

Technical Disclosure Commons

Defensive Publications Series

March 19, 2019

SOFTWARE FOR DRIVERLESS DISTRIBUTED PRINTING ON LARGE FORMAT DESIGNJET PRINTERS

HP INC

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

INC, HP, "SOFTWARE FOR DRIVERLESS DISTRIBUTED PRINTING ON LARGE FORMAT DESIGNJET PRINTERS",
Technical Disclosure Commons, (March 19, 2019)
https://www.tdcommons.org/dpubs_series/2040



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Software for driverless distributed printing Large Format DesignJet Printers.

The program herein disclosed allows printing from a computer with the minimum required user actions without any Application or Printer Driver processing and to all the available printers installed in the user computer at once. The program also allows for speeding up the Printer Firmware upgrade or patching process.

Most users print files from their computers in two ways. The first involve the actions of opening the file with an application capable of loading the file and then following through the applications print dialog selecting a printer and finally submitting the file to print. Another faster way of accomplishing this task is by selecting directly the files and making use of some Operating Systems (OS) support for Printing by right click print menu options or dragging and dropping the files over the Printer Icon (Windows for example). Both of the previously mentioned printing paths involve opening the associated application rendering the file to print by the application and translating the rendered content into a printing language through a complex function call process that involves the Application, the OS Application Programming Interface (API) and the Printer Driver installed. A simplified description of the steps involved in this process is shown in the Figure 1 attached. This process limits the user interaction to one printer at a time and requires the OS to open as many application interfaces as file types; so if the user wants to make use of several printers the operation must be repeated for each printer individually.

This program offers the users with certain control over the file content or the printer settings available through the Designjet printers Front Panel the fastest way to accomplish the task of printing, minimizing their interactions and impact over OS resources and taking the advantage of file formats that provide the printer with the required size and color information as for example PDF. The software opens a direct printing path enabling simultaneous print job submittal to multiple LFP DesingJet Printers. The program makes use of DesignJets printers Firmware capability of rendering PDF, JPEG and TIFF file formats (among many other). Thus, the file to print is not required to be rendered by an Application and Driver at the user computer side but it is simply sent through the chosen printer interface installed in the system freeing the user computer OS from any processing task. The software operation process is shown in the Figure 2 attached.

A Proof of Concept solution has been developed for Windows OS as a Windows Presentation Forms application; the application icon has chosen as an image of a folder to visually represent a file drop destination to the user. Consider the following C# code snippets to illustrate the main calls during the printing process.

```
//Main User Interface code.
public partial class App : Application {
    private void Application_Startup(object sender, StartupEventArgs e) {
        //The Application Window opens.
        MainWindow wnd = new MainWindow();
        wnd.Show();

        //The Application builds a list of all compatible printers installed.
        if (printersPanel.loadInstalledPrinters(wnd.PrintersStackPanel)) {
            printer.files = new List<String>();

            //Dropped files are passed to the application as arguments by the OS.
            if (e.Args.Length > 0) {
                for (int n = 0; n < e.Args.Length; n++) {
                    printer.files.Add(e.Args.ElementAt(n));
                }
            }
        }
    }
}
```

Once the available printers are listed in the program UI, the user selects the printers to which the file should be sent, the following function is then executed recursively for each of the printers selected.

```
//Printing function called recursively for all check box selected printers.
static public string printFile(string printerName, string filePath, series printerSeries) {
    IntPtr printerHandle;
    PRINTER_DEFAULTS defaults = new PRINTER_DEFAULTS();
    uint dwWritten=0;

    //Get a handle of the installed printer.
    if (SpoolApi.OpenPrinter(printerName, out printerHandle, ref defaults)) {
        DOC_INFO_1 pDocInfo = new DOC_INFO_1();
        pDocInfo.pDocName = "HP Print Folder Job";
        pDocInfo.pOutputFile = null;
        pDocInfo.pDatatype = "RAW";

        //Notify the Spooler for sending data to the printer.
        if (SpoolApi.StartDocPrinter(printerHandle, 1, ref pDocInfo) > 0) {

            //As file will be rendered by the printer Firmware make a one-page call.
            if (SpoolApi.StartPagePrinter(printerHandle)) {
                byte[] bytes = readFile(filePath);

                //Check the file header to validate the supported format.
                if (isSupportedFormat(bytes, printerSeries)) {

                    //Write the data directly to the printer.
                    if (SpoolApi.WritePrinter(printerHandle,
                        bytes,
                        (uint)bytes.Length, out dwWritten)) {
                        SpoolApi.EndPagePrinter(printerHandle);
                        SpoolApi.EndDocPrinter(printerHandle);
                    }
                }
            }

            //Finish the process.
            SpoolApi.ClosePrinter(printerHandle);
        }
    }
}

//Return some information.
return dwWritten + " Bytes written.";
}
```

Note the SpoolApi class implements the DllImport calls for API functions for using them from C#.

Building over these two simple functions, it is possible to implement several iteration models for job ticketing, copies settings or selective job distribution if required.

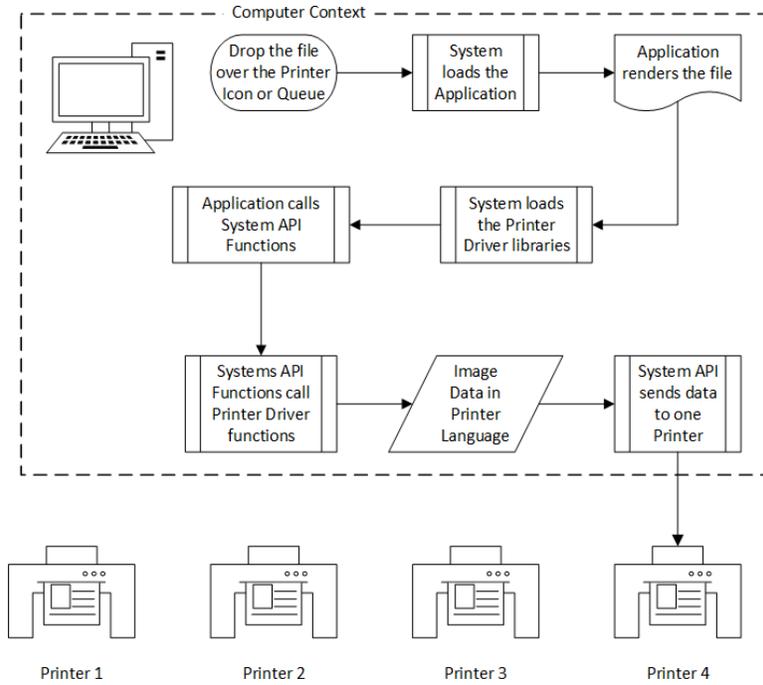


Figure 1. Standard Printing with Application and Driver processing.

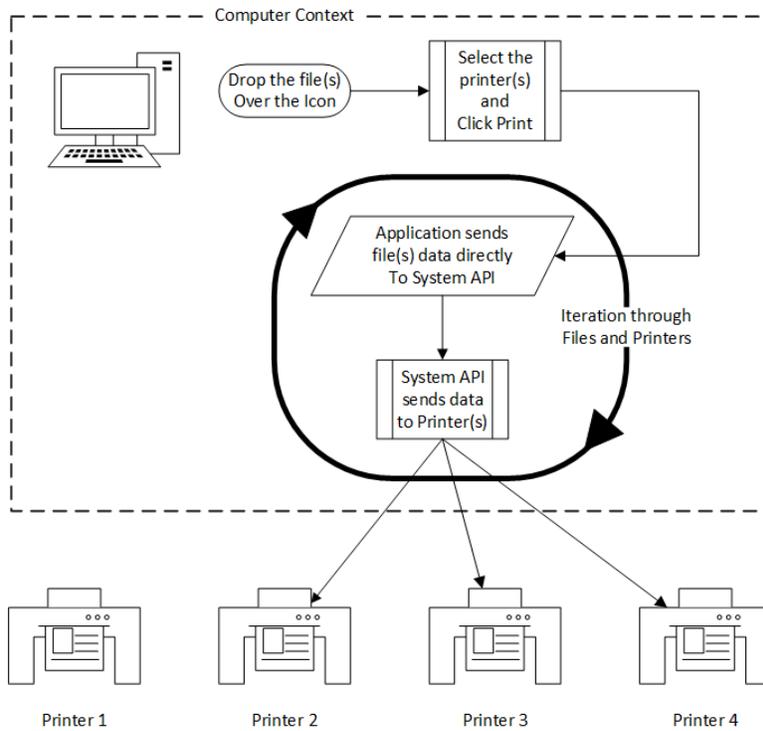


Figure 2. Printing without Application processing (Driverless).

Disclosed by Bladimir de la Hoz Matveeva, HP Inc.