

DynamicPaper with **DP Archives**



DynamicPaper

User Guide

Revision 1.4

**For Product releases
DynamicPaper 1.2.2
DynamicXport 2.3**

6/01/2005

Information in this document is subject to change without notice and does not represent a commitment on the part of Dynamic Concepts, Inc. (Dynamic). Every attempt was made to present this document in a complete and accurate form. Dynamic shall not be responsible for any damages (including, but not limited to consequential) caused by the use of or reliance upon the product(s) described herein.

The software described in this document is furnished under a license agreement or nondisclosure agreement. The purchaser may use and/or copy the software only in accordance with the terms of the agreement. No part of this manual may be reproduced in any way, shape or form, for any purpose, without the express written consent of Dynamic.

© Copyright 2003-2005 Dynamic Concepts, Inc. (Dynamic). All rights reserved.

Dynamic Concepts, Incorporated
18-B Journey
Aliso Viejo, CA. 92656

www.dynamic.com

DynamicPaper is a trademark of Dynamic Concepts, Inc.

dL4 is a trademark of Dynamic Concepts, Inc.

UniBasic is a trademark of Dynamic Concepts, Inc.

DynamicXport is a trademark of Dynamic Concepts, Inc.

UNIX is a registered trademark of UNIX Systems Laboratories.

Microsoft, MS, and MS-DOS are registered trademarks, and Windows and Windows NT are trademarks of Microsoft Corporation in the USA and other countries.

JetPCL is a registered trademark of Tech Know Systems, Inc.

Table of Contents

General.....	1
<i>Product Definition</i>	1
<i>Product Components</i>	2
DynamicPaper	2
DP Archives	3
Installation, Configuration, Setup, and Checkout.....	4
<i>Installation of software</i>	4
<i>System configuration changes</i>	4
<i>dL4 and UniBasic Printer Driver Script Configuration</i>	9
<i>PATH and LUST set up</i>	9
<i>Post Installation Checkout</i>	10
<i>DP Archive Configuration</i>	11
Using DynamicPaper	13
<i>Making forms</i>	13
<i>The API</i>	17
<i>Environment Variables</i>	18
<i>Embedded Forms</i>	19
<i>Alternate Outputs</i>	19
<i>Examples</i>	20
Example 1:	20
Example 2:	24
Example 3:	24
Example 4:	25
<i>Useful tools</i>	25
DP Archives.....	27
<i>Database Initialization and definition</i>	27
dpcreate.dl4 and dpsystem.txt control file	27
DPSYSTEM Parameters File.....	29
Report dataset.....	30
Archive dataset.....	32
UserOptions dataset	34
User dataset.....	34
<i>Administration</i>	35
Reports	35
Archives	37
User's Reports.....	39
User's Subscriptions	40
<i>Using DPArchives</i>	42
Subscribing to reports	42
Searching the Archives	42
Viewing Archived Reports	43
Download Archived Report in PDF format	44
Email Archived Report in PDF format	45
Download Archived Report in Text format	45

Sticky Notes	46
<i>The DPArchive API</i>	48
DPADD.DL4	48
CONDITIONS.LIB.....	49
Appendix A – Error Messages and Logging.....	52
<i>From DynamicPaper Driver</i>	52
<i>From DPArchive under dynamicXport</i>	53
<i>From DPADD subroutine</i>	53
Appendix B – Directory Lists	54
<i>Driver Files List</i>	54
<i>Archives Files List</i>	54
<i>HTML Files List</i>	55
Standard HTML files delivered with DynamicPaper Archives	55
<i>Stylesheet Files List</i>	55
<i>Image Files List</i>	56
Inactive graphics:	56
Active graphics:	56
Appendix C – JetPCL®	57
<i>JetPCL® version 9.0 Integrated into DynamicPaper</i>	57
<i>Purchasing a ‘JetPCL® for DynamicPaper’ Runtime License</i>	57
<i>Installing a ‘JetPCL® for DynamicPaper’ Runtime License</i>	57
<i>Contacting Tech Know Systems, Inc.</i>	57

General

Product Definition

This product provides a printer driver and a form processor which supports creation of business forms in MS® Word, MS® Publisher or other third party tools, which can then be utilized by dL4 and UniBasic applications to overlay an application's output onto the forms. These forms can have logos and other graphics just as though they were ordered from a commercial printer. Now, print the forms and the data at the same time and at a fraction of the cost of ordering preprinted forms. Changing address, phone numbers, or terms regularly printed on forms, are quick and easy updates to forms files. No more throwing away obsolete forms. Never print with an out of date form again.

DynamicPaper processes forms files in HP PCL format compatible with any printer which can print HP PCL coded documents. All the basic PCL functions are supported up to PCL-5.

Graphics and text are combined to create a form in MS Word or other tool and then printed to a PCL format .prn file. The DynamicPaper tool "makeforms" processes the .prn file to create one or two .dpf form files. Two form files, a front and a back, are produced when the form is two-sided.

The application identifies which form(s) to use in the printer OPEN statement or print stream and prints its data in the appropriate page positions to fit the form. DynamicPaper merges the two outputs in the printer's memory before printing.

Subsequent pages can be printed with the same or different forms without the need to change forms in the paper trays or tie up the printer while others wait for plain paper to be reloaded. This time saving is a major contribution to the cost savings brought by DynamicPaper.

DynamicPaper can also produce a ".prn" file of merged form and data and convert it to PDF with JetPCL®, a separately purchased third party utility (see Appendix C). These PDF files are ready for emailing or archiving on disk. Thus, further savings can be realized with DynamicPaper in moving from a paper environment to the reduced labor, supplies, and storage costs of a "paperless" office.

To complete the required functionality of the "paperless" office, the DynamicPaper Archiving Option for accessing and managing document archives is available with dL4 GUI-based, and DynamicXport browser based interfaces. It supports viewing of archived documents, downloading their content and printable pdf images as well as emailing pdf formatted copies of documents on demand. The DP Archives supports a subscription function to automatically email copies of a report to users who are subscribed to it when either the report is archived via its creation by an application or when it is released for distribution by an authorizing agent. In addition to the facility to archive reports created with the DynamicPaper driver, the product's API supports applications adding plain text and pdf format reports to the archive via a dpadd call.

Product Components

DynamicPaper

DynamicPaper, a special windows printer driver, produces the properly formatted PCL forms files when “printed” to a .prn file from MS Word or other tools.

makeforms, a UNIX® application, is a utility to process a .prn file and create form file{s}.

dynpaper is the DynamicPaper dL4/UniBasic printer driver as a UNIX® script file. This is the driver that your application opens to print. It accepts the special form file names and other parameters as options.

formmerg is a compiled support module called by the dynpaper script to process the form and data, merging them into the requested output.

formmerg.dat is a PCL file used by formmerg.

JetPCL® is a third party UNIX® filter application which converts “.prn” (PCL encoded) files into PDF documents and is optionally called by the formmerg module to produce a PDF output (see Appendix C).

PCLFilter is a text file for dL4’s lptfilter and is used by formmerg when a form is not specified.

dpmail, is a dl4 application which sends email either from the dynpaper driver, from a user application, or as a stand alone GUI input Unix program.

DP Archives

conditions.lib is a stubbed out user customizable library of filter functions for controlling retrieval of archived reports.

dpadd.dl4 is the callable subroutine to add a report to the archive

dpadmin.dl4 is a dynamicXport application server module supporting maintenance of the dparchives reports and archives database entries managing report access and subscriptions.

dparchive.inc is the DP Archives database record definitions for report, archive, user and organization datasets.

dparchive.lib is a the support library of dl4 subroutines and functions for DP Archives

dpcreate.dl4 is the creation, backup, and initializer for the DP Archives database.

dpmail.dl4 is the stand alone program for emailing archived reports from the DP Archives database.

dpsystem.txt is a control file with running parameters for the DP Archives modules.

dpview.dl4 is a dynamicXport application server module supporting searching for and viewing, downloading, and emailing reports archived in the DP Archives database.

foremail is a compiled module of DynamicPaper which supports the preparation of reports for emailing.

Installation, Configuration, Setup, and Checkout

Installation of software

Logon as “root” to do the installation. The DynamicPaper distribution file should be copied to a temporary directory such as “/usr/tmp” and if it has a “.Z” extension uncompressed with the “uncompress” command. Expand the delivery archive with the command “cpio -imcdv <filename” and initiate the install with the command “./install”. The install script will load the software into the directory “/usr/lib/dynpaper”.

The “readme.txt” file included with the DynamicPaper distribution has the most current information on the product and step-by-step instructions for installing the product’s software components. The following describes changes required in the target UNIX® system and the MS® Windows system used for creating forms files.

If the DP Archives feature is being added, its distribution file should be copied into a temporary directory and uncompressed with the “uncompress” command if it has the “.Z” extension. Then, the delivery archive must be expanded with the following command: “cpio -imcdv <filename” and installed with the command “./install”. An additional “readme2.txt” file is included with the most current information on the product and additional step-by-step instructions on the installation of the software.

System configuration changes

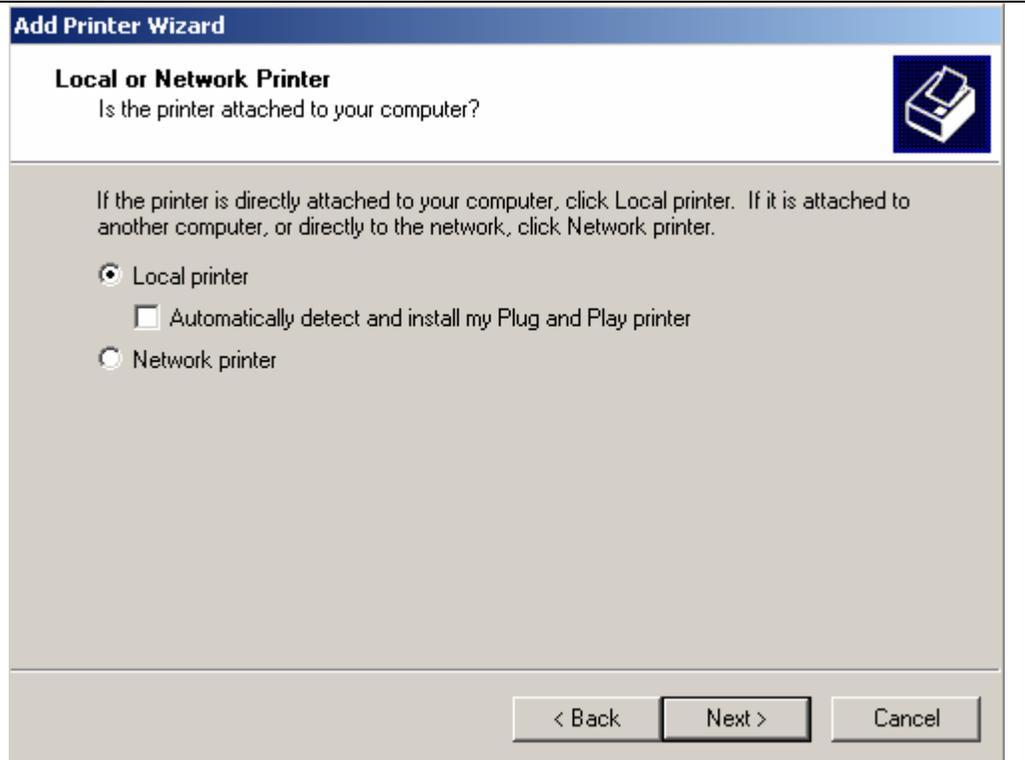
There are no system configuration changes required on a UNIX® system to make use of DynamicPaper’s capabilities including those of its optional feature of DP Archives.

DynamicPaper uses form files created on MS® Windows platforms. To support forms development, a special printer driver needs to be added to the installed drivers on any PC which will be used for designing forms. The following are step-by-step instructions on how to install and configure the printer driver.

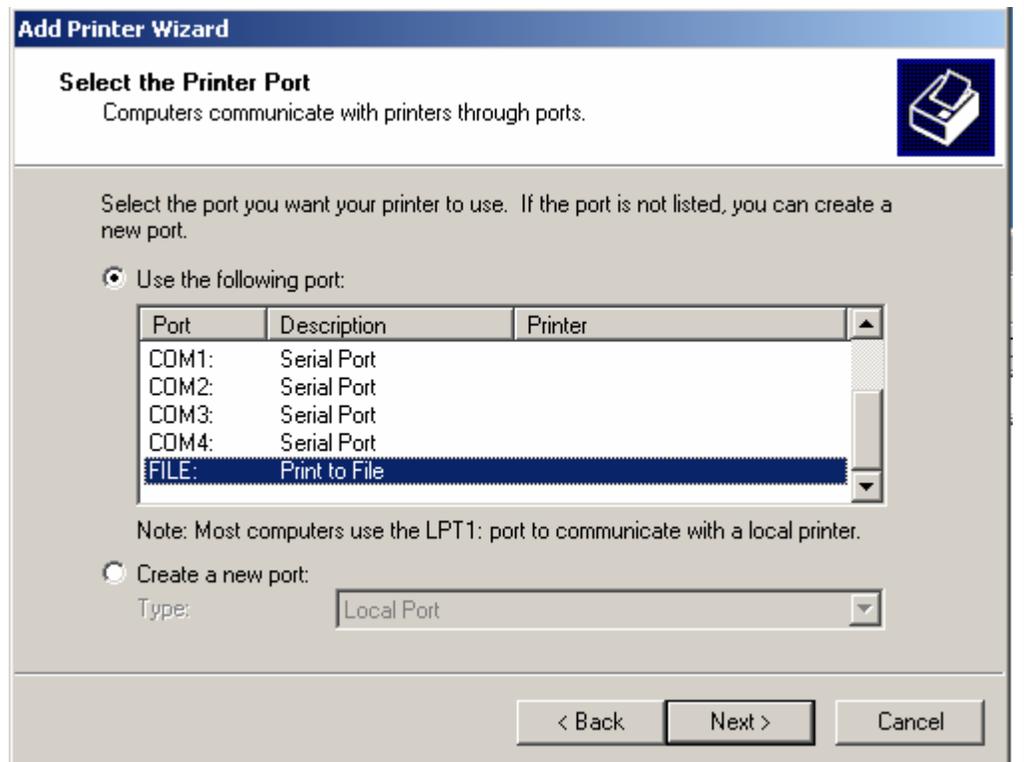
These instructions describe the process under Windows 2000 and XP. There could be some differences with other versions of Windows. If something different is encountered and remains unresolved, Dynamic’s Customer Support should be contacted for assistance. It is important to read these instructions before starting the process.

First, from the “Start” menu opened by clicking on Window’s “Start” button or pressing the Windows key, select “Settings” and then “Printers”. In the Printers folder double-click the “Add Printer” icon to start the “Add Printer Wizard”. After the “Welcome” window appears and “Next” is clicked, the following window will appear.

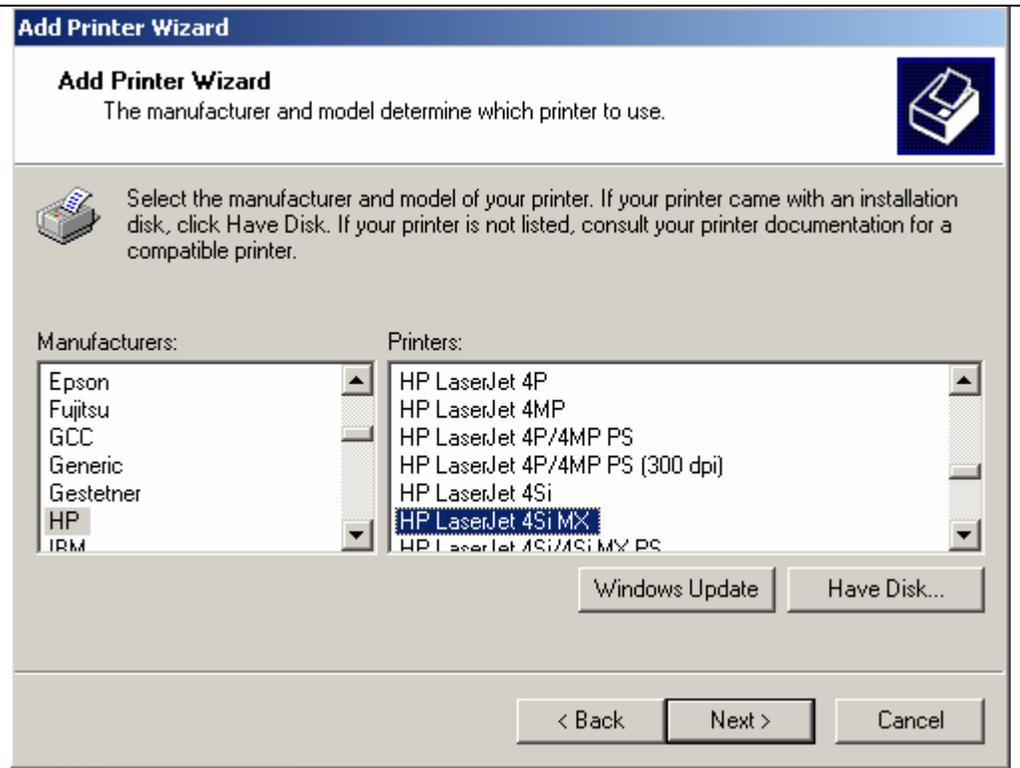
Select “Local” and make sure the “Automatically detect and install my Plug and Play printer” checkbox is NOT checked. Then click “Next”.



This takes you to the “Select the Printer Port” screen. Select the “Use the following port” choice, scroll down the selection box to the “FILE” port choice, and click to highlight (select) that port. Then click “Next”.

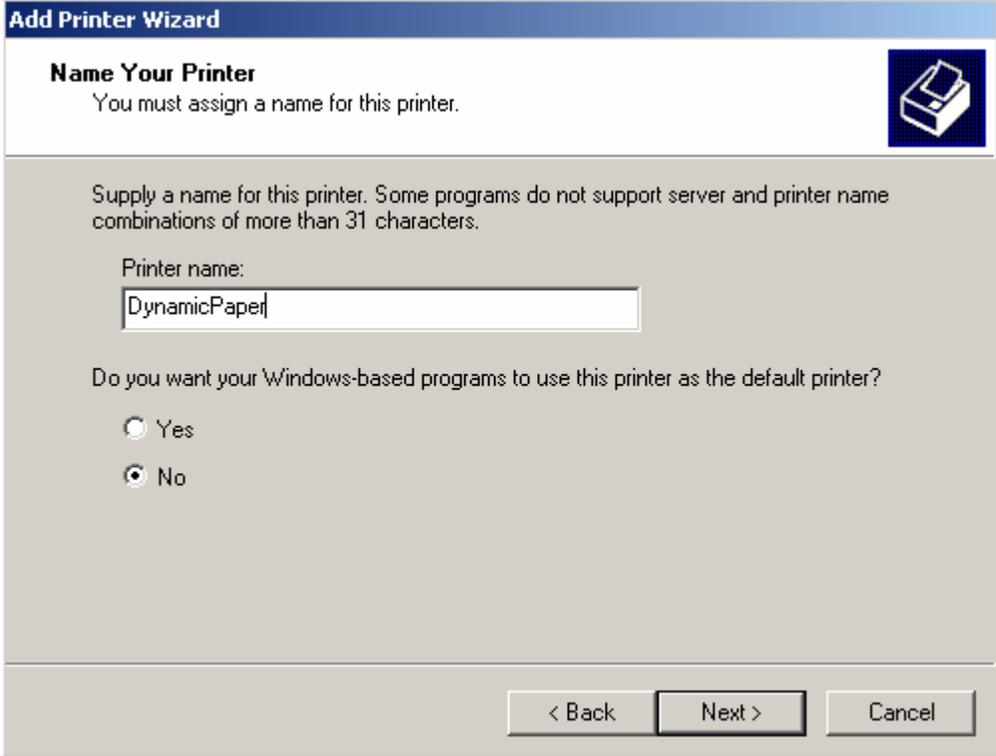
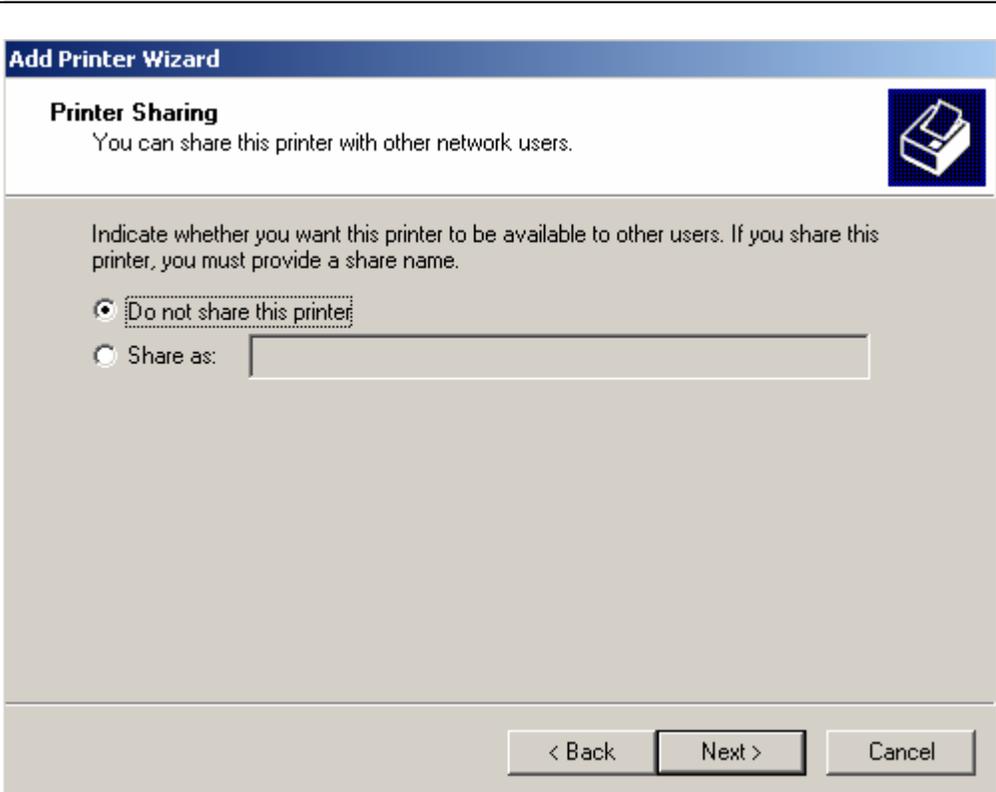


The next screen “Add Printer Wizard” is for selecting the specific printer driver. Select the manufacturer first as “HP”. This sets up the printer selection box. There select the “HP LaserJet 4SiMX” printer. (Do not pick the “PS” version!) Then click “Next”.

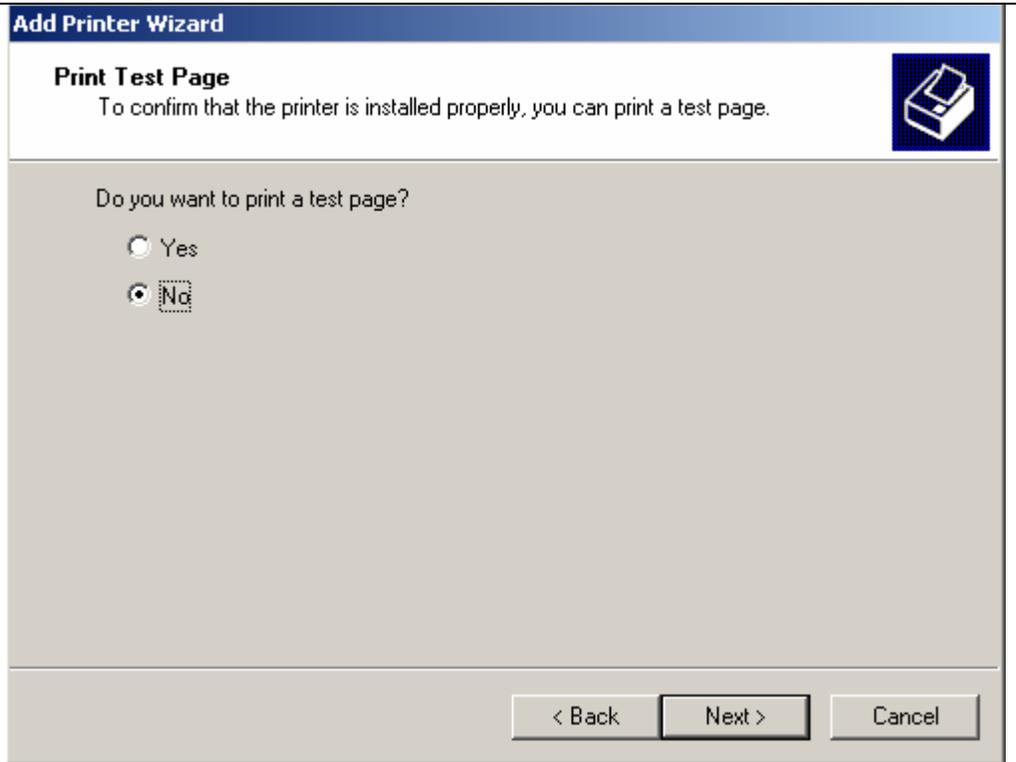


The next step is to choose to “Use Existing Driver”. Select the “Keep existing driver (recommended)” choice and then click “Next”.



<p>This step is to “Name Your Printer”. This will initially have the name of the HP 4siMX printer in the box. Replace it with “DynamicPaper” and select “No” for the choice to use as the default printer. Then click “Next”.</p>	
<p>The “Printer Sharing” choice is next and you select the “Do not share this printer” option. Then click “Next”</p>	

This is the “Print Test Page” option choice. Select “No” and then click “Next”.



The final step is to check the choices shown on the last wizard box and then click “Finish” to complete the driver installation.



When this setup is completed there will be the following new icon in the Printers folder shown for Windows 2000 and XP respectively.



Note that this shows the printer as outputting to a disk file.

dL4 and UniBasic Printer Driver Script Configuration

This is a description of the “dynpaper” driver script file which may be tailored to suit the application requirements and existing environment. The name of the driver script may be changed. Delivered as “dynpaper”, the driver script could just as easily be named “lpthp”. However, staying with the standard can facilitate future upgrades and customer support.

A few environment variables are set in the script file to default values for the printer parameters.

The environment variable DPFORMDIR is set to the path where the compiled module, formmerg, expects to find the forms it will merge with the application’s printed output. Its setting can be done in the user’s profile. It can also be set in the application with a system call before opening the driver script. If necessary, the “dynpaper” script can be changed with the consideration that product updates require more effort. As delivered, the script file will set DPFORMDIR to the default of “dpforms” if it has not been set.

Another environment variable DPTEMDIR, specifies the directory path where the temporary and final outputs are put. If unset, DPTEMDIR is set by dynpaper to the default of \$HOME.

The FILTERFILE environment variable identifies a file to be used if no form is declared. It is a file which dL4’s “lptfilter” uses in translating print control mnemonics. As delivered, the script file sets this variable to “PCLFilter,” a small translation file holding only two entries for translating ‘BX’ and ‘EX’ mnemonics. It can also be set in the user’s profile; but, the line setting it in the printer script must then be commented out.

PATH and LUST set up

The environment variable PATH needs to be set up for running the DynamicPaper driver with either a dL4 or UniBasic application. When running UniBasic, the LUST environment variable must also be set. This is usually done by editing the user’s “.profile” file. What needs to be added is the path to the driver. Edit in at the beginning of the PATH and LUST settings “/usr/lib/dynpaper:” (without the quotes).

If needed this can be accomplished by command line shell command entries as follows:

```
$ PATH=/usr/lib/dynpaper:$PATH export PATH
$ LUST=/usr/lib/dynpaper:$LUST export LUST
```

Post Installation Checkout

After the installation and configuration process is complete, DynamicPaper can be checked out using the example programs and form files included with the product.

Create a temporary directory and copy the following files from /usr/lib/dynpaper into it.

```
DGIinvoice.prn  dgiinv.dl4  dgiinv.ub
```

Test the DynamicPaper form processor by entering the following command:

```
$ makeforms DGIinvoice.prn
```

Makeforms should announce that it has created the form file DGIinvoice.dpf which should be identical to the same named file in /usr/lib/dynpaper. Copy the new form file into the forms directory.

```
$ cp DGIinvoice.dpf $DPFORMDIR/.
```

To check using dl4, enter run dgiinv.dl4 at the prompt and then use the Adobe reader to view the resulting pdf report file.

```
$ run dgiinv.dl4
```

To check using UniBasic start it and enter dgiinv.ub at its # prompt and check the resulting pdf file.

```
$ unibasic
....
# dginv.ub
```

Note that the programs say the results are in \$HOME which assumes that DPTEMPDIR is going to be the default.

If there is a problem with the results, check the error files “DPerror” and “jeterror” in the local directory. They should be zero length when this works. If they are non-zero length, cat them or view them with an editor. “jetout” is also normally zero and if not might contain additional diagnostic information.

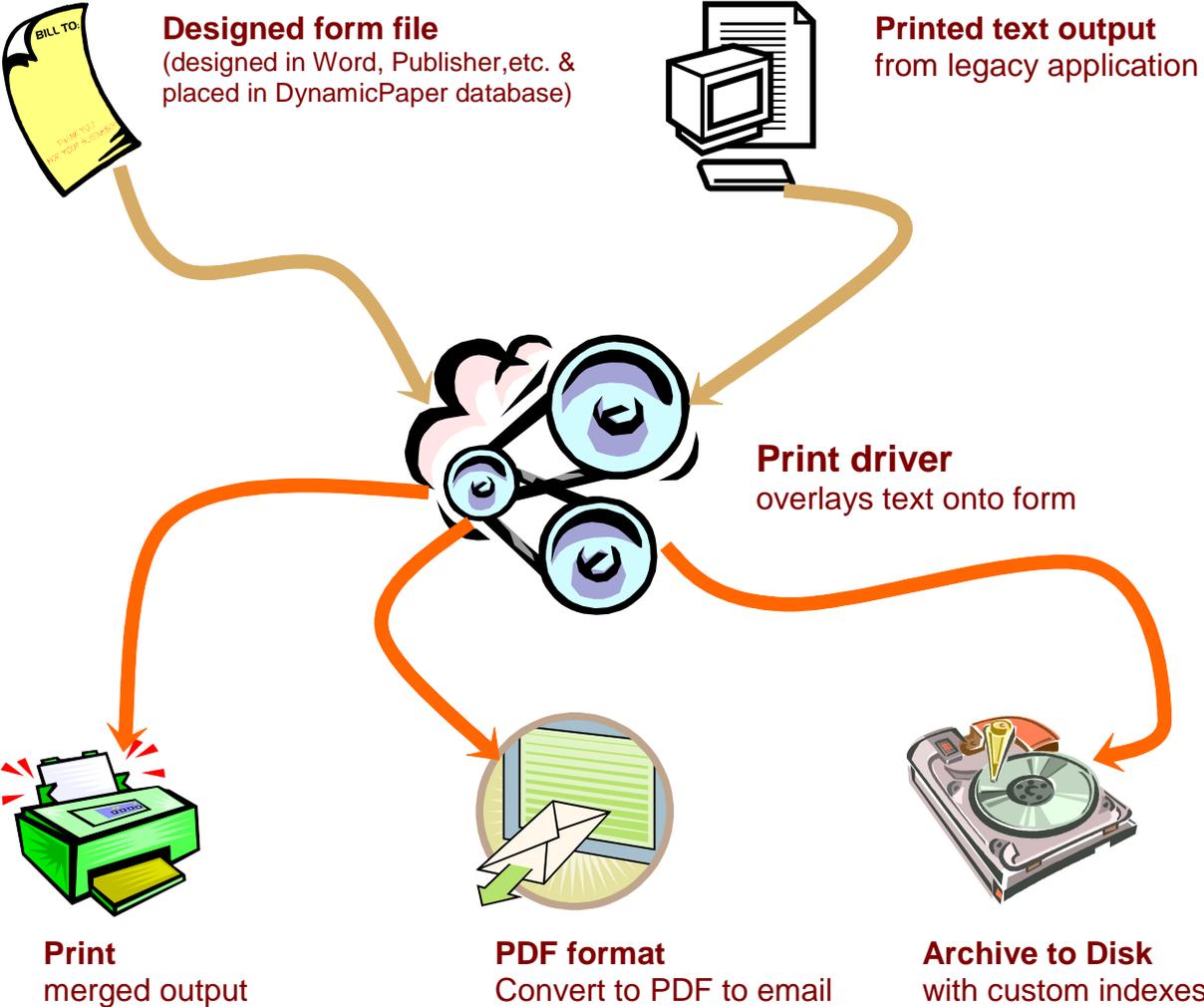
DP Archive Configuration

Configuration of DP Archives is handled by the editable text file `dpsystem.txt` which must be in the same director (folder) as the `dl4` application modules. Its format and the initial supplied default parameters are found below.

```
FileType=F          File type values are:
                    F=FoxPro,
                    S=Microsoft SQL (Not Supported),
                    U=Universal (Not Supported)

FormDir=forms
FilesDir=dpfiles/
TempDir=/usr/tmp
Report=dpreports
Archive=dparchive
Code=dpcode
User=dxuser
UserOptions=dxuseroptions
ErrorLog=dperrorlog
DPDriver=$dynpaper
```

DynamicPaper form merge process



Using DynamicPaper

Making forms

To create a form file for an application, use MS-Word® ,or any other word processor which can print to a file via our designated MS PCL-5 printer driver, to design the form and add graphics. If the form being created is replacing a previously pre-printed form, the pre-printed form can be used as a spacing template by printing copies of the new form and overlaying the old to see how the output fields will align. If Word was used to design the pre-printed form, that file might be taken as is to create the DynamicPaper form file saving considerable time aligning the form and output.

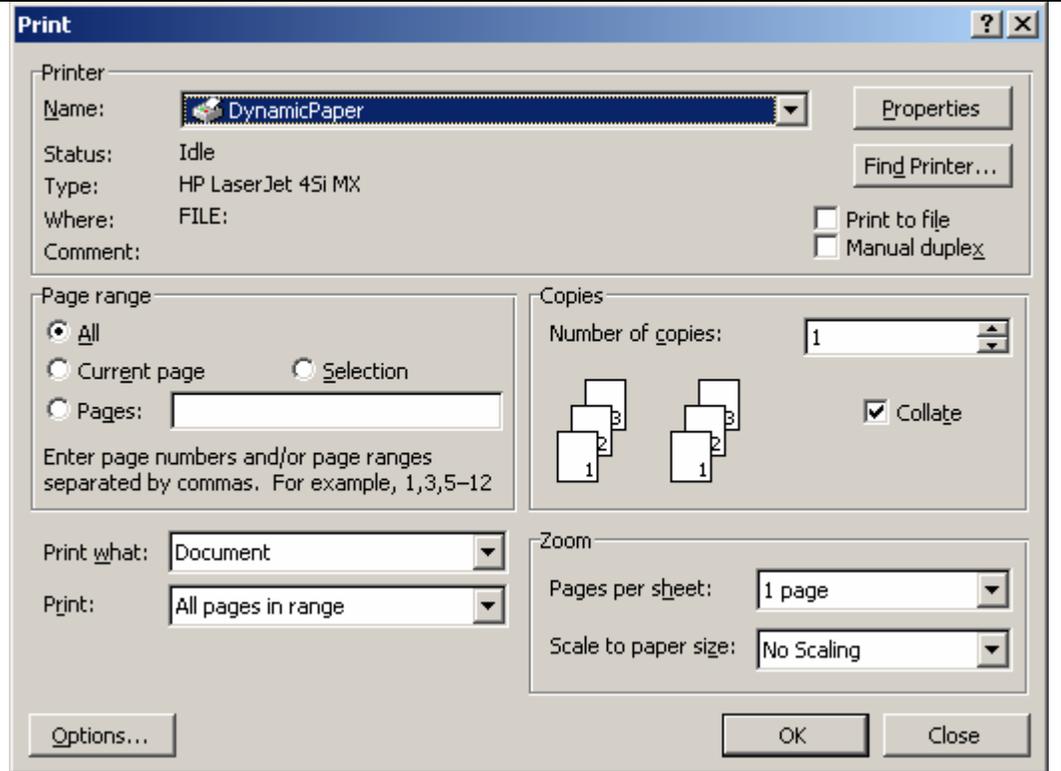
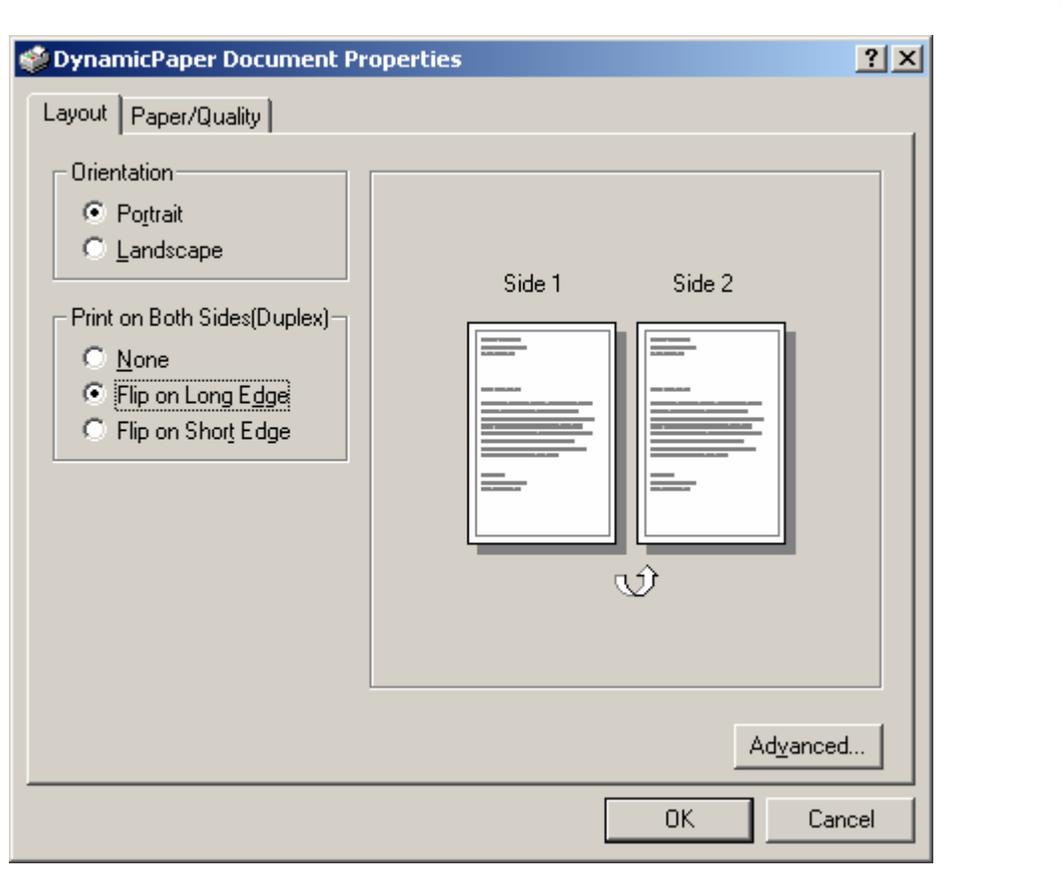
By varying font sizes for certain lines the text and graphics on the new form can be adjusted to fit the application output. The application can be modified to adjust the field placement as well.

One thing to avoid is using Word's graphic tools such as text boxes. While they are compatible and can be used, their size and position is independent of font size and alignment. This means that their use can force all of the text around them to be printed as graphics dramatically increasing the size of the form file and slowing down print performance. Generally speaking, graphics such as logos and headers can be pasted in without causing this problem.

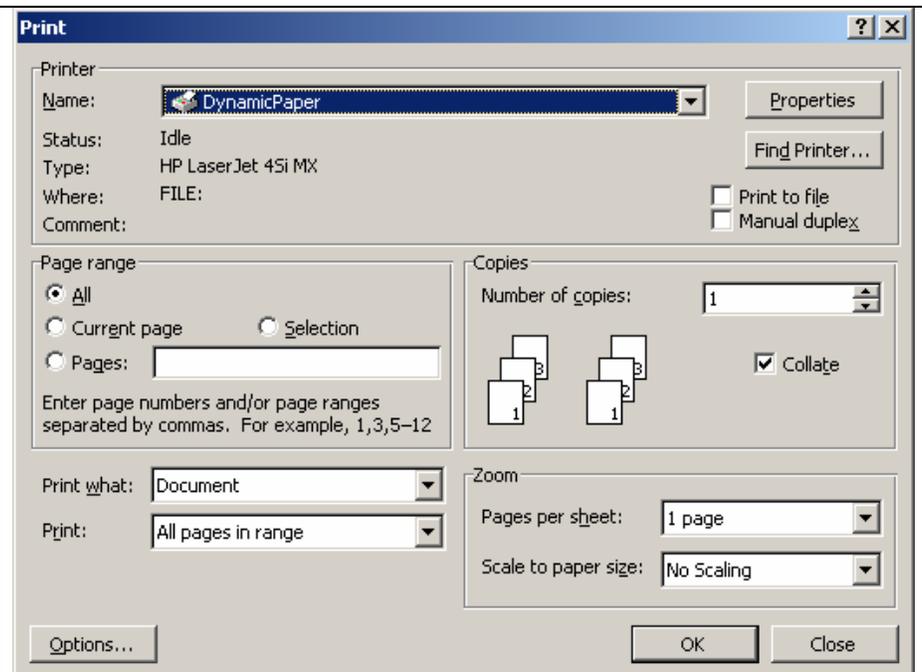
Once a satisfactory form layout has been created, coordinated and aligned with the application's output, print the form to a file, as described below, creating a .prn file. Double sided forms are done as one file with two pages, the first page for the front and the second page for the back. Single or double sided forms, are printed to a single ".prn" file.

When finished designing the form, in Word, select Print from the File menu to get the following window.

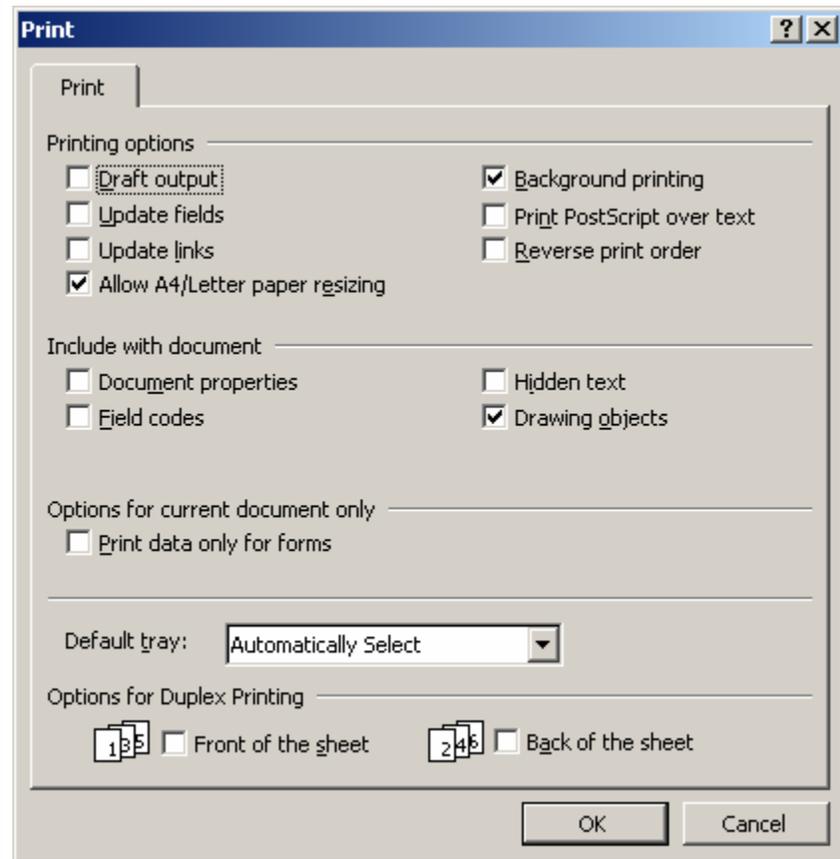
Note that the DynamicPaper printer driver shown selected below must be setup as described in the Installation and Setup section above.

<p>Select the DynamicPaper driver as your Printer Name</p> <p>Print to file does not need to be checked.</p> <p>For double sided, click the Properties button in the top right corner of the screen.</p>	
<p>Select the appropriate flip for “duplex” print option, “Flip on Long Edge” is normal for the Portrait orientation and the other for Landscape orientation. For single side printing “None”. Click the tab “Paper/Quality” for paper tray selection. Then click “OK”</p>	

Clicking the “Options” button in the lower left of the main Print window brings up the following window.



Print options important for forms are Drawing objects, Background printing, and Default tray select. Background printing should be enabled. Drawing objects does not mean graphics pasted into the page and can be unselected without removing all graphics. Default tray should remain as “Automatically Select”. Tray selection is done via “Paper/Quality” tab in Properties window.



If DynamicPaper is used on UNIX® the “.prn” files must be transferred to the system they will be used on. They should be put in a directory where they can be processed with the makeforms tool.

On the UNIX® system, create the “.dpf” forms file(s) by processing the “.prn” file with makeforms. The syntax for running makeforms is simply:

```
makeforms filename
```

where the filename is of a “.prn” file. If the filename is entered without the “.prn” extension and cannot be found when opened, makeforms will try to open it with the “.prn” extension.

If the input file is as a single page form, makeforms will create the form file with the same name but with a “.dpf” file extension. This “.dpf” extension is not essential to making the form file work with DynamicPaper. The form file can be renamed to anything else and still be functional within the application.

If the input file is for a double sided form, makeforms will create two form files. The first will be for the front page and have the same filename with a “.dpf” extension. The second file for the back page will have the same filename with a 2 appended and then a “.dpf” extension. See “Embedded Forms” below for instructions on handling the back page of a double sided form.

For example, formfile.prn would be input by makeforms which would output formfile.dpf if the form is single sided or formfile.dpf and formfile2.dpf if the form is double sided.

The forms created for use with DynamicPaper can be placed in any directory and the environment variable “DPFORMDIR” must be set via user profile, script file, application, or direct command line input to tell DynamicPaper where to find the forms called for in the applications.

The API

The following describes the applications printer driver parameters passed in the “OPEN” statement.

Parameters are passed to the printer driver in the format parameter letter=parameter separated by a space

Syntax:

```
OPEN #chan.no, “$drivername {parameter list} “
```

Parameters:

c=	#of copies	1-n; (defaults to 1)
o=	select other printer options (most correlate to UNIX® lp -o options)	
	“double”	Double sided (duplex) printing
	“lower”	Select lower paper input tray
	etc.	refer to man lp
p=	lp printer device name (defaults to “lpt”)	
	“lptselect”	for transparent print to user’s terminal
	“report”	for PCL output to file report.prn
	“pdf”	for PDF output to file report.pdf
	“email”	for PDF output and automatic emailing
d=	description to print in UNIX® print queue	
f=	filename of form to merge with	
rf=	filename of form to merge repeatedly at every form feed *	
N=	.pdf or .prn report output filename (defaults to “report”	
cpi=	Set base font - Courier at variable characters per inch**	
	“10”	for 10cpi
	“10.5”	for 10.5 cpi
	“12”	for 12cpi
	“16”	for 16cpi
lpi=	Set base line spacing in lines per inch**	
	“6”	for 6lpi
	“8”	for 8lpi
lpp=	Set base line spacing in lines per page**	
	“68”	for 68lpp

(the following parameters support the optional archiving features)

a= filename of archive file to save
id= used to archive using this Report.ID
s= archive subtype (folder)
r1= archive file reference #1
r2= archive file reference #2
r3= archive file reference #3
r4= archive file reference #4

*The repetition initiated by this argument applies to embedded forms (see below). It supersedes any “f=” parameter given.

**These parameters can be customized by changing the dynpaper driver script.

Parameter list example:

OPEN #1, “\$dynpaper c=1 d=AR REPORT a=M0007131200”

Environment Variables

There are three environment variables which can or must be setup to run.

DPFORMDIR	Where the “.dpf “form files are stored (default=\$HOME/dpforms)
DPTEMPDIR	Where the “.prn and .pfd output files are left and temporary working files are created (default=\$HOME)
DPEMAIL	Parameter string for dpmail which must be set when the OPEN parameter string has “p=email”. (no default). It should be as follows: “from=”{sender’s email address}”,to=”{destination email address}”,subject=”{subject line}”[“,msg=”{message text}”] This is set in the application before doing the OPEN. Please, see details below.
DXCODEDIR	Needed only for archiving with DPArchives, it specifies the path to the DynamicXport code directory for dppost.
FILTERFILE	The mnemonic to PCL translation file used only when a form file is not.

Embedded Forms

Form files can also be embedded into the text document stream using the format {f=formfilename}. The printer driver will replace {f=formfilename} with the form.

Embedded forms allow you to produce a multi-page form such as a multi-page invoice in a single file. Another important use of the embedded form is for duplex printing to set the form for the second side of the page.

With no limit on the number of embedded forms used, this function is helpful for creating multiple documents in a single print job. For example, an invoice, packing slip, special product instructions, warranty, and package label can all be printed together in proper sequence facilitating processing, saving collation time and reducing errors.

If the original OPEN statement parameter specifying a form is done with an “rf=” instead of the plain “f=”, all embedded forms will be repeated for subsequent pages which are separated with form feeds. Page overflows will not be overlaid.

Alternate Outputs

DynamicPaper produces a PCL output file which can be directed to a local or remote printer, print spooler, or saved for later printing on any PCL compatible printer. The output file is stored in the directory specified by the environment variable DPTEMPDIR which, if not set, defaults to the user’s HOME directory. When the application requests this function by setting “p=report” in the parameter string on “OPEN”, the driver puts this file in the directory specified by DPTEMPDIR. If an output filename isn’t specified with the “N=” open parameter, the primary output filename will be either “report.prn” or “report.pdf”. Within the application or from the command line, this file can be renamed and transferred to another directory. Unless the applications for storing in an archive, or for similar deferred printing needs. out with the name “report.prn”

When the application specifies “p=email” in the driver’s parameter string on “OPEN” the driver initially outputs the results in a PCL file (e.g. “report.prn”) in the directory specified by DPTEMPDIR. Then, the driver executes a command to translate that file with the optional utility “JetPCL®” (see Appendix C) to produce a pdf output file (e.g. “report.pdf”). This file can be renamed and moved to another directory by the application. As a PDF format file, it can be sent via email or placed on a web site for viewing or downloading.

If the application specifies “p=lptselect” in the “OPEN” parameters for the driver, the driver sends the printer output to the *aux printer* port on the user’s “terminal”.

When the application specifies “a=<archivefilename>” in the driver’s “OPEN” parameter string, the driver, in addition to its other functions, zips the application’s output into a file with that name plus a date and time stamp. In doing this with the raw output from the application, the file does not contain any form selection or control parameter information from the “OPEN” statement. It does however

contain all the embedded form file specifications in the {f=<formfilename>} format. When forms are specified within the output, they can be recombined with the data in the future

With c=0 and a=<a filename>, the application can produce an archive file which can be printed later with DynamicPaper form merging capability enabled. Then, embedded form file declarations will be processed and the data will be merged with the forms.

The driver passes the application output file through “pfilter” with the standard filter file “pcl.prf”. PCLFilter supplied with the product specifies mnemonic translation strings in pairs for the ‘BX’ and ‘EX’ mnemonics and can be modified to add other mnemonic translations or it can be replaced with another filter file by setting the environment variable FILTERFILE to an alternative filename.

Examples

Example 1:

The following shows a sample form, example application code for using it and the results of running the example code output with DynamicPaper.

This sample form and the example application code were all done from scratch in about two hours. Of course, the code is trivial with constants where variables would normally be used. However, it gives an idea of how easy it is to create forms and applications to take advantage of the cost saving power of DynamicPaper.


```
10 Rem dL4 code for running a DynamicPaper example
20 Dim Name$(30),Addr$(30),CityState$(30),PO$(8),OrderDate$(8),ShipDate$(8)
30 Dim Invoice$(8),Cost$(10),Shipping$(10),Tax$(10),Total$(10)
40 Dim I%,I,Billto,Shipto,item,quan,sku,desc,cost
50 Open #1,"$dynpaper f=DGIinvoice.dpf p=report"
60 Name$ = "Maxi-Mouse Bioengineering";Addr$ = "Park Avenue, Suite 1024"
70 CityState$ = "Anaheim, CA 92624"
80 PO$ = "10034";OrderDate$ = "10/03/04"
90 ShipDate$ = "10/05/04";Invoice$ = "100651"
100 Cost$ = "18400.00";Shipping$ = " 50.00"
110 Tax$ = " 1472.00";Total$ = "19920.00"
120 Billto = 7;Shipto = 45
130 item = 5;quan = 7;sku = 18;desc = 28;cost = 67
140 For I = 1 To 14 \ Print #1; \ Next I
150 Print #1; Tab (Billto);Name$; Tab (Shipto);Name$
160 Print #1; Tab (Billto);Addr$; Tab (Shipto);Addr$
170 Print #1; Tab (Billto);CityState$; Tab (Shipto);CityState$
180 Print #1;'LF';'LF'
190 Print #1; Tab (7);PO$; Tab (31);OrderDate$;
200 Print #1; Tab (50);ShipDate$; Tab (68);Invoice$
210 Print #1;'LF';'LF'
220 Print #1; Tab (item);"1"; Tab (quan);".5 M"; Tab (sku);"TM09";
230 Print #1; Tab (desc);"Transgenetic Mouse 9 DNA"; Tab (cost);" 1650.00"
240 Print #1; Tab (item);"2"; Tab (quan);".25 M"; Tab (sku);"KKad8";
250 Print #1; Tab (desc);"Kong growth genes"; Tab (cost);" 2200.00"
260 Print #1; Tab (item);"3"; Tab (quan);".75 M"; Tab (sku);"FC42";
270 Print #1; Tab (desc);"FrankenCorn Patented gene"; Tab (cost);"14550.00"
280 For I = 30 To 48 \ Print #1; \ Next I
290 Print #1; Tab (cost);Cost$
300 Print #1; Tab (cost);Tax$
310 Print #1; Tab (cost);Shipping$
320 Print #1; Tab (cost);Total$
330 Close #1
340 End
```


The following are additional examples of OPEN statements with descriptions of the DynamicPaper functionality they produce.

Example 2:

```
OPEN#0, "$dynpaper f=invoice1.dpf p=email"
```

This statement sets up to produce the file "report.pdf" in the user's home directory which will contain the results of merging the form file "invoice1.pdf" from the forms directory selected by the user's environment variable DPFORMDIR with the output of the application.

Example 3:

```
10 Open #1, "$dynpaper f= salesrep.dpf o=double p= hp8100dnp234 d=Sales"
20 For I = 1 To 14 \ Print #1; \ Next I
30 Print #1; Tab (20); "Slick Williams"
40 Print #1; 'LF'; Tab (20); "Industrial Widgits"; Tab (60); "$30,000"
50 Print #1; 'LF'; 'LF'; Tab (35); "Sudsville, WI"
60 Print #1; 'LF'; 'LF'; 'LF'; "Page 1 of 2"
70 Print #1; "{f= salesrep2.dpf}"
80 For I = 1 To 14 \ Print #1; \ Next I
90 Print #1; Tab (20); "Willy Lohman"
100 Print #1; 'LF'; Tab (20); "Mercury Shoes"; Tab (60); "$53"
110 Print #1; 'LF'; 'LF'; Tab (35); "Fayette, GA"
120 Print #1; 'LF'; 'LF'; 'LF'; "Page 2 of 2"
130 Close #1
```

This code example sets up to print data from the application merged with the form file "salesrep.dpf" as a double sided form on the system printer named "hp8100dnp234". The form file used for the back side of the form is defined by the application in statement number 60 with the embedded form directive "{f= salesrep2.dpf }". In the UNIX® print queue, the print job will have the name "Sales".

Example 4:

OPEN#0, "\$dynpaper f=expenses.dpf p=report a=May2003"

This statement would set up to put the application output zipped in the archive file "May2003.zip" and the application output merged with the form file "expenses.dpf" in the file "report.prn" in the user's home directory.

Useful tools

A useful tool when working with DynamicPaper forms and applications is a Windows utility for viewing the generated PCL files. You can use the feature of DynamicPaper which provides output to a file "report.prn". The results can then be viewed with Visual PCL/2 Version 2.1 which also uses the JetPCL® utility to produce its screen images of PCL documents.

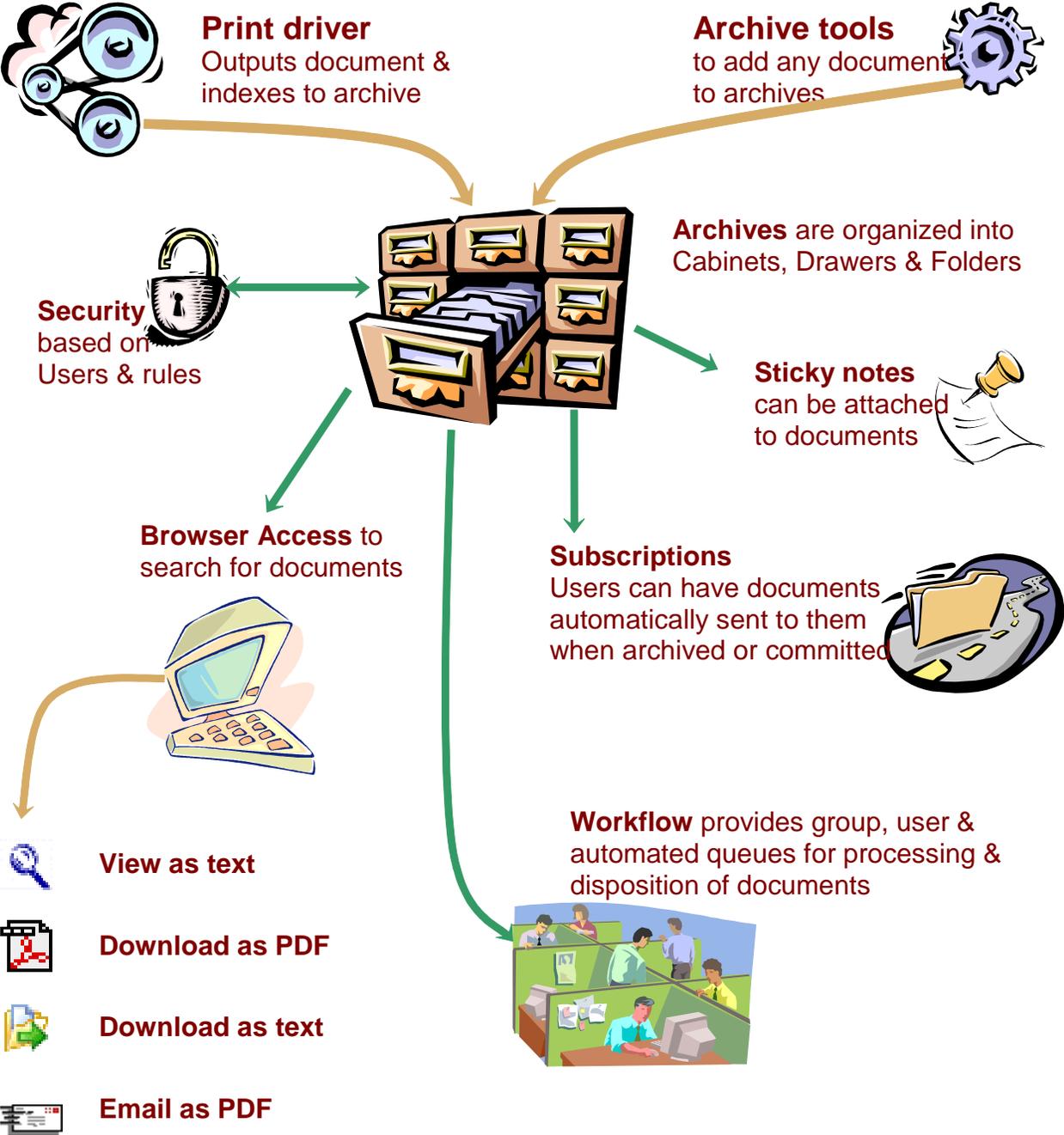
The product is available from Visual Software whose web site's URL is:

<http://www.visual.co.uk/>

Contact information in the UK is:

Visual Software
PO Box 399
DORKING RH5 4WX
United Kingdom
Telephone **+44 7092 383 702**
Fax **+44 20 7691 7007**
E-mail geddes@visual.co.uk

DynamicPaper archive process



DP Archives

DynamicPaper Archives provides ability to conveniently and automatically archive documents and reports when they are printed. The report content is stored efficiently in compressed format with any required report forms stored separately. Documents or reports not created with DynamicPaper can also be archived and managed with DP Archives.

DP Archives organizes documents into virtual cabinets and drawers creating indexes for quick retrieval. The standard supported search allows retrieval of reports in a selected date range from a selected drawer such as “Invoices” in a selected cabinet such as “Billing”. The user can define optional indexing of selected content from several report fields for easy retrieval of any report. The additional user defined indexed fields could be, for example, the invoice number, customer name, or PO number.

An obvious impediment to progress towards the adoption of the “paperless office” is the mistaken belief that there are certain important capabilities no longer possible without the paper copy. An important example is the need to make annotations on reports and documents. DP Archives cleanly solves this problem with Sticky Notes which allow the user to add virtual notes “stuck to” any selected archived report.

DP Archives is implemented with a dynamicXport application interface for full web-based access. Local or remote, users can retrieve reports display as text in which they can search for keywords or browse from page to page. They can also download a report as text or pdf or email a copy of the report in pdf format.

DP Archives also supports security based on assigning report access rights to users. Further, by modifying the conditions.lib dl4 code the user can write additional code stubs to add further access control defined in terms of report content

The problem of automatic report distribution is solved by DPArchives with report subscriber lists. Users can subscribe or be subscribed by an administrator to receive an emailed pdf copy of any document they have access to whenever it is generated or released for distribution by a workflow commit.

.

Database Initialization and definition

The stand alone program dpcreate.dl4 is used to create, initialize, and backup the DPArchive database.

dpcreate.dl4 and dpsystem.txt control file

Before running dpcreate, the default settings for the configuration file dpsystem.txt should be checked to see if they need to be modified. The following shows what an initial unexceptional run of the dpcreate utility looks like. Note the default settings indicated. If they need to be changed, a “y”

response brings up the file in the editor through which it can be edited and saved. The program automatically restarts when this option is taken.

```
#dpcreate.dl4
DPCREATE: Building DynamicPaper Archive Database
```

The system control file (dpsystem.txt) has been created.

The dpsystem.txt configuration file currently is:

FileType=F

FilesDir=dpfiles

FormDir=forms

TempDir=/usr/tmp

Report=dpreports

Archive=dparchive

Code=dpcode

User=dxuser

UserOptions=dxuseroptions

ErrorLog=dperrorlog

DPDriver=\$dynpaper

You can view/edit using the editor defined in your EDITOR environment variable.

If no editor is defined, vi is used (notepad on Windows).

Use the editor commands to edit or save changes and return to finish the install process.

Do you want to view/edit the dpsystem.txt file with UNIX editor? [Y/N] n

Creating database files for DynamicPaper Archive Server!

Building: <666> dpfilesdpreports!

Building: <666> dpfilesdparchive!

Building: <666> dpfilesdpcode!

Building: <666> dpfilesdperrorlog!

DynamicPaper Archive Server database files created!

When run subsequent to the initial setup of the database, dpcreate backups up the existing database before building new datasets. The following appears prior to "Creating database files..."

Backing up old database files to olddpfiles/ !

Copying dpfiles/dpreports.dbf olddpfiles/dpreports.dbf

Copying dpfiles/dparchive.dbf olddpfiles/dparchive.dbf

Copying dpfiles/dpcode.dbf olddpfiles/dpcode.dbf

Copying dpfiles/dperrorlog olddpfiles/dperrorlog

These backup files overwrite previously created backup files created by dpcreate.

DPSYSTEM Parameters File

The DPSYSTEM file “**dpsystem.txt**” contains the global configuration parameter settings for DPArchives. The file is a text file maintainable with any text editor. The file is located on the application server in the same directory as the DPArchive application programs. The format of the file is Parameter Name = Parameter Setting.

The parameters are as follows :

FileType	=	F=Foxpro Full-ISAM (default) S=Microsoft SQL	
FilesDir	=	dpfiles	Directory to find DP Archives files on Application Server
FormDir	=	forms	Directory to find DP forms files on Application Server
TempDir	=	/usr/tmp	Directory use for temp files on Application Server
Report	=	dpreports	Report index and information*
Archive	=	dparchive	Archive index and information*
Code	=	dpcode	(reserved for future use)
User	=	<path>/dxsystem.txt	DynamicXport system parameters file to get dxuser
UserOptions	=	<path>/dxsystem.txt	Same as prior, to get dxuseroptions dataset
ErrorLog	=	dperrorlog	Log file for errors for trouble shooting reference
DPDriver	=	\$dynpaper	Name of DynamicPaper driver

* These datasets are created by dpcreate and described below under Administration.

Report dataset

The report dataset “**dpreports**” contains control information about reports defined for the application. Its records link reports to their virtual drawers and cabinets where the archived reports are stored. The records also link to the report forms which are merged with the report content by the DynamicPaper driver. The dataset is implemented as a FoxPro full ISAM file.

Report file field definitions:

ID Report # (or sections of arch) (Drawer)

The report ID can be up to 8 characters in length and can consist of alphanumeric characters, comma, period, dash and underscore only. The ID is the unique identifier of the report and is case sensitive.

OrgID (future use)

The organization id can be up to 8 characters to describe the Organization such as a department name or company name.

ParentID (cabinet)

The ParentID is the report ID of the parent of this report and can be up to 8 characters in length and is the ID of another report record which is organizationally the parent of this report.

If null this report has no parent and is therefore a level 1 “report class” considered the virtual Cabinet for all reports for which it is their parent. Otherwise, it is the id of a level 1 report class

Title

The Title can be up to 40 characters in length and is the title of report and default report description.

OptionID

The OptionID can be up to 40 characters in length and ties this report to the program that creates it (Program name in SF.PROGRAMS for now)

Form

The Form can be up to 60 characters in length and is the path and Filename of the form file to merge with the archived contents to create this report.

Directory

The Directory can be up to 60 characters in length and is the Archive Directory where these reports are filed.

Ref1

The Ref1 field can be up to 20 characters in length and is the description of first reference (key 3)

Ref2

The Ref2 field can be up to 20 characters in length and is the description of second reference (key 4)

Ref3

The Ref3 field can be up to 20 characters in length and is the description of third reference (no key)

Ref4

The Ref4 field can be up to 20 characters in length and is the description of fourth reference (no key)

Subtype

Subtype is a single character Y/N flag specifying whether a subtype is required.

OptionString

The OptionString can be up to 60 characters in length and contains anything that might be sent to the DynamicPaper printer driver. It can also contain options for the application which produces this report.

secure1

The Secure1 field can be up to 10 characters in length and can be one of three values.

“High” (must be on userOptions=dpuseroptions list)

“Med” (user must be logged in)

“Low” (none)

conditions

The conditions field can be up to 60 characters in length and defines conditions filtering retrieved archived copies of this report. (condition rules format defined by user). It is used to limit what reports in a drawer a user has access to, based on their OrgID, AppID etc. For example it could contain “AppID is CustomerID” where CustomerID is the content of the Ref1 field meaning users can only see this report when their AppID=Ref1.

Duplicates

The duplicates field is a “Yes” or “no” flag field of up to 3 characters which must be “YES” to allow multiple dparchive entries pointing to the same report file.

Sendwhen

The Sendwhen field is a single character code defining when it is OK to send this report by email to subscribers. It values can be:

“A” = as archived (default)

“C” = as committed

Emailstring

The Emailstring field can be up to 900 characters for the default additional email parameters (for dpadd)

Archive dataset

The archive dataset “**dparchive**” contains the index and information about the report archive files. Each record indexes a particular report file in a virtual drawer in a virtual cabinet defined by the report file record for the particular report. The archive record contains some content information from the archived report used for user indexed retrieval of the specific report.

The archive dataset record fields are defined as follows:

ReportID

The ReportID is up to 8 characters in length and links the archive record to its report in the report dataset matching the report ID(Drawer) field.

Type

The output type field is a 1 character code identifying this archived report record encoded as:

- E=e-mail
- F=fax
- P=print
- N=note

Date

The date is 6 numeric characters formatted as yymmdd and is the identifying date when the report was archived.

Time

The timestamp is 6 numeric characters formatted as hhmmss and is the identifying time when the report was archived.

File

The File field is up to 60 characters in length is the filename including dir path of the archived report.
(default format is <Report.Dir>/<Report.ID><date><time>)
(default followed by .nt if it is an attached sticky note)

Desc

The description field can be up to 60 characters in length..

UserID

The UserID field can be up to 8 characters in length and is the ID of the user who archived the report file. It is from the DynamicXport user ID or from their Unix LOGNAME

Ref1

This field can be up to 20 characters in length and contains the search key (3) content as defined in the report dataset record. It should contain the key providing the most direct access such as an invoice number or PO number.

Ref2

This field can be up to 20 characters in length and contains the search key (4) content as defined in the report dataset record. It should contain a secondary key for the archived report such as customer number or supplier name.

Ref3

This field can be up to 20 characters in length and contains report record content for additional search text (no key) and can also be used in access control filtering through rules implemented in a conditions.lib routine.

Ref4

This field can be up to 20 characters in length and contains report record content for additional search text (no key) and can also be used in access control filtering through rules implemented in a conditions.lib routine.

Kommit

This field of 12 numeric characters reserved for future use as the timestamp of commit formatted as yymmddhhmmss and marks the point after which the archive can no longer be changed or deleted. It will also be the time when the report requiring a commit is released and emailed to subscribers.

Email

This field can be up to 60 characters long is the E-mail address to which the report was e-mailed.

Name

This field can be up to 30 characters long is the name of person to whom the report was e-mailed or faxed.

Subtype

The subtype field is three characters long as a Subtype code(Folder)

Form

This field can be up to 60 characters in length and is the absolute path and filename of the form file for this report. It might be null since some reports do not have forms. It might be different from the Form field of the report dataset record because forms change over time. It is the form with which the report content was originally mated and merged to ensure a true copy of the original report can be reproduced.

UserOptions dataset

The useroptions dataset “**dxuseroptions**” is the same one created and used by DynamicXport and is located as defined in dxsystem.txt. It is used by DPArchives for the management of user subscriptions and report access privileges with two new record types added as shown below.

UserID

The UserID field can be up to 8 characters in length. It is the user for whom the content of the record applies.

ID

The ID field can be up to 8 characters in length and identifies the report this record relates to.

Type

The type field is a single character identifying what kind of record. The new types added for DPArchives are 7 and 8.

Type=7 for reports user has access to

Type=8 for reports user is subscribed to

Privilege

This is a single character field not relevant to DPArchive use.

Seq

The sequence field can be up to 3 numeric characters in length and identifies the sequence in which the user’s accessible or subscribed reports is maintained and displayed.

User dataset

The user dataset “**dxuser**” is the same one created and used by DynamicXport and is located as defined in dxsystem.txt. It is used by DPArchives for the management of user subscriptions and report access privileges. This is done by an alternate use of the User record Device field to contain the two fields described below.

CopyUserReportID (Device 1-8)

This field of up to 8 characters in length is the User ID of another user whose report access privileges are conferred on this user

CopyUserSubscribeID. (Device 9-16)

This field of up to 8 characters in length is the User ID of another user whose report subscriptions are added to this users.

For the description of other fields and their usage refer to the dynamicXport User Guide.

Administration

The administration of DPArchives is managed by a dynamicXport application servicing four options under a sub-menu of “Paper Administration” in the main dynamicXport menu. Access to these options is limited by dynamicXport to specific privileged users. The general user is serviced by a dynamicXport application servicing the View Archives option on the main menu.



Reports

Under Paper Administration the “Reports” option goes to a display of a list of all the available report records.

Reports			
Report ID	<input type="text"/>	<input type="button" value="Add Another"/>	<input type="button" value="Search"/>

Search Results				
Parent ID (Cabinet)	Report ID (Drawer)	Title	Option ID (Program)	
	AP	Accounts Payable		<input type="button" value="Edit"/>
AP	AP001	A/P Checks	AP.PRINT	<input type="button" value="Edit"/>
	AR	Accounts Receivable		<input type="button" value="Edit"/>
AR	AR001	A/R Statements	AR.STATEMENT	<input type="button" value="Edit"/>
AR	AR002	Customer Master List	AR.CUSTLIST	<input type="button" value="Edit"/>
AR	AR003	Daily Cash Receipts Journal	AR.CASHJOURN	<input type="button" value="Edit"/>
AR	AR004	A/R Trial Balance	AR.TRIAL	<input type="button" value="Edit"/>
AR	AR005	Monthly Cash Receipts Journal	AR.CASHJOURNM	<input type="button" value="Edit"/>
AR	AR006	Monthly A/R Manual Posting Register	AR.OPENLDREGM	<input type="button" value="Edit"/>
AR	AR007	A/R Detail Aging	AR.AGINGD	<input type="button" value="Edit"/>
	BI	Billing		<input type="button" value="Edit"/>
BI	BI001	Invoices	BI.PRINT	<input type="button" value="Edit"/>
BI	BI003	Daily Invoice Register	BI.INVREGD	<input type="button" value="Edit"/>
BI	BI004	Monthly Invoice Totals Report	BI.INVREGM1	<input type="button" value="Edit"/>
BI	BI005	Recurring Billing Audit Report	BI.BILLAUDIT	<input type="button" value="Edit"/>

This listing shows the cabinet and drawer organizational hierarchy. The add and selection of a report to edit functions are supported from this page. Delete and update are supported from the edit page shown next.

Entering a new report ID and clicking the “Add” button initials a new entry. The add function produces the following screen which shows an example of what might be a new report record.

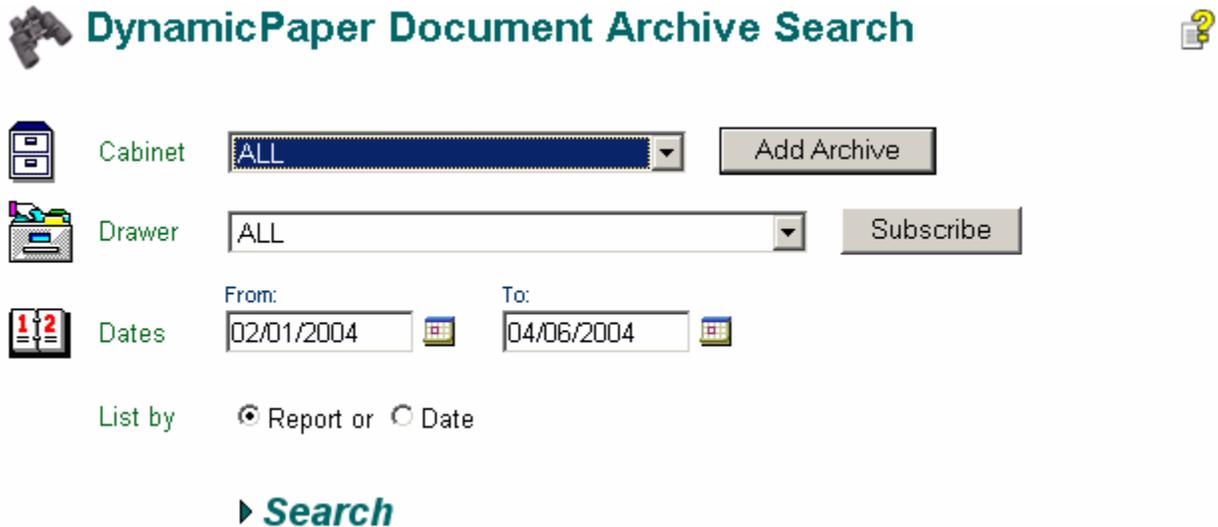
Report Definition	
Report ID	AP002
Organization ID	(MASTER) ▾
Parent ID (Cabinet)	AP ▾
Title	Direct bank payment
Option ID (program)	directpay
Form	bankpayment
Directory	DARCHIVE\$/ap/ap002
Reference 1	PO number
Reference 2	vendor id
Reference 3	vendor invoice #
Reference 4	vendor invoice date
Subtype Required ?	No ▾
Options string	
Email string	
Send subscriptions when ?	Archived ▾
Allow multiple indexes to same file ?	Yes ▾

Save Back

The same input page is presented for editing an existing report with the addition of “Delete” button at the bottom.

Archives

The archive records are maintained through selection the “Archives” option from the sub-menu and the first page presented is a search criteria selection screen shown following.



DynamicPaper Document Archive Search 

Cabinet

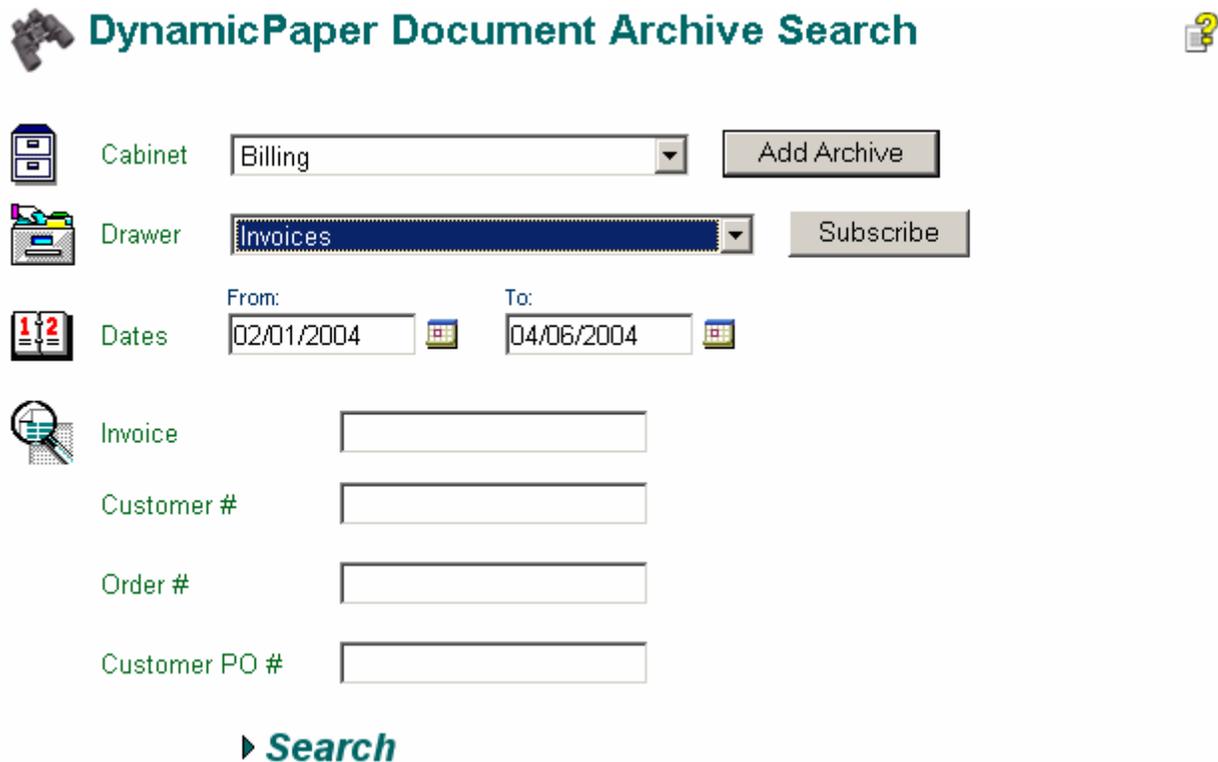
Drawer

Dates From: To:

List by Report or Date

▶ Search

This represents the simplest case for this page with a setup to retrieve all archived reports in a date range. The more useful and more targeted example shown next has the additional advantage of being significantly quicker in the retrieval since its search space is vastly more limited.



DynamicPaper Document Archive Search 

Cabinet

Drawer

Dates From: To:

Invoice

Customer #

Order #

Customer PO #

▶ Search

By selecting a particular report (drawer), the page becomes augmented to include the report specific search fields added from the report record which as for example “invoice” in Ref1, “Customer #” in Ref2, “Order #” in Ref3, and “Customer PO # “ in Ref4. The archive records for archived invoice reports will have the report content specified for these fields because the DynamicPaper driver will have been given it as parameters in the open call that produced the report. Alternately, that content for these fields might have been from an application calling the dpadd routine to place a report in the archives with a parameter string including the ref1, ref2, ref3, and ref4 content.

This search in our sample database returned the following data displayed in the archive administration record list page.

 **DynamicPaper Document Archive Search**  [▶ New Search](#)

Search results							
Options	Cabinet	Drawer	Folder	Date	Description	User	Notes
<input type="button" value="Edit"/>	BI	BI001		02/06/04 02:36:43PM	994374 MBOOK - Mondragon Bookstore	admin	<input type="button" value="Add"/>
<input type="button" value="Edit"/>	BI	BI001		02/06/04 02:39:37PM	651377 DGI050 - Designer Genes Inc.	admin	<input type="button" value="Add"/>

[▶ New Search](#)

Clicking on Edit for the first entry brings up the following page for editing the record.:

Archive Record	
Report ID	BI001
Type	print
Date/Time	02/06/2004 02:36:43 PM
Filename	/home/richard/icn/dparchive/reports/bi/bi00
Form	invoice
Description	994374 MBOOK - Mondragon Bookstore
User ID	admin
Reference 1 Invoice	994374
Reference 2 Customer #	MBOOK
Reference 3 Order #	157481
Reference 4 Customer PO #	350263/TOP PRIOR
Subtype	

From this page, the record can be either deleted (click Delete) or updated (click Save) after editing.

User's Reports

The access to reports is controlled by UserOptions records of type 7. The records are not directly accessed for maintenance. Instead, a list selection scheme manages these. The first page displayed when selecting this option for maintenance is the following screen for selecting the target user.

User Reports

Enter a full or partial field value to search

Search by : User Name ▼ Search for :

Search

Search results

Org ID	User ID	Name	App ID	More Info	E-mail
ICN	JOEL	test case sens		Reports	
(MASTER)	admin	Administrator		Reports	joel@dynamic.com
DCI	dchad	Douglas Chadwick	DCIINT	Reports	
COLLMGR	kgrotz	Joel Grotzinger		Reports	
omaha	omaha	test hidden		Reports	
COLLECT	qwcol	qwcol		Reports	
abc	tire	Demo tire user	K400	Reports	
ubs	ubs	Chris Raffo		Reports	

Selecting one by clicking on the link “Reports” brings up the access list editing page. This page has an available list and the user’s currently accessible list. The organization of the lists follows the cabinet and drawer organization described before.

To manipulate the lists a report entry which is to be moved to the other list is selected by clicking on it. Then the “<” or “>” button is clicked to move the selection in the appropriate direction. Whole cabinets can be moved by selecting the cabinet heading instead of an individual drawer. The entire list can be moved in either direction by clicking either the “<<” or “>>” button. Other than by cabinet or by the whole list, only single selections can be moved at one time.

To the right of the user’s accessible list is pair of up and down arrows. These are for changing the sequence of the elements of that list. Presently, these can be ignored because the order of the list does not have any current relevance. However, the possibility exists that a future functionality will make use of the sequencing. This warrants having the capability; but, ignores it for the present.

Just below the list of the user’s accessible reports is a box for indicating another user ID whose accessible reports should also be made accessible to the target user. This does not cause a copy of that user’s accessible list to be made at the present time. Instead, when access privileges are determined for a user the copy from user’s accessible reports at that time become added to those directly declared in this user’s list. That copying only works for one level. If user A is given access to user B’s reports and user B has been given access to user C’s by this means, user A will not gain access to user C’s reports through this copy access rights mechanism.

On the bottom are buttons for saving or canceling the changes made in the user's list by this page. It is essential to remember to click "Save" instead of the browser's back button for navigation to update the lists.

User Reports
Help

Available reports :

Accounts Payable
A/P Checks
Accounts Receivable
Customer Master List
Daily Cash Receipts Journal
A/R Trial Balance
Monthly Cash Receipts Journal
Monthly A/R Manual Posting Register
A/R Detail Aging
Billing
Daily Invoice Register
Monthly Invoice Totals Report
Recurring Billing Audit Report
Recurring Billing Product Summary

>
<

>>
<<

icn test 's reports :
Accounts Receivable
A/R Statements
Billing
Invoices
Credit Card Charges Journal
Order Processing
Printed Orders

↑
↓

AND Copy User Reports of User ID:

Save **Cancel**

User's Subscriptions

The user's subscription to reports is controlled by UserOptions records of type 8. The records are not directly accessed for maintenance. Instead, a list selection scheme manages these. The first page displayed when selecting this option for maintenance is almost the same as that displayed for selecting the target user for the access list shown above (on the previous page).

User Subscriptions

Enter a full or partial field value to search

Search by : Search for :

Search

Search results

Org ID	User ID	Name	App ID	More Info	E-mail
ICN	JOEL	test case sens		Subscriptions	
(MASTER)	admin	Administrator		Subscriptions	joel@dynamic.com
DCI	dchad	Douglas Chadwick	DCIINT	Subscriptions	
COLLMGR	kgrotz	Joel Grotzinger		Subscriptions	
omaha	omaha	test hidden		Subscriptions	
COLLECT	qwcol	qwcol		Subscriptions	
abc	tire	Demo tire user	K400	Subscriptions	
ubs	ubs	Chris Raffo		Subscriptions	

Click on “subscriptions” to select the user to be serviced. This will bring up the following page for manipulating the user’s subscription list.

Report Subscriptions

[Help](#)

Available reports :

- Accounts Payable
- A/P Checks
- Accounts Receivable
- Customer Master List
- Daily Cash Receipts Journal
- Monthly Cash Receipts Journal
- Monthly A/R Manual Posting Register
- A/R Detail Aging
- Billing
- Daily Invoice Register
- Monthly Invoice Totals Report
- Recurring Billing Audit Report
- Recurring Billing Product Summary
- Suspended Customers Report

rcnsicnt 's reports :

- Accounts Receivable
- A/R Statements
- A/R Trial Balance
- Billing
- Invoices
- Sales Tax Report

AND Copy Report Subscriptions of User ID:

Save **Cancel**

This works the same as the page for managing the user’s list of accessible reports. Edit and click on the “Save” link to update the subscription list records.

Using DPArchives

The dynamicXport menu selection to “View Archives” is the access to search for, subscribe to, copy, examine, and send reports from the archives. Clicking on the menu selection “View Archives” brings up almost the same search selection page shown above for administering the DPARCHIVES dataset. However, the “Add Archive” button is not made available.

Subscribing to reports

Reports can be subscribed to by selecting the report (drawer) from the drop down list and then clicking on the “Subscribe” button.

Searching the Archives

The same instructions for searching described in the administration section above apply here. Selecting a drawer to narrow the search significantly reduces the search effort and time to retrieve the desired reports. When the search link is clicked the retrieved reports are displayed in the following formatted page.

Options	Cabinet	Drawer	Folder	Date	Description	User	Notes
	BI	BI001		02/06/04 02:36:43PM	994374 MBOOK - Mondragon Bookstore	admin	Add
	BI	BI001		02/06/04 02:39:37PM	651377 DGI050 - Designer Genes Inc.	admin	Add

Although similar to the page view in the administrative mode, the left column is now offering different options and the “Edit” option is no longer available.

Viewing Archived Reports

To select a specific report from the list of those retrieved and view it in text mode, click on the magnifying glass icon for that report. This will access the selected report and display it in the following page.

DynamicPaper Document View



The screenshot shows a document viewer window titled "DynamicPaper Document View". The main content area displays a report page with the following text:

```

PAGE-1
DESIGNER GENRES NT.
1234 Biotech Blvd.
Somerville CT 06901

PRE05D      06/26/03      157763      SAME                      06/26/03      351377

Recurring Billing CREDIT CARD      Sep 2003 Service

1 Monthly Services for: pre-loans.ccm
2 SOLO Billing for NT Monthly Service      1      1      35.00      35.00
3 Bill to: 4460-XXXX-XXXX-7001 Exp: 01/06
4 Thank You For Your Business !
  
```

At the bottom of the window, there is a navigation and search interface:

- Navigation buttons: <<, < Prev, Page 1 (dropdown), Next >, >>, Close, and a help icon.
- Page information: 1 - 4 of 4 pages
- Search controls: Find what: [text box], Find, Case Sensitive, and an "Add Note" button with a sticky note icon.
- Page range controls: Pages from: [text box] to: [text box], with icons for print, refresh, and save.

This page comes up in a new window and has several significant functional options. The report page can be scrolled through and any page of the report can be selected for viewing with the control functions on the lower left corner of the window. From left to right, “<<” selects the first page, “<Prev” selects the previous sequential page, the drop down list of available page numbers allows jumping to a specific page, “Next>” selects the next sequential page and “>>” selects the last page. To the right of these controls are the “Close” button and help icon. The first closes this window.

Below the report navigation buttons is a search box and “Find” button. This is for searching the report itself for specific content and the search can be done as case insensitive or sensitive by setting the check box to the right.

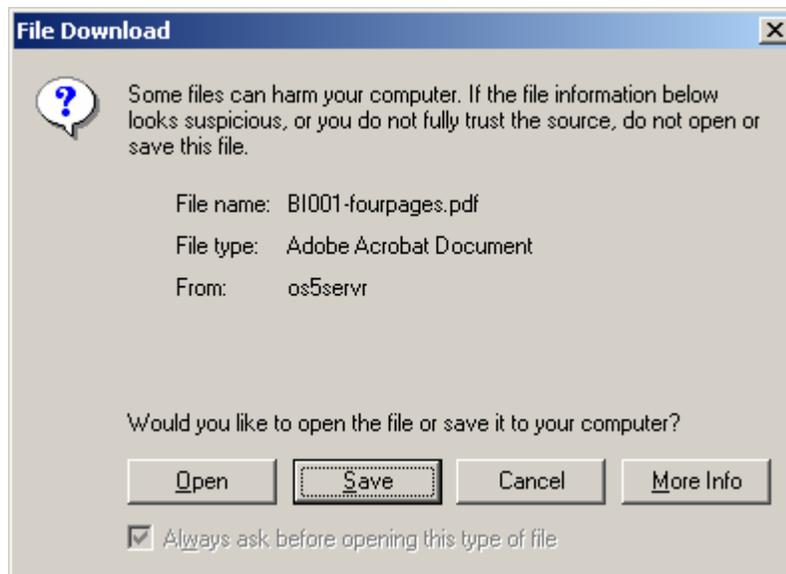
A little to the right of center under the page of pages display is a icon and “Add Note” button for adding a sticky note. Sticky notes and their functions are described below.

In the bottom right corner of the page is a box for selecting downloading or email options. It has first and last page boxes for selection of a few pages from a larger report. Below the page range selection boxes are option icons like those in the options column of the archived reports list. From left to right these icons and their options are as follows. The Adobe acrobat PDF icon is for downloading the report in PDF format. The envelope icon is for emailing a copy of the report in PDF format. The last icon is a document in a folder overlaid with a green arrow to the right. This icon is for requesting the report content to be downloaded in raw text format without an overlaid report form. All three options are described in more detail below since they are options which can also be requested on the archived report list page.

Download Archived Report in PDF format

In the options column on the left of the archived reports list the next icon to the right of the magnifying glass is the Adobe acrobat PDF icon. Click on it for requesting a download of the report in PDF format. This produces a copy of the originally sent report with the archived content overlaid on any associated forms.

The download function is partially handled by the standard mechanism in the user's browser. Since there are several browsers available and even several versions of Internet Explorer which could be used, the following is described using Microsoft's Internet Explorer version 6.0 under Windows 2000 Professional Edition. When the icon is clicked the browser gets the information to setup the download and displays the following dialog box.

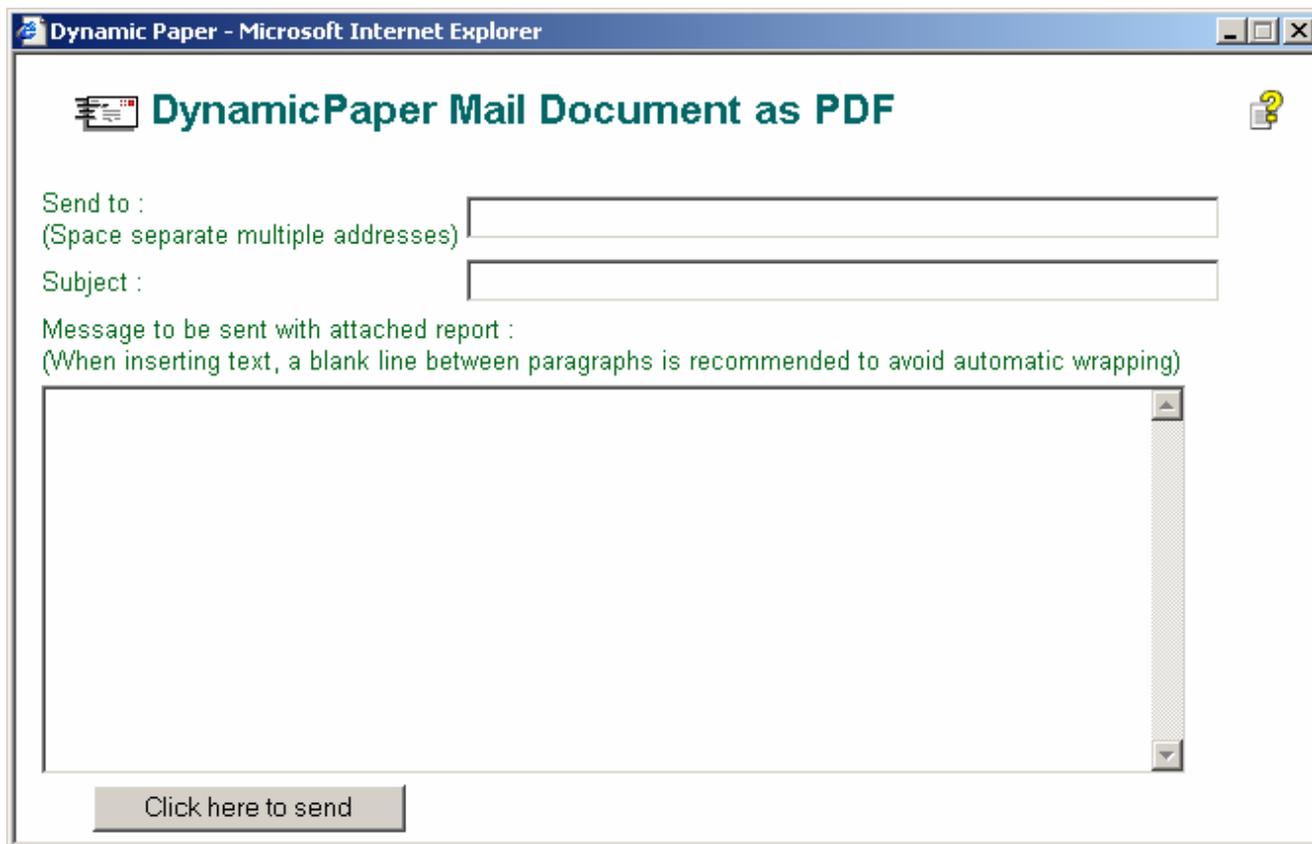


From this point it is assumed that no further instructions are needed to get the downloaded report into its desired location on the user's system. Just click the "Save" button and follow the usual steps to complete the download operation.

Note: This function requires the availability of third party PCL to PDF conversion software optionally supplied through Dynamic but licensed directly from the vendor.

Email Archived Report in PDF format

To the right of the PDF icon is an envelope icon for emailing a copy of the report in PDF format. Clicking it brings up the email window to fill out destination, subject, and comments.



Dynamic Paper - Microsoft Internet Explorer

DynamicPaper Mail Document as PDF

Send to :
(Space separate multiple addresses)

Subject :

Message to be sent with attached report :
(When inserting text, a blank line between paragraphs is recommended to avoid automatic wrapping)

Click here to send

If the subject line is left blank it will be filled in with the report description and the date and time it was archived. For example:651377 DGI050 - Designer Genes Inc. 02/06/04 02:39:37PM

The mailed “From” email address is the address in the user’s dxuser dataset record and cannot be blank.

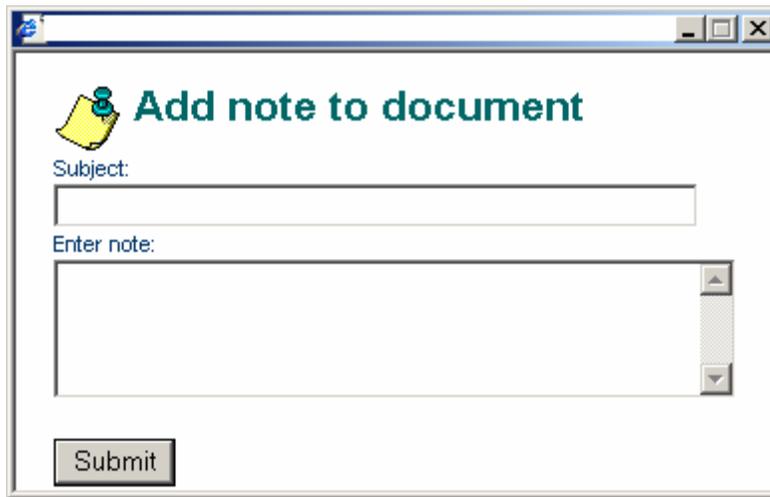
After the email has been sent, a “Sent” indication is presented in the window with a “Close” button to continue.

Download Archived Report in Text format

The rightmost icon in the options column is a document in a folder overlaid with a green arrow to the right. This icon is for requesting the report content to be downloaded in raw text format without an overlaid report form. See the PDF downloading instructions above for functional details.

Sticky Notes

On both archived reports list pages, the rightmost column is “Notes” with “Add” buttons. On the page for viewing an archived report near the center of the bottom of the page is an “Add Note” button next to a Sticky Note icon. Clicking any of these “Add” buttons will pop up a window for adding a sticky note to the selected report. The pop-up Sticky Note entry window is shown below.



After entering the Subject field and note content click the “Submit” and the note will be added (linked) to the selected report. The confirmation that the note was added is posted in the entry window which can then be closed by clicking the “X” close button in the top right corner of the window.

After a sticky note had been added to a archive record, a Sticky Note icon appears in the “Notes” column next to the “Add” button as seen in the example display below.

DynamicPaper Document Archive Search

[▶ New Search](#)

Search results								
Options	Cabinet	Drawer	Folder	Date	Description	User	Notes	
	BI	BI001		02/06/04 02:36:43PM	994374 MBOOK - Mondragon Bookstore	admin	Add	
	BI	BI001		02/06/04 02:39:37PM	651377 DGI050 - Designer Genes Inc.	admin	Add	

[▶ New Search](#)

Clicking on the Sticky Note icon will display in a pop-up window all sticky notes which have been added to the selected archived report as shown below.



Notes

Print Close

Date	User	Note
04/07/04 05:05:19PM	admin	This is a sample Sticky Note The content of the note can be anything including, for example, special instructions, paid date, seen by initials, approvals, tracking #, etc.
04/07/04 05:38:38PM	admin	A second sticky note for the same report This could be a notation about sending a copy of this archived report to the customer, bank, or a fulfillment center, for example.

Once a sticky note is added to an archived report, it is there as long as the report remains in the archive. There is no way to delete it. This way, it mimics the writing of notes directly on a filed paper copy of a report except that copies of the report still are produced without the notes. If necessary, a copy of the notes can be sent by printing this display screen on paper or to a file to be attached to the transmitted report.

The content of selected notes can be sent along with a emailed report simply by displaying the notes for a report before initiating the email. Then, when both the sticky notes display and the email input windows are displayed together, selected text from the notes display can be dragged and dropped into the message box of the email input window. This is just a sampling of the many kinds of new capabilities to be found in moving to a paperless office with DPArchives.

The DPArchive API

DPADD.DL4

In addition to the provision for adding reports to the archive through the DynamicPaper printer driver, there is support for directly adding reports to the archive from any user's application via the *dpadd* subroutine. DPADD is used only if the file to archive already exists.

Setup:

```
External Lib "dpadd.dl4"
Declare External Sub DPADD
```

Syntax:

```
CALL DPADD(E$,Z3$,...)
```

Parameters:

```
:
E$= irrelevant - passed in to return status
Z3$= DynamicPaper string of parameters, same syntax as OPEN parameters
    a= file id= reportid d= "desc" s= subtype
    r1= "ref1" r2= "ref2" r3= "ref3" r4= "ref4"
Z4$= Optional additional string of parameters,
    t= override TYPE code, default to 'P' (printed)
    d= override data and time YYMMDDHHMMSS, default to current.
    u= override USERID, up to 8 char., default to UNIX LOGIN
```

Expects:

```
"dpsystem.txt" configuration file in local directory
    contains dataset type, path and filename parameters
ARCHDIR environment variable set to absolute root path to archive directory
```

Returns:

```
E$ is returned status
0=success
1=failure - unexpected error logged
2=failure - duplicate key (not logged)
3=error message failure - logging of error message failed
```

CONDITIONS.LIB

The archive administrator can specify that access to a report is restricted by conditions defined by custom code which can be added to the **conditions.lib** dl4 module supplied with the DPArchives option. As delivered, “**conditionslib.txt**” and its loaded and saved version contains one switching routine “oncondition” and nine stubs for use in adding code to test the custom conditions for a report. The following is the partial listing of “oncondition” which is called when conditions have been specified in the Report record..

```

! Name:
!   oncondition()
!
! Synopsis:
!   oncondition(conditions$, archive. As Archive)
!
! Description:
!   Called to determine access privileges for user to specific report
!
! Return value:
!   !Zero if access denied, non-zero permitted
!
External Function oncondition(conditions$, archive. As Archive)
Dim passes,pcol, selector$[12], parameters[100]
Try
pcol = pos(conditions$,=":")
if pcol > 1
  selector$ = UCase$(conditions$[1,pcol-1])
  parameters$ = conditions$[pcol+1]
  Select Case Selector$
    Case "C1"
      return(Condition1(parameters$, archive.))

```

This switching function can be extended to include as many conditions as might be required by any specific DPArchive application. This “oncondition” function is set up to handle conditions named “C1” through “C9” which can be specified in the Report record “conditions” field. The report.conditions field entry is in the following format.

<condition name>:<condition expression and/or parameters>

Examples:

```

C1:Ref1$=123*
C2:UserID$=admin and Ref2$>54

```

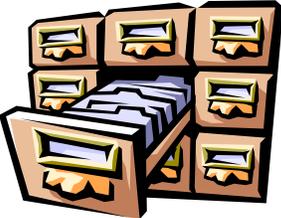
Since the code for processing these conditions can be anything desired or required the format of the conditional expression or parameter is defined only by that code written to fill out the stubs. In

addition to passing the condition parameter string, DPView passes the archive record currently being processed to test if it is viewable by the current user.

The following is the code for one of the stubs.

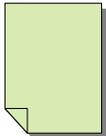
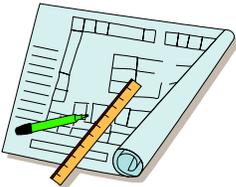
```
!  
! Name:  
!     condition1() - Access filter for ICN applications  
!  
!Synopsis:  
!     condition1(Parameters$, archive. As Archive)  
!  
!Description:  
!     Called to determine access privileges for user to specific report  
!  
!Return value:  
!     Zero if access denied, non-zero permitted  
!  
External Function condition1(Parameters$, archive. As Archive)  
Dim passes  
Try  
  if Parameters$  
    passes = 1 ! replace with parsing and evaluation code  
  else  
    passes = 1  
  end if  
else  
end try  
end function passes ! condition1
```

DynamicPaper workflow process



Archives contain new documents to be processed

Workflow Definitions are unique flows defined for each 'Folder'. Each work phase is defined with multiple dispositions. Customized code stubs can be executed during the flow.



New documents get routed to Phase 1

Management Tools monitor workload & progress



In each phase a document can be routed to a workgroup, a user, users, or automated process for processing



Automatic Alerts on delayed processes



During review there can be custom prompts and entries based on document and/or disposition. Notes can also be entered.

Audit Trail of entire process



Documents are routed to the next phase based on disposition.

Appendix A – Error Messages and Logging

From DynamicPaper Driver

Since DynamicPaper is a driver, the error output must be sent to a log file. In this case, the following error messages are concatenated to “DPerrors” file in the caller’s active directory.

```
DynamicPaper aborted: Demo Mode failure mode 2! Call Customer Support.
DynamicPaper failed to open form: <form file path-name>
DynamicPaper: Error reading form file: <form file path-name>
DynamicPaper: Error closing <form file path-name>
DynamicPaper aborted: Write failure.
DynamicPaper aborted: Embedded filename too long.
DynamicPaper aborted: Failure parsing embedded formfile name!
DynamicPaper aborted: Error found in CopyForm.
DynamicPaper aborted: Error reading application output.
Licensing error <variable msg>
```

When the driver is requested to output a PDF format file, JetPCL (a third party software component, see Appendix C) is called and if it generates errors they are logged to jeterror in the caller’s active directory.

If the formmerg module is executed from the command line instead of via the driver “dynpaper” it presents the following message to the user and then aborts.

```
formmerg DynamicPaper driver module.
<current revision information>
All Rights Reserved. Copyright (C) 1994-2003 by:
Dynamic Concepts, Inc. Aliso Viejo, California USA
formmerg is a DynamicPaper driver component only
executed from a dl4/UniBasic driver script file.
See DynamicPaper User Guide for API information!
There is no other help available!
```

If the foremail module is executed from the command line instead of via the driver “dynpaper” it presents the following message to the user and then aborts.

```
foremail DynamicPaper driver module.
<current revision information>
All Rights Reserved. Copyright (C) 1994-2003 by:
Dynamic Concepts, Inc. Aliso Viejo, California USA
foremail is a DynamicPaper driver component only
executed from the formmerg component.
See DynamicPaper User Guide for information!
There is no other help available!
```

From DPArchive under dynamicXport

Error messages from DPArchive support under dynamicXport are display on message pages which are typically blank except for the message. After noting the message the prior page is most often returned to via the browser's back button

dpadmin received bad count
You must enter a new Report ID!
Invalid report ID: <report id>
Report record no longer accessible.
Requested ReportID <report id> is already in use!
Archive record not found!
Archive record no longer accessible.
Sorry, you cannot email! Your email from address is not in your User record.

Other error messages can appear when unanticipated IO errors or code failures occur. The are in the general form:

{dl4 error number} <dl4 error message>” at “<statement number>” in “ <code identifier>

When this kind of error appears, it is important to notify Dynamic technical support even if a work around is found

From DPADD subroutine

Failure accessing dpsystem.txt file.

Appendix B – Directory Lists

Driver Files List

/usr/lib/dl4/printer/dynpaper	DynamicPaper driver script file
/usr/lib/dynpaper/dynpaper	Copy of DynamicPaper driver script file
/usr/lib/dynpaper/formmerg	DynamicPaper compiled driver component
/usr/lib/dynpaper/formmerg.dat	PCL control data for formmerg
/usr/lib/dynpaper/foremail	DynamicPaper component for pdf and email support
/usr/lib/dynpaper/JetPcl5	Third party PCL to PDF converter (demo)
/usr/lib/dynpaper/license.txt	License terms and conditions
/usr/lib/dynpaper/makeforms	DynamicPaper .prn to .dpf (PCL to forms file) tool
/usr/lib/dynpaper/readme.txt	This file
/usr/lib/dynpaper/PCLFilter	PCL conversion table for printing
/usr/lib/dynpaper/dpmail	email support: callable subprogram & GUI application
/usr/lib/dynpaper/dpmail.bas	Text copy of dl4 test program
/usr/lib/dynpaper/dgiinv.dl4	Sample dl4 test program (example from User Guide)
/usr/lib/dynpaper/dgiinv.bas	Text copy of dl4 test program
/usr/lib/dynpaper/dgiinv.ub	Sample UniBasic test program (from User Guide)
/usr/lib/dynpaper/dgiinv.ubs	Text copy of UniBasic test program
/usr/lib/dynpaper/DGInvoice.dpf	Form file for sample test program
/usr/lib/dynpaper/DGInvoice.prn	Original prn file used to make test form file
/usr/lib/dynpaper/sub.ptr	New subroutine replaces sub.ptropen
/usr/lib/dynpaper/sub.ptr.ubs	Text copy of sub.ptr
/usr/lib/dynpaper/sub.ptrc	New subroutine replaces sub.ptrclose
/usr/lib/dynpaper/sub.ptrc.ubs	Text copy of sub.ptrc
/usr/lib/dynpaper/ubloadsavesh	UniBasic utility loadsave script

Archives Files List

Filename	Purpose
dx/dp/conditions.lib	User customizable access restriction tests
dx/dp/dpadd.dl4	Callable routine for applications to archive documents
dx/dp/dpadmin.dl4	dxport application supporting maintenance options
dx/dp/dparchive.lib	support library for all dparchive dl4 modules
dx/dp/dpcreate.dl4	stand-alone to create, backup, initialize database
dx/dp/dpview.dl4	dxport application supporting view archives options
dx/dp/dpsystem.txt	Control parameter file

HTML Files List

Standard HTML files delivered with DynamicPaper Archives

Filename	Purpose (page to)
dx/dp/dparchivemnt.html	enter, view, edit, save or delete an archive record
dx/dp/dpclose.html	finish operations by closing the popped up window
dx/dp/dphelp.html	display help content
dx/dp/dpmail.html	input email parameters and send report
dx/dp/dpnote.html	enter and submit a new sticky note
dx/dp/dpnotepr.html	display all sticky notes attached to a report
dx/dp/dpreportmnt.html	enter, view, edit, save or delete a report record
dx/dp/dpreportpr.html	display all reports to select on for editing or add a new one
dx/dp/dpuserpr.html	display users to select one to manage access or subscriptions
dx/dp/dpuserrpmnt.html	manipulate a user's report access or subscriptions
dx/dp/dpview.html	view a archived report as a plain text
dx/dp/dpviewlist.html	displays results of search for archived reports as list
dx/dp/dpviewpr.html	inputs archive search criteria and initiates search, or subscribes

Stylesheet Files List

dx/styles/dpstd.css
dx/styles/dxstd.css
dx/styles/icnstd.css

Image Files List

Inactive graphics:

Filename	Purpose
dx/dp/images/dpbinoc.gif	Function cue
dx/dp/images/dpcabinet.gif	Function cue
dx/dp/images/dpcal.gif	Function cue
dx/dp/images/dpdrawer.gif	Function cue
dx/dp/images/dpfview.gif	Function cue

Active graphics:

Filename	ALT	Purpose
dx/admin/images/dxcal.gif	Calendar	popup date selector
dx/dp/images/dpdownload.gif	Download as text	down load selected file
dx/dp/images/dphelp.gif	Display Help	popup help window
dx/dp/images/dpmagglass.gif	View as text	display selected archive
dx/dp/images/dpmail.gif	Email as PDF	convert to and email pdf copy
dx/dp/images/dpnewsearch.gif	Start a new search	links to reselect search criteria
dx/dp/images/dpNext.gif		display next search results page
dx/dp/images/dpnote.gif		popup note display window
dx/dp/images/dppdf.gif	Download as PDF	convert to pdf and download copy
dx/dp/images/dpprev.gif		display previous search results page
dx/dp/images/dpsearchb.gif	Display Search Results	initiate selected search

Appendix C – JetPCL®

JetPCL® version 9.0 Integrated into DynamicPaper

JetPCL® is a third party software tool supplied by Tech Know Systems Inc. The demo version of this product supplied with DynamicPaper prints a watermark notice overlaid on the PDF document. Obtain an end user license for JetPCL to install the production version and create PDF documents without the watermark for printing, storing, or emailing.

Purchasing a ‘JetPCL® for DynamicPaper’ Runtime License

JetPCL® for DynamicPaper runtime licenses are available directly from Tech Know Systems. Pricing is based on the number of DynamicPaper servers and concurrent users (based on the DynamicPaper license that will be using JetPCL®). Pricing is as follows:

- \$1000 per server machine,
- plus \$100 per concurrent user.

Visit our web site at www.techknowsystems.com and click on the “JetPCL for DynamicPaper” link on the JetPCL® Products page for more information.

Installing a ‘JetPCL® for DynamicPaper’ Runtime License

Once a **JetPCL® for DynamicPaper** runtime licenses is purchased installation can easily be accomplished by either an in-house IT person familiar with the Unix command prompt or by a qualified Dynamic Concepts distribution consultant. Installation involves these steps:

- receiving an activation key from Tech Know Systems via email,
- receiving a new JetPCL® replacement executable via email,

Easy to follow, step by step, instructions will explain how to update the JetPCL® executable and install the JetPCL® activation key to enable the integrated JetPCL® module.

Contacting Tech Know Systems, Inc.

Further information can be obtained directly from the creators and marketers of JetPCL®.

Tech Know Systems, Inc.

P.O. Box 548
Whitefish, MT 59937

406-862-1435
650-745-1039 fax

<mailto:sales@techknowsystems.com>
<http://www.techknowsystems.com/products.shtml>

This information is included in the pdf document, “JETPCLDP.pdf” supplied with DynamicPaper.