

Braille2000, LLC

6801 Southfork Cir, Lincoln, NE 68516

Phone: (402) 423-4782

<http://www.braille2000.com>

Embossing to a ViewPlus® Embosser

Summary

The ViewPlus line of graphics embossers (as well as similar products they manufacture for other brand names) involves an innovative tactile technology that operates on graphic image data of a general nature: each output page is a graphic; areas that are shaded get dimpled on paper. The general dot resolution is 20 dots per inch horizontally and vertically (some models may position dots with greater or lesser precision). In the abstract, the technology mimics dot-matrix printing at 20 dpi with dimpling replacing inking. There is also a feature for ink overprinting, giving both tactile and inked output on the same sheet. This approach to embossing is herein termed “raster” embossing. (Other, older, traditional approaches are herein termed “legacy” embossing.)

Braille2000 outputs to legacy embossers via the Emboss function. Braille2000 outputs to raster embossers via the Print function. Thus, you use Print for a ViewPlus Columbia or an APH PixBlaster, and you use Emboss for a Juliet 120, or an Index Basic D, or a Braillo 300, etc.

(History: Legacy embossers receive an ASCII-braille text stream that manufacturers expect to come from a dumb pass-through print driver known as the "Generic Text-Only" driver. Braille2000 is more modern and utilizes a smart print driver for legacy embossers, a print driver that comes with Braille2000 and that is added to the system via a setup process in Braille2000. The Braille2000 Emboss function always uses the Braille2000 smart driver, for legacy embossers only. ViewPlus embossers have their own print driver. Braille2000 uses it via the Print function. Some ViewPlus embossers can emulate a legacy embosser and can thereby be used via the Emboss function, but that is not the modern approach.)

Some ViewPlus models can also ink print. Braille2000 supports the "InkConnect" behavior of those models (another innovative design from ViewPlus). Braille2000 supports the "2 in 1" strategy in which each page is rendered twice, once for tactile output and then again for inked output.

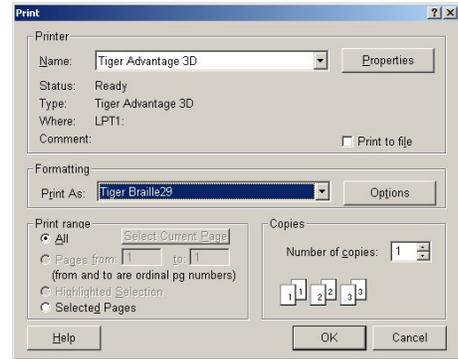
(Note: unfortunately, the ViewPlus print driver does not conform to Windows' standard behaviors. This is an ongoing problem manifest in some but not all manners of use. Beginning with Braille2000 version 2.35, accommodation for this non-conformation has been built into Braille2000 and so the problem (that does need to be fixed by ViewPlus) is mostly academic.)

Embossing to a ViewPlus embosser

To use your ViewPlus embosser with Braille2000, install the embosser as indicated by the installation instructions that came with the embosser. Be sure to install the “ViewPlus fonts” on your system... they are a part of the embosser’s Software Suite. The embosser will appear as a printer in the system’s Printers and Scanners panel, and you can test the device by printing to it from Word or WordPad (but remember that black items are dimpled on the paper rather than inked).

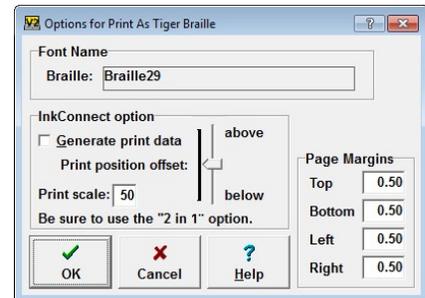
In Braille2000, click the Print button. The print dialog box will appear. If necessary, change the Name box to the embosser identifier (shown here for the “Tiger Advantage 3D” model). Change the Print As selection to “Tiger Braille29” (There may be embosser models or situations when the number will be other than 29, e.g., Tiger Braille17).

(Note: if Print As offers no “Tiger Braille##” choices, it is because the ViewPlus fonts have not been installed. They come with the embosser and you must install them.)

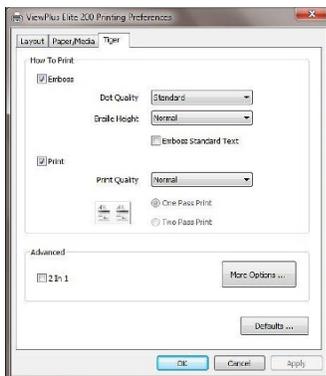


Configuration of the embossed braille is handled by two buttons: Options and Properties.

The Options settings control how Braille2000 “draws” braille on output pages. It is via Options that you can adjust the page margins. The margin (in inches) is the amount of space at the edge of the paper that comprises a border around the embossed area. The default values (half an inch) are generally appropriate. You can move the embossed area left or right on the page by adjusting the values of the Left and Right margin amounts (add to one while subtracting from the other).



Some ViewPlus models have the ability to ink print, called the InkConnect option. If your model has that feature, and you wish to use it, put a checkmark at “Generate print data”. This causes Braille2000 to draw each output page twice, once for tactile output and then for ink output. This is called “2 in 1” mode, and must also be activated at the embosser itself, via Properties. You can adjust “Print scale” to change the size of the print: 50 denotes 50 percent relative to cell height. The position of the print lines, relative to the braille lines, can be adjusted using the slider. The middle position ink-prints right on top of the braille. Move the slider up to raise the print position, down to lower the print position.



The Properties button in the Print dialog causes the print driver to show its model-specific settings. It is via Properties that you can declare paper size and activate the embosser’s “2 in 1” behavior, and other things. (The snapshot shown is for the ViewPlus Elite 200, your model will most likely be different.)

(Reminder: if you are using the InkConnect feature, you need “Generate print data” in Options, **and** the “2 in 1” feature in Properties.)

Background on Windows Printing (may be of technical interest to some readers)

The bulk of printer support today is still the “type 3 printer driver” architecture of Windows 2000. It has application programs (e.g., Braille2000) writing print pages abstractly via GDI (the Graphics Device Interface, a comprehensive set of data functions for “drawing” on abstract visual surfaces such as the main display screen). Via GDI, programs can draw text in any font/size/color and draw colored lines and shapes and pictures, onto digital abstract “surfaces” including the computer screen, a sheet of paper in a series of pages to be printed, or to a photo file such as a .jpg or .png.

To be a Windows “printer”, the device must be able to receive data from the system (USB cable, network link, old-style parallel cable, etc.) and there must be software that can offer a GDI-like understanding by which GDI data functions can get converted into the data the printer device needs to draw on real paper the image a program “drew” abstractly on the digital surface (one drawing per page of output). Thus, an application program wanting to print three pages of “whatever” needs to “open” a printer (gaining indirect access to its GDI capabilities), then declare “begin print” to access a blank surface on which it can “draw” via GDI functions, then declare “next page” to tell the driver that that surface is done (to send its data to the printer) and to get another blank surface for the next page, on which it “draws” page two, then another “next page”, then more “drawing”, and finally “end print”. The application program is communicating with the Windows system, that, via “open printer” is communicating with the printer-specific print driver (software) that is converting the abstract GDI drawing the application has done onto page surfaces into printer-specific data to cause the real output to happen on the device.

(Note: in addition to this versatile “drawing” abstraction, the system also performs “print spooling”, a strategy by which printer data goes first into a holding file, to be transferred to the printer via a background process when the printer is ready to receive it. In this way, via “spooling”, you can print additional documents while the printer is busy with earlier ones... the new output is held in files until the printer is free to receive it.)

Software modules are packaged in “libraries” often in .dll files. If a printer is to have “type 3” capabilities (what was described above), it has to come with two .dll software libraries, one for GDI processing and another (the “ui” library) for user control panels. (The Properties snapshot above came from the “ui” dll for the ViewPlus Elite 200 printer.) When an application “opens” a particular printer, it gains indirect access to the capabilities of both dll libraries. (When you “install” a new printer, those two dll files (and usually additional dll support libraries) get added to Windows for use with that printer.) Because each printer has its own dll libraries, each can handle abstract surface “drawing” differently and each can have different Properties panels.

The Windows system manages all of this. It includes a set of “Common Dialogs” that include dialog boxes for “Open” (opening a file), “SaveAs” (saving a file), and “Print” (output to a printer). An application can make use of the dialogs as-is, or it can use them with minor customization, or it can use its own dialog completely. A modern trend is for applications to use their own dialog for print operations, and thus there is no “standard” of behavior. Braille2000 uses the Windows standard Print dialog, and it is in that use that the ViewPlus print driver (its dll files) are non-conforming. The issue has to do with how Properties are managed (not how GDI is handled).

The typical printer can be used in different ways, for example, an ink printer might be able to handle letter size, legal size, and envelope size paper. The print driver has to know which size because paper size determines the size of the abstract drawing surface the application has to “draw” on. And the way the driver converts abstract GDI operations into printer data may change too. There are some universal printer properties (such as paper size) that relate to all printers, and of course, a particular printer will have its own unique properties (such as dimple height for an embosser).

Printer settings are organized into a data record (data collection) known as the DevMode. It has a universal (generic) part and a device-specific part, such that device behavior is completely specified by the DevMode: with the same DevMode, an application’s GDI drawing will yield the same final result (the DevMode provides all control settings). Each printer has a master DevMode record (stored in the Registry) containing the printer’s default settings. Windows supports multiple user accounts (though some personal computers have only one login). Each user can have their own default printer properties (for each installed printer). Those per-user DevModes are also stored in the Registry. But that is not all...!

For each application using a printer, that application gets a copy of the DevMode for that printer, for it to use during its operation (to give continuity to repeated print requests in that application). The intent is this: when an application does its first Print request (or if you select a different printer), the DevMode is the user's default for that printer (if there is a user DevMode for that printer in the Registry) or else it is the system default for that printer (from the system area in the Registry). Via the Print dialog, that DevMode is provided to the application (it will use it when "opening" the printer to use its GDI functions). If the user clicks the Properties button in the Print dialog, the print driver's "ui" dll offers properties pages by which the user can adjust device settings, and it is the adjusted DevMode that then goes to the application. If the user of that application uses Print again, the application will normally start with the same DevMode used previously, i.e., the settings will be unchanged, unless the user again clicks Properties and changes something. Note that in this interaction, neither DevMode in the Registry is supposed to be changed: if the user launches another application, or closes and relaunches the same application, the same default DevMode as before should apply (not affected by using the Properties button).

(Note: the DevMode settings in the Registry are supposed to be set only via the "Printers and Scanners" system control panel. The setting for "Printing Defaults" stores a DevMode in the "Local Machine" (system) part of the registry (applies to all users). The setting for "Printing Preferences" stores a DevMode in the "Current User" part of the registry (applies to just the current user). The Properties button in a Print dialog should never change either registry setting.)

The "problem" with the ViewPlus print driver

The ViewPlus "ui" dll library does three things wrong:

1. It does not always set the "dmFields" member of DevMode properly
The generic portion of DevMode has redundant settings for paper size by category (e.g., letter vs legal vs. A4) and in tenths of millimeters; the dmFields setting indicates which of the various settings are valid and which are not; the driver always sets the size category dmPaperSize to "letter" and sets dmPaperLength and dmPaperWidth to correct paper size values (tenths of mm), but it sets dmFields to say that dmPaperSize is valid but dmPaperLength and dmPaperWidth are not. The result (if DevMode is properly interpreted) is a paper size always 8½x11 (i.e., "letter" size), regardless of user choice (and most of the time braille paper size is 11½x11_ . The GDI tools operate with the wrong surface size.
2. It does not copy the adjusted DevMode to the Windows "devmodout" variable
Some requests to "ui" functions (in the "ui" dll of the ViewPlus driver) ask for a copy of the DevMode to be provided upon completion of the request. This replaces the previous DevMode. The standard Windows print dialog returns the updated DevMode to the application. By not providing the updated DevMode, user interactions via the ViewPlus "ui" dll (i.e., settings changes) are never returned to the application, when using the Windows print dialog... i.e., using the Properties button has no effect.
3. It updates the Current User DevMode in the registry when it should not
The ViewPlus "ui" dll always writes an updated DevMode to the "Current User" location in the Registry. This means that the use of the Properties button in one application will manifest in other applications launched afterwards. Though sometimes useful, it is simply wrong. (This behavior, with non-system print dialogs, might mask the deficiency of (2).)

(Note: beginning with Braille2000 version 2.35, these problems are mitigated, and no special user behaviors are required.)