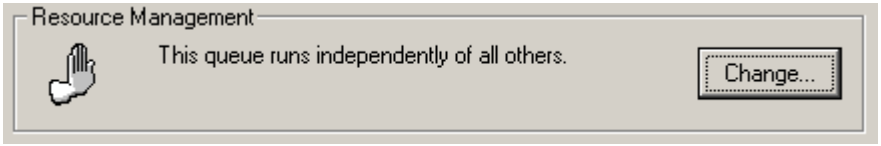
To set the clean-up options:

1. Click on the **Change...** button to display the 'Clean-Up' dialog:



2. Do **not** select the options in the 'Error handling' panel.
3. In the 'Jobs that complete successfully' panel, select **Pass job to another queue (specified below)**.
4. In the 'Next queue' panel, click on the **Browse...** button and choose the **Completed** queue from the 'Choose a Queue' dialog. Click on **OK**.
5. Do not set any other options in this dialog (the previous dialog shows the correct settings).
6. When you have finished, click on **OK**. You will now see the clean-up directory listed in the 'Overview' dialog.

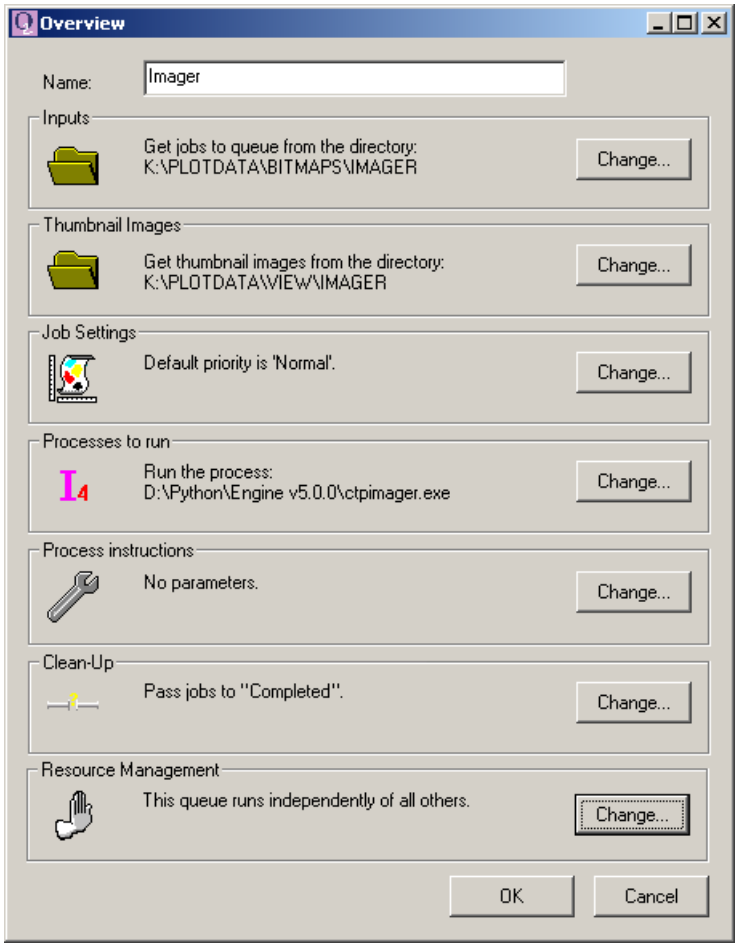
Resource Management



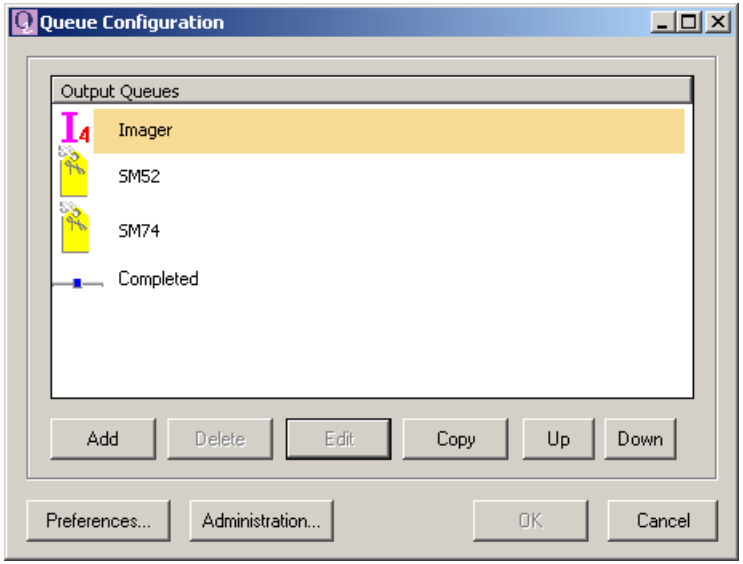
No 'Resource Management' options need to be set for the Imager queue.

A.4 The finished queue

When you have finished creating the Imager queue, the completed 'Overview' dialog will look like this:



Click on **OK** to return to the 'Queue Configuration' dialog. You will now see the Imager queue listed in the 'Queue Configuration' window:



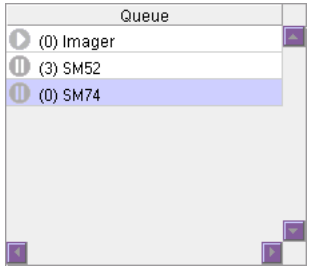
Note: If there are no TicketMaker queues then these should be created now using the Queue Configuration application (refer to chapter 14 for full instructions).

When you have finished creating the required queues, you can quit out of the 'Queue Configuration' dialog by clicking on the **OK** button.

A.5 Checking the new queue

Now, you should make sure that the new Imager queue is set up correctly in the Python Console. To do this:

1. Go to the Python Console's 'Queues' tab. You should see the new Imager queue you have just created:



2. The Imager queue is now ready to output jobs to the Python platesetter.

Note: When you have finished creating new queues, we recommend that you save the new information. Section 16.2 shows you how to do this.

Appendix B. Creating the Completed queue

This appendix shows you how to create the Completed queue (in the Python Console, jobs are moved to the 'Completed' queue once they have been successfully processed. You can see all successfully processed jobs in the 'Completed plates' tab).

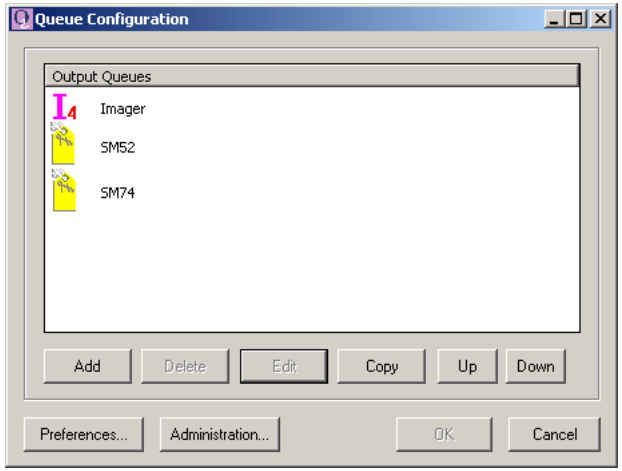
There must be only **one** Completed queue and, typically, it has already been created by the installation engineer. However, if your system data gets lost, becomes corrupted or you inadvertently delete or edit the Completed queue, you will need to re-create it. You can do this either by copying configuration files (which contain definitions for the Completed queue) to a specific location, or by manually creating the Completed queue using the Queue Configuration application.

This appendix shows you how to create the Completed queue and includes the following sections:

B.1 Checking for a Completed queue

To check if the Completed queue exists:

1. Launch the Queue Configuration application, as described in section 7.7. The 'Queue Configuration' window opens showing all the queues that have been created, for example:



2. If there is no Completed queue, then you need to create it, as described in this chapter. Alternatively, if there is a Completed queue but you think its settings may be wrong you can check them from the instructions given in section B.3.

B.2 Creating the Completed queue by copying configuration files

You can create the Completed queue by copying specific configuration files into the required directory. However, in doing this, all existing TicketMaker queues will be deleted, which means you will have to re-create them. If you have a lot of TicketMaker queues or do not wish to

re-create them, then create the Completed queue manually, as described in the next section. Otherwise, to create the Completed queue by copying configuration files:

1. In Windows Explorer, navigate to D:\Python\Engine v5.x.x\preconfigured settings\Q2 v4.4 Config Files.
2. Copy the following files to D:\Python\Q2v4.4:

   Applications.cfg

   Preferences.cfg

   Queues.cfg

   Note: If you are asked to overwrite existing files, click on **Yes**.

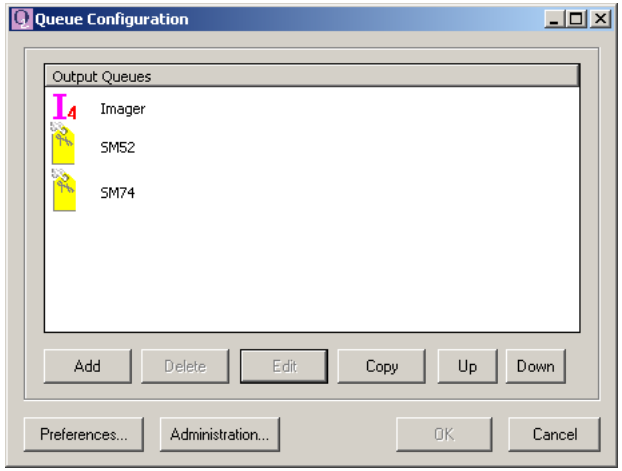
   WARNING: Make sure you COPY rather than MOVE the required files to D:\Python\Q2v4.4 otherwise you will lose the backup copies.

3. If necessary, re-create any TicketMaker queues, as described in chapter 14.

B.3 Creating the Completed queue using the Queue Configuration application

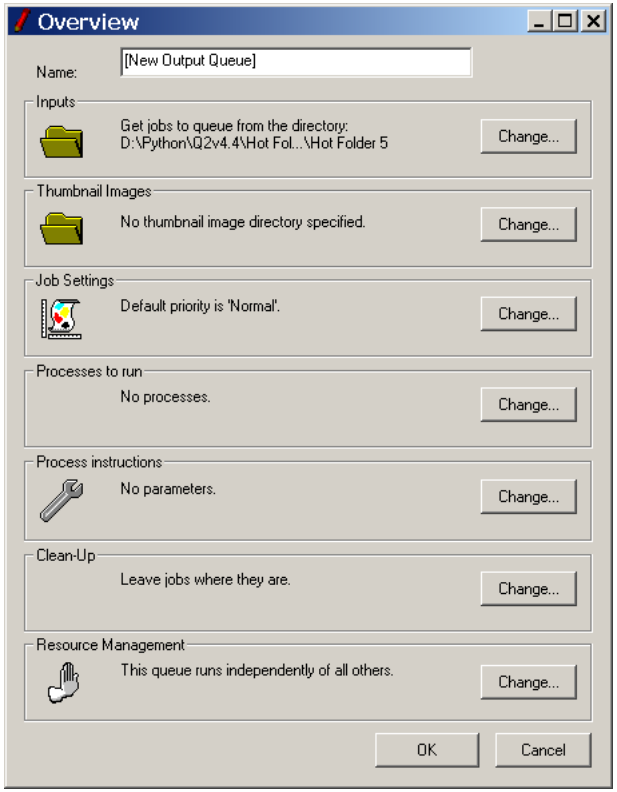
To create a new Completed queue using the Queue Configuration application:

1. If necessary, launch the Queue Configuration application, as described in section 7.7. The 'Queue Configuration' window opens:



Note: The 'Queue Configuration' window shows any queues that have already been created. In this example, the Imager queue and two TicketMaker queues are shown.

2. Click on the **Add** button to open the 'Overview' dialog:



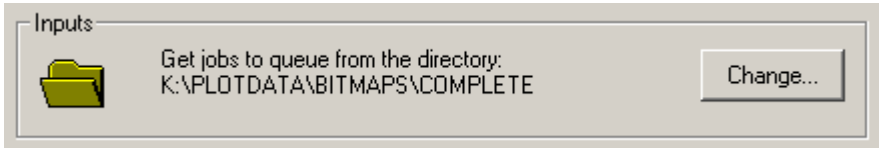
This dialog contains a number of panels, which are described in the following sub-sections.

Name

In the **Name** field, type **Completed** for the queue's name.

WARNING: This queue must be called Completed**.**

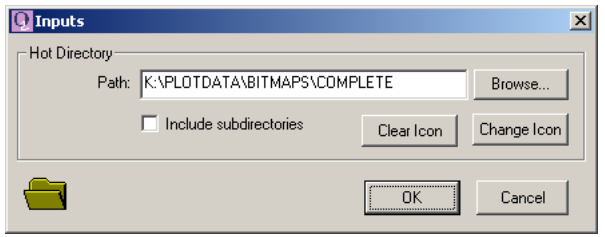
Inputs



The 'Inputs' panel specifies the directory where the Python Console should look for files.

To specify the directory:

1. Click on the **Change...** button to display the following dialog:



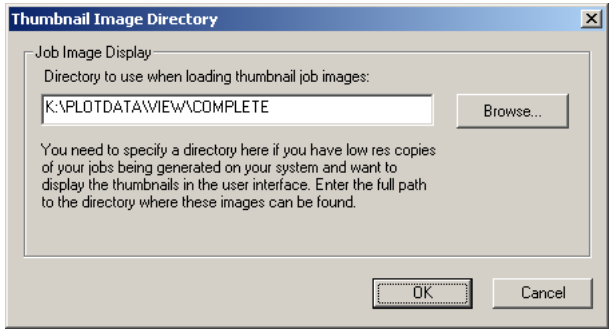
2. Here, you need to specify the K:\PLOTDATA\BITMAPS\COMPLETE directory. Either type the full pathname of this directory into the **Path** field, or click on the **Browse...** button to display the 'Queue Hot Directory' dialog, then locate and highlight the directory and click on **OK**.
3. In the 'Inputs' dialog, check that the correct directory name is now shown.
4. Click on **OK** when you have finished. You will now see the directory listed in the 'Overview' dialog.

Thumbnail Images

The LRG application creates job thumbnail images for display in the Python Console (see section 5.4.1 for more details about the LRG application).

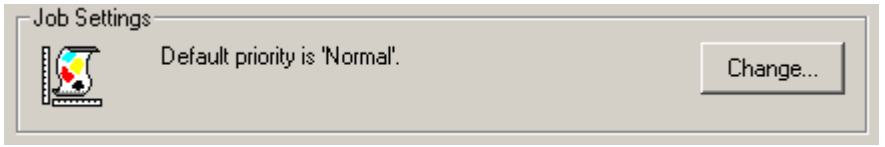
To specify the location of the thumbnail images:

1. Click on the **Change...** button to display the following dialog:



2. Click on the **Browse...** button and locate the thumbnails directory, K:\PLOTDATA\VIEW\COMPLETE.
3. Click on **OK** when you have finished. You will now see the thumbnail images directory listed in the 'Overview' dialog.

Job Settings

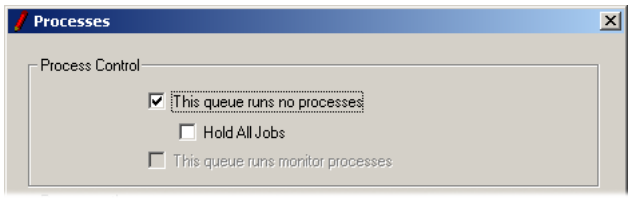


No 'Job Settings' need to be set for the Completed queue.

Processes to run

The 'Processes to run' panel specifies the application that processes your jobs:

1. In the 'Processes to run' panel, click on the **Change...** button.
2. In the 'Process Control' panel, select the **This queue runs no processes** option.

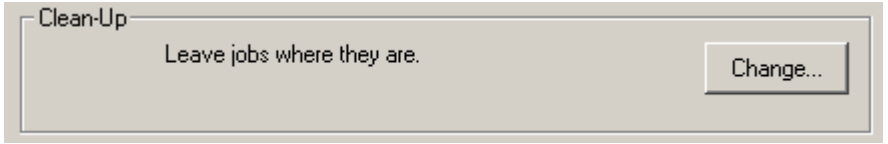


3. Click on **OK** to return to the 'Overview' dialog.

Process instructions

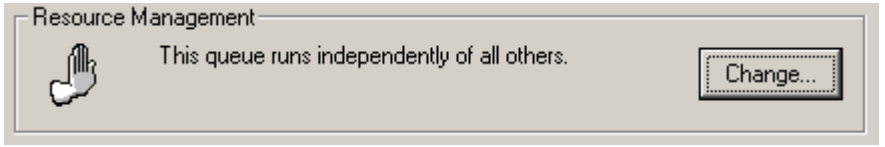
For the Completed queue, there is no need to set any 'Process instructions'.

Clean-Up



For the Completed queue, there is no need to set any 'Clean-Up' options (they should be set to **Leave jobs where they are**).

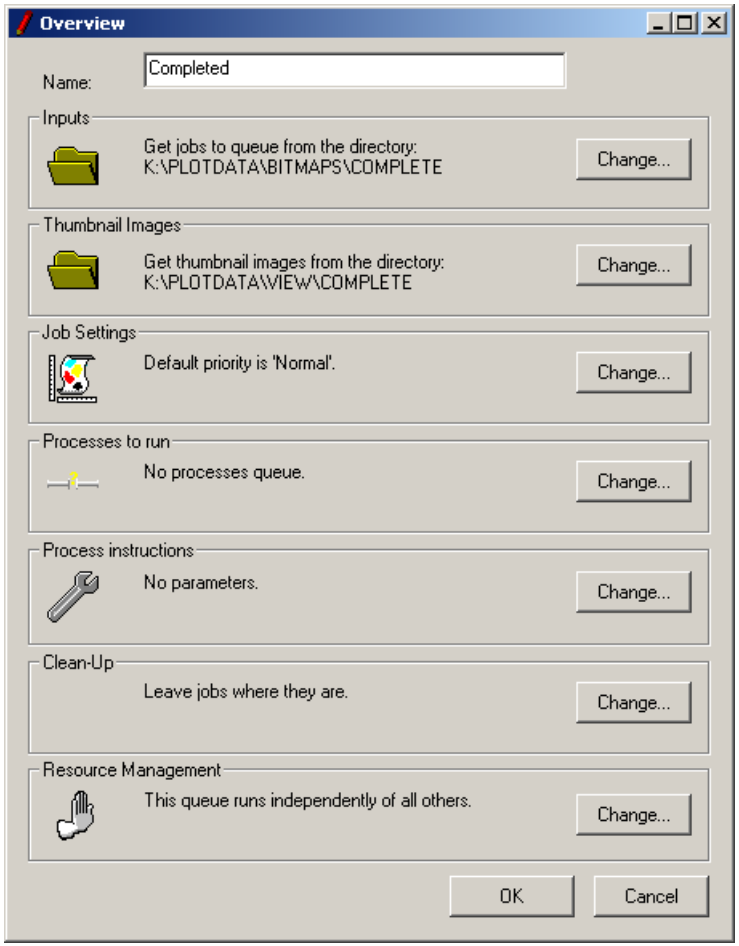
Resource Management



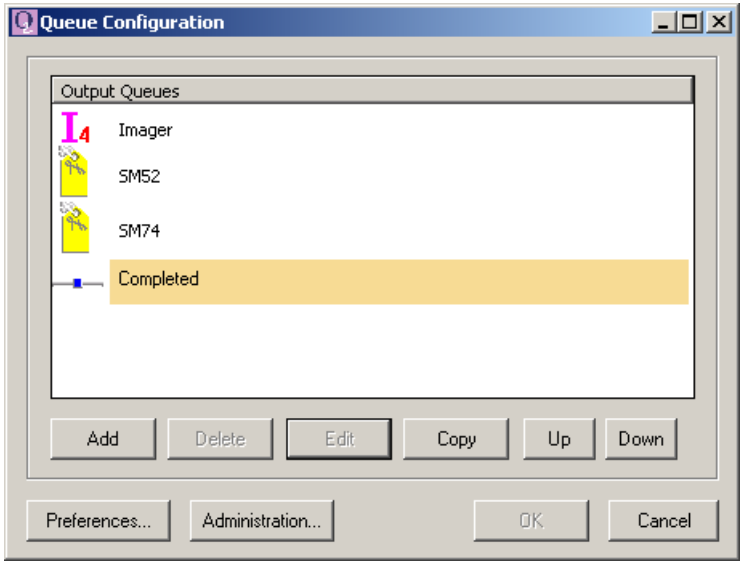
For the Completed queue, there is no need to set any 'Resource Management' options.

B.4 The finished queue

When you have finished creating the Completed queue, the 'Overview' dialog will look like this:



Click on **OK** to return to the 'Queue Configuration' dialog. You will now see the Completed queue in the 'Queue Configuration' window:



If there is no Imager queue then this should be created now using the Queue Configuration application (see Appendix A for instructions).

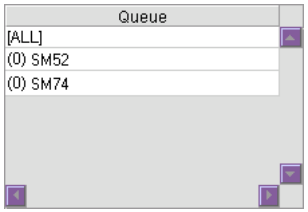
If there are no TicketMaker queues then these should be created now using the Queue Configuration application (see chapter 14 for instructions).

When you have finished, you can quit out of the 'Queue Configuration' dialog by clicking on the **OK** button. You will be returned to the Python Console.

B.5 Checking the new queue

Now, you should make sure that the Completed queue is set up correctly in the Python Console. To do this:

1. Go to the Python Console's 'Completed plates' tab.
2. If you have set up the Completed queue correctly, you will see any TicketMaker queues listed (assuming some have been created), for example:



Note: When you have finished creating new queues, it is a good idea to save the new information. Section 16.2 shows you how to do this.

Appendix C. Setting up Barcode Plate Requeue

The Barcode Plate Requeue (BPR) program adds a barcode to your plates so that you can locate and re-output jobs with ease.

This chapter describes:

Note: Using BPR to re-output plates to the Python platesetter is covered in section 7.5.

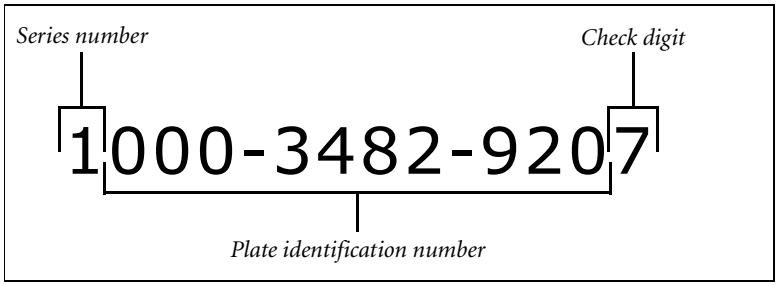
C.1 An introduction to BPR

BPR lets you add a barcode with a unique reference number to jobs that are output to the Python platesetter. If a job needs to be requeued in the Python Console (for instance, if a plate gets damaged), you can simply do this by scanning the barcode off the plate or manually typing in the barcode number.

The barcode is 80 x 6 mm in size and looks like this:

It is made up of the barcode itself and a 12 digit number representing the barcode.

The 12 digit number is made up as follows:

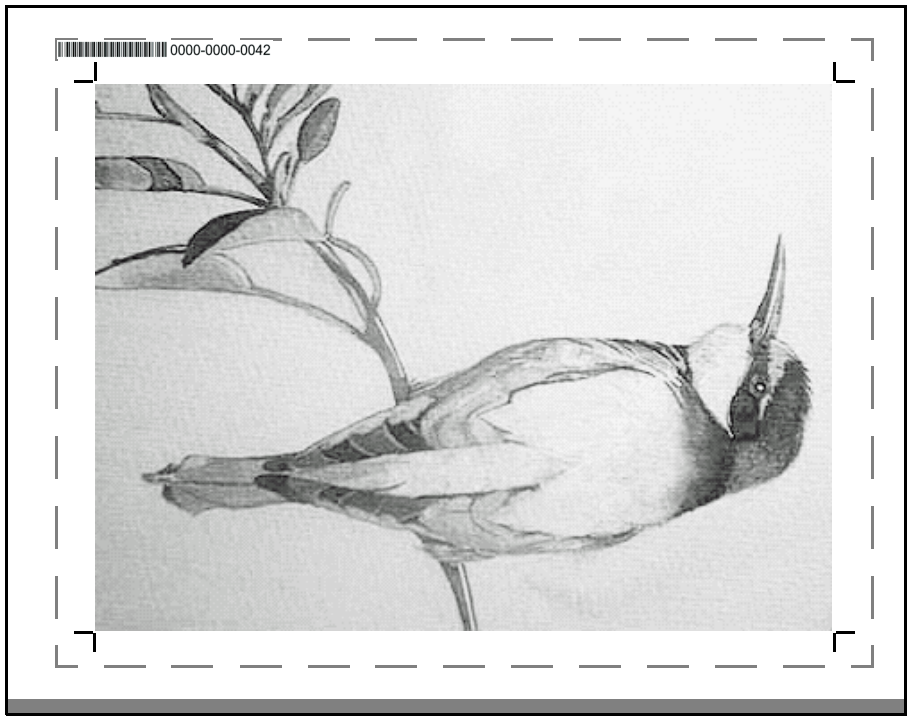


- The **series number** (0-9) allows you to have up to ten systems running BPR, all applying unique barcode numbers. The series number also lets you identify which system a plate has come from. (See section C.3 for more information about setting the series number.)
- The **plate identification number** is used to identify the plate/image. This number starts at 0 and increments by 1 for each subsequent output.
- The final number in the barcode is the **check digit**.

The record of a plate's barcode is held for a certain amount of time since the plate was last output (this time limit is specified in the BPRConfig Preferences, as described in section C.2). If this time limit is exceeded, a 'Cannot find the plate...' message is displayed when you attempt to requeue a plate using its barcode in the Python Console. Refer to section 7.5 for more details about this.

C.1.1 Barcode positioning

The barcode can be positioned anywhere in the image that will be exposed to the plate (regardless of how the image is oriented on the plate), for example:

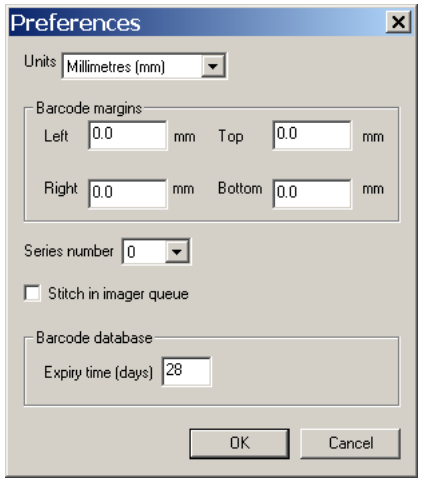


However, you need to position the barcode so that it avoids crop marks, registration marks and the image itself. (Positioning the barcode is covered in section C.3.)

C.2 Setting the BPR preferences

To set the BPR preferences:

1. Launch the BPR Config software (refer to section 7.7 for details). The main 'BPRConfig' dialog is displayed.
2. Click on the **Preferences...** button to display the following dialog:



3. Select the measurement **Units** (choose from **Millimetres** or **Inches**).
4. It should not be necessary to specify margins to be added around the barcode. However, if you want to set margins, use the **Left**, **Right**, **Top** and **Bottom** fields in the 'Barcode margins' panel.

   Note: If you set margins, ensure that the barcode will not print over the job image or any crop or registration marks.
5. If you wish to use BPR on more than one system, with each system generating unique barcodes, select a **Series number** (between **0** and **9**) from the pull-down list. This series number will be the first digit on all barcodes output from the particular Python machine.

6. **Do not** select the **Stitch in imager queue** option.
7. Set the **Expiry time (days)** field to specify how long the plate's barcode will be held since the plate was last output (this will allow you to requeue the plate, if required, as described in section 7.5).

   Note: When this time limit is exceeded, a 'Cannot find the plate...' message is displayed when you attempt to requeue a plate using its barcode in the Python Console.

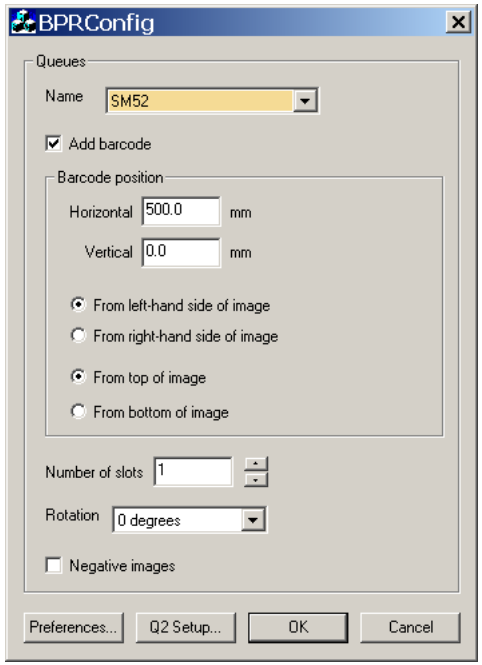
When you have finished setting the preferences, click on **OK** to return to the 'BPRConfig' dialog.

C.3 Configuring BPR

Note: Before you start, make sure that you have created the required Python Console queues, as described in chapter 14.

To configure BPR to add barcodes to the jobs output by the Python platesetter:

1. Launch BPR Config if it is not already running (refer to section 7.7 for details on how to do this). The main 'BPRConfig' dialog is displayed:

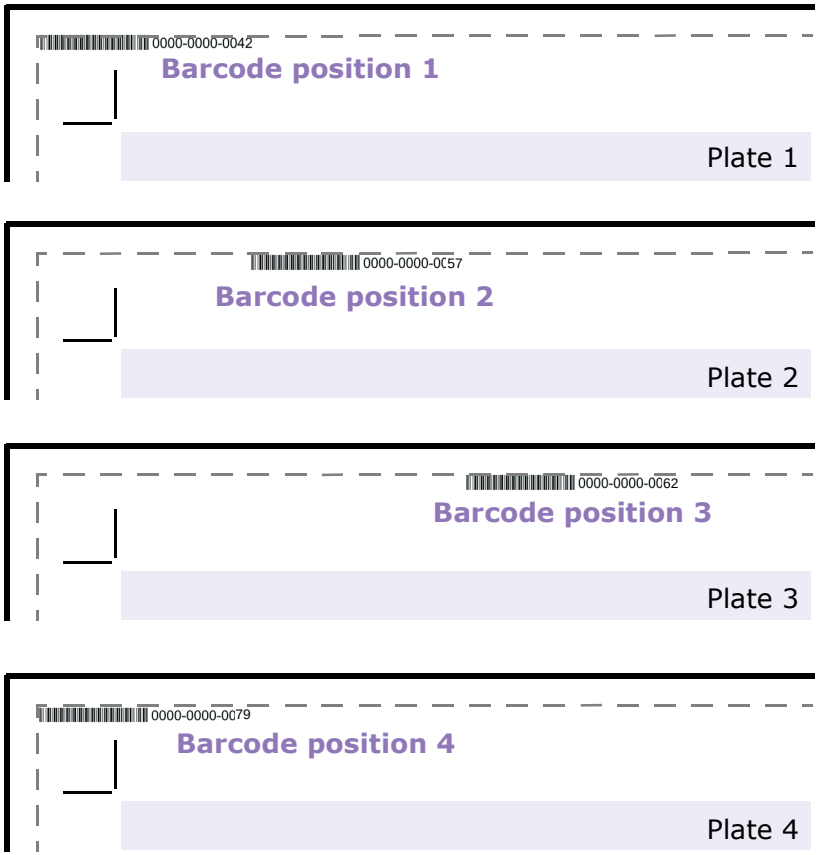


2. From the **Name** pull-down list, select the required TicketMaker queue (a barcode will be added to all plates in this queue).
3. Check the **Add barcode** box.

4. In the 'Barcode position' panel, use the **Horizontal**, **Vertical**, **From left-hand side of image**, **From right-hand side of image**, **From top of image** and **From bottom of image** fields to specify the position of the barcode. These let you place the barcode in any position and relative to any plate edge.

   WARNING: Position the barcode so that it does not print over crop or registration marks, or the image itself. The barcode is 80 x 6 mm in size. Ensure that no barcode will print outside the image area. If this occurs, the image area for that plate will be increased to include the barcode.

5. The **Number of slots** option lets you place subsequent barcodes in adjacent positions on the plate. (If you require all barcodes to be placed in the same position on the plate, as with previous releases of this software, then just specify **1** slot). If you specify 3 slots, for example, the barcode on subsequent plate outputs will be placed in one of three positions, as shown below:



Notes: Before specifying the number of slots, make sure there is enough clear space on every plate image for the number of barcode positions required. The barcode is 80 x 6 mm in size so:

- For 2 barcode slots you will need a 160 x 6 mm clear space.
- For 3 barcode slots you will need a 240 x 6 mm clear space.
- For 4 barcode slots you will need a 320 x 6 mm clear space.

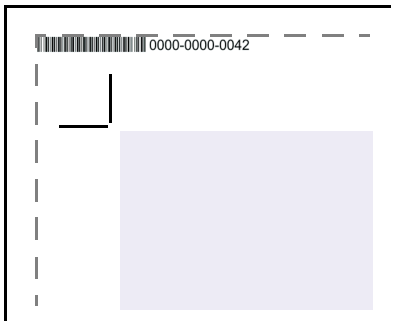
The location of the clear space on the plate for the staggered barcodes depends on the position and rotation of the first barcode (as specified in the main 'BPRConfig' dialog). Within the group of slots, each subsequent barcode will be placed in the same orientation at the end of the previous barcode position.

You can set any number of slots but, in practice, you probably won't require more than 6.

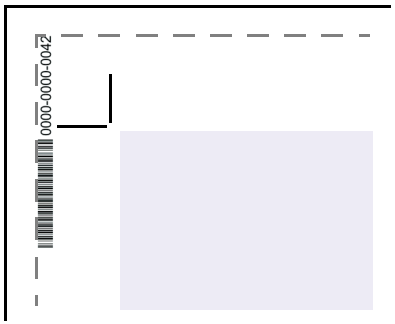
WARNING: If in doubt about specifying the number of slots and where the barcodes will be positioned, you should view the plate images for all slots in HWRoam before the plates are output. Ensure that no barcode will print outside the image area. If this occurs, the image area for that plate will be increased to include the barcode.

---

6. Select the barcode **Rotation** from the pull-down menu. Choose from **0**, **90**, **180** and **270** degrees:

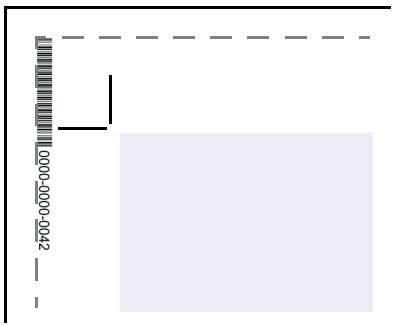


0° rotation



90° rotation

180° rotation



270° rotation

7. Check the **Negative images** box if the barcode is to be output in negative.

   Note: If the barcode is to be read with a barcode reader, it must appear as black stripes and text on a light background on the plate. In nearly all cases, the **Negative images** option should not be selected.

8. If you wish to add barcodes to other Python queues follow the instructions in this section (steps 2 to 7) again.

9. When you have finished, click on **OK** to close the BPR Config program.

A barcode will be added to all plates arriving in the TicketMaker queues that have been configured to add a barcode.

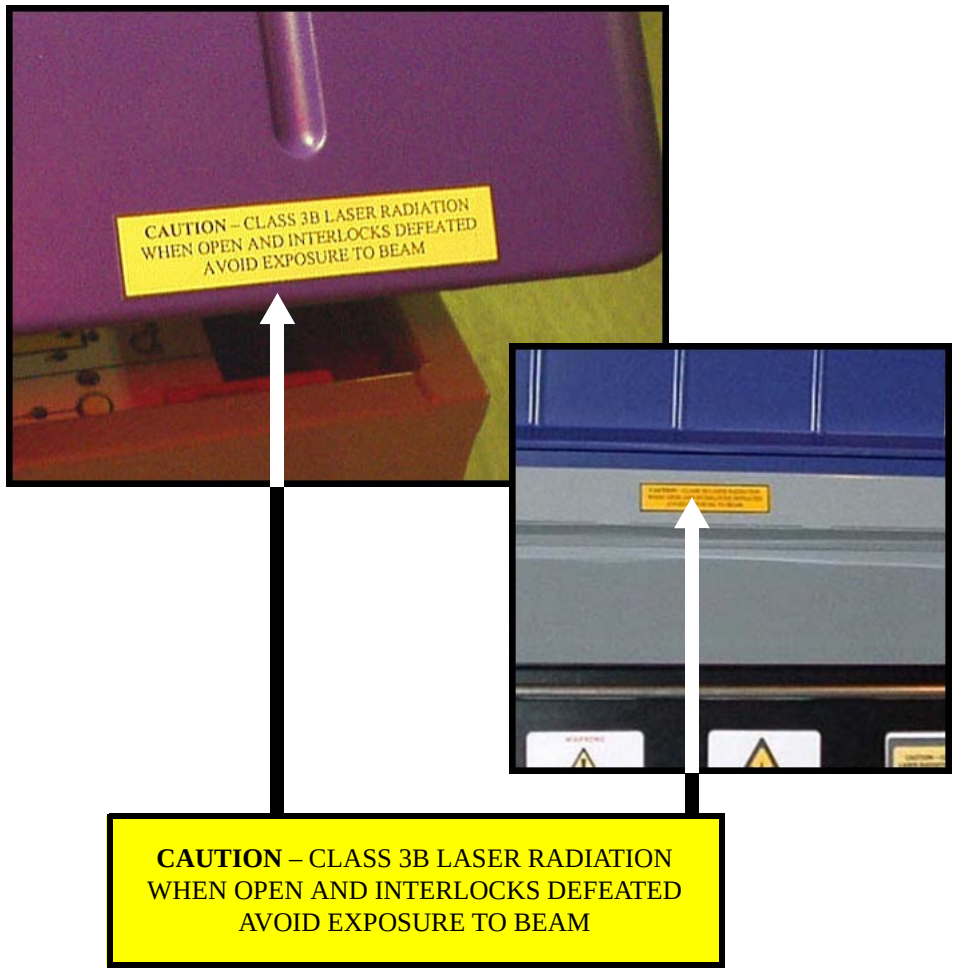
Note: When you configure a queue to add a barcode to the plates, the queue's configuration (as set up using the Queue Configuration program) will be modified. You will see this if you open the 'Processes to Run' panel, which will now contain an additional process (AddBarcode.exe) to run in series with the TicketMaker.exe application.

Appendix D. Python safety labels and conformity certificates

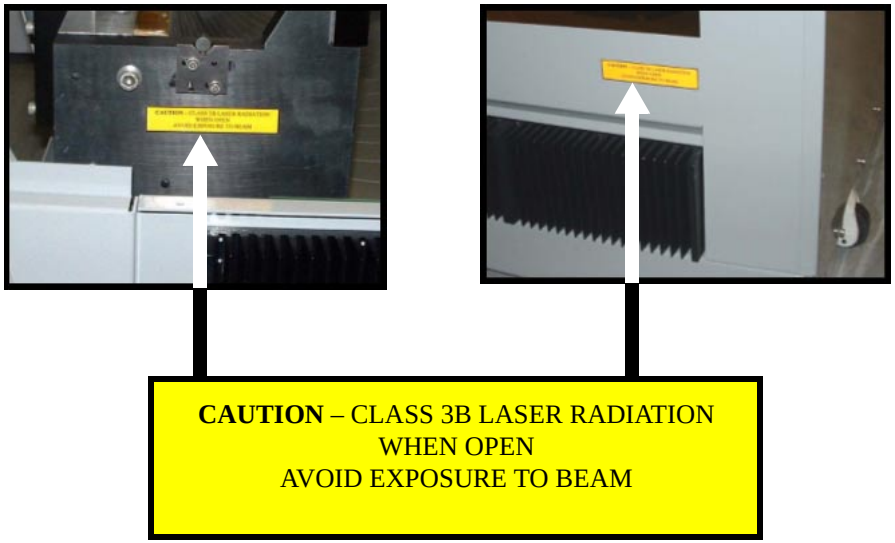
This appendix shows all the labels that are present on the Python engine:

Please ensure that you follow all the necessary safety precautions on the labels.

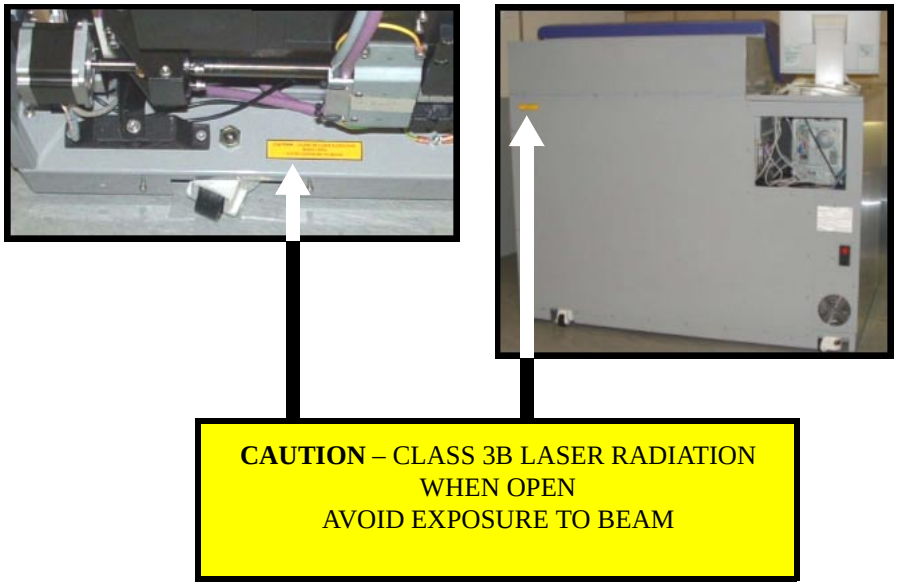
D.1 On Python's lid (inside and outside)



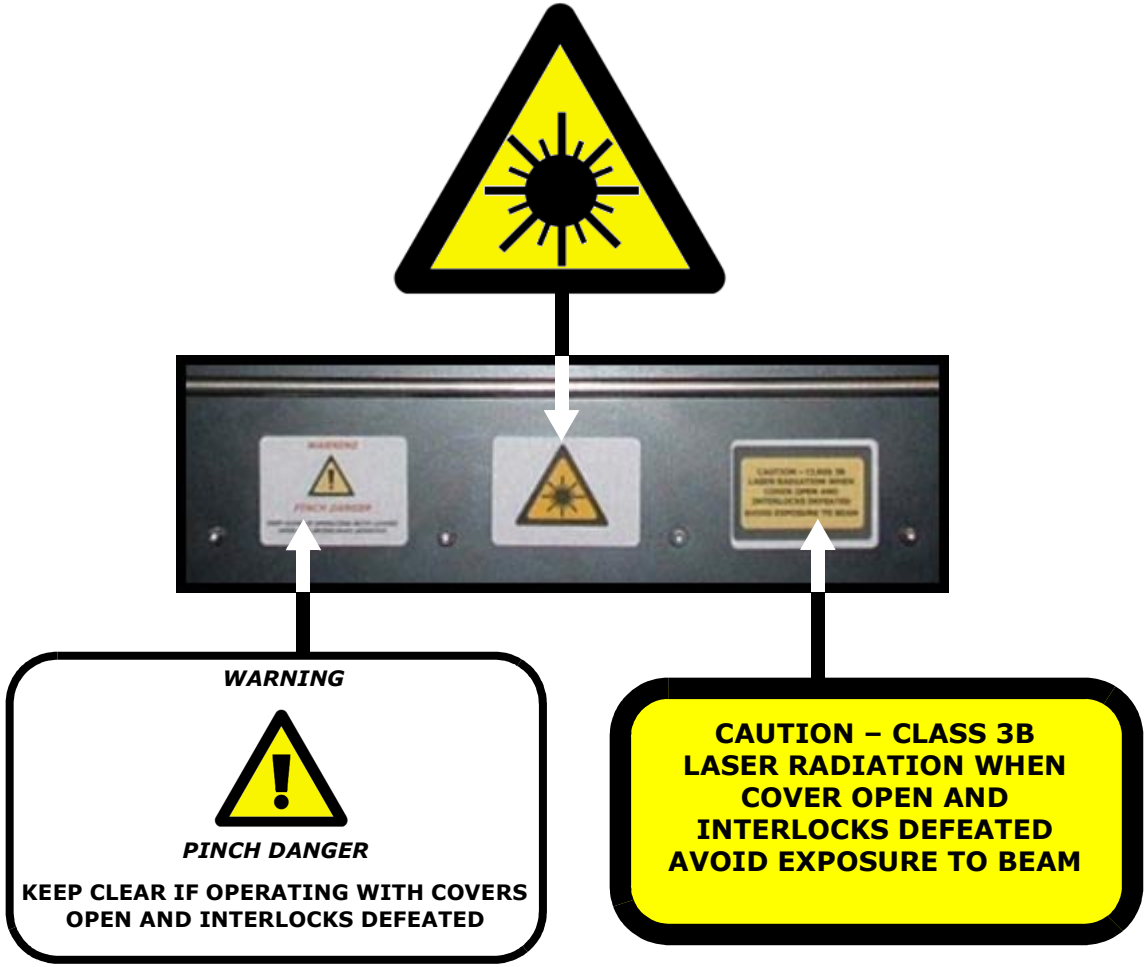
D.2 On the right hand side panel (inside and outside)



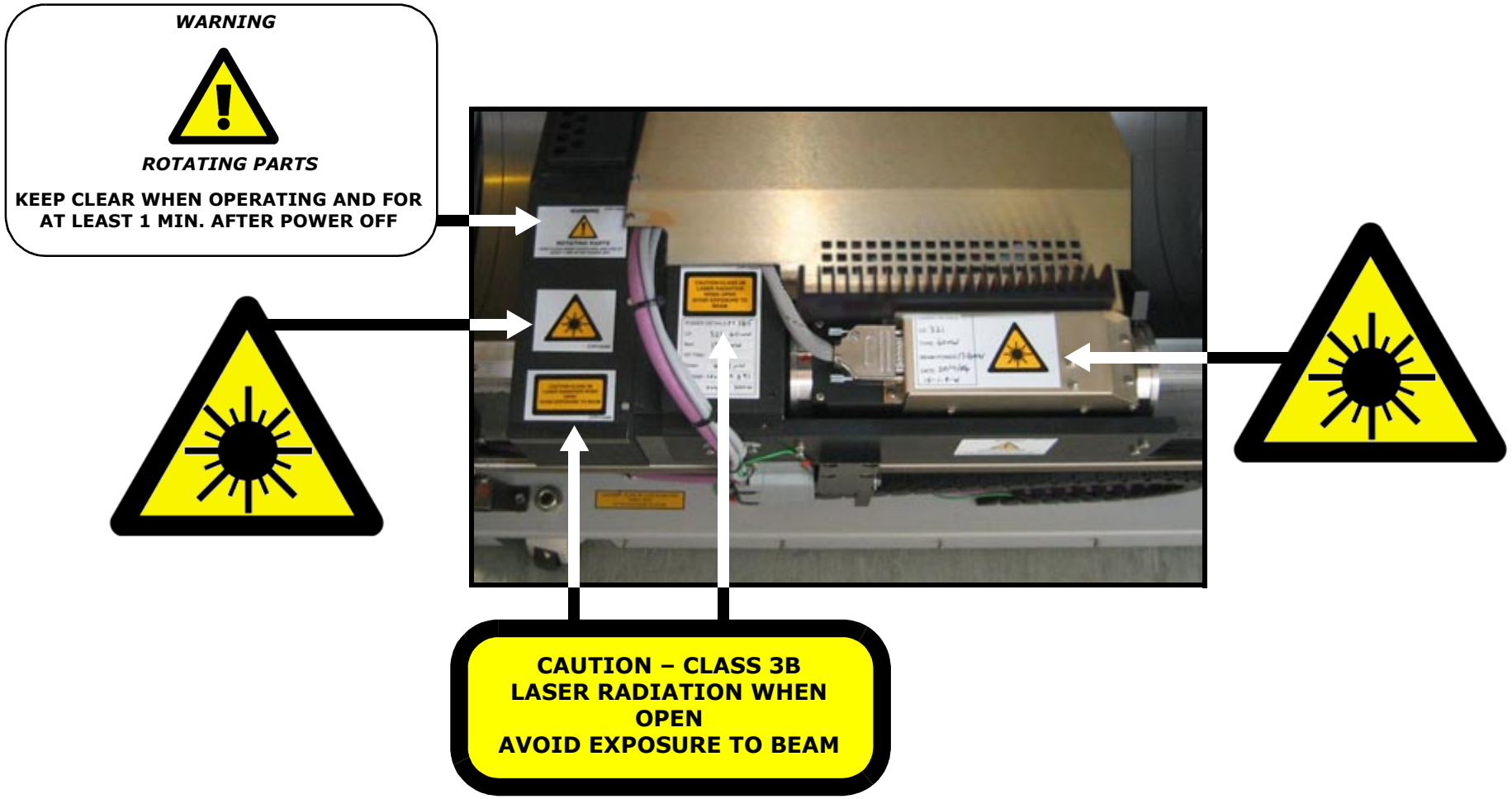
D.3 On the rear panel (inside and outside)



D.4 On the underside of the tilt-table



D.5 On the carriage



D.6 Product ID and Certification labels (EN and IEC versions)

Python 74BV30EN

CE

Computer to Plate Equipment

		Serial Number	PY 397
Rating	230V 50Hz 6 Amps **Fuse in IEC Inlet**	**Date of Manufacture**	November 2004

CAUTION For continued protection against risk of fire, replace only with the same type and rating of fuse

ATTENTION Pour ne pas compromettre la protection contre les risques d'incendie, remplacer par un fusible de même type et de mêmes caractéristiques nominales.

HighWater Designs Ltd
Head Office: St George's Business Park, Alstone Lane, Cheltenham, Glos, GL51 8HF
Tel +44 (0) 1242-542100, Fax +44 (0) 1242-251600
http://www.highwater.co.uk, Email: Info@highwater.co.uk

See installation instructions before connecting to supply
Voir la notice d'installation avant de raccorder au reseau

Complies with
BS EN 60825-1:1994+A1&A2

Fuse 230V – T6.3AH250V

Python 74-BV30-IEC
Computer to Plate Equipment

Rating	110V 60Hz 10 Amps Fuse in IEC Inlet

SERIAL NUMBER	**PY 398**
Date of Manufacture	**November 2004**

CAUTION: For continued protection against risk of fire, replace only with the same type and rating of fuse

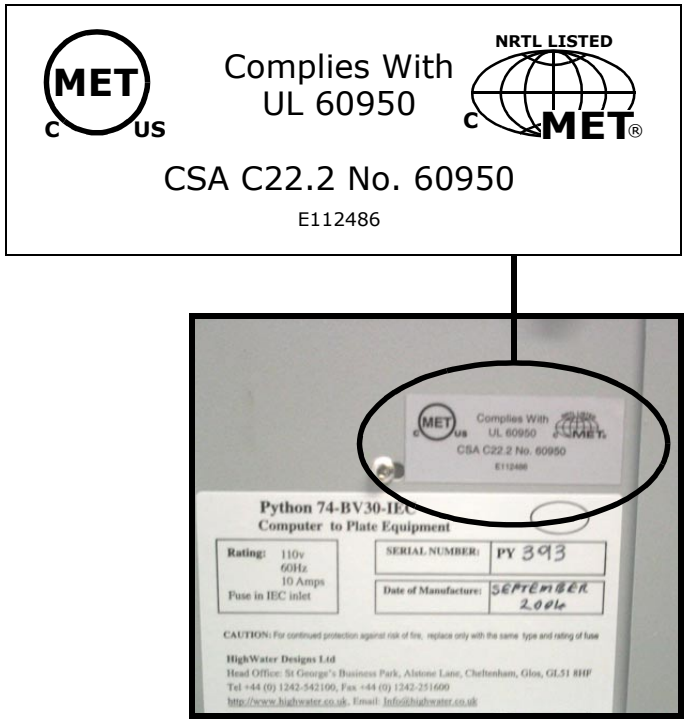
HighWater Designs Ltd
Head Office: St George's Business Park, Alstone Lane, Cheltenham, Glos, GL51 8HF
Tel +44 (0) 1242-542100, Fax +44 (0) 1242-251600
http://www.highwater.co.uk, Email: Info@highwater.co.uk

Complies with: BS EN 60825-1:1994 + A1 and A2 and with 21 CFR 1040.10 and 1040.11

Fuse 110V – T10AH110V

D.7 MET Label

This label is present on 110V machines:



LOW VOLTAGE AND EMC DIRECTIVES

Declaration of Conformity

Manufactured by:

HighWater Designs Ltd.,
St. George's Business Park,
Alstone Lane,
CHELTENHAM,
Gloucestershire
GL51 8HF

PRODUCT CATEGORY: **Computer to Plate Imaging Engine**

PRODUCT NAME: **PYTHON 74-BV-30**

DIRECTIVES COMPLIED WITH:

EMC: 89/336/EEC
LVD: 73/23/EEC as amended by 93/68/EEC

STANDARDS COMPLIED WITH:

EMC	**EN61000-6-4:2001**
	EN61000-3-2:2001
	EN61000-3-3:1995
	FCC CFR47: March 2003 Part 15 Limit A
LVD	**EN60950:1999**
	EN60950:2000
LASER SAFETY	**BS 60825-1**
	CFR 1040.10 & 1040.11

HighWater Designs Ltd declare that the product listed above:

(i) Complies with the protection requirements of the Council EMC Directive, based on the procedures laid down in Article 10 (2) of the EMC Directive and tested to the above standards.

(ii) Complies with the principal safety objective requirements and tested to the above standards.

Signed. .....R G Warren, Managing Director

Full copies of the Low Voltage Directive certificate, plus the EMC, RFI and RFI Immunity certificates are available from HighWater Designs Limited on request.

MACHINERY DIRECTIVE

98/37/EC

Declaration of Conformity

Manufactured by:

HighWater Designs Ltd.,
St. George's Business Park,
Alstone Lane,
CHELTENHAM,
Gloucestershire
GL51 8HF

PRODUCT CATEGORY: **Computer to Plate Imaging Engine**

PRODUCT NAME: **PYTHON 74-BV-30**

DIRECTIVES COMPLIED WITH:

98/37/EC (89/392/EEC, 91/368/EEC, 93/44/EEC, 93/68/EEC)

STANDARDS COMPLIED WITH:

EN61000
EN60950:1999
EN60950:2000
FCC CFR47
BS 60825-1
CFR 1040.10 & 1040.11

HighWater Designs Ltd declare that the product listed above conforms with the Essential Health and Safety Requirements as specified in Annex I of Directive 89/392/EEC as amended and is safe.

Signed:  .....R G Warren, Managing Director

Full copies of the Low Voltage Directive certificate, plus the EMC, RFI and RFI Immunity certificates are available from HighWater Designs Limited on request.

MET Laboratories, Inc. *Safety Certification - EMI - Telecom - Environmental Simulation - NEBS*
914 WEST PATAPSCO AVENUE • BALTIMORE, MARYLAND 21230-3432 • PHONE (410) 354-3300 • FAX (410) 354-3313

June10, 2004

Mr. Cliff Hancox
High Water Designs Ltd
1-6 St, George's Business Park
Alstone Lane
Cheltenham
GLDS
GLSI 8HF

Subject: Product: Laser Imaging System model Python 74BV05IEC
Listing Number: MET Project Number 15045
Safety Standards:
- UL60950, Third Edition," Information Technology Equipment "
- CSA C22.2 No. 60950, Third Edition "Information Technology Equipment

Dear Mr. Hancox

Congratulations on successfully completing the MET Certification process for the Python 74BV05EN. Highwater Designs Ltd may begin to apply the MET Mark on the above stated product at this time in accordance with the MET Mark Utilization Agreement. The reports covering the above stated product will be forthcoming.

Production line testing is required. Refer to the attached excerpt from the report. It is your responsibility to make sure you understand the requirements imposed on manufacturing before the MET certification mark can be applied. If you have any questions, please contact your project engineer prior to producing and labeling the first product.

Thank you for the opportunity to perform this service for Highwater Designs. We look forward to future opportunities with your company.

Sincerely,
MET LABORATORIES, INC.

Tariq Latif
Project Engineer,
Safety Laboratory

Reviewed by:

Rick Cooper
Director of Laboratory Operations,
Safety Laboratory

/ll

Index

Q

R

S

T