Plug-in Manual

Barco Crescent

Version 2.4.0.0
December, 2004

Xitron Part Number Doc-1002 12/04

Overview

Xitron's Navigator PostScript RIP and Raster Blaster TIFF Catcher rely on software modules called plug-ins to communicate with imagesetters, platesetters, and proofers. These plug-ins are written and compiled to a format known as Dynamic Link Libraries, or DLLs. They act as device drivers for the software and control most actions of the output devices. Some of these actions include checking device status, device setup, and advancing and cutting material. In addition, the plug-in relays all the physical characteristics of an engine such as supported resolutions and imageable area.

During the launch sequence, both Navigator and Raster Blaster scan a directory called "devices" for plug-in files. The software loads each plug-in it finds, and then queries them for a description of the capabilities of the supported devices. In this manner the plug-in configures the RIP to output a bitmap to these devices.

Each plug-in controls a particular family of recorders and is able to understand most messages and errors communicated by the output device. Plug-ins for use with Windows-based platforms consist of three software modules. The first module is the core plug-in written specifically for a particular device. This DLL is 32-bit code and runs under Windows NT, Windows 2000 Server, Windows 2000 Professional, Windows 2003 Server and Windows XP. The second module is a kernel mode device driver. This module communicates with the Xitron interface boards and moves the bitmap data from the PC to the output device's interface. The third module is a 'helper' DLL that translates calls from the plug-in to the Windows device driver.

When a page is sent to an output device for imaging, the Xitron software loads the correct plug-in and begins a series of steps prior to output. The plug-in first initializes the engine and checks that it is ready. After receiving the proper signal, the plug-in will begin reading bitmap data from the platform's hard drive into a "printer buffer." When the printer buffer is full, the plug-in starts communicating the data to the output device. As the output device consumes the data, the plug-in relays this information to the software, which then refills the buffer. This continues until all of the data has been communicated to the output device. The plug-in tells the software the job is complete and waits for an indicator that the recorder has finished. This process is repeated for each page being output.

Raster Blaster

Plug-ins used by Xitron's Raster Blaster have the same functionality as those for the Navigator RIP and the same options are available for configuration. Therefore, unless otherwise specified, the information in this manual will apply to both products. See the Raster Blaster Manual for specific configuration information.

Configuring Devices

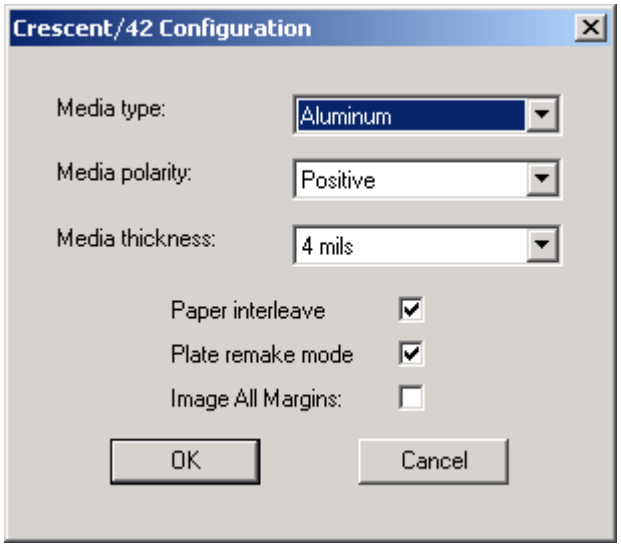
Xitron's plug-in for the Barco (Gerber) Crescent supports the Crescent42, Crescent32, and Crescent3030 using the Waverly PCI interface card. **Note: The Waverly card and WaveC42 plug-in will not work with Windows 2000, Windows Server 2003, or Windows XP. Because of this limitation, the Barco plug-in has reached end-of-life status.**

> ***Important:*** *Xitron distributes the Crescent plug-in configured to drive the Crescent/42. The plug-in is configured using the WaveC42.ini, which is installed in the Windows WINNT directory. The WaveC42.ini file may not be installed into your Windows WINNT directory by default; you must verify the file's existence there. Locate the file by searching the hard drive, then place it in your Windows directory (c:\winnt\). Enabling the plug-in to drive one of the other two devices requires editing the WaveC42.ini file. Comments at the top of the file contain the details. The plug-in can only be configured to drive one device.*

Please see the section at the end of this document for an example of the Wave42.ini file.

Barco Crescent

After installing the Barco Crescent plug-in you will be able to create Page Setups the Page Setup Manager. Select the appropriate resolution, density, and page orientation from the main window of the Page Setup Manager. You should also configure the options specific to the Barco Crescent plug-ins. Click on "Configure Device" under the Device Type list box. The following dialog box will appear:



For Raster Blaster, see the Raster Blaster manual section on *Creating New Devices.*

From this dialog box you can configure the following options:

•**Media type:** Options are Aluminum, Film, and Polyester. Select the type loaded in the platesetter.

•**Media polarity:** This determines whether the pixels are inverted or negated by the Crescent when they are imaged on the media. For positive plates this would typically be off.

•**Media thickness:** Select the appropriate thickness to ensure accurate focus.

•**Paper interleave:** Check this box if there is a slipsheet inserted between each plate.

•**Plate remake mode:** Checking this option places the Crescent in Plate remake mode.

•**Image All Margins:** This causes all four margins (top, left, right, and bottom) to be imaged. If you are imaging film and you want it burned out (imaged to the edges) check this box.

Attaching the Barco Crescent

Please see the documentation that came with the Crescent for precise details on this procedure. This section is a short overview of the process.

The plug-in for the Crescent utilizes two channels of communication with the Crescent. The video data is sent via the Waverly PCI card. This card is connected using the cable supplied with the Crescent. Command and control are handled using a serial cable, also supplied with the platesetter. This cable is connected to the PC's COM 1 port and the port labeled PORT A on the Crescent. PORT B on the Crescent is a debugging channel and should not be connected to COM 1 on the PC.

Replacing an Autologic SoftPIP or GrafixRIP

If a previous connection to the Crescent exists through an Autologic (AIII) PIP converter box, you will need to switch the serial cable connection to the Crescent. The PIP converter box can be connected to the Crescent's PORT B.

Sample WaveC42.ini File

The following is an example of the standard WaveC42.ini file, which may require editing depending on your installation:

```
; $Log: WaveC42.ini $
; Revision 1.6  2000/10/24 21:13:11  Mark
; Final version for 2.4 release.
; Revision 1.5  2000/10/24 20:45:35  Mark
; Removed slash from 3030 entry
; Revision 1.4  2000/10/24 20:44:04  Mark
; Add in C3030 entry.
; Revision 1.3  2000/10/17 18:45:57  Mark
; Add byteSwitch flag for 2.4 plugin release. Default is OFF/0
; Revision 1.2  2000/10/17 18:22:58  Mark
; Added bitReverse flag and xitronDebug flag for 2.3.
; Also, Kern's change for eliminating the forward slash in the
; name which was causing grief in Calibration Manager.
; Revision 1.1  2000/06/05 17:50:05  Mark
```

```
; Initial revision
# created 01-22-97 sjpflederer from GerbC42.ini:
#       Revision 1.1.2.1.2.1  1996/01/15  13:27:38  hope
;


; ------------------------------------------------------------------------------
;
; The currentDevice entry in [Plugin] section indicates that the remaining
; information should be taken from the specified section.
;
; The following entries are supported.
;
;  debugState                   Non zero logs internal state transitions
;  debugSerial                  Non zero logs serial i/o
;  debugError                   Non zero logs error conditions
;  simulator                    Non zero disables serial I/O
;  debugXitron                 Turn on Xitron messages
;  bitReverse                  Flip bits on the interface card
;  byteSwitch                  Send the BS command to the engine, 0=OFF, 1=ON
;
;  resolution0, resolution1...  Supported resolutions
;  frequency0, frequency1...    Default screen frequencies
;
;  timeoutCA                    Override timeout for CA command
;  timeoutDW                    Override timeout for DW command
;  timeoutME                    Override timeout for ME command
;  timeoutMP                    Override timeout for MP command
;  timeoutPI                    Override timeout for PI command
;  timeoutPO                    Override timeout for PO command
;  timeoutPR                    Override timeout for PR command
;  timeoutRS                    Override timeout for RS command
;  timeoutRU                    Override timeout for RU command
;  timeoutSR                    Override timeout for SR command
;  timeoutXL                    Override timeout for XL command
;  timeoutXT                    Override timeout for XT command
;  timeoutXW                    Override timeout for XW command
;  timeoutRQ                    Override timeout for RQ command
;  timeoutFO                    Override timeout for FO command
;  timeoutOF                    Override timeout for OF command
;
;  debugQueue = verbose + range; verbose = 100 for ON, = 0 for OFF
;                      range (>2) is number of buffers to go empty BEFORE debugging.
;       This may be useful in tracking underruns as it can show them about to happen.
;  driverQueue = KB of buffers to allocate for printahead to driver.
;  testmode = 0 for production; = 1 to simulate presence of imager (Waverly only!)
;
; ------------------------------------------------------------------------------


[Plugin]
currentDevice=Crescent42
;;currentDevice=Crescent3030

[Crescent42]
debugState=0
debugSerial=0
debugError=1
debugQueue=0
resolution0=1270
resolution1=1905
resolution2=2540
resolution3=3810
frequency0=8500000
frequency1=8500000
frequency2=13300000
frequency3=15000000
timeoutCA=5
timeoutDW=5
timeoutME=10
timeoutMP=5
timeoutPI=5
timeoutPO=5
timeoutPR=5
timeoutRS=5
timeoutRU=600
;timeoutRU=60
timeoutSR=15
timeoutXL=5
timeoutXT=40
timeoutXW=5
```

```
simulator=0
testmode=0
driverQueue=4096


[Crescent3030]
debugState=0
debugSerial=0
debugError=1
debugQueue=0
resolution0=1270
resolution1=1905
resolution2=2540
resolution3=3810
frequency0=8500000
frequency1=8500000
frequency2=13300000
frequency3=15000000
timeoutCA=5
timeoutDW=5
timeoutME=10
timeoutMP=5
timeoutPI=5
timeoutPO=5
timeoutPR=5
timeoutRS=5
timeoutRU=600
;timeoutRU=60
timeoutSR=15
timeoutXL=5
timeoutXT=40
timeoutXW=5
simulator=0
testmode=0
driverQueue=4096
```