User Manual

6/13/2012

Variable Data Printing

Table of Contents

Hardware Requirements 3
VDP Designer Overview 4
Installation 4
Reform Licensing and Registration 6
  License Manager 8
Creating your First Form using a template 10
Working with Your Data 24
  Design Layer 24
    Common Icons and Controls 25
  Text Layer 28
    Mapping Area 29
    Design Tips & Techniques 30
    VDP Designer Keyboard Shortcuts 32
    Edit Menu 35
    Options Menu 35
    Special Fields Menu 37
    Form Key Field 38
    Wizards Menu 41
  Special Functions 43
  Designing Objects 48
    Paragraph Object 52
    Rich Text Object 53
    Shape Object 56
    Barcode Object 56
    Image Object 57
Output Devices 62
What is the Workflow Designer? 63
  Benefits of using the Workflow Designer 65
  Multiple Destinations Tutorial – Printing and Archiving Simultaneously 65
Advanced Features in Reform 89
  Decision Maker 89
  Scripting 95
    Adding Your Own Send-to-Device 96
    Commands 97
      Functions 109
    Running Scripts as Services 127
Inbound Connectivity 128
    Reform Printer Installation 129
    Reform Printer Driver 135
    Generic Text File 139
    Serial, Parallel, and TCP/IP Ports 139
    TCP/IP (Port 9100) 142
    AS/400, System 36, System 38 143
    LPD Windows 2000/XP/2003 148
    Unix and Linux Operating Systems 150
      AIX LPR Setup instructions 152
Maintenance 157
Troubleshooting 157
Reform Issue Ticket 164

System Requirements

- Reform 14 Enterprise or Reform 14 PDC
- For Evaluation and Demonstration: Windows XP, Windows 2003 Server (32Bit), Windows Vista, Windows 7 or Windows 2008 Server (32Bit and 64 Bit Operating Systems are Supported)
- For Production Systems: Windows 2003 Server (32Bit) or Windows 2008 Server (32Bit and 64 Bit Operating Systems are Supported)

All required Reform software downloads can be found at www.fabsoft.com >> Downloads section.

Hardware Requirements

- Production System Recommendation (These suggestions are only estimations. High load environments may require multiple servers, or additional hardware.)
    1. Pentium 4 Processor
    2. 4 GB RAM
    3. HDD Space 100 GB (Redundancy recommended)
    4. Battery Backup (UPS)
    5. Dedicated server, no other software products except for virus and security programs.
- Minimum System Requirements for demonstration purposes only
    1. Pentium 4 Processor
    2. 2 GB RAM
    3. HDD Space 30 GB

Hardware specifications depend greatly on the workload and throughput that is required. Higher end processors and greater amounts of memory will increase the amount of information and data that Reform can process thus improving overall system throughput. For load balancing or failover, multiple servers can also be used. (Note: A license for Reform and each Plug-in is required for each server running the software.) Before obtaining the server that you plan on utilizing for Reform, please check the Plug-ins' user manuals for System Requirements as well as they may require additional hardware. For production systems, we also recommend redundant hard drives utilizing the mirroring features of RAID (Redundant Array of Independent Disks) to minimize data loss and expedite system recovery in the event of a hardware failure. Battery backups are also suggested to maintain system uptime and reliability in case of power failures.

VDP Designer Overview

VDP Designer is a program that allows users to enhance forms generated by your application. Because of its simple "what-you-see-is-what-you-get" approach, the VDP Designer is easy to use. The objects that you add to your form on the Design layer reflect exactly what your form will look like if it were printed out. In order for Reform to generate the printout that you need, you must create a template for each of your forms. The template serves as a new layout for each of the forms generated by your application. It is similar to using a pre-printed form, where the text is overlaid on the form.

If you need to remotely design your forms, use VNC, PCAnywhere, www.LogMeIn.com, or other remote software that will let you view the server at a console level (as if you were standing in front of the server).

Installation

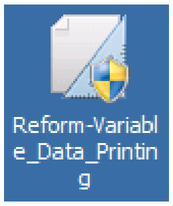
All required Reform software downloads can be found at www.fabsoft.com in the Downloads section.

The default installation directory for Reform and its Plug-ins is

C:\Program Files\Reform…\Variable_Data_Printing

If you have changed the path during the Reform installation routine, or if you are using a 64 bit operating system, please refer to the appropriate path instead of the default one referenced throughout this documentation.

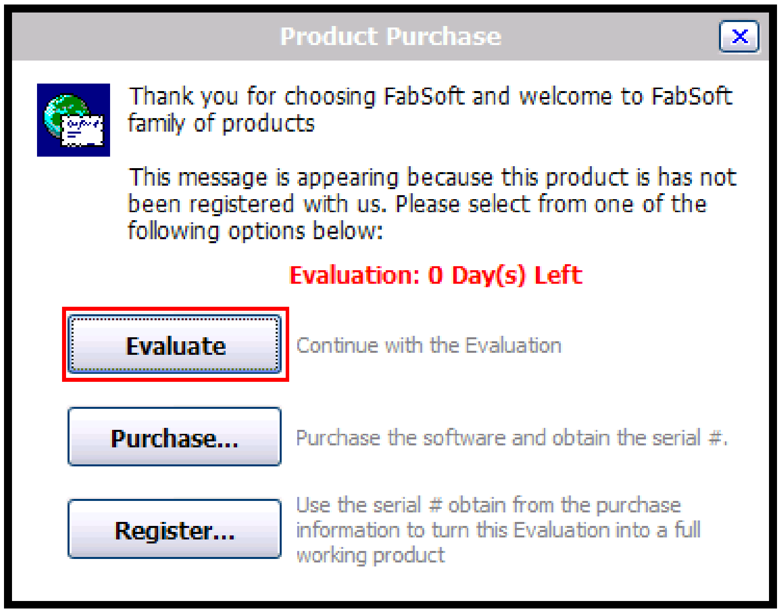
To install the Reform_VDP Designer, download and run the installer.



Step through the prompts, accept the License Agreement, and the installation will begin. Allow the installation to finish and then continue onto the next section.

The default installation directory for Reform and its Plug-ins is **C:\Program Files\Reform…\**. If you have changed the path during the installation routine, or if you are using a 64 bit operating system, please refer to the appropriate path instead of the default one referenced throughout this documentation.

Next, you must register Reform. We are going to be obtaining a 60-day evaluation key. If you don't see the registration window, run the Spooler by going to **Start > Programs > Reform... > Reform Spooler**. It will ask you to register your copy of Reform. Click the **Evaluate** button.


Product Purchase

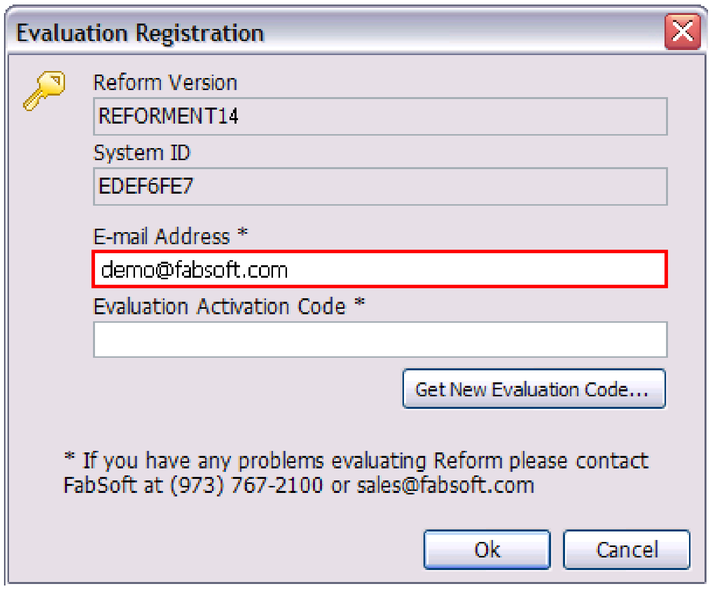
Thank you for choosing FabSoft and welcome to FabSoft family of products

This message is appearing because this product is has not been registered with us. Please select from one of the following options below:

Evaluation: 0 Day(s) Left

Evaluate
Continue with the Evaluation
Purchase...
Purchase the software and obtain the serial #.
Register...
Use the serial # obtain from the purchase information to turn this Evaluation into a full working product

Enter your email address, and then click **Get New Evaluation Code**. This will launch an Internet Explorer window in which will let you request a 60-day evaluation key to be emailed to you.


Evaluation Registration

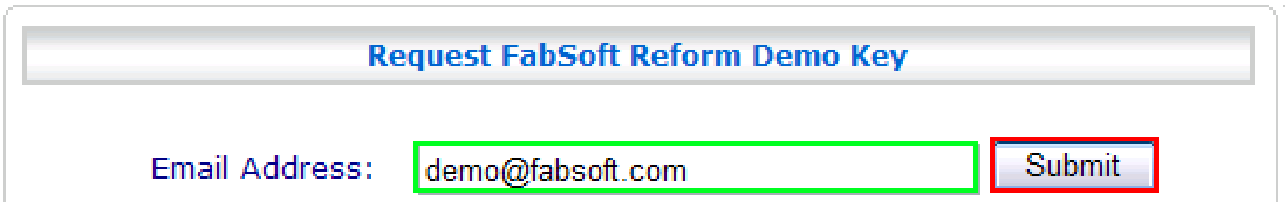
Reform Version: REFORMENT14

System ID: EDEF6FE7

E-mail Address *: demo@fabsoft.com

Evaluation Activation Code *:

* If you have any problems evaluating Reform please contact FabSoft at (973) 767-2100 or sales@fabsoft.com

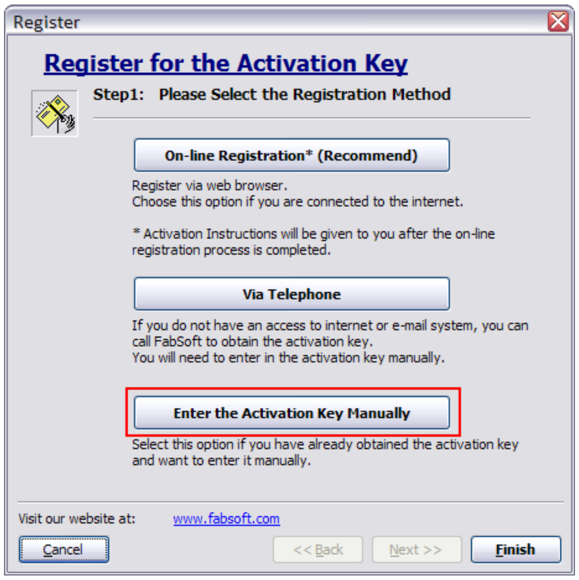


Click the **Submit** button. If it says that an evaluation key has been mailed to you, check your email for a key to arrive. Then, in the **Evaluation Registration** window, paste the key from your email into the **Activation Code** box and click **OK**. You now have an unrestricted 60 days of Reform usage.

Reform Licensing and Registration

To obtain a registration key, call FabSoft at **973-767-2100**. Once you obtain the key, open the Spooler application (if it is not already open), go to **Help | About**, and click the **Register** button.

1. Click **Enter the Activation Key Manually**.


Register
Register for the Activation Key
Step1: Please Select the Registration Method
On-line Registration* (Recommend)
Via Telephone
Enter the Activation Key Manually
Visit our website at:
www.fabsoft.com
Cancel
<< Back
Next >>
Finish

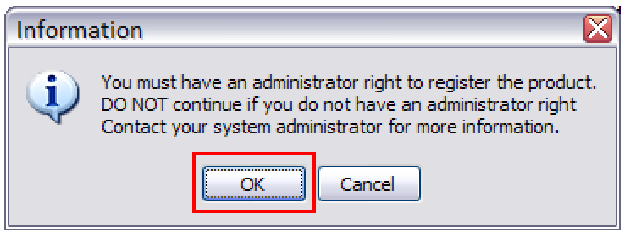
Register via web browser.
Choose this option if you are connected to the internet.

* Activation Instructions will be given to you after the on-line registration process is completed.

If you do not have an access to internet or e-mail system, you can call FabSoft to obtain the activation key.
You will need to enter in the activation key manually.

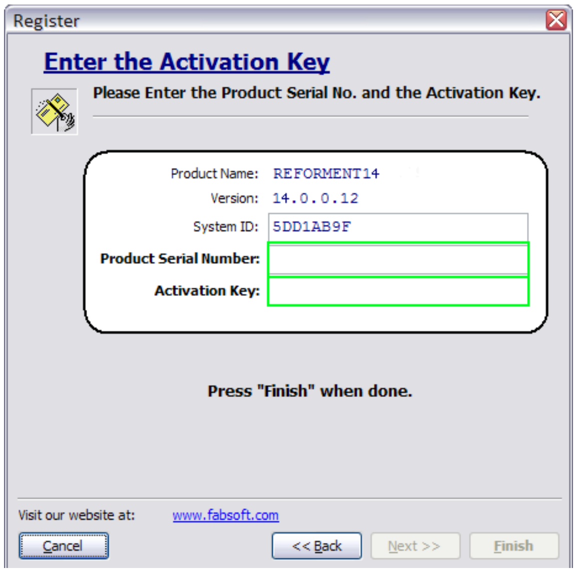
Select this option if you have already obtained the activation key and want to enter it manually.

a. If the following window is displayed, click **OK**.



You must have an administrator right to register the product.
DO NOT continue if you do not have an administrator right
Contact your system administrator for more information.

2. Enter the **Serial Number** and your **Activation Key** and click **Finish** to permanently register your copy of Reform.


Register

Enter the Activation Key

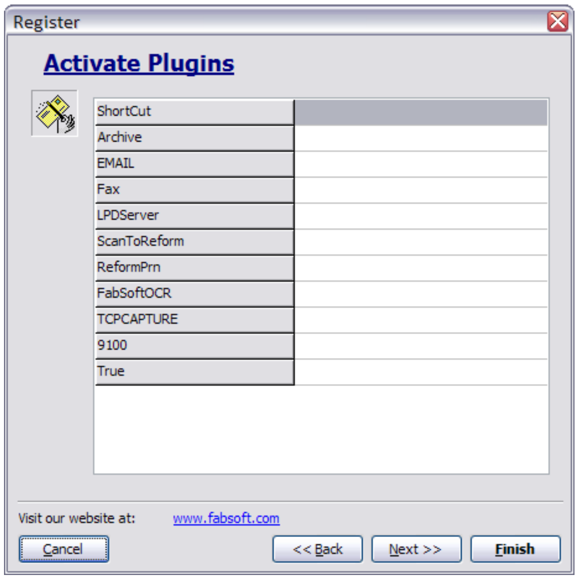
Please Enter the Product Serial No. and the Activation Key.

Product Name:	REFORMENT14
Version:	14.0.0.12
System ID:	5DD1AB9F
Product Serial Number:	
Activation Key:	

Press "Finish" when done.

Visit our website at: www.fabsoft.com

3. You now have a fully licensed copy of Reform.
4. If you need to register additional Reform Plug-ins:
   a. Follow the same registration process to get to the **Register** window, shown in Step 2.
   b. Press **Alt+R**.
   c. A window will be displayed that will allow you to enter serial numbers for one or more Plug-ins. Once you have entered the desired keys, click **Finish** to register the Plug-in(s).


Register

Activate Plugins

ShortCut	
Archive	
EMAIL	
Fax	
LPDServer	
ScanToReform	
ReformPrn	
FabSoftOCR	
TCPCAPTURE	
9100	
True	

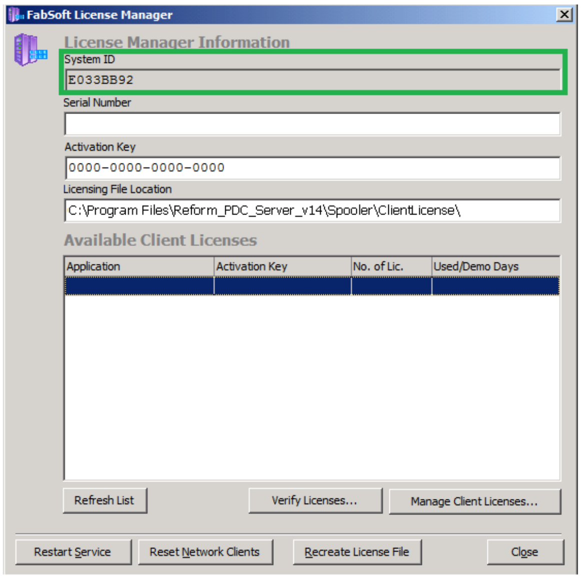
Visit our website at: www.fabsoft.com

5. Once all the activation codes are entered and verified, you will need to restart the Reform Spooler. Please see the Reform Spooler > Reform Spooler Service section of this manual for more information on how to restart.
    a. If any Plug-ins are also installed, they may also need to be restarted. Please see each Plug-in's user manual for more information on how to restart them.

License Manager

If you are using the License Manager, stop all services including the License Manager Service before opening the program and adding the keys. Once all the services have been stopped please go to: to **Start > Programs > Reform...> License Manager**

You will need to provide the **System ID** to FabSoft to generate the activation keys for the software.


FabSoft License Manager

License Manager Information

System ID: E033BB92

Serial Number:

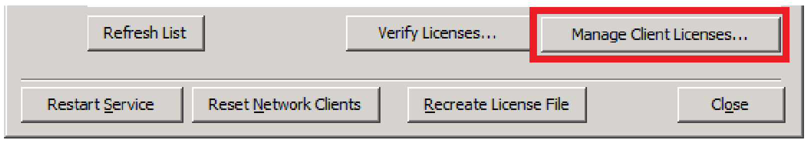
Activation Key: 0000-0000-0000-0000

Licensing File Location: C:\Program Files\Reform_PDC_Server_v14\Spooler\ClientLicense\

Available Client Licenses

Application	Activation Key	No. of Lic.	Used/Demo Days

Once you have the list of Activation Keys for each Plugin you have installed, please click the **"Manage Client Licenses"** button to add/update any desired activation keys.



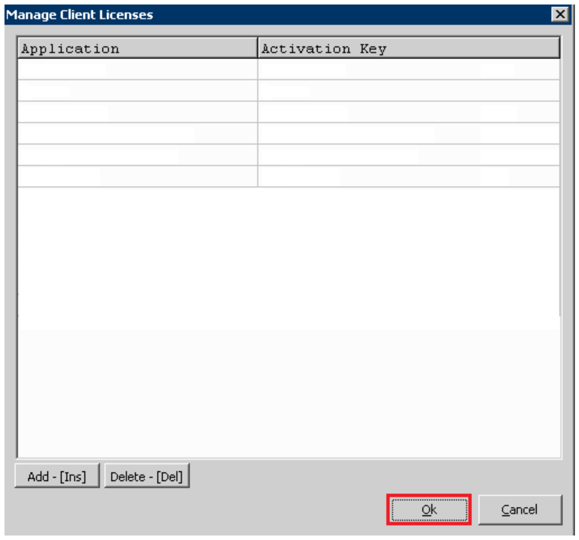
a. If a change or addition is made, you should see a "Last Refresh" message when you close the manage licenses windows.
b. If an update to an existing code was made (For example increasing the number of licensed clients) – Follow these steps as well.

i. Click the "Reset Network Clients" button.

ii. Select the license that was updated.

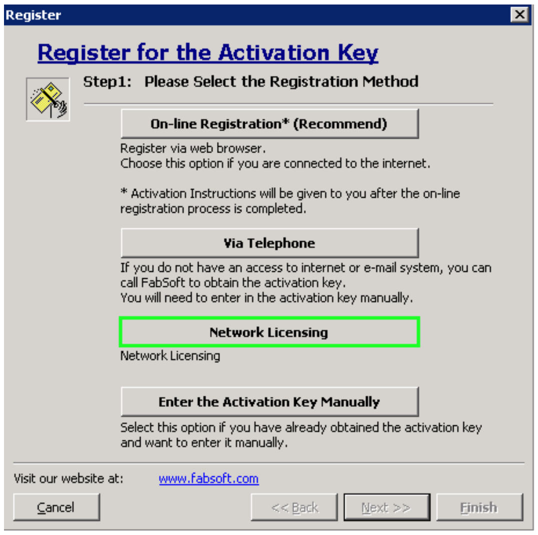
iii. This will cause the license file to regenerate with the new number of clients available.

Please add the correct Application Names in the left side and the activation key on the right side. After you are done click the **OK** button.



Close the License Manager Interface. Then restart the License Manager Service.

The next time you open the VDP Designer, MOST or other Plugin, it will ask you to specify the Network Licensing File. Click in the **Network Licensing** button.


Register
Register for the Activation Key
Step1: Please Select the Registration Method
On-line Registration* (Recommend)
Via Telephone
Network Licensing
Enter the Activation Key Manually
Cancel
<< Back
Next >>
Finish

Register via web browser.
Choose this option if you are connected to the internet.

* Activation Instructions will be given to you after the on-line registration process is completed.

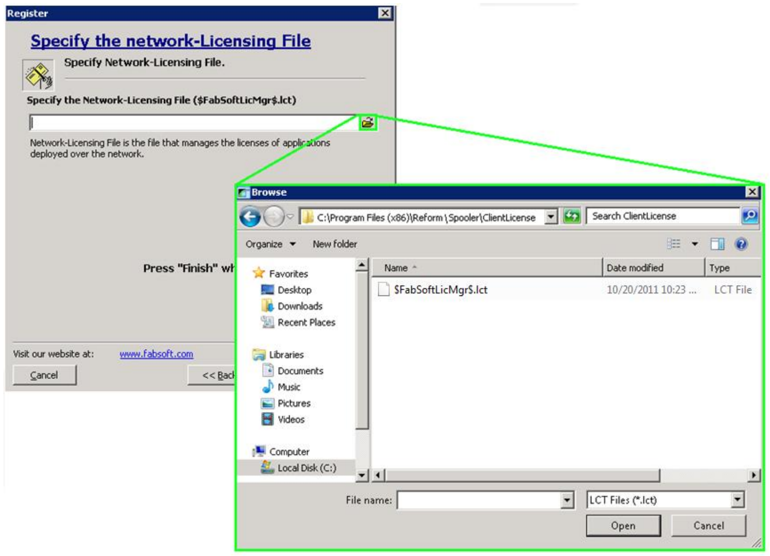
If you do not have an access to internet or e-mail system, you can call FabSoft to obtain the activation key.
You will need to enter in the activation key manually.

Network Licensing

Select this option if you have already obtained the activation key and want to enter it manually.

Visit our website at: www.fabsoft.com

Look for the Network-Licensing file. It will be in **Programs\Reform…\Spooler\ClientLicense\**



Specify the Network-Licensing File ($FabSoftLicMgr$.lct)

Network-Licensing File is the file that manages the licenses of applications deployed over the network.

Visit our website at: www.fabsoft.com

C:\Program Files (x86)\Reform\Spooler\ClientLicense

Search ClientLicense

Name	Date modified	Type
$FabSoftLicMgr$.lct	10/20/2011 10:23 ...	LCT File

LCT Files (*.lct)

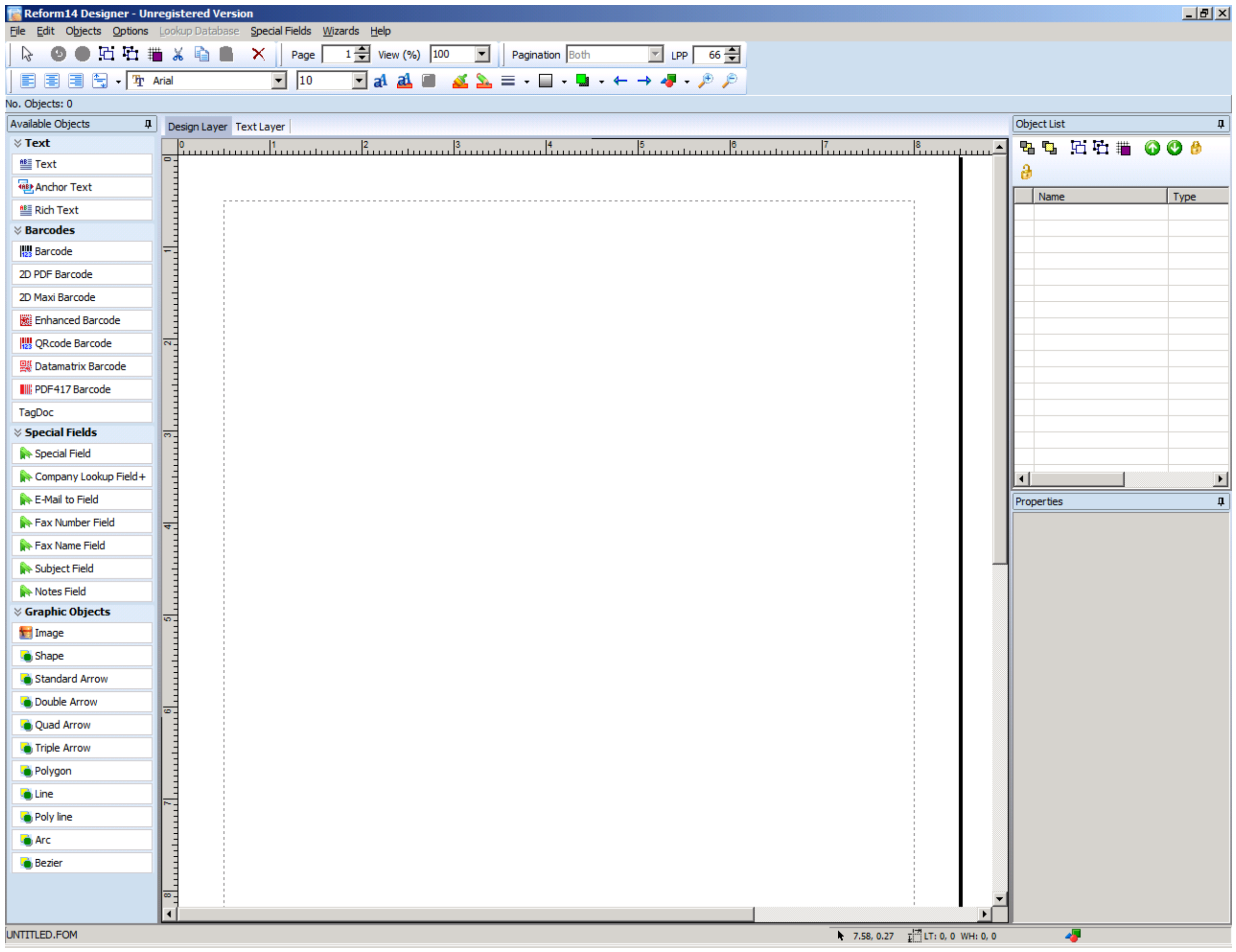
Once you have done that with all the Plug-ins, the next time you open them will not ask for the network license file.

Now, you can restart all the services.

Creating your First Form using a template

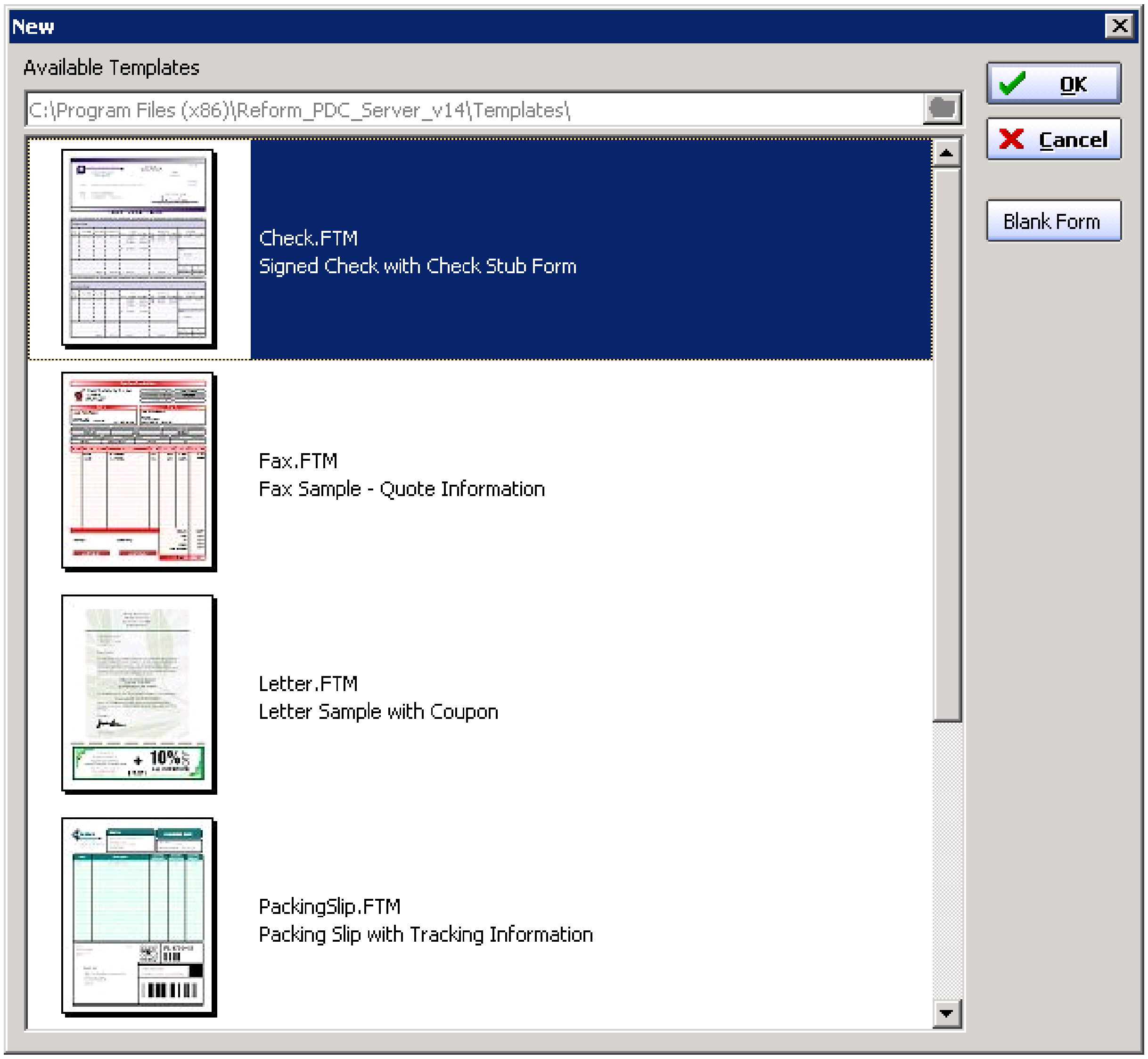
1) If the VDP Designer is not already open, Go to **Start > Programs > Reform… > Variable_Data_Printing > VDP Designer**. This loads the VDP Designer.

   The **Design Layer tab** is where you create your forms. The simple drag-and-drop interface allows you to add and alter data, text, graphics, shapes, bar codes, etc.



The default view of the VDP Designer is a traditional view, which uses 2 tabs that represent the Design Layer and the Text Layer. Click each tab to switch back and forth between the Design Layer and the Text Layer.

2) Click **File > New > New…** to see the **Available Templates**



C:\Program Files (x86)\Reform_PDC_Server_v14\Templates\

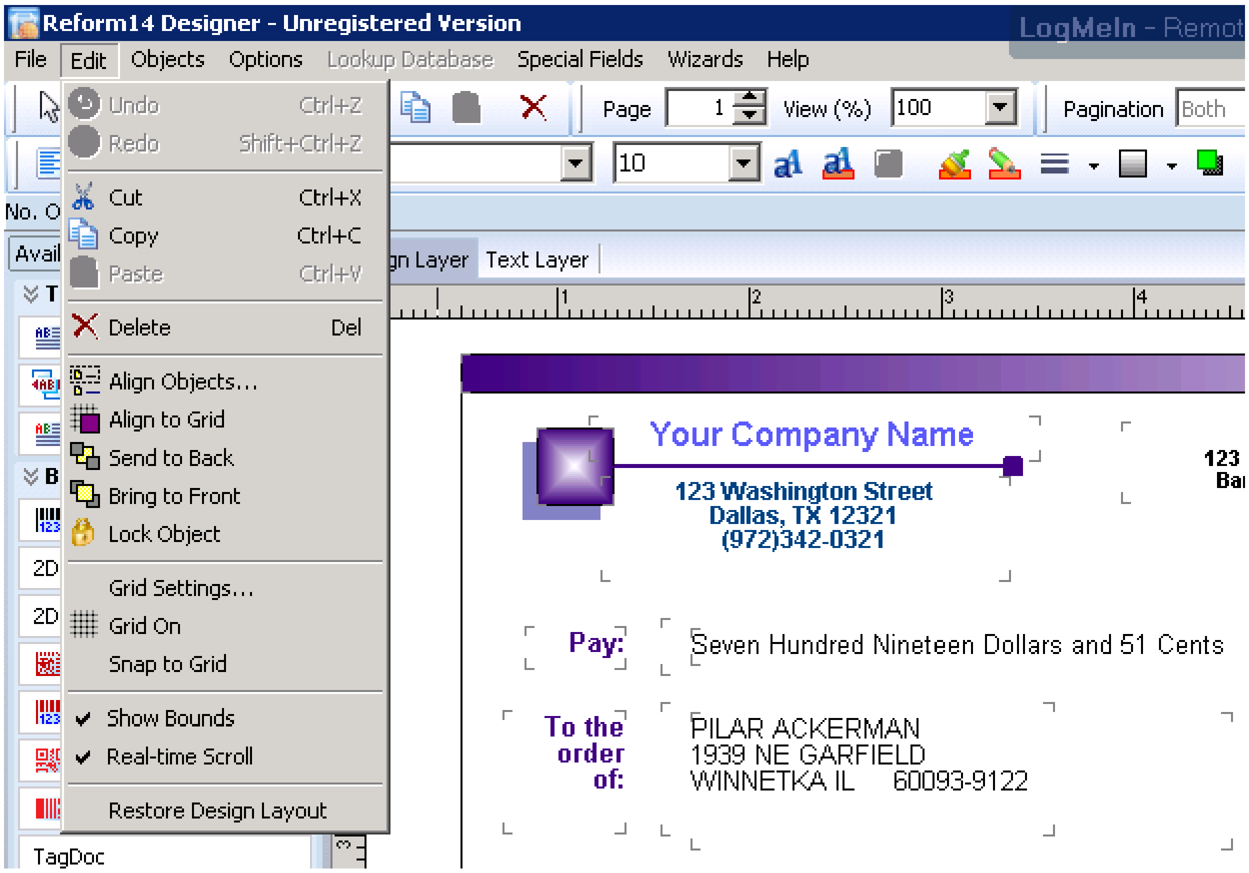
Check.FTM
Signed Check with Check Stub Form

Fax.FTM
Fax Sample - Quote Information

Letter.FTM
Letter Sample with Coupon

PackingSlip.FTM
Packing Slip with Tracking Information

3) Select **Check.FTM** and click the **OK** button.
4) You will be asked if you want to open a text/ASCII file (also known as spool file) for designing. Click **Yes**, now it will ask you to select a file. Go back to the **root directory** and browse to **Reform…\Backups** and select **Check.out** to load the file. The templates come with sample spool files. Outside the tutorial your spool files will come from your Application.
5) In the **Designer Layer**, you will notice that all our fields (**Pay, To the order of, Amount, etc..)** are populated.
6) The **Design Layer** displays both the **Design Layer objects and Text Layer objects**. Go to **Edit>click Show Bounds** to display the border of each field in the form.

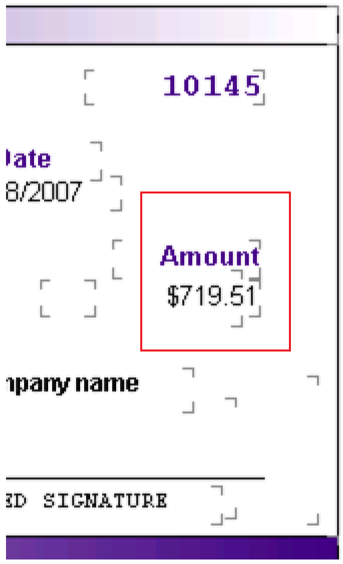


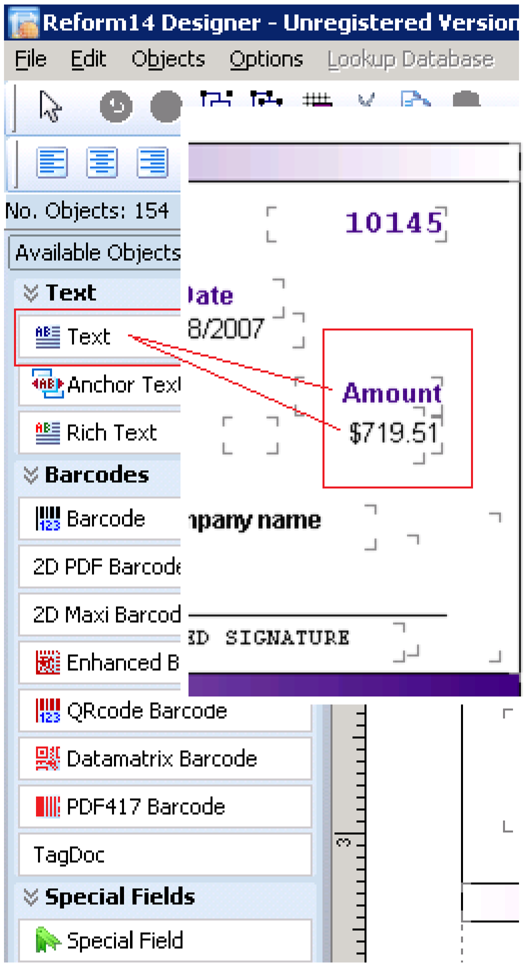
Your Company Name
123 Washington Street
Dallas, TX 12321
(972)342-0321

Pay: Seven Hundred Nineteen Dollars and 51 Cents

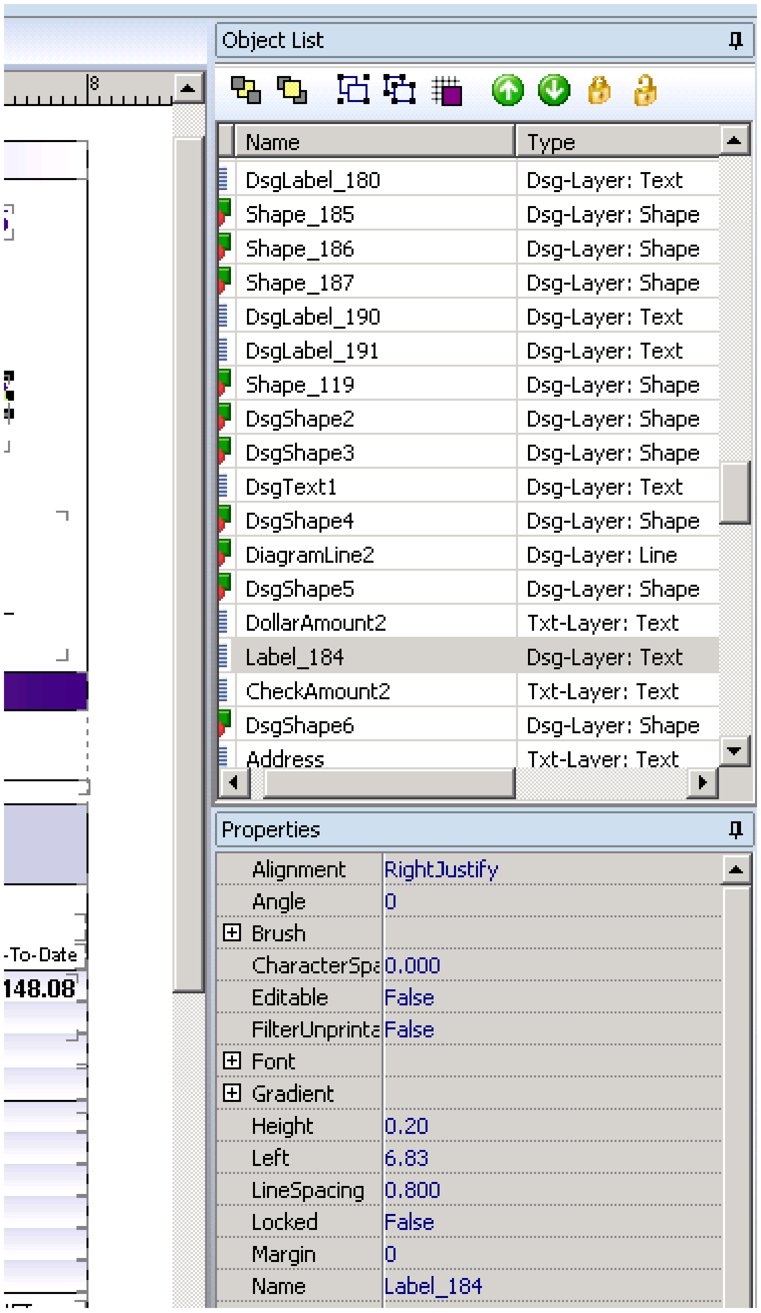
To the order of: PILAR ACKERMAN
1939 NE GARFIELD
WINNETKA IL 60093-9122

7) Let's take a closer look at these two objects, **Amount** and **$719.51.** Both objects are created using the **Text object.**





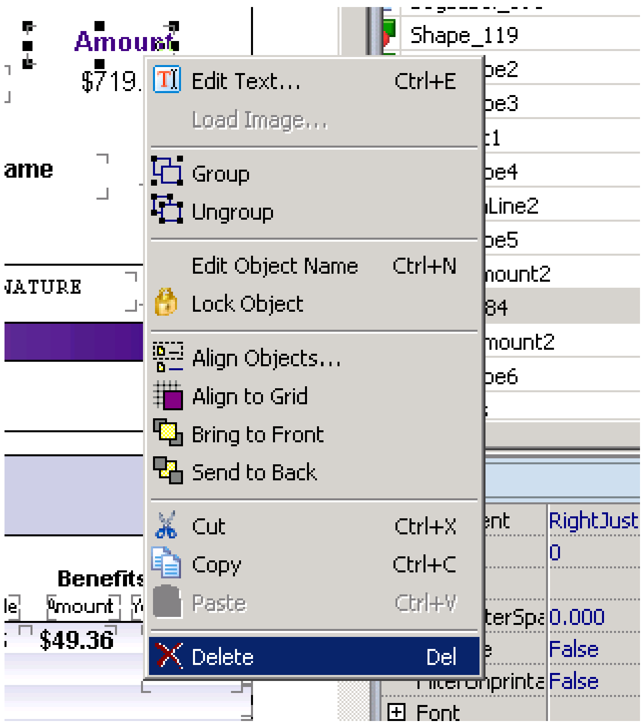
8) Click the object that has the word **Amount**. You will notice on the right side of the VDP Designer, under **Object List**, **Label_184** is the object name and next to it is the object type **Dsg-Layer:Text**. Dsg-Layer-Text tells you that it's a text object in the Design Layer. Every time you create an object the VDP Designer will assign a name to it, but you can always rename it by going to the **Properties Window**.



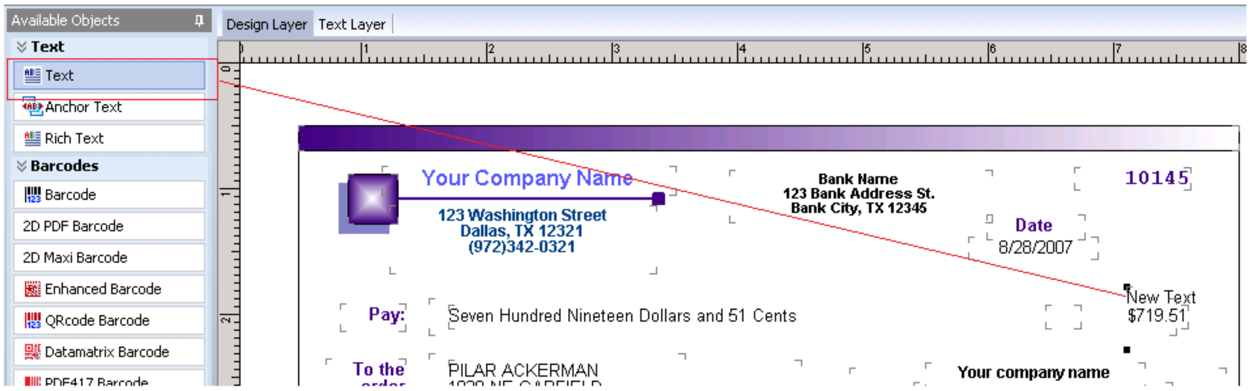
Name	Type
DsgLabel_180	Dsg-Layer: Text
Shape_185	Dsg-Layer: Shape
Shape_186	Dsg-Layer: Shape
Shape_187	Dsg-Layer: Shape
DsgLabel_190	Dsg-Layer: Text
DsgLabel_191	Dsg-Layer: Text
Shape_119	Dsg-Layer: Shape
DsgShape2	Dsg-Layer: Shape
DsgShape3	Dsg-Layer: Shape
DsgText1	Dsg-Layer: Text
DsgShape4	Dsg-Layer: Shape
DiagramLine2	Dsg-Layer: Line
DsgShape5	Dsg-Layer: Shape
DollarAmount2	Txt-Layer: Text
Label_184	Dsg-Layer: Text
CheckAmount2	Txt-Layer: Text
DsgShape6	Dsg-Layer: Shape
Address	Txt-Layer: Text

Alignment	RightJustify
Angle	0
⊞ Brush	
CharacterSpa	0.000
Editable	False
FilterUnprinta	False
⊞ Font	
⊞ Gradient	
Height	0.20
Left	6.83
LineSpacing	0.800
Locked	False
Margin	0
Name	Label_184

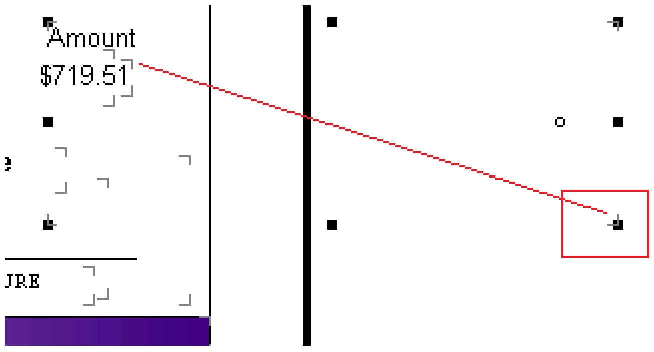
9) Let’s learn how to add a label. First, let’s delete the label Amount, Right Click on the object and Click Delete to remove the object.

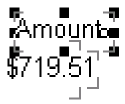


10) Now, go ahead and put back the label for Amount. Go to the Available Objects Window, and Click Text. You will see a White Cross next to your cursor, Right Click where you want to add the Text. A New Text object has been created.

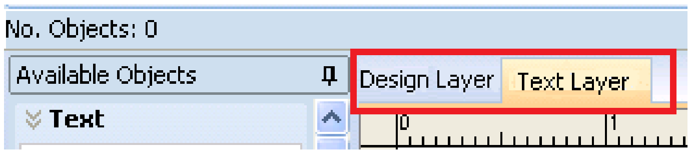


11) Right Click on the Text object, Click Edit, and replace New Text with the word Amount. Let's also resize the Text object so it's just big enough for the word Amount. Right Click on where the red box is drawn, hold it down, and drag it closer to the word Amount.

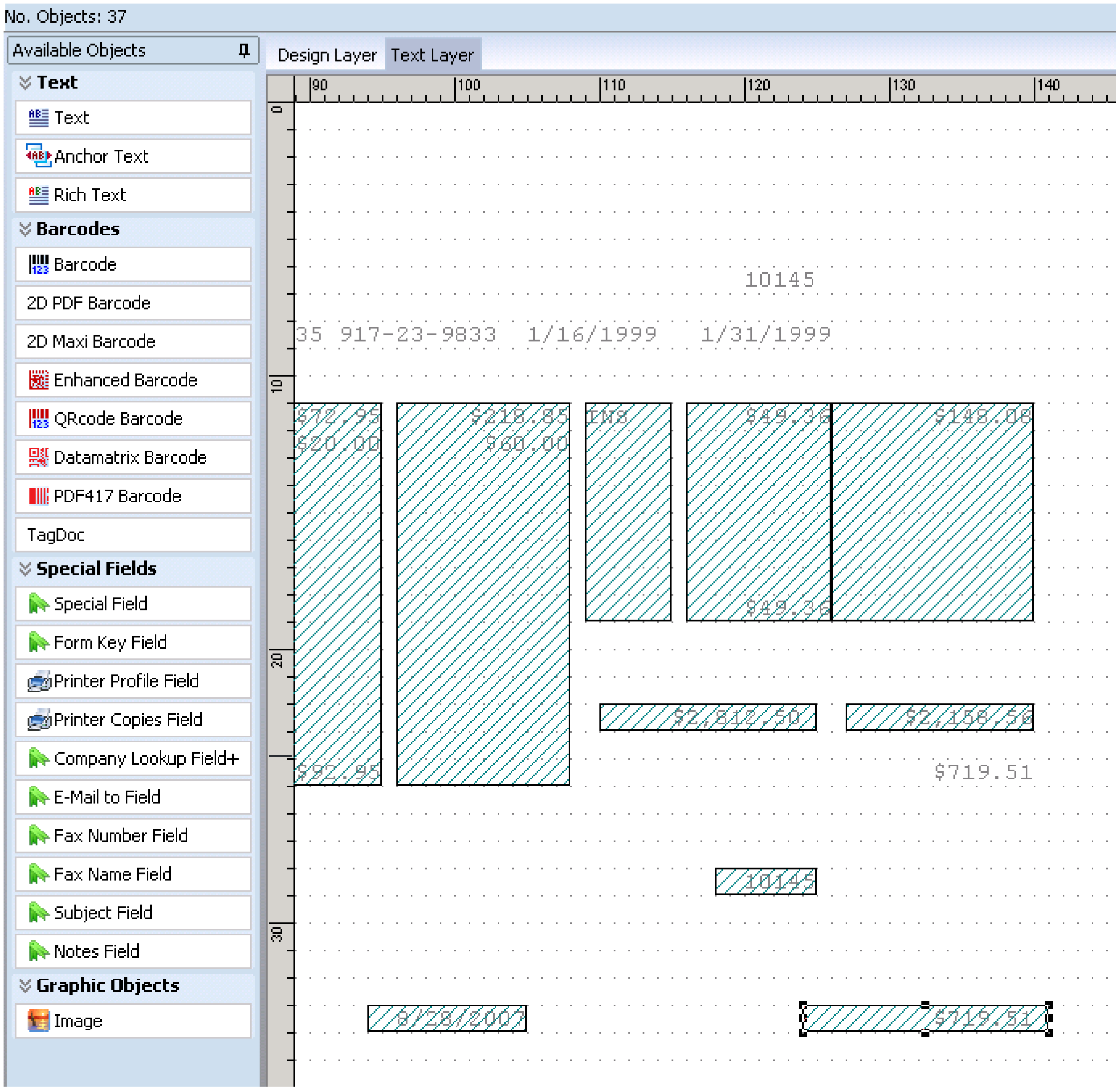




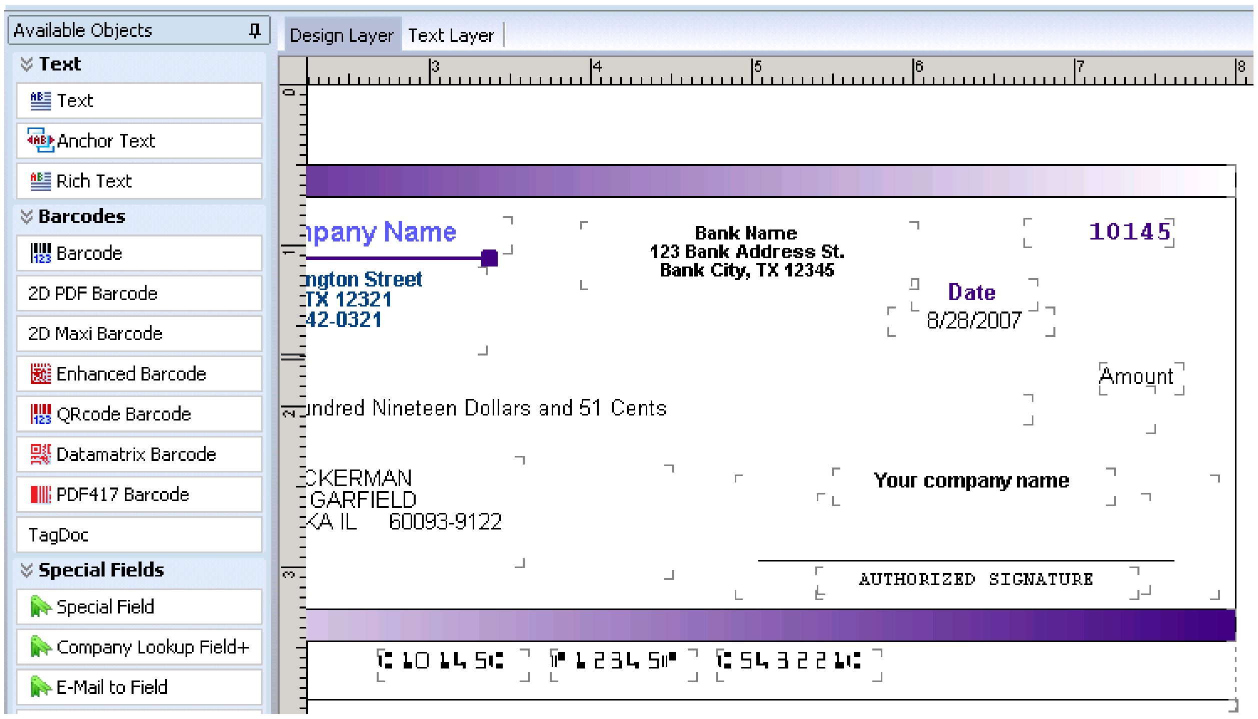
12) So far you've learned how to manipulate the Text object in the Design Layer. Now let's try doing it in the Text Layer. Click on the object that displays **$719.51**. While the dollar amount is selected let's switch to the Text Layer by Clicking on the **Text Layer**.



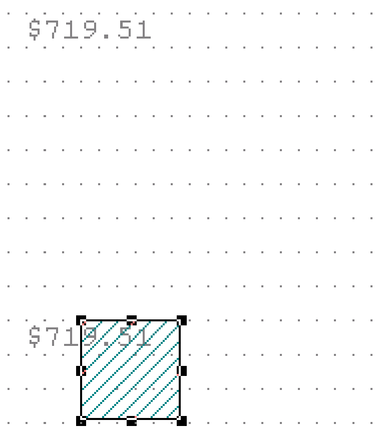
13) You will see that $719.51 has Text object over it, the object name is **CheckAmount2**, and the object Type is **Txt-Layer:Text**. While the object is still highlighted click Delete on your keyboard.



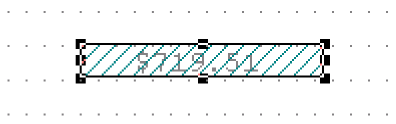
14) Switch back to the **Design Layer**. You will notice the dollar amount $719.51 is no longer displayed.



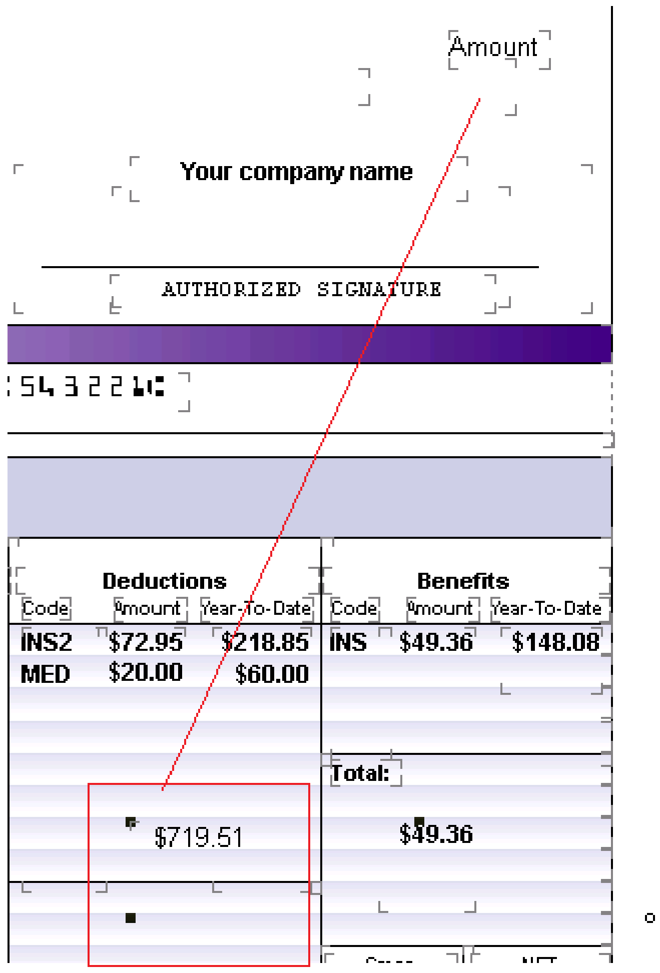
15) Let's put it back by **mapping** the Text object to the dollar amount field in our Text Layer. Switch back to the Text Layer, go to the **Available Objects** click on **Text**, Right Click on $719.51, to create the Text object. You may need to move the object so the data falls inside the text object. Right click on the Text object; hold it until it's mapped correctly. If you want to move the object you need to click inside the box, if you want to resize the object you need to click on the tiny boxes on the border.



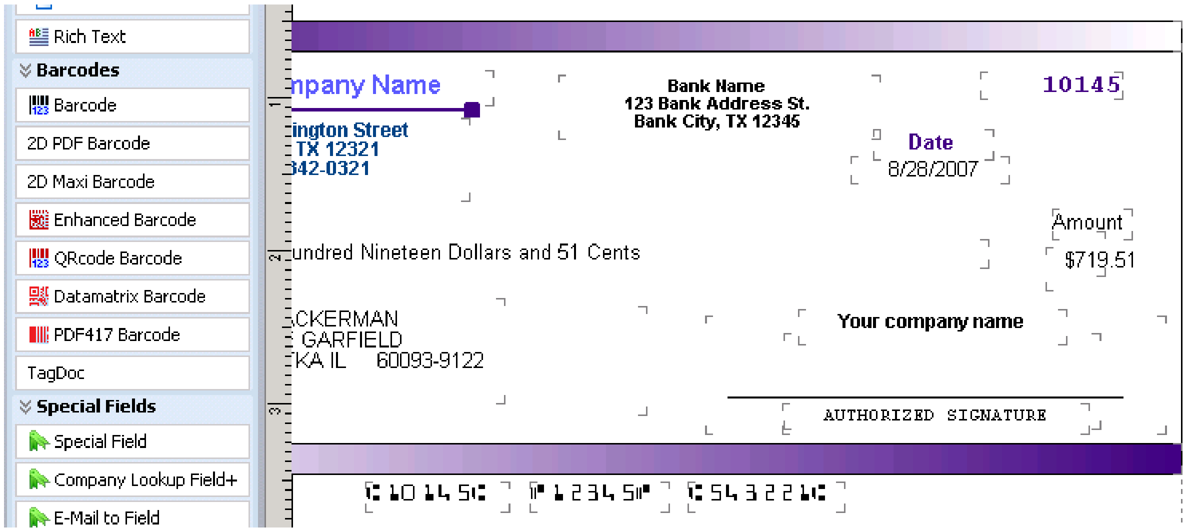
16) You also want to keep in mind, when mapping your data, you have to assume that the information can grow and shrink, like our dollar amount. For that reason our text object is big enough to accommodate larger dollar amount.



17) Switch back to the Design Layer, and look for the object we just created. Since the text layer, we added is somewhere in the middle of the canvas, it will be in the same location when you switch to the Design Layer. If the text layer you created is in the bottom left, expect it to be in the bottom left of your VDP Designer screen. After finding the Text object $719.51 move it under Amount.



18) After moving the object your form should look like the image below.



npany Name
ington Street
TX 12321
342-0321

Bank Name
123 Bank Address St.
Bank City, TX 12345

10145

Date
8/28/2007

Amount
$719.51

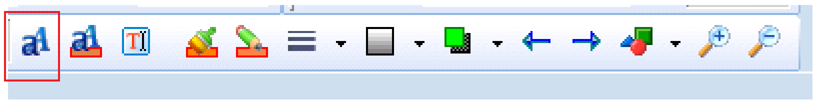
undred Nineteen Dollars and 51 Cents

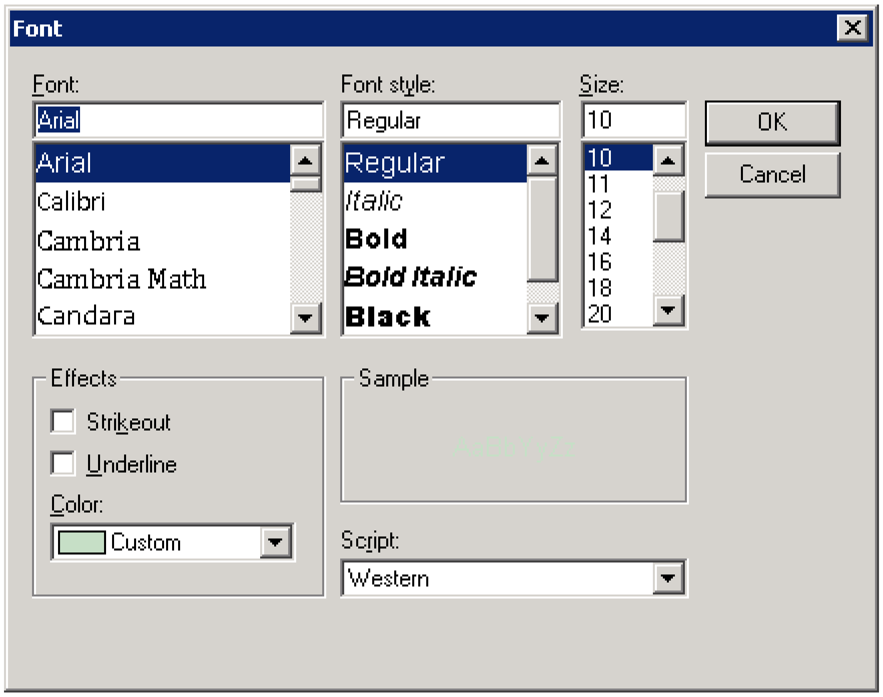
CKERMAN
GARFIELD
KA IL 60093-9122

Your company name

AUTHORIZED SIGNATURE

19) By selecting the object and clicking on the **Text Tool box**, you can **change the property of the text**.





Font

Font: Arial

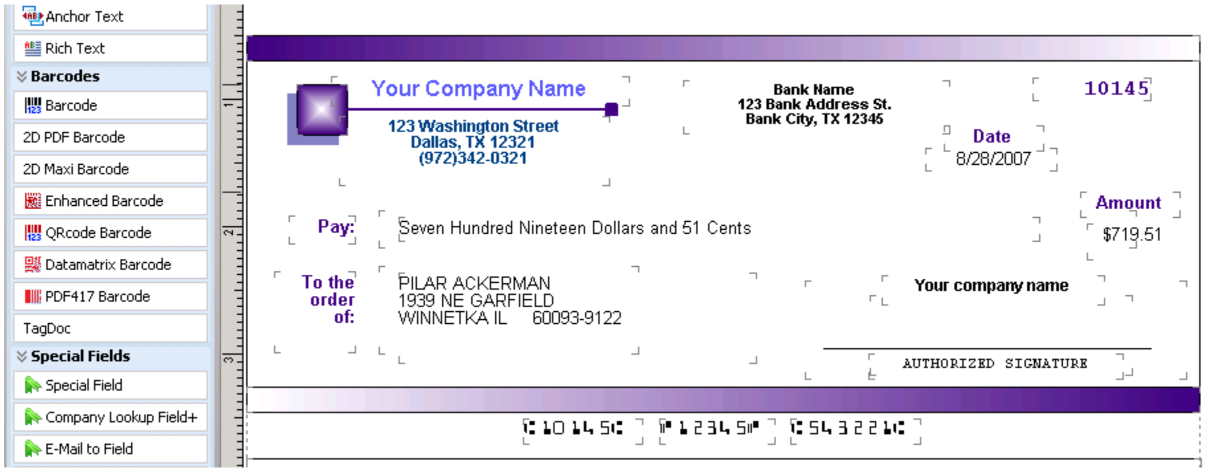
Font style: Regular

Size: 10

Color: Custom

Script: Western

20) By selecting the right color and making the text bold, the label matches the rest of the labels on the check.



Your Company Name
123 Washington Street
Dallas, TX 12321
(972)342-0321

Bank Name
123 Bank Address St.
Bank City, TX 12345

10145

Date
8/28/2007

Amount
$719.51

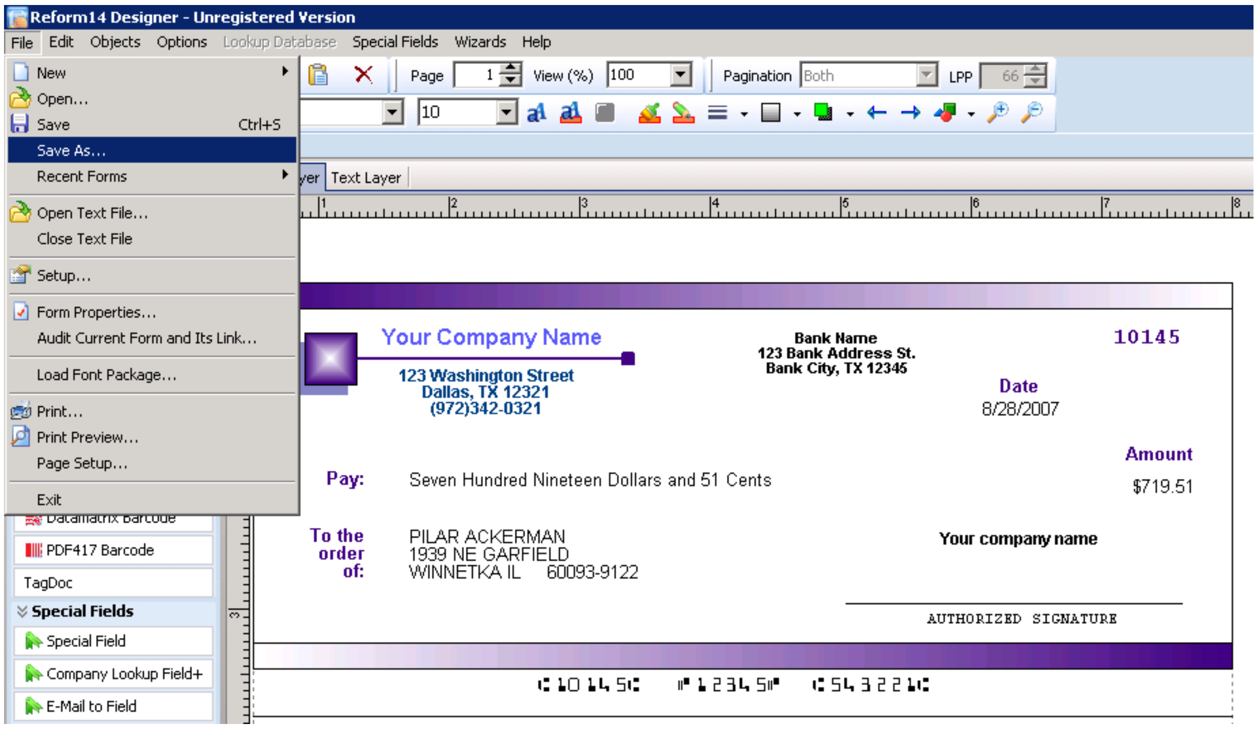
Pay: Seven Hundred Nineteen Dollars and 51 Cents

To the order of: PILAR ACKERMAN
1939 NE GARFIELD
WINNETKA IL 60093-9122

Your company name

AUTHORIZED SIGNATURE

21) Now go ahead and save the form by going **to File>Save As.**



Your Company Name
123 Washington Street
Dallas, TX 12321
(972)342-0321

Bank Name
123 Bank Address St.
Bank City, TX 12345

10145

Date
8/28/2007

Amount
$719.51

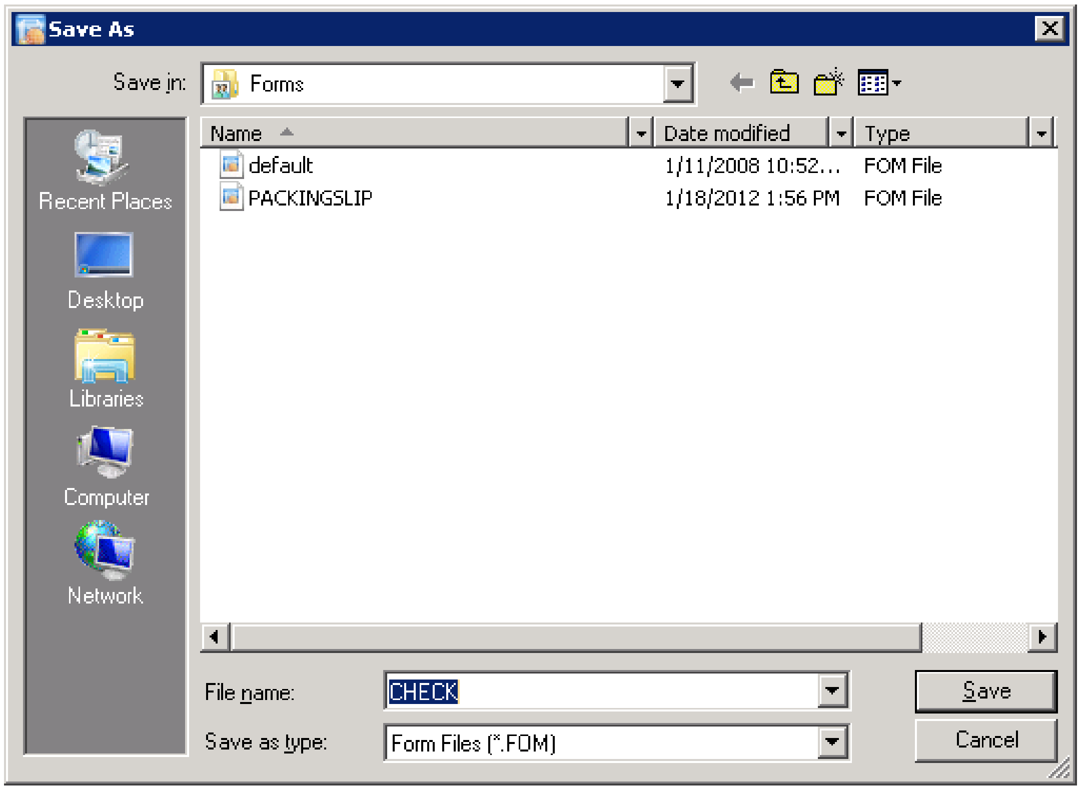
Pay: Seven Hundred Nineteen Dollars and 51 Cents

To the order of: PILAR ACKERMAN
1939 NE GARFIELD
WINNETKA IL 60093-9122

Your company name

AUTHORIZED SIGNATURE

22) Type in **CHECK, and click SAVE**. Congratulations! You just created your first form.



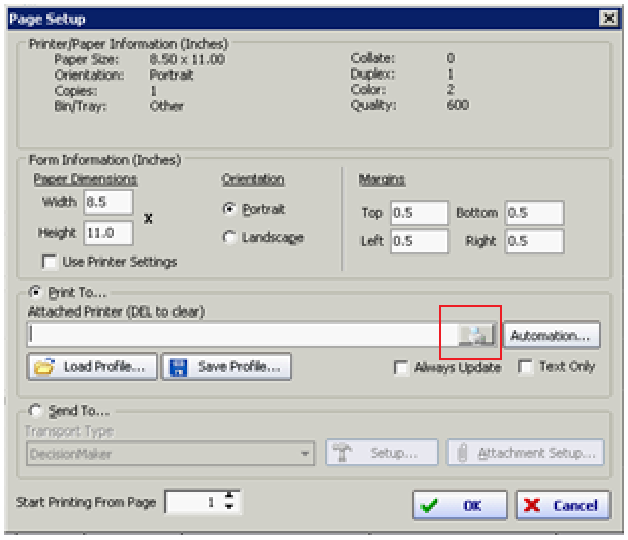
Save in: Forms

Name	Date modified	Type
default	1/11/2008 10:52…	FOM File
PACKINGSLIP	1/18/2012 1:56 PM	FOM File

File name: CHECK

Save as type: Form Files (*.FOM)

23) You can go to **File>Preview…** to preview the Check form. Or if you want to Print the form go to **File>Page Setup…** to open the **Page Setup Window**. Click on the **Printer Icon** to view the available Printers on your machine.



Printer/Paper Information (Inches)

Paper Size:	8.50 x 11.00	Collate:	0
Orientation:	Portrait	Duplex:	1
Copies:	1	Color:	2
Bin/Tray:	Other	Quality:	600

Form Information (Inches)

Paper Dimensions: Width 8.5 x Height 11.0

Use Printer Settings [ ]

Orientation: [✓ Portrait] Landscape

Margins: Top 0.5, Bottom 0.5, Left 0.5, Right 0.5

[✓ Print To…]

Attached Printer (DEL to clear)

Always Update [ ] Text Only [ ]

Send To…

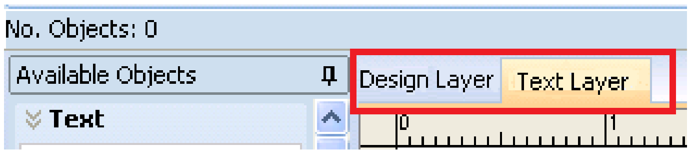
Transport Type: DecisionMaker

Start Printing From Page 1

24) Select the printer you want to use to print the Check. **Click OK**. Save the form. Go to **File>Print** and the Check should print from the selected Printer.

Working with Your Data

Once you have opened the VDP Designer, you will see two tabs: Text Layer & Design Layer.



You will be using the VDP Designer to create or edit the Form overlay for the final output. Once the Form overlay is created, it will be used by the Reform Spooler each time the related text/ASCII file appears in the Spooler directory.

Important: Please refer to the Reform_Server Manual to learn more about the Reform Spooler.

To Start

Start by selecting **New** from the File menu, and choose a form template from the list that matches the type of information you are using. If none of the templates matches your information, click the **Blank Form** button.

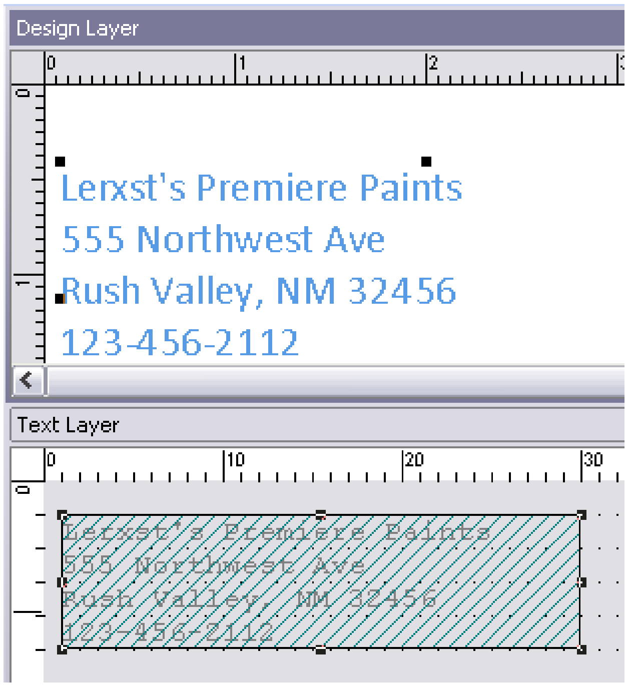
Text Layer contains the information generated by your application.

Design Layer defines the appearance and layout of the form that you design.

See **Tips & Techniques** for ideas about how to create your own desired Reform formats/overlays.

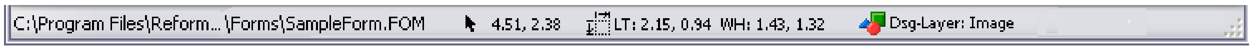
Design Layer

The Design Layer is the layer that holds graphical objects on the form. It works like a design canvas where you can add borders, frames, text and images (such as your company logo). The final output will resemble the layout and content of the Design Layer.



Lerxst's Premiere Paints
555 Northwest Ave
Rush Valley, NM 32456
123-456-2112

The **Status Bar** provides information about the file name and the object currently selected. The right-hand side of the status bar will always display the type of object that is currently selected. This is helpful when there are many objects on the screen and you want to know what it is you are selecting.



C:\Program Files\Reform...\Forms\SampleForm.FOM 4.51, 2.38 LT: 2.15, 0.94 WH: 1.43, 1.32 Dsg-Layer: Image

Common Icons and Controls

Align to Grid
Aligns the selected objects to the grid.

Delete Object
Deletes the selected object(s). (See **Object Basics**)

View Scale Selector View (%) 100
Adjusts the scaling factor (%) to achieve the desired design view.

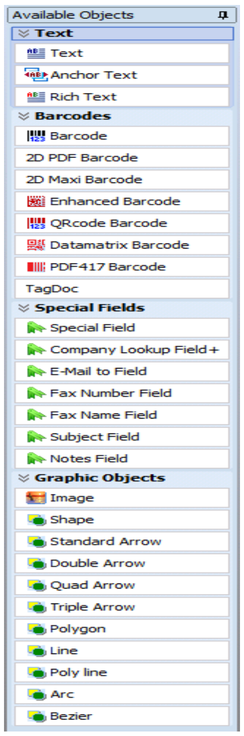
Page Number Page 2
The page number selector advances through the different pages on the Text Layer. By doing so, you can make sure that the text appears and is properly distributed in the Design Layer.

Object Inspector

The new object inspector allows you to easily select and edit form objects. Within the object inspector you edit all properties such as color, size, font, brush etc. You can also reorder, group and align objects.

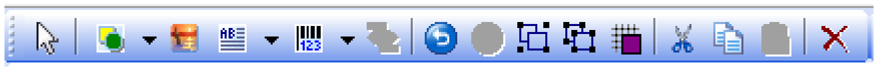
Available Objects

The available objects are easily accessible on the left side of the VDP Designer.



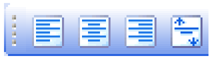
VDP Designer Toolbar

The VDP Designer Toolbar puts all of the frequently used design features in a convenient location so they can be quickly accessed.



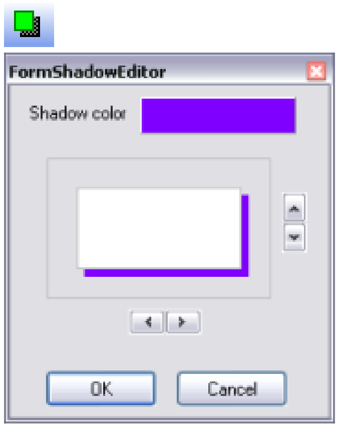
Text-Alignment Options

This part of the toolbar will allow you to align your text horizontally and vertically. Select the Design Text object you would like to align and click the appropriate button to align to the left, right, or center. Click the vertical alignment button to change the position of text within its boundaries.



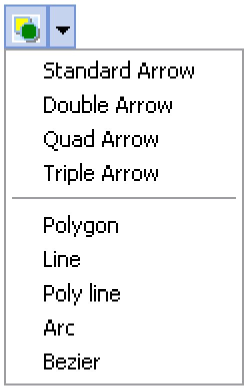
Shadows

Click the Shadow button to bring up a menu that allows you to place a shadow on any object that is visible in the Design Layer. To change the color, click the colored box and bring up the palette. To change the angle and size of the shadow, click the directional arrows to change the object's dimensions.

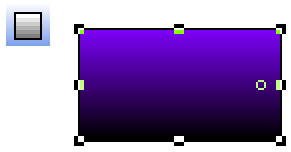


Shapes

To easily create different shapes on your forms, click the Shapes button on the toolbar to bring up the Shapes menu. Click one of the shapes, and draw it out on your form. This will only work in the Design Layer.



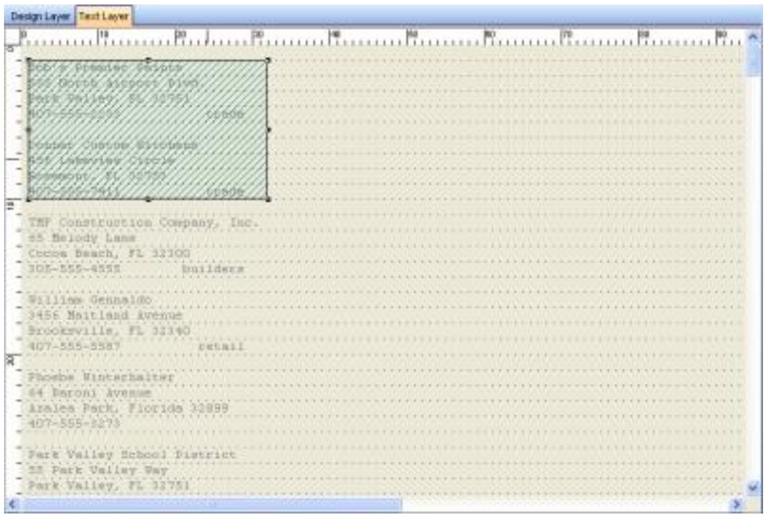
Gradients



First, click on the object you want to add a gradient for and set the Gradient->Visible property to True. Click the Gradient button on the toolbar, click **Start Color**, and then pick a color to start the gradient with in the dialog that shows up. Then, click **End Color**, and choose the ending color. Select a direction for the gradient to fade. The gradient should now be visible. You can change the gradient colors and direction at any time, either by using the buttons on the toolbar or the properties in the properties panel.

Text Layer

The Purpose of the Text Layer is to provide a visual area in which to view the content of a spooler file, and map specific information for inclusion in a form overlay during the design process. The Text Layer displays text exactly as it appears in the text file, which generally reflects the positioning and layout of text generated by the printing application.

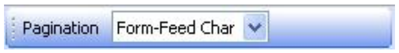


```
[illegible]
[illegible] North Airport Blvd
Park Valley, FL 32751
[illegible]                    trade

[illegible] Custom Kitchens
[illegible] Lakeview Circle
[illegible]
[illegible]                    trade

TMF Construction Company, Inc.
65 Melody Lane
Cocoa Beach, FL 32300
305-555-4555        builders

William Gennaldo
3456 Maitland Avenue
Brooksville, FL 32340
407-555-5587          retail

Phoebe Winterhalter
64 Baroni Avenue
Azalea Park, Florida 32899
407-555-3273

Park Valley School District
55 Park Valley Way
Park Valley, FL 32751
```

To obtain the spooler file generated from a print job for the purposes of creating or adjusting a form overlay, select **Open Text File** from the File menu, and locate your file. If you have no spooler file available in the **Reform…\Spooler** directory, make sure the Reform Spooler is turned off and you have printed from your application to the Reform Server. Printing to the Reform Server can be done either by using the Reform Printer Driver, or a method for capturing data from a host system). The text file should be located in the Reform Spooler directory.

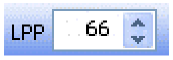
Any text on this layer can be mapped and manipulated from the Design Layer by using the various design objects provided. (See **Available Objects**)

Paginate Method



The Paginate Method dropdown list allows you to configure Reform to look for Form-Feed characters or use a specific number of lines per page to break the spooler file into individual pages. **Form-Feed Char** will look for the character (ASCII value 12) that signifies the end of a page. **Lines Per Page** will use the LPP number you choose and paginate each print stream every X number of lines. Choosing **Both** will attempt to break it up by Lines Per Page, but if a Form Feed character is found before it reaches the Xth line, it will break the page at the form feed character.

Lines Per Page

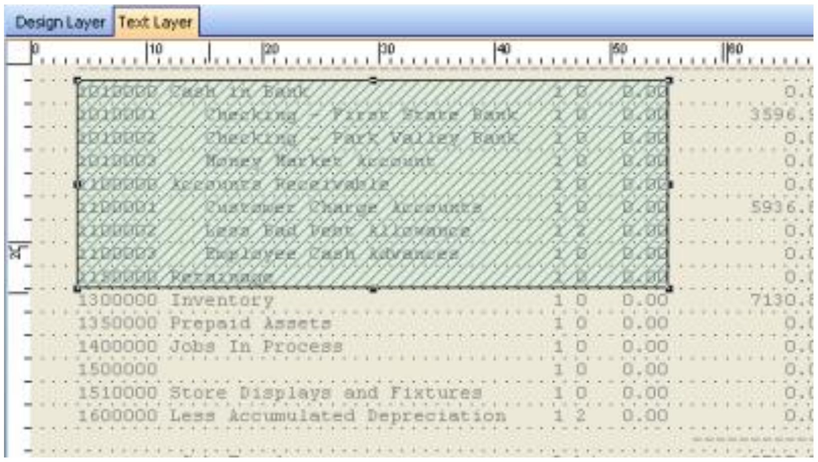


Use this control to adjust the number of lines per page of the text file. When the Spooler processes a print stream on this form, the print stream will be broken up into separate pages by increments of this number. For instance, if you send a 150 line print stream through to this form, it will break the stream up into 3 pages, the first two containing 66 lines each, and the third containing 18 (totaling 150). This is used primarily if you print stream is not able to insert form feed characters between pages. This allows multi-page print streams to be broken up based on the number of lines per page. Be sure to adjust Lines per Page correctly, or your final output page may be misaligned.

Page Selector

The Page Selector allows you to scroll forward and backward through the pages of text in the Text Layer. It also allows you to zoom in to precisely move objects on the Design Layer. It uses the Lines per Page setting and/or form feed characters embedded in the text file to determine where pages begin and end.

Mapping Area



```
1010000 Cash in Bank                          1 0   0.00        0.0
1010001    Checking - First State Bank        1 0   0.00     3596.
1010002    Checking - Park Valley Bank        1 0   0.00        0.0
1010003    Money Market Account               1 0   0.00        0.0
1100000 Accounts Receivable                   1 0   0.00        0.0
1100001    Customer Charge Accounts           1 0   0.00     5936.
1100002    Less Bad Debt Allowance            1 2   0.00        0.0
1100003    Employee Cash Advances             1 0   0.00        0.0
1150000 Retainage                             1 0   0.00        0.0
1300000 Inventory                             1 0   0.00     7130.
1350000 Prepaid Assets                        1 0   0.00        0.0
1400000 Jobs In Process                       1 0   0.00        0.0
1500000                                       1 0   0.00        0.0
1510000 Store Displays and Fixtures           1 0   0.00        0.0
1600000 Less Accumulated Depreciation         1 2   0.00        0.0
```

The Mapping Area is the rectangular region that causes the text beneath it to appear on the Design Layer. To crop portions of the text document, while you are in the Text layer, create a new Link Paragraph object and drag/size it over the text you want to use on the design layer.

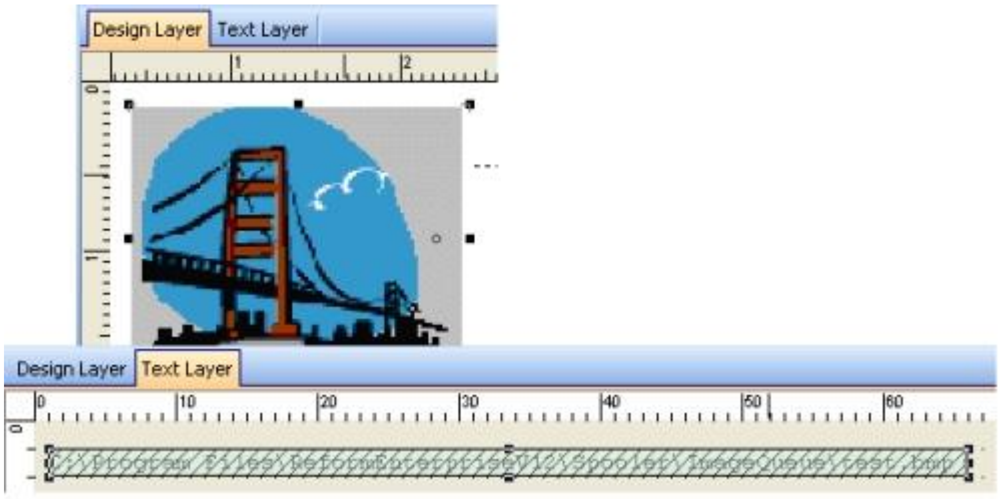
Retain Images

When enabled, this feature will save digital copies of any documents that are printed to the FabSoft Reform Printer. When the print stream is generated, the first line of the file will be a path to the saved image. This path can be mapped using a Link Image object on the Text Layer to display the saved image on your form.

To Use this Function

1. In the Control Panel, select **Reform Port Monitor Setup**. In the Advance tab, check the box labeled **Retain Images**.
2. Print from your application to the **FabSoft Reform Printer.**
3. In VDP Designer, select **Open Text File** from the File menu.

Note: At this point you should see a file created in the Spooler directory, and an image file for each page in the print stream will appear in the Spooler\ImageQueue directory. If you were to create a Link Image object in the Text Layer and map it over the path to the file, the image will be displayed in the Design Layer.



Design Tips & Techniques

Tips

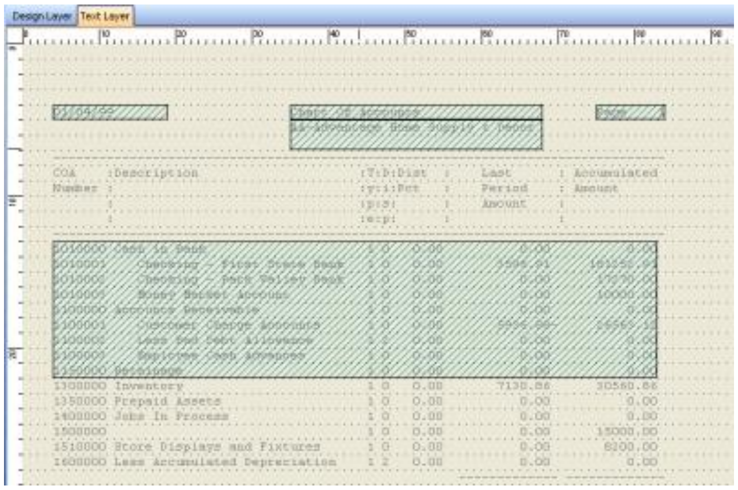
When designing a form overlay, it is always a good idea to print a test file from your application that fills up the entire length of each of its data fields. For example, if the customer name on your test file is FabSoft and it is 7 characters in length, but the user prompt can contain a customer name that is 20 characters in length, filling up the user prompt with 20 characters will help avoid any truncations in the final output of your print jobs.

The header, body and footer areas should remain the same height on both pages of a two-page document. To avoid any possible format changes in the second page, a test file should take up more than one page. If they change, you may need to use the Scripting Plug-in to make adjustments to the text positioning using a Reform Page Process Script. If more advanced adjustments are required, a Decision Maker script may also be used.

Techniques

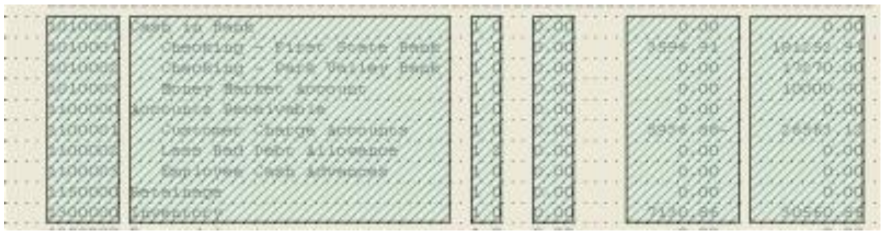
Item Detail Section: If the Text Layer contains information spread over a large area, you will need to map the maximum area allowed to avoid any truncations. Reform uses a copy and paste method when mapping text from the text layer to the design layer, so it will not matter if the item detail area has one item or the maximum number of items.

The easiest and **least** flexible mapping method is to select the entire area using one paragraph object, while using a mono-spaced font such as Courier New or an after market mono-spaced font. Mono-spaced fonts make it easier to retain the text formatting generated by the printing application because the character spacing is consistent.



```
01/04/99              Chart Of Accounts                     Page
                      [illegible]

COA     :Description                 :T:D:Dist  :    Last      : Accumulated
Number  :                            :y:i:Pct   :    Period    : Amount
        :                            :p:s:      :    Amount    :
        :                            :e:p:      :              :

1010000 Cash in Bank                  1 0   0.00          0.00         0.00
1010001    Checking - First State Bank 1 0  0.00       5556.91    [illegible]
1010002    Checking - Park Valley Bank 1 0  0.00          0.00    [illegible]
1010003    Money Market Account        1 0  0.00          0.00     10000.00
1100000 Accounts Receivable           1 0   0.00          0.00         0.00
1100001    Customer Charge Accounts    1 0  0.00    [illegible]   [illegible]
1100002    Less Bad Debt Allowance     1 2  0.00          0.00         0.00
1100003    Employee Cash Advances      1 0  0.00          0.00         0.00
1150000 Retainage                     1 0   0.00          0.00         0.00
1300000 Inventory                     1 0   0.00       7130.86     30560.86
1350000 Prepaid Assets                1 0   0.00          0.00         0.00
1400000 Jobs In Process               1 0   0.00          0.00         0.00
1500000                               1 0   0.00          0.00     15000.00
1510000 Store Displays and Fixtures   1 0   0.00          0.00      8200.00
1600000 Less Accumulated Depreciation 1 2   0.00          0.00         0.00
```

The **recommended method** of covering a large area is to map each column individually.



This will give you the flexibility to:

1. Change the font to any type.

2. Justify the text left, right and center. Certain columns look much neater when they are center-justified, while other columns will look better left/right-justified.
3. Rearrange the columns on the Design Layer.
4. Remove columns from appearing in the Design Layer. If, for instance, the difference between an order and a picking ticket is that one contains the price and one does not, you can simply delete the price column from the Design Layer when designing the picking ticket. This alleviates having to re-map entire regions of information over and over during the design process.

VDP Designer Keyboard Shortcuts

F1

Opens the Reform help file.

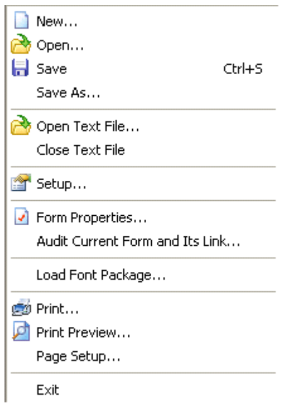
CTRL + Arrow Keys

You can use the combination of CTRL and the Arrow keys to move selected objects one pixel at a time. This can be useful in fine tuning the placement of the objects in your forms.

SHIFT + Arrow Keys

You can use the combination of SHIFT and the Arrow keys to resize the selected objects one pixel at a time.

File Menu



Open Text File...

Opens the text file to be used for the form design process. The text file's contents will appear on the Text Layer, ready to be mapped. ***Note:** When a text file is loaded into the Text layer in Reform, the file will be locked, and as such, cannot be modified or deleted. If you are trying to delete a spooler file manually and it is giving you an error, make sure that file isn't still open in Reform.

Close Text File
Closes a text file and removes it from the text layer.

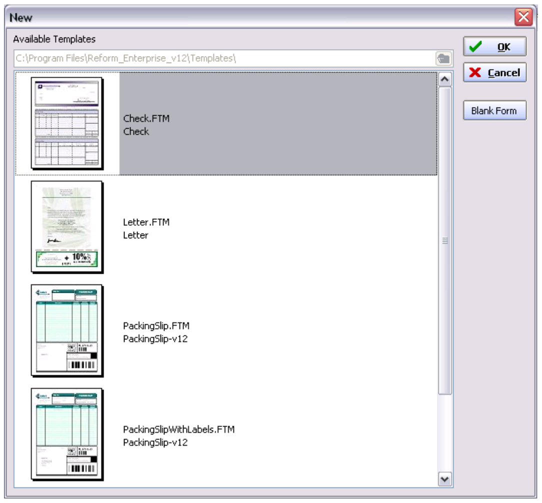
Setup
Change some of the back-end options for Reform, including application paths and special settings.

Form Properties
Change several form settings, such as Page-Process Scripts, Form Linking, Background Image etc.

Load Font Package…
You can manually load a font package created from a Retained Images print action. Font packages are usually found in the Spooler\ImageQueue directory.

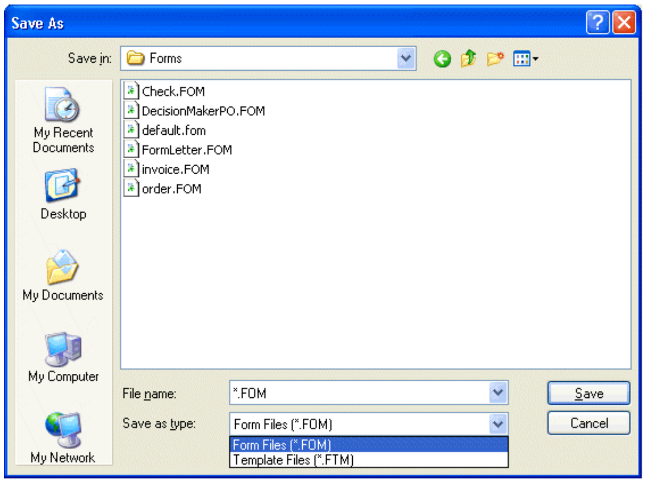
Page Setup
In here you can choose if you want to send your document to a printer or to a device script. You can also configure the corresponding settings for each. This includes the page dimensions, margins, and printer profiles (if a printer is desired). Attachment settings are also available such as PDF, TIF, or other image settings (if a device is desired).

New...



When you choose **New** from the **File** menu, Reform displays a dialog prompting you to make a selection from a list of available templates. If none of the available templates suit your purpose, click **Blank Form** to create a new form or template. When you select a template, Reform creates a copy of the template as a new form overlay for you to edit and modify.

Save & Save As



Check.FOM
DecisionMakerPO.FOM
default.fom
FormLetter.FOM
invoice.FOM
order.FOM

File name: *.FOM
Save as type: Form Files (*.FOM)
Form Files (*.FOM)
Template Files (*.FTM)

When you select **Save** or **Save As** from the **File** menu, Reform prompts you to save the file. There are 2 file types that you can use when saving your form:

Form Files (*.FOM)

Form files are the files that the Spooler uses in conjunction with the text files generated by your application to generate the final output of your print job. They should always be associated with a text file.

Note: We recommend that you save all forms in \REFORM…\FORMS directory.

Template Files (*.FTM)

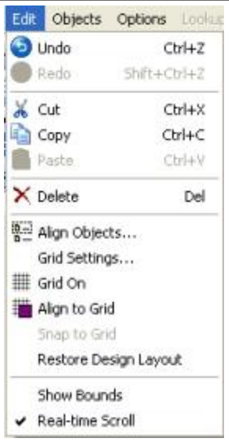
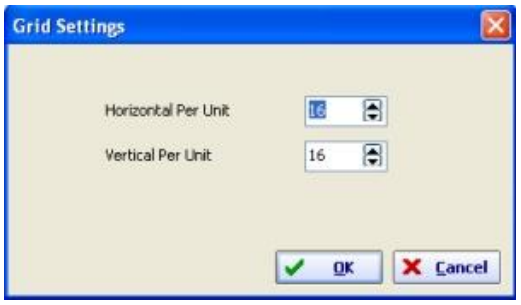
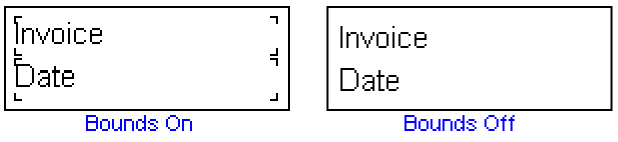
New forms can be derived using the Template file forms. These are the forms that will appear under "New..." dialog. They must be saved in the Templates folder in the Reform directory to be displayed in the Template list.

Note: You can change a form overlay into a template by simply changing its extension to .FTM and placing it in the Templates folder.

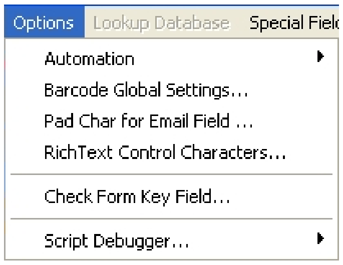
Print

The Design Layer displays exactly what the printer will print out. Clicking **Print** will allow you to process your form overlay with your print stream (if a text file is open) and have it sent to the Send-To Device or Printer that you selected in Page Setup. This manual way of printing can be used for the purpose of testing and perfecting the look of your file output. This is essentially what is happening “behind the scenes” when the Reform Spooler matches up a print stream with a form.

Edit Menu

Undo Undo the last action. **Cut** Removes the selected object(s). Cut objects can be pasted back onto the form using the Paste command. **Copy** Copies the selected object. Copied objects can be pasted multiple times to facilitate easier form design. **Paste** Paste back the previously cut or copied object(s). You can paste multiple times (before performing another cut or copy action) to create duplicates of those objects. **Delete** Deletes the selected object(s). **Grid…** Brings up the Grid Settings dialog. This dialog allows you to adjust the grid spacing per inch on the Design Layer. **Align to Grid** Align the selected object(s) to the grid. **Restore Design Layout** Reset all of the frames in the Design layer to their default positions. **Show Bounds** Toggle on/off the bounds indicator of each object. **Real-time Scroll** This option is on by default. Check this off to improve the response when scrolling the page.	 Edit Objects Options Lookup Undo Ctrl+Z Redo Shift+Ctrl+Z Cut Ctrl+X Copy Ctrl+C Paste Ctrl+V Delete Del Align Objects... Grid Settings... Grid On Align to Grid Snap to Grid Restore Design Layout Show Bounds Real-time Scroll Grid Settings Horizontal Per Unit 16 Vertical Per Unit 16 

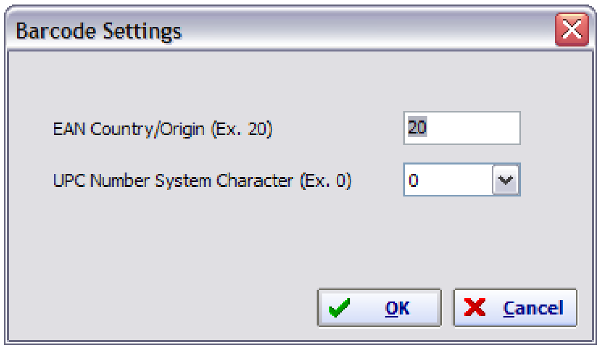
Options Menu



Automation

This allows you to manually select a cover page to be used with faxes.

Barcode Settings



EAN Country/Origin (Ex. 20) 20

UPC Number System Character (Ex. 0) 0

Allows you to change barcode-related settings for barcodes that are on your current form. See **Rich Text Paragraph for more information**

Pad Char for Email Field

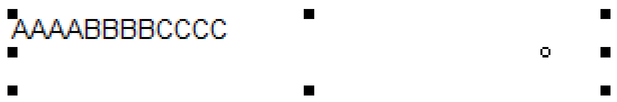
If you are using email integration on your form, the email addresses you are sending to may have a specific prefix or suffix that you need to change. Rather than hard-coding this into the script, you can type the prefix or suffix into this window to automatically reformat email addresses.

RichText Control Characters

This allows you to change the ascii values that represent different types of formatting: bold, italics, and underline. By settings these ascii values, if you are modifying the text content of an object on the Design Layer in a script, you can append these different ascii characters to change the formatting on the fly. For instance, if you have an object on the form called RichText and you set it's value to 'AAAABBBBCCCC' using the script, the script will look like this:

```
_richtext = "AAAA" + "BBBB" + "CCCC"
```

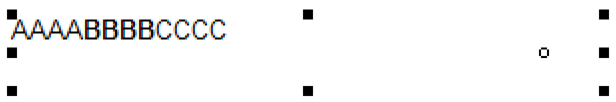
The end result being:


AAAABBBBCCCC

However, if you wanted to make the B’s bold, you could append your selected ascii number for Bold (as defined in the Control Characters window, which is 16) using the Chr() function in the script, so that it looks like this:

_richtext = "AAAA" + chr(18) + "BBBB" + chr(18) + "CCCC"

The final result being:


AAAA**BBBB**CCCC

As you can see, you can change the text formatting by encasing the chosen text in the appropriate ascii characters. It is used in an open/close fashion.

Check Form Key Field...

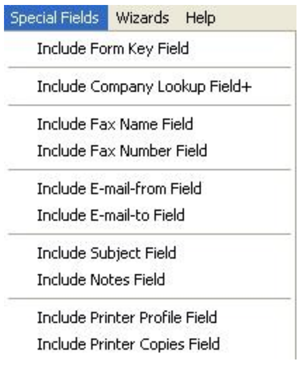
This feature reduces common errors that occur when improperly using the Form Key Field. Check Form Key Field checks the validity of a Form Key Field. If the content of this field does not match the form name, Reform will prompt you with instructions on how to resolve the problem. See **Include Form Key Field** for more Information.

Script Debugger

From this sub-menu, you can launch the Script Debugger and open all scripts associated/used with the current form, only open the page-process script, or only open the device script.

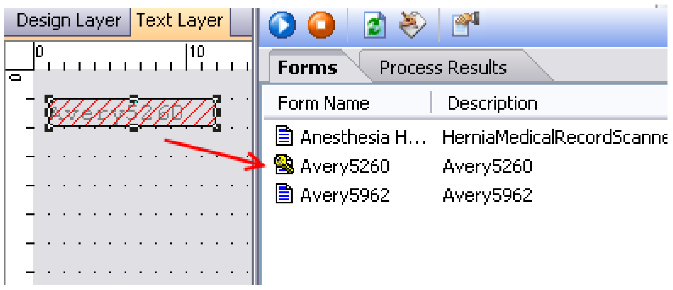
Special Fields Menu

Special fields are used to pass information to the devices that Reform communicates with, such as printers, fax systems, email systems, etc. The only special field available with **Reform Standard** is the **Form Key Field**; all other fields are **Enterprise/PDC only**.



Form Key Field

The Form Key Field is used by the Reform Spooler to associate a print stream with a form when there is a recurring string in the text file at a specific spot. When a text file is detected in the Spooler directory, Reform opens the text file and compares the file contents underneath the Form Key Field against the list of available forms. If the content matches one of the form names, the Reform Spooler uses that form to process the text file. When you add the Form Key Field, you should notice a red rectangular region on the Text Layer.



Form Name	Description
Anesthesia H...	HerniaMedicalRecordScanne
Avery5260	Avery5260
Avery5962	Avery5962

How the Spooler Looks for the Proper Form to Use

The Spooler needs to reference an ASCII text file and apply it to a Reform form. The Spooler looks for the text files by cycling through three different methods in the following order:

1. Using the Form Key Field

The Spooler opens the Text/ASCII file and searches through the list of Form Keys. To set up the Form Key option, click on **Include Form Key Field** under **Special Fields** in VDP Designer. To verify that you have set the reference correctly, go to **Check Form Key Field** under **Options**.

2. Matching File Names

The Spooler attempts to match a Text/ASCII file with a form of the same name. The file extension is ignored. If a report is printed to a file called ORDER.OUT in the C:\Program Files\Reform…\Spooler\ directory, the Spooler looks for a corresponding ORDER.FOM file in the Reform…\Forms\ directory. When found, it will open ORDER.OUT and ORDER.FOM for processing. **Note**: The filename will be ignored if the Spooler finds a form and a text file that meet the criteria for the Form Key Field.

3. Using the Default form

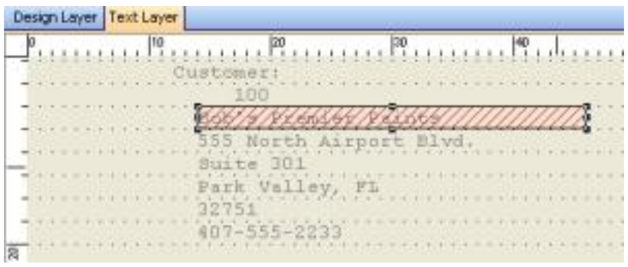
If the Text/ASCII file does not qualify for the first two methods, the default form (default.fom) will be used. If the default form does not exist, the Spooler will ignore the Text/ASCII file.

Company-Lookup Field +

If you are planning to use a fax or email device, this field must be included on the form overlay. This field is used to look up the company name in the Lookup Database. When you add Company-Lookup Field to a form, a red mapping box will appear. The Company-Lookup field is invisible to Reform; therefore, you can place this field anywhere on the Design Layer and it will not show up in the final output. In addition, it can be used to fill in Company Name for your email or fax device as well.

To enable the lookup abilities, the file “LookupAdo.fpg.remove” in the Reform…\Plug-ins\ directory must be renamed to “LookupAdo.fpg” Once this is done, the VDP Designer and Spooler will need to be closed and opened (or restarted).

When you add it onto the Text Layer, its content is determined by the text enclosed within the mapping region. Therefore, you cannot edit the content of this field. Map the entire company name section on the text file, and the form will look for the company name in the database every time it is used. If a corresponding company name is found, it will then be able to use any associated information (such as fax numbers or email addresses) that it finds.



```
Customer:
    100
Bob's Premier Paints
555 North Airport Blvd.
Suite 301
Park Valley, FL
32751
407-555-2233
```

Company-Lookup Field added onto the Text Layer

Fax Fields

These fields are used to extract fax information directly from the text content. They are invisible to Reform, and thus can be placed anywhere on the Design Layer without appearing on the final output. Certain fax devices may or may not use all of the fields. Any fields that are not applicable will be ignored.

Note: You must include the Company-Lookup field in your form overlay in order to initiate the fax automation feature.

Fax Number Field

Include this field on the Text Layer if you want Reform to grab the fax number directly from the text content. Reform will use this fax number instead of looking it up in the database. However, if the fax number is empty, Reform will try to lookup the fax number from its database using the Company-Lookup Field. You should map the entire fax number, including the area code. The valid fax number formats are:

(973)334-0720
973-334-0720
973 3340720
973 334 0720
973-3340720
334-0720

Fax Name Field

This field is used to fill in the contact name when applicable.

Subject Field

This field is used to fill in the Subject when applicable.

Notes Field

This field is used to fill in notes when applicable.

Email Fields

In addition to the Company-Lookup Field, you can also include these email fields in your form overlay. Email fields are a group of fields that help manage Reform’s email functions. These fields are invisible to Reform, and can be placed anywhere on the Design Layer. These fields are used to fill in necessary information required by the email system.

Note: You must include the Company-Lookup field in your form in order to use the email feature.

Email-from Field

Include this field on your form if the sender's name appears in the Text/ASCII file. This field does not have to be an email address; it can be anything that indicates the sender's name. This field is only applicable when using the SMTP feature of the email Plug-in. This field is ignored when using MAPI.

Email-to Field

Include this field if the recipient's email address appears in the Text/ASCII file. Reform will use this field to get the email address instead of looking for one in the Lookup Database.

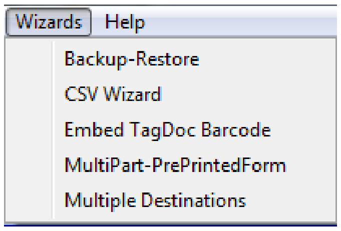
Subject Field

This field is used to fill in the subject of the email. Any text that this field contains will be used as the email subject.

Notes Field

This field is used to fill in the body of the email. Any text included in this field will appear as a message in the email body.

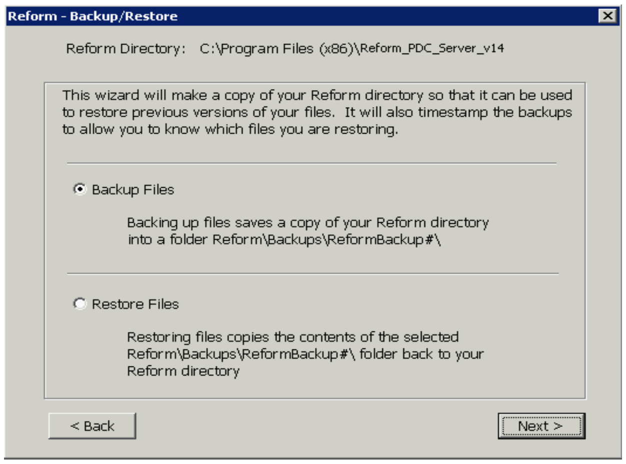
Wizards Menu



Wizards can automate tasks that may be tedious or require many steps to set up properly. They all use a very simple format to gather information from you and set up your form.

Backup-Restore

This wizard will create, manage, and restore backups of your entire Reform directory. This includes forms, backup files, page scripts, device scripts, and settings. This is useful if you have a large workflow consisting of heavily modified scripts and forms that you need to back up often.



Reform Directory: C:\Program Files (x86)\Reform_PDC_Server_v14

This wizard will make a copy of your Reform directory so that it can be used to restore previous versions of your files. It will also timestamp the backups to allow you to know which files you are restoring.

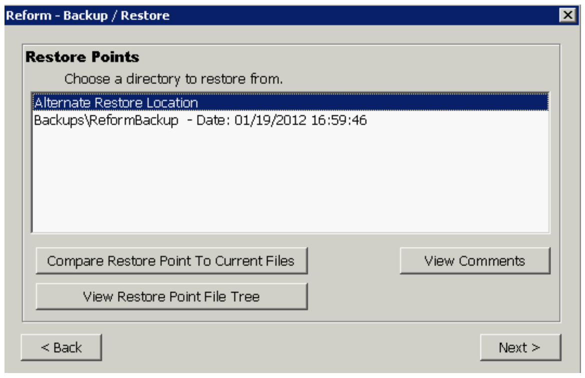
(●) Backup Files

Backing up files saves a copy of your Reform directory into a folder Reform\Backups\ReformBackup#\

( ) Restore Files

Restoring files copies the contents of the selected Reform\Backups\ReformBackup#\ folder back to your Reform directory

On the first screen, it will ask you whether want to make a backup or restore a backup (The Restore Files option is not visible if there are no backups currently saved). Step through the screens and the backup will begin. If you select Restore Files, it will allow you to select a date and time that you would like to restore to.


Reform - Backup / Restore
Restore Points
Compare Restore Point To Current Files
View Comments
View Restore Point File Tree
< Back
Next >

Choose a directory to restore from.

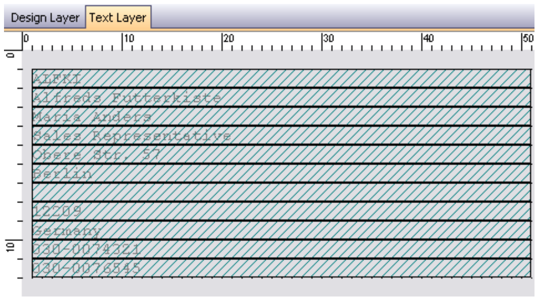
Alternate Restore Location
Backups\ReformBackup - Date: 01/19/2012 16:59:46

Select an item and click **Next** to begin the restore process.

CSV Wizard

The CSV Wizard will allow you to select a CSV file, and it will then create a special field box over each line of text in the selected CSV file. The CSV file will be broken down into separate pages by line, and then added as the Text File in Reform. The text maps can all be modified in the VDP Designer.

This is the SampleCSV.csv file, found in the /Reform…/Misc folder, after it has been processed by the CSV Wizard. Each page in the text layer is now a row of the CSV file.

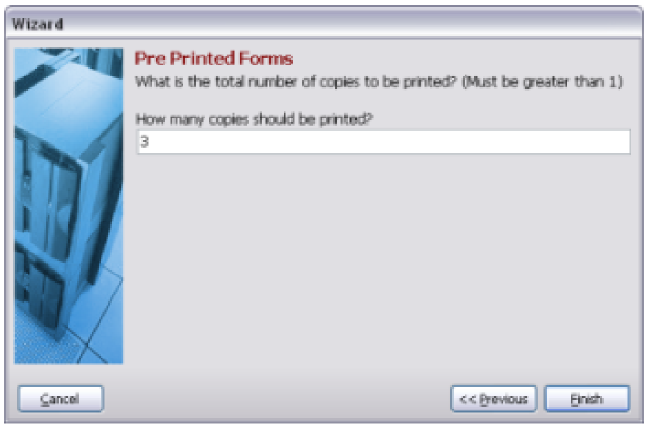


ALFKI
Alfreds Futterkiste
Maria Anders
Sales Representative
Obere Str. 57
Berlin

12209
Germany
030-0074321
030-0076545

MultiPart PrePrintedForm

This wizard will create multiple pages for forms that look very similar but perform different tasks. For instance, you could run this wizard to create several forms that create a customer copy, an internal copy, an archive copy, etc.



What is the total number of copies to be printed? (Must be greater than 1)

How many copies should be printed?
3

It also allows you to select the send-to devices right from the wizard screens, as well as change the amount of copies to be made.

On this screen, you must enter how many copies of your original form you would like. If you enter 4, for example, using a form called "Check", the form link chain will look like this:

Check.fom → Checkpart2 → Checkpart3 → Checkpart4

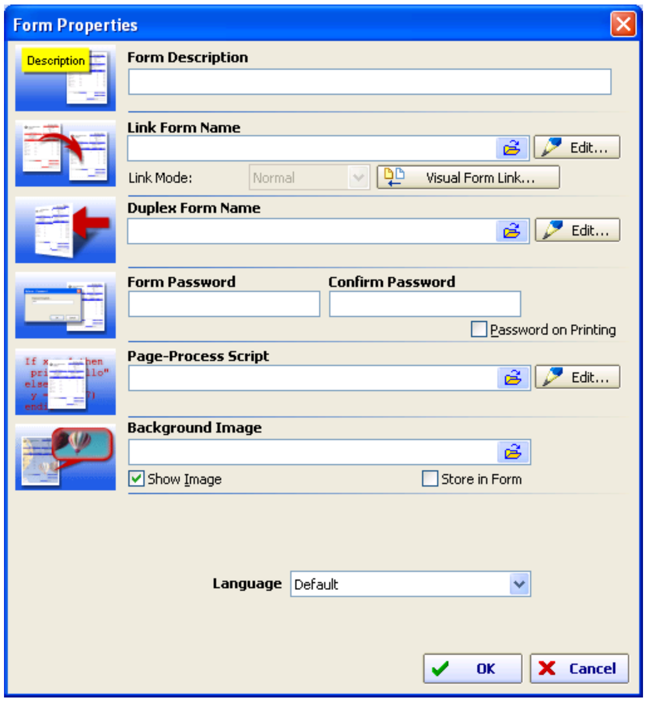
You can modify the Send-To Device for each one. Copying and linking multiple forms can be time consuming, but the Pre-Printed Forms Wizard does all of that for you—you simply enter the number of copies needed.

Multiple Destinations

This wizard will allow you to select several destinations that you want to send a form to. It will repeatedly let you choose different destinations to send your form to, and then it will create all of the necessary forms, set and configure the scripts, and link the forms together.

Special Functions

Form Properties



Link Mode: Normal

☐ Password on Printing

☑ Show Image ☐ Store in Form

Language Default

Form Description

Enter the text that describes the form overlay; this is viewable in the Reform Spooler.

Link Form Name

Specifies the form overlay to be linked to the current form overlay. This signals Reform to perform a Form Link process.

Link Mode

Specifies the method that Reform will use to print the linked forms:

1. **Normal** - The current form is used to print all the text pages. Once finished, the linked form is then used to print all the text pages. The process repeats until all linked forms are processed. See **How Form Link Works** for more information. Note: In order to prevent an endless link process, circular linking is not allowed. Reform automatically detects a circular link and will end the process.
2. **Collate -** Processes each linked form for each page in the text layer; this method reproduces pre-printed multi-part forms. See the section on **Collating** in **How Form Link Works** for more information.
3. **Advance Page** - This allows Reform to print using the current form for the first text page, then the linked forms for the next consecutive pages. This continues until there are no more text pages. See the section on **Advance Page** in **How Form Link Works** for more information.

Duplex Form Name

Use Duplex printing to print on both sides of a page. Certain printers are capable of printing in duplex mode. If your printer supports duplex printing then you can take advantage of this feature. The form specified in the Duplex Form Name box is the form that will be printed on the backside of the page. This information is stored along with your main form; therefore, each form can have a different duplex form attached to it.

Note: Both forms must have the same printer definition set in **Printer Properties**, and the duplex option must be activated in the properties setting of each printer.

Form Password

Form Password allows you to password-protect an important form overlay (such as a check or purchase order) from any unauthorized modification and/or printing. To password protect your form, simply type the password you would like to use in the **Enter Password** and **Confirm Password** boxes. To prevent unauthorized printing, check the **Password on Printing** box. Otherwise, leave Password on Printing unchecked so that any user can print the form; however, you will not be able to modify it without the password.

Background Image

The Background Image is an image that occupies the entire page. If you have a pre-printed form that you need to import into VDP Designer, simply scan the form you have into an image and use the image as a background. You can also use the background image as an embedded watermark.

Language

Use the language drop down list to select the language for the form.

How Form Linking works

Reform normally prints every page of a Text/ASCII file using only one designed form; however, using the Form Link feature allows Reform to print multiple pages to as many forms as the job requires without having to perform a new print job for each form. To facilitate this, all applicable forms must be linked together during the design process. To Link two or more forms together:

1. Open the Form Properties dialog box by selecting **Form Properties** from the File menu.
2. Click on the folder icon to the right of the text field titled **Link Form Name**. This opens the Form List window, which displays every available form.
3. Scroll through the list of available forms until you find the one you're looking for, and highlight the form by clicking on it.
4. Click **OK**. The selected form will now appear in the Link Form Name field in Form Properties.
5. Click **OK** again to return to VDP Designer.
6. Save the form by selecting **Save** from the File menu. The two forms are now linked together. If you wish to add more forms to your Form Link, repeat this process by creating a link reference to the third form in the second form, to the fourth form in the third form, and so on, until all forms are properly linked. You may link as many forms as you wish.
7. You can test the form link by selecting **Print** from the File menu in the first form. If every linked form prints out in the desired order, your Form Link is working.

Advantages to Using Form Link

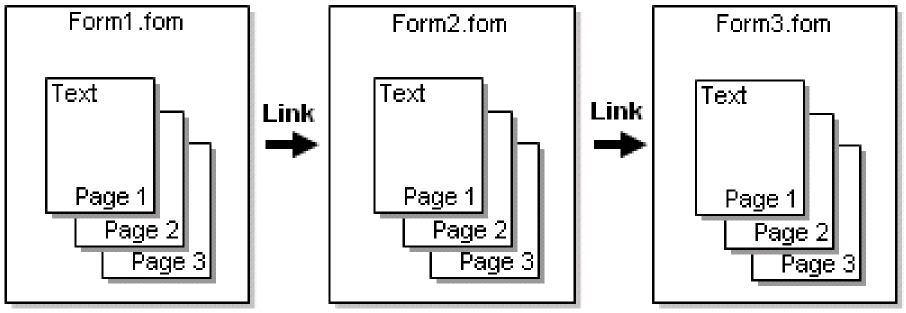
Form Link is a powerful feature, and when used correctly it can save a person, department or company hours of work that would otherwise be spent manually sending faxes, composing emails and printing hardcopies. With Form Link, all these interrelated tasks can be accomplished at the touch of a single button.

Each form in a series of linked forms can be sent to a different output device. For example, the first form can be sent to a laser printer, the second form can be sent to a fax machine, and the third can be sent to the shipping printer in the warehouse. More forms can be added to the chain and automatically routed to multiple different departments, email addresses, fax numbers, etc., facilitating the accomplishment of potentially hundreds of otherwise tedious activities.

There are three different linking methods that can be used, each of which is designed to produce output in a different manner. They are called **Form Link Normal**, **Form Link with Collating** and **Form Link with Advance Page**. Following are detailed descriptions about how to use Form Link to print using these different link methods:

Form Link Normal

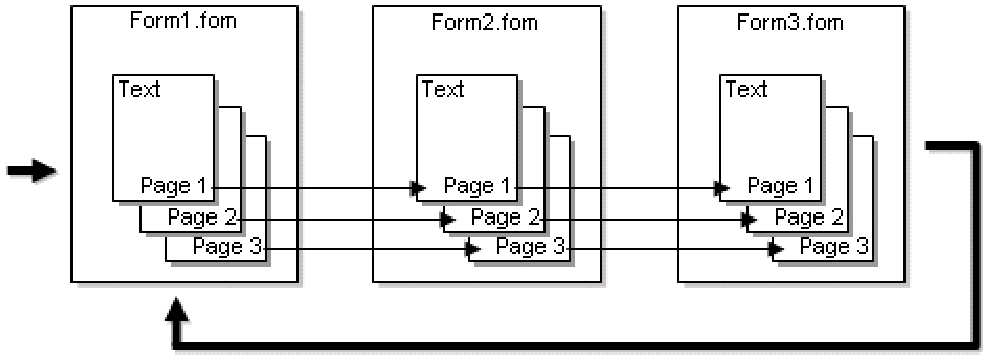
This is the standard method of using Form Link, which processes each text page for each form before moving on to the next. Assume that you want to generate a 3-part output and you've already designed 3 different form layouts using the same information: FORM1.FOM, FORM2.FOM and FORM3.FOM. You would link FORM1.FOM to FORM2.FOM and FORM2.FOM to FORM3.FOM per the instructions at the beginning of this section. FORM1.FOM is the initial form that you use to print your document. When the printing process begins, Reform uses FORM1.FOM to print every text page, then FORM2.FOM for every text page, and finally FORM3.FOM for every text page. Because FORM3.FOM has no reference specified in its Form Properties, the printing stops after FORM3.FOM is processed.



Form Link Normal

Form Link with Collating

Form Link can be set to collate your printed forms. Using this method, each page of your document will be printed on every linked form before Reform moves on to the next document page. This simulates pre-printed forms that you would typically find on a line printer.



To set the forms to collate, the following steps must be taken:

1. Make sure your forms are properly linked together.
2. With Form 1 loaded into the VDP Designer, go to **File** on the menu bar and click **Form Properties**.
3. In the Link Mode dropdown box, select **Collate**.
4. Click on the **Visual Form Link** button next to the dropdown box, and a pictorial representation of your linked forms should appear, cascading from left to right:
5. Click **OK** to close the Visual Link window, and click **OK** again to close the Form Properties window.
6. Save your form to insure that all changes take effect.

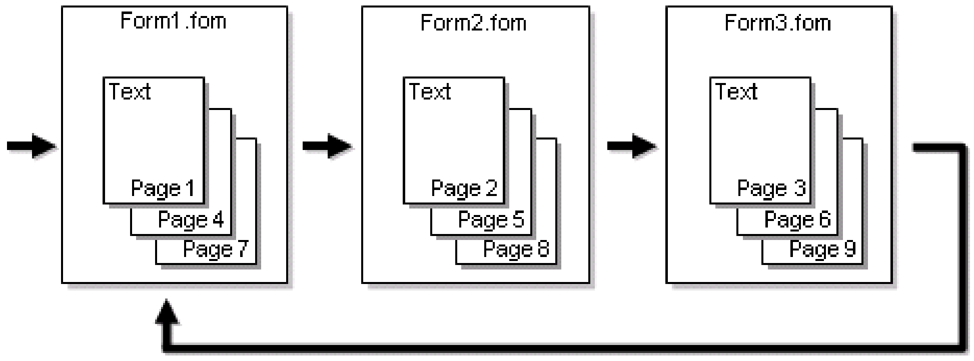
When the above steps have been completed, your forms will print out collated.

Note: It is only necessary to activate this feature on Form 1. The Link Mode for all other linked forms should be set to **Normal**. To stop the forms from collating, simply set the Link Mode on Form 1 back to Normal.

You can also print collated forms from separate printer trays. See **Printing from Multiple Trays** for more information.

Form Link with Advance Page

Form Link can also be set to print using a different form for each consecutive page of your Text/ASCII file. The first page is printed to one form and the next page is printed using the linked form. The process continues until there are no more text pages to process.



Form Link with Advance Page

Assume that you want to print using a different form for each consecutive text page: FORM1.FOM for page 1, FORM2.FOM for page 2 and FORM3.FOM for page 3. You would link FORM1.FOM to FORM2.FOM and FORM2.FOM to FORM3.FOM as usual, but instead of leaving the Form Link reference for FORM3.FOM empty, set it to link back to FORM1.FOM, causing it to cycle through every form again, in a perpetual cycle until it runs out of text pages. To achieve this, the **Link Mode** for **each form** must be set to **Advance Page**.

More on Form Link with Advance Page

If any of your linked forms' Link Mode is not set to Advance Page, the form will loop on itself until it reaches the final page of the Text/ASCII file.

The loop back can be referenced to any linked form. For example, if you link FORM3.FOM in the above example to FORM2.FOM, the forms will continue to loop from FORM2.FOM to FORM3.FOM, bypassing FORM1.FOM for every page except the first.

A Common Scenario: If your text file contains a cover or start page, you can create a form for the cover page, and a second form for the accompanying information. After linking the two forms together, set the first form's Link Mode to Advance Page. In the second form, do not select a **Link Form Name**. This will cause the cover page to be processed for the first text page, and the second form will loop on itself for the remaining pages of the print job.

Forms can also be used to process a print stream using the LoadForm() function with the Scripting Plugin. This will forward the print stream to the form you wish, and it will be processed.

Designing Objects

Object Basics

The Form Editor uses an object-oriented approach for form design. Each element on the form is treated as an object. There are four basic objects you can add to a form: Shape, Image/Graphic, Barcode, and Paragraph Text.

Available Objects

Shape Object

Image / Linked Image Object

Paragraph Object / Rich Paragraph Object

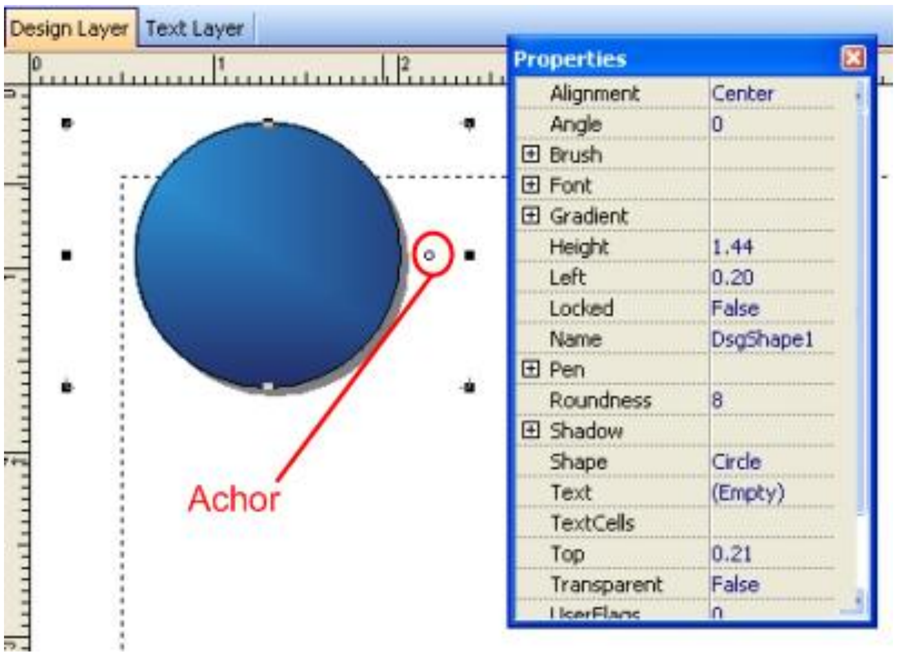
Barcode Object

Delete and Align-to-Grid buttons

Selecting an Object

To select an object, simply click on the object you want to manipulate. If two or more objects overlap one another, you can select the object underneath by clicking the left mouse button while holding down the **CTRL** key. Alternately, you can right-click the top object and select **Send to Back**. This will cause the object beneath it to become fully visible. Once the intended object is in front, it can be selected with a simple mouse click.

Each object has its own properties (color, line width, text alignment, etc). You can access the properties of an object by selecting the object as described above. The properties for that particular type of object will be populated into the Properties panel.



Alignment	Center
Angle	0
⊞ Brush	
⊞ Font	
⊞ Gradient	
Height	1.44
Left	0.20
Locked	False
Name	DsgShape1
⊞ Pen	
Roundness	8
⊞ Shadow	
Shape	Circle
Text	(Empty)
TextCells	
Top	0.21
Transparent	False
UserFlags	0

The Object Properties dialog box.

You can move, resize, and rotate objects by using the mouse. To move or resize an object, first select it by clicking the left mouse button. A map mark will appear around the object, indicating that it is selected and ready to be moved or sized. To rotate an object, click and drag the anchor on the object.

Adding an Object

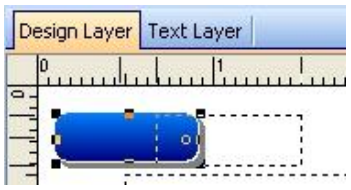
To add an object to a form, click one of the object-creation buttons (see **Available Objects**, above), and click and drag out the object on the Text or the Design Layer.

Deleting an Object

To delete an object, select the object you want to delete, and then select **Delete** from the Edit menu. You can also press the Delete key on your keyboard. Alternatively, you may right-click the object and select **Cut** from the dropdown menu, or press **CTRL-X**.

Moving an Object

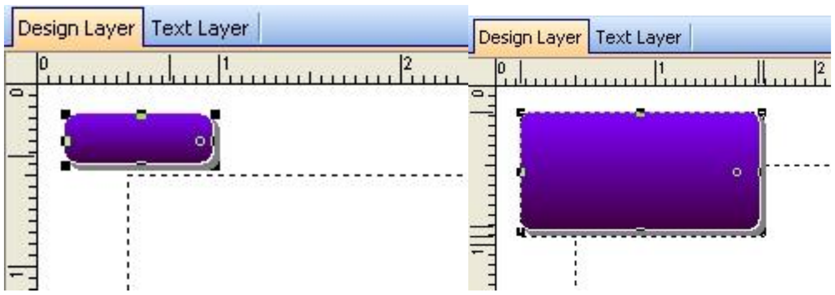
To move an object, place the mouse inside the boundaries of a selected object, hold down the left mouse button and drag the object to the desired location.



You may also use the combination of **CTRL** and the **Arrow keys** to move the selected object one pixel at a time. This can be useful in fine-tuning the placement of the objects.

Sizing an Object

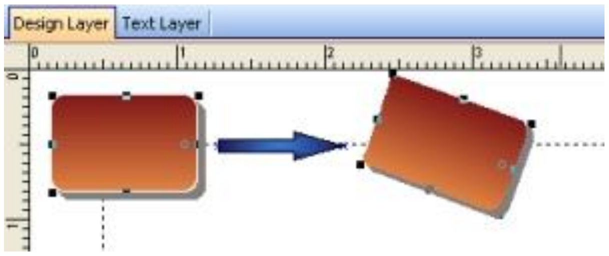
To size an object, move the mouse toward the edge of the object until the cursor changes to a diagonal arrow shape. Hold down the left mouse button and drag the corner or side of the object. It will grow and shrink with the movement of the mouse. When the object reaches the desired size, you can release the left mouse button.



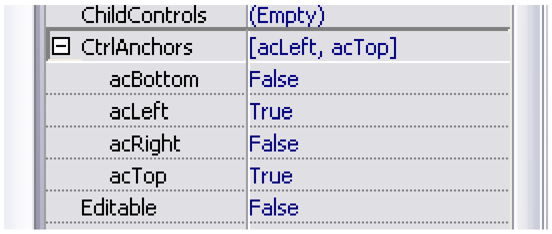
You may also use the combination of **SHIFT** and the **Arrow keys** to size the selected object(s) one pixel at a time. This can be useful when fine-tuning the size of an object.

Rotating an Object

To rotate any object, click and drag on the circular anchor. In the image below, it is on the right-middle of the box. When you click and drag it, you can rotate it to any position you want.



Using the Linked Anchor Text Object



ChildControls	(Empty)
⊟ CtrlAnchors	[acLeft, acTop]
acBottom	False
acLeft	True
acRight	False
acTop	True
Editable	False

A Linked Anchor Text can only be created on the text layer. A Linked Anchor Text object is used when the text that you wish to map is shifting around in the print stream. The Linked Anchor Text Object has several features:

1. It is hidden. The text selected by the Linked Anchor Text Object is not displayed at the time of print.
2. It is used in association with a Linked Paragraph object. To associate a Linked Label Object with the Linked Anchor Text Object, click on the **Attach Objects**… the object you wish to attach, then click on **OK**.
3. It can be used to control display of the Linked Label Object based on keywords. This is enabled by checking the "Hide Attached Obj. When Not Found" option. An object must be attached for this option to work.
4. It can track keywords within a specific column (vertical search) or row (horizontal search).
5. Tracking can be restricted by locking search positions on the left, right, top or bottom of the Linked Anchor Text Object. For instance, if you know the keywords only appear below the placement of the object and to the right of the object, the left and top side of the object can be locked. If the text to track always appears in the same location, you may restrict tracking by locking all sides of the object. If you want tracking enabled on all sides, uncheck all lock options.

Example:

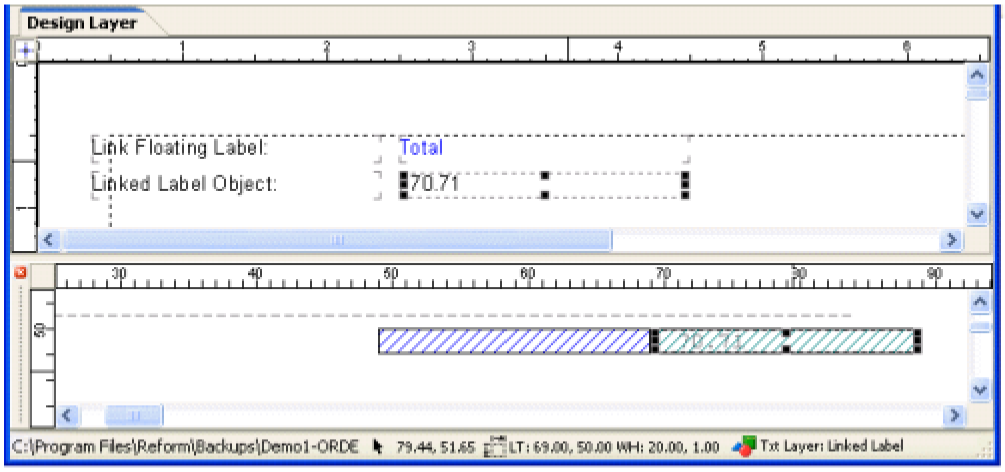
A company prints a batch of multi-page purchase orders which always displayed the subtotal on each page and displays only the final total on the last page. The last page also shows the word "Total" before the actual total amount is displayed. Furthermore, the total amount displayed shifts vertically and does not always appear at the bottom of the page. Instead of printing the subtotal on each page, the company wants to display only the final total at the end of the purchase order. This company chooses to use the Linked Anchor Text object to accomplish the task.

To use the Linked Anchor Text Object in the scenario above:

1. Make sure you are working in the Text Layer.
2. Click on the **Label Object** icon to create a new Linked Label Object. Follow the rest of the instructions above to move, size and position the Linked Label Object. Please remember to give the Linked Label Object a name.
3. Next, create a Linked Anchor Text Object on the text layer.
4. Move and size the Linked Anchor Text Object until it overlays the desired text content.
5. Click on the **Linked Anchor Text** object and look at the properties panel on the bottom left of the screen. Change the anchor properties.
6. Name the Linked Anchor Text appropriately.
7. Enter the text to track. In this case, the user enters the word "Total". If the word "Total" shifts vertically, the user may choose to restrict the text track to vertical

movement by locking the horizontal movement (lock the left and right sides of the object).

8. Next, click on the **Attach Objects…** button. Select the Linked Object Label that you wish to attach to this Linked Anchor Text Object. In the example above, the user would attach the object selecting the subtotal/total amount field.
9. For the example above, the user will then check "Hide Attached Obj. When not Found." Using this option, if the Linked Anchor Text Object does not find the word "Total", the attached object (which is highlighting the subtotal/total amount), will not be displayed. However, if the Linked Anchor Text Object finds the track word "Total", the Linked Object Label displaying the dollar amount will be displayed.



C:\Program Files\Reform\Backups\Demo1-ORDE 79.44, 51.65 LT: 69.00, 50.00 WH: 20.00, 1.00 Txt Layer: Linked Label

To preview the results, go to **File>Print Preview**.

Paragraph Object

The Paragraph Object is used to create single and multiple lines of text. Depending on which layer you create this object in, the behavior can be very different. Please read on for more information.

Using the Paragraph Object

In the **Design Layer**, the Paragraph Object is used when you want to add multiple lines of text to a form that does not appear in the Text/ASCII file. The text in the Paragraph Object remains the same regardless of the content of the Text/ASCII file, and Reform stores it along with the form. To use the Paragraph Object for this purpose:

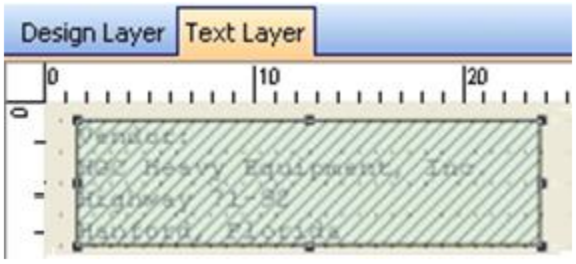
1. Make sure you are working in the Design Layer.
2. Click on the **Paragraph Object** icon, and click and drag on the Design Layer to draw a **Paragraph Object**.

3. The properties pane on the bottom right will change and allow you to change multiple properties of the **Paragraph Object.**
4. Modify the **Text** property by clicking the **...** button on the Text property line.
5. Click **OK**. The Paragraph Object will now contain the text you typed in step 4. Position this anywhere on the screen. Modify the Text property at any time to adjust the appearance of the text.

In the **Text Layer**, the Paragraph Object is referred to as the **Linked Paragraph Object**, because it is used to link information from the Text Layer to the Design Layer. The Linked Paragraph Object differs from the standard Paragraph Object in that its contents are dictated by the text content on the Text Layer. To use the Linked Paragraph Object:

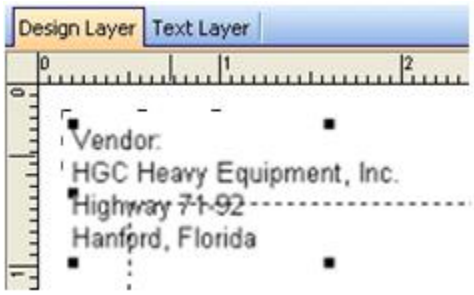
1. Make sure you are working in the Text Layer.
2. Click on the **Paragraph Object** icon, and click and drag on the Design Layer to draw a **Paragraph Object**.
3. Move and size the Paragraph Object until it overlays the desired content.
4. Go to the Design Layer. The Linked Paragraph Object and the text it overlays will be displayed on the form. You may alter the position or properties of this Paragraph Object the same way you would a standard Paragraph Object.

Note: If the position or font properties of the Linked Paragraph Object are altered on the Design Layer, it will not affect the position or appearance of the same text in the Text Layer.



Vendor:
HGC Heavy Equipment, Inc
Highway 71-92
Hanford, Florida

Linked Paragraph Object on the Text Layer



Vendor:
HGC Heavy Equipment, Inc.
Highway 71-92
Hanford, Florida

Linked Paragraph Object on the Design Layer

Rich Text Object

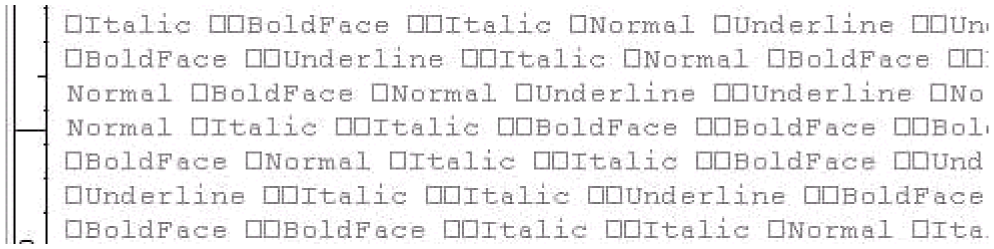
The Rich Text Object is a specialized object used to map text that contains ASCII control characters. These control characters are used for various formatting functions, such as bold, underline and italics. If your application is capable of printing these characters, the Rich Paragraph Object must be used to maintain the formatting implied by the use of the control characters.

Example:

In the following example, the parenthesized numbers are representative of control character number 16, which is interpreted by Reform as a bold formatting character. All control characters will actually appear as small rectangles in the text layer.

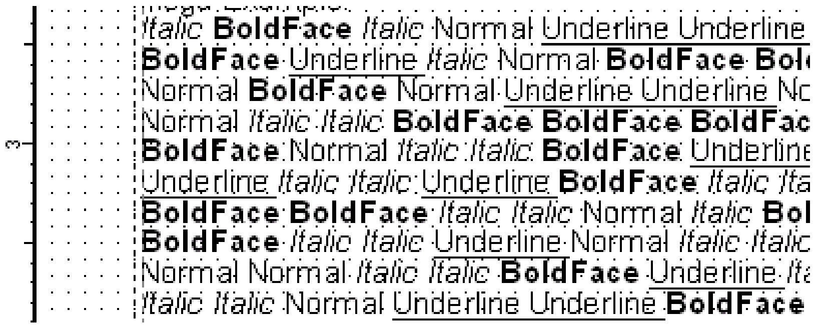
1. Text content: 'An opened product (#16) cannot (#16) be returned.'
2. Final Output: 'An opened product **cannot** be returned.'

An example of a text file or print stream with Rich Text formatting in the Text Layer is as follows:



```
□Italic □□BoldFace □□Italic □Normal □Underline □□Un
□BoldFace □□Underline □□Italic □Normal □BoldFace □□
Normal □BoldFace □Normal □Underline □□Underline □No
Normal □Italic □□Italic □□BoldFace □□BoldFace □□Bol
□BoldFace □Normal □Italic □□Italic □□BoldFace □□Und
□Underline □□Italic □□Italic □□Underline □□BoldFace
□BoldFace □□BoldFace □□Italic □□Italic □Normal □Ita
```

In the Design Layer, the mapped text will appear properly formatted, as shown below:

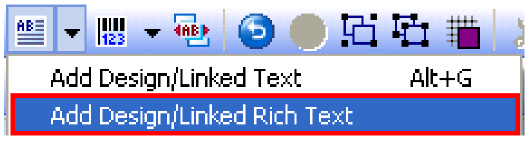


Italic **BoldFace** *Italic* Normal Underline Underline
BoldFace Underline *Italic* Normal **BoldFace** **Bol**
Normal **BoldFace** Normal Underline Underline Nc
Normal *Italic* *Italic* **BoldFace** **BoldFace** **BoldFac**
BoldFace Normal *Italic* *Italic* **BoldFace** Underline
Underline *Italic* *Italic* Underline **BoldFace** *Italic* *Ita*
BoldFace **BoldFace** *Italic* *Italic* Normal *Italic* **Bol**
BoldFace *Italic* *Italic* Underline Normal *Italic* *Italic*
Normal Normal *Italic* *Italic* **BoldFace** Underline *Ita*
Italic *Italic* Normal Underline Underline **BoldFace**

Creating a Rich Text Object

In the **Text Layer**, the Rich Paragraph Object is referred to as the **Linked Rich Paragraph Object**, because it is used to link information from the Text Layer to the Design Layer. The Linked Rich Paragraph Object differs from the standard Rich Paragraph Object in that its contents are dictated by the text content on the Text Layer. To use the Linked Rich Paragraph Object:

1. Make sure you are working in the Text Layer.
2. Click on the arrow next to the **Paragraph Object** icon to create a new Linked Rich Paragraph Object.

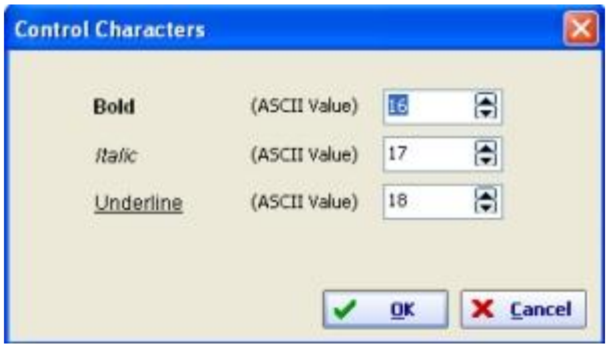


3. Move and size the object until it overlays the desired content.
4. Go to the Design Layer. The Linked Rich Paragraph Object and the text it overlays will be displayed on the form. You may alter the position or properties of this object the same way you would a standard Rich Paragraph Object.

Note: If the position or font properties of the Linked Rich Paragraph Object are altered in the Design Layer, it will not affect the position or appearance of the same text in the Text Layer.

Modifying the Control Characters

The control characters can be changed to fit your needs. To access the Control Characters dialog, select **Control Chars** from the Options menu. The dialog box pictured below will appear. To modify the ASCII values used to represent different formatting options, simply click the up or down arrows to the right of the number boxes.


Control Characters

Bold	(ASCII Value)	16
Italic	(ASCII Value)	17
Underline	(ASCII Value)	18

Keep in mind that certain control characters are reserved by the system. Make sure that you do not use those control characters. The reserved control characters and their assigned functions are listed below:

#8	Back Space
#9	Tab Character
#10	Line Feed
#12	Form Feed
#13	Carriage Return

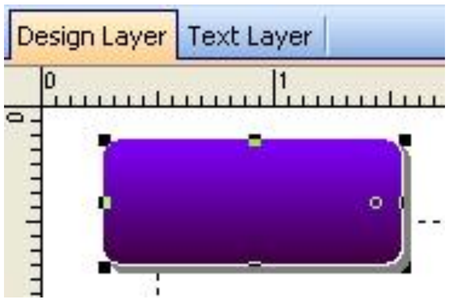
#27 Escape

Shape Object

The Shape Object is used to create frames, lines, rectangles, squares, arrows, and circles.

Creating a Shape Object

1. Make sure you are working in the Design Layer.
2. Click on the **Shape Object** icon. Click and drag to create a box.
3. You may change the shape and properties of the Shape Object by clicking on it and then modifying the properties in the Properties panel.
4. Alternately, if you want to simply create a shape without having to change properties first, click the little arrow next to the Shape Object button and you can easily create several different types of shapes.
5.



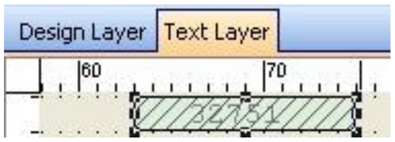
Shape Object Sample (Rounded Rectangle with Gradient applied)

Barcode Object

The Barcode Object converts text content on the Text Layer into a specified barcode representation on the Design Layer.

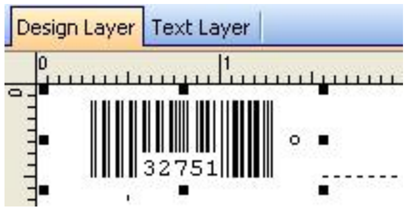
Creating a Barcode Object

1. Make sure you are working in the Text Layer.
2. Click on the **Barcode Object** icon. Click and drag on the text layer to create the **Barcode Object**. The object appears as a bluish mapping box.
3. Drag the Barcode Object so that it overlays the desired text content.



Barcode Object on the Text Layer

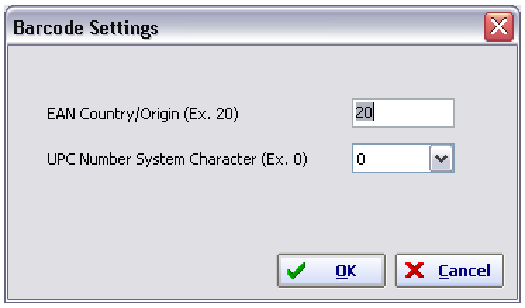
4. Go to the Design Layer. The Barcode Object will appear as a graphical barcode.



Barcode Object on the Design Layer

5. Click on the object and modify the properties to alter the barcode's type, size, caption, appearance and other properties.

The Barcode Settings (which differ from its properties) may be accessed and modified by selecting **Barcode Global Settings** from the Options menu.



EAN Country/Origin (Ex. 20) 20
UPC Number System Character (Ex. 0) 0

Barcode Default Settings:

Note: Certain barcode types require that you specify additional information:

EAN 8 and EAN 13 require 2-digit country or origin code.

1. UPC requires a number system character from 0 - 9.

This additional information will be applied to the newly added barcodes (depending on the type of barcode you choose), eliminating the need to individually specify this information for each of the barcodes you add. After the barcode is added, you can manually alter these values in the Barcode Global Settings dialog.

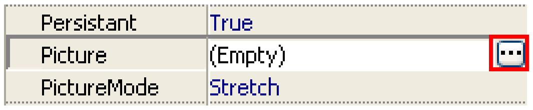
Image Object

The Image Object is used to add graphics to your form. The supported graphic types are BMP, JPG, GIF, TIFF and EMF, however, the recommended image type is BMP. You can use this object to import your company's logo, backgrounds, digital signatures, etc.

Using the Image Object

1. Make sure you are working in the Design Layer.
2. Click and drag out an image object in the Design Layer.

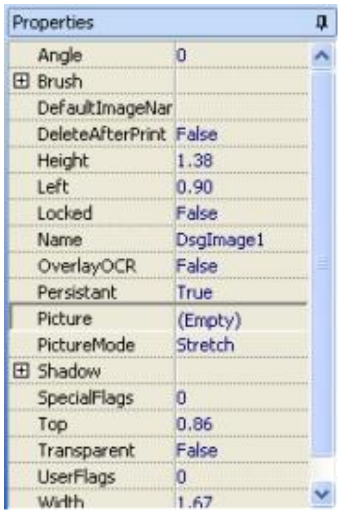
3. Click on the **image object**.
4. In the Properties panel, change the Picture property.



Persistant	True
Picture	(Empty)
PictureMode	Stretch

5. When you find the desired file, click on the file name and press **Open**. The image that you selected will now be displayed in the Properties panel.
6. The desired image will now appear in the Image Object on the Design Layer.

Note: If the box labeled **Stretch** has been checked in Properties, you may resize the picture to your specifications. In Stretch-mode, resizing the Image Object causes the image to grow and shrink accordingly. If the Stretch box is unchecked, however, the image will appear at its standard size, and adjusting the size of the Image Object will not affect the size of the image. Stretch may slow down the processing speed of Reform, since the image has to be resized each time. To optimize an image, use a graphing program to resize the image and reduce colors.



Properties

Angle	0
⊞ Brush	
DefaultImageNar	
DeleteAfterPrint	False
Height	1.38
Left	0.90
Locked	False
Name	DsgImage1
OverlayOCR	False
Persistant	True
Picture	(Empty)
PictureMode	Stretch
⊞ Shadow	
SpecialFlags	0
Top	0.86
Transparent	False
UserFlags	0
Width	1.67

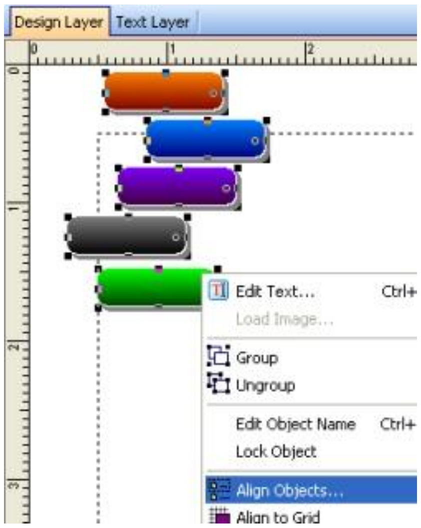
The Properties panel when you have an image object selected.

Object Alignment

Sometimes objects appear out of alignment, and it becomes necessary to align them to each other in order to maintain a uniform and professional appearance. In VDP Designer, this is a quick and easy process.

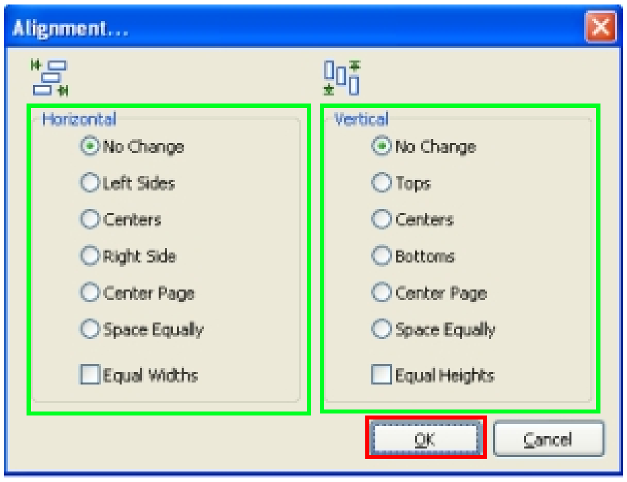
Aligning a Group of Objects

1. Select the objects that you wish to align. You may select multiple objects by holding down the Shift key and clicking on each object you wish to select or by dragging your mouse over multiple objects.
2. Right-click on the object that you wish the others to align to, and select **Align Objects** from the popup menu.



Objects selected for alignment

3. In the Alignment dialog box, select the alignment method that you would like to use.

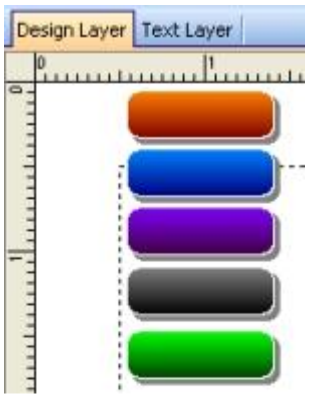

Alignment...

Horizontal
- [✓ No Change]
- Left Sides
- Centers
- Right Side
- Center Page
- Space Equally
- ☐ Equal Widths

Vertical
- [✓ No Change]
- Tops
- Centers
- Bottoms
- Center Page
- Space Equally
- ☐ Equal Heights

OK Cancel

4. Click **OK**. When you return to VDP Designer, the objects will be perfectly in line.



The objects are now all aligned to the green button, because the green button was first clicked on.

Image Linking

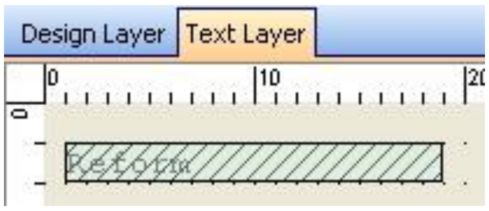
The Linked Image Object is a unique and powerful feature that causes images on a form to change based on text content in the Text Layer. If, for example, you wish to include graphical representations of products on a specific form, the Linked Image Object can be set to display a product's image by reading its name or product ID from the Text/ASCII file during the printing process.

Using the Linked Image Object

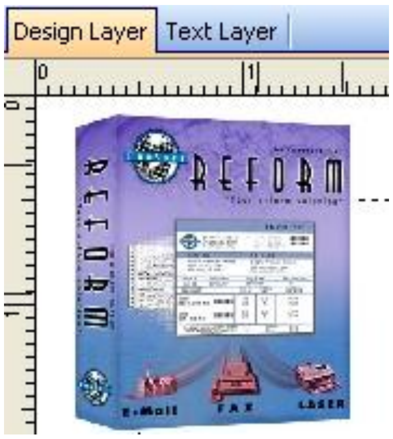
To use this feature, the name of the graphic file must be the same as the text content mapped by the Linked Image Object. For example, if you would like to have pictures of your inventory displayed on the invoice, you could use the product's item number as the image file name. If the item number is E10034, for instance, and you have a bitmap image of the item, then the bitmap should be saved as E10034.BMP. When Reform processes a form containing Linked Image Objects, it looks for the first image with a name that matches the text content. Follow the steps below to configure a form to use a Linked Image Object.

1. To set Reform to retrieve the pictures, you must first specify the location of the graphic files. To do this, Select **Setup** from the **File** menu to call up the Setup dialog.
2. In the Setup dialog, make sure the Image Directory points to the correct folder. The default folder is **C:\Program Files\Reform…\Images**. You may use this folder or specify a new one.
3. Click **OK** to return to VDP Designer.
4. Now you are ready to create your Linked Image. Make sure you are in the Text Layer, and click the **Linked Image Object** icon. Click and drag a box on the text layer to create your **Linked Image Object**.
5. Move and size the box so that it covers the desired text content.

6. Go back to the Design Layer and right-click the object to view its Object Properties. **Default Image** allows you to select a default image for use if Reform cannot locate a file specified by the mapped text. To disable this, simply leave the field blank.
7. The image displayed in the Linked Image Object can also be controlled from a Page Process Script.



Linked Image on Text Layer



Linked Image on Design Layer

Tips and Suggestions for the Use of Linked Image Object

1. Display a salesperson's picture on the Invoice using their ID.
2. Display pictures of items on an invoice using the item numbers or SKU.
3. Display different signatures for checks and purchase orders.
4. Display different company logos for multi company systems.

Printer Profile

In order to use the Printer Profile field:

1. In the VDP Designer >> Special Fields >> Printer Profile, map the printer profile name on the text layer or define it from the page script.
2. Go to VDP Designer >> File >> Page Setup and select a printer and set all the properties (tray, finishing etc) and select **Save Profile** and use the same name as the printer profile name in #1.

This will allow users to pass a printer name in their data stream that will control the printer and destination.

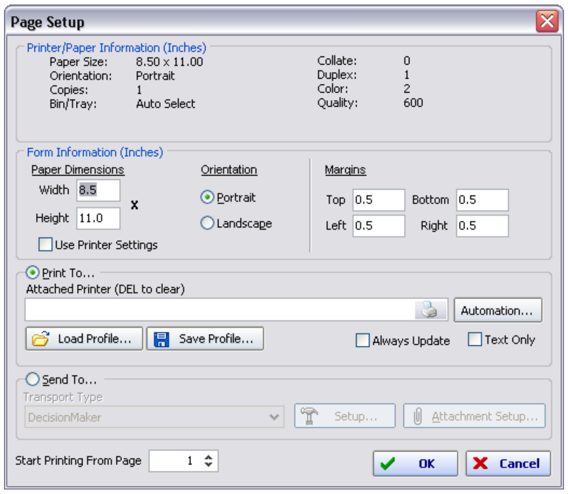
Also, if there is text in the text layer which matches the printer name and there is no other printer profile selected, it will use that printer profile.

You can also modify the printer profile and control it from a page-process script by setting the variable for the object "_PrinterProfile". For instance, if you have a printer profile named "MainPrinter", you can change the printer profile to print to MainPrinter by adding in the page-process script the following code:

```
_PrinterProfile="MainPrinter"
```

Output Devices

Page Setup



Page Setup

Printer/Paper Information (Inches)

Paper Size:	8.50 x 11.00	Collate:	0
Orientation:	Portrait	Duplex:	1
Copies:	1	Color:	2
Bin/Tray:	Auto Select	Quality:	600

Form Information (Inches)

Paper Dimensions: Width 8.5 x Height 11.0

[ ] Use Printer Settings

Orientation: [✓ Portrait] Landscape

Margins: Top 0.5, Bottom 0.5, Left 0.5, Right 0.5

[✓ Print To...]

Attached Printer (DEL to clear)

[ ] Always Update [ ] Text Only

Send To...

Transport Type: DecisionMaker

Start Printing From Page 1

Printer/Paper Information

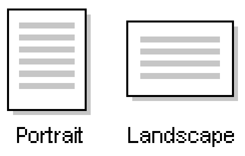
This section displays the current settings of the selected printer.

Form Information

This section displays the current settings of the form overlay. Changes to these settings will affect your form layout. Paper dimensions and orientation can be set independently from the printer settings; however, in most cases you will want these settings to be the same as the printer page settings. Checking **Use Printer Settings** will keep the form's dimensions and orientation equal to that of the printer settings. To change the printer settings, click on the Printer button 🖨 next to the selected printer.

Paper Orientation

Select either Portrait or Landscape.



Note: The **Print to** and **Send to** settings are mutually exclusive. Depending on the printing method you choose, selecting one will disable the other.

The **Start Printing From Page** option lets you choose the print start page.

Always Update

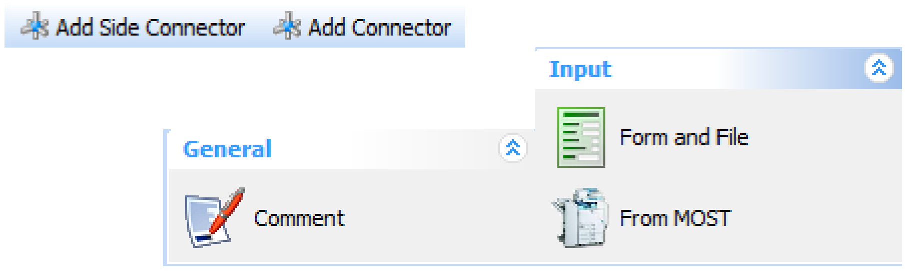
The Always Update checkbox, if selected, will automatically push the printer settings to all forms that are linked to the current form through the Form Linking setting in the Form Properties. It will create different print jobs in the printer queue. This setting can include different printer trays, print settings such as duplexing, collating, stapling together multi-part forms, etc.

Text Only

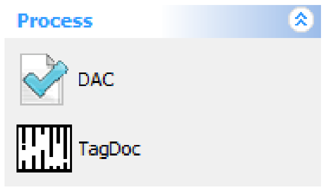
If selected, the form overlay will not be applied to the form at the time of printing.

What is the Workflow Designer?

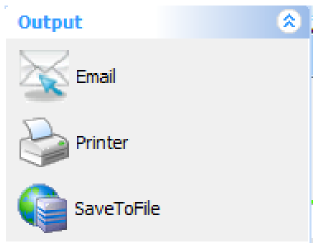
Reform's Workflow Designer is visually enhanced wizard approach to designing and implementing a workflow. It gives users the ability to drag and drop different steps of the workflow process onto a digital canvas which then automatically makes all the changes necessary to the chosen input data in order to satisfy what the user sees on the screen. There are five main types of objects that can be added to the workflow: Arrow Connectors, Comments, Inputs, Processes and Outputs. Arrow connectors are used to link between Inputs, Processes and Outputs and provide the visual logic to a workflow.



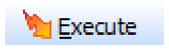
Comments are used to add descriptions to different parts of the workflow in order to more accurately depict what is being done at each object or group of objects. There are two different types of inputs which depend on which Plug-ins a user currently has installed: “Form and File” and “From MOST”. “Form and File” is your typical Reform Form and data stream and is available to all users. “From MOST” is only available if Reform’s MOST Plug-in is installed and allows the user to configure a workflow around an MFP (Multi Functional Printer). A user is limited to one input per workflow.



Processes include DAC (Document Authority Control) and Tag Doc (a bar-coding and document routing system) and are available depending on which Plug-ins a user currently has installed. The DAC Plug-in is required to be able to add the DAC process onto a workflow. Adding a DAC processes onto the workflow gives the user the ability to route documents based on a hierarchy. The DAC process requires an Input and at least one Output type. The Tag Doc Plug-in is required to have the capability to add the TagDoc process onto a workflow. Adding a Tag Doc process onto the workflow gives the user the ability to add barcodes onto their forms and subsequently decide what to do with those documents once they are scanned.



Currently there are 8 different available Outputs depending on which Plug-ins are installed: Printer, SaveToFile, Email, RightFax, Ricoh LanFax, Microsoft's SharePoint, DocumentMall and Alchemy. Printer and SaveToFile are installed by default with Reform. All other Outputs require their respective Plug-ins. All outputs require at least one Input to be configured but can also be a combination of an Input and any number of processes. For example, one workflow can have a "Form and File" input, and then be routed to DAC. Once approved in DAC, the document can then be sent through the Tag Doc process and printed with a barcode. Finally, the Tag Doc process would be connected to the desired Output. All of the Outputs have intelligent and newly redesigned setup wizards to appropriately configure each respective output and have control over different settings. The settings in each output can be statically entered or can use variable data from the data stream or MOST.



Once a workflow is complete, the execute button on the toolbar is clicked in order to fulfill what is represented on the screen. During this process, the form will be backed up to a Workflow Designer Original Forms directory which is located in the same directory as the selected form. Each Process and Output will have their own form which is also created in the same directory as the selected form. All forms will be linked together in the appropriate manner and all form modifications (adding special fields, configuring the send to device, or printer) will take place. Each newly created form is copied from the original backed up form so that the user can easily make additions to the workflow without having to go delete any forms in the forms directory. Every change can be verified by opening up the forms in VDP Designer.

Benefits of using the Workflow Designer

- The Workflow Designer improves upon the current way of trying to configure a workflow. Presently, it is very easy to get lost while sending a form to multiple destinations or through different processes.
- The Workflow Designer simplifies the whole process of trying to recreate a workflow which could have been originally been drawn with pen and paper.
- The user has control over different Processes and Outputs through the use of either static or variable data.
- With the Workflow Designer, you now have a visual representation of your workflow which can be easily modified by adding, removing or reconfiguring different Processes and/or Outputs.
  Have the ability to save a workflow and then later go back and make changes without having to configure parameters all over again.

Multiple Destinations Tutorial – Printing and Archiving Simultaneously

This tutorial will walk you through the steps of creating a simple workflow in Reform. It will demonstrate how to setup Reform to print a document, while at the same time archiving the file to your hard drive. The workflow will be created using the Workflow Designer.

First, the VDP Designer must be started. Go to **Start > Programs > Reform… > Variable_Data_Printing > VDP Designer.**

1. We will be creating a new form with an existing template. Go to **File > New > New**. Select the **SaveToFile** template (for demonstration purposes) and click **OK.**

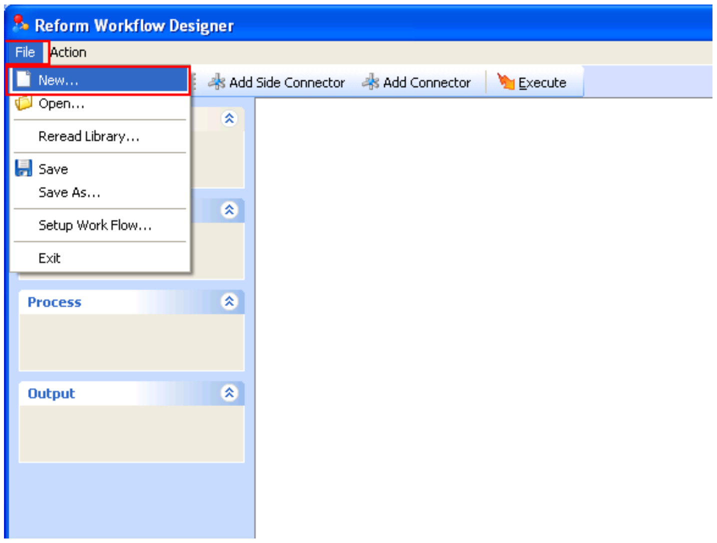
SaveToFile.FTM
SaveToFile Sample - Invoice Form

2. When it asks if you would like to open a text file for designing, select **No.**
3. Next, go to **File>Save** to save your form. Name it **SaveToFile.FOM**
4. You can now close the VDP Designer since we have our form which will be used to configure and route the document.

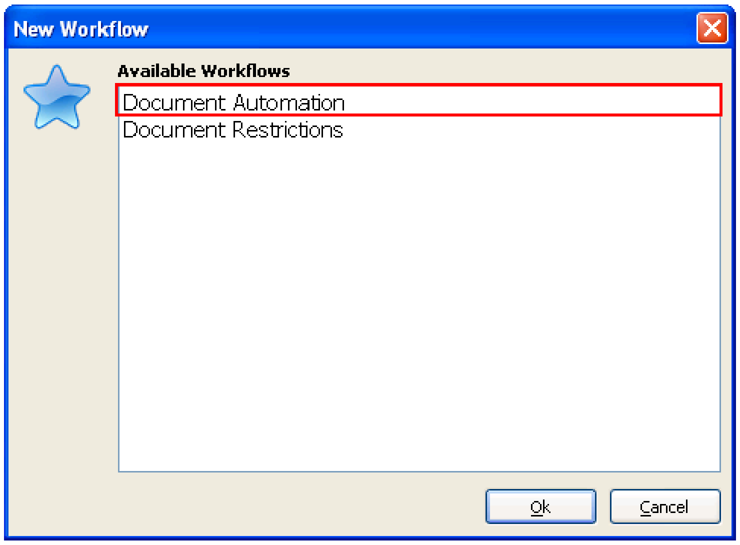
Now that we have our form, we can begin to fulfill the desired workflow using the **Workflow Designer**. First, the Workflow Designer must be started. Go to:

Start > Programs > Reform… > Workflow Designer

5. Once open, we must go to File > New…

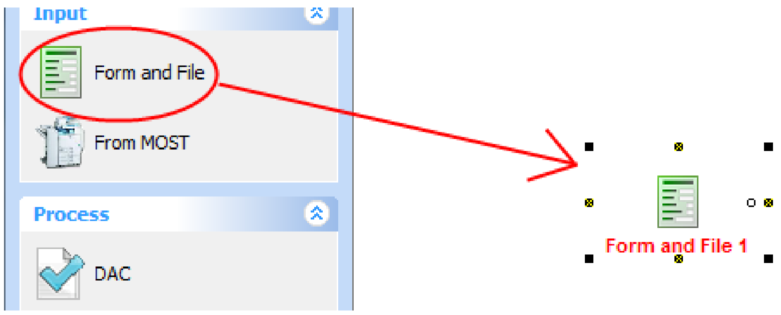


6. Select "**Document Automation**" and click **Ok**.

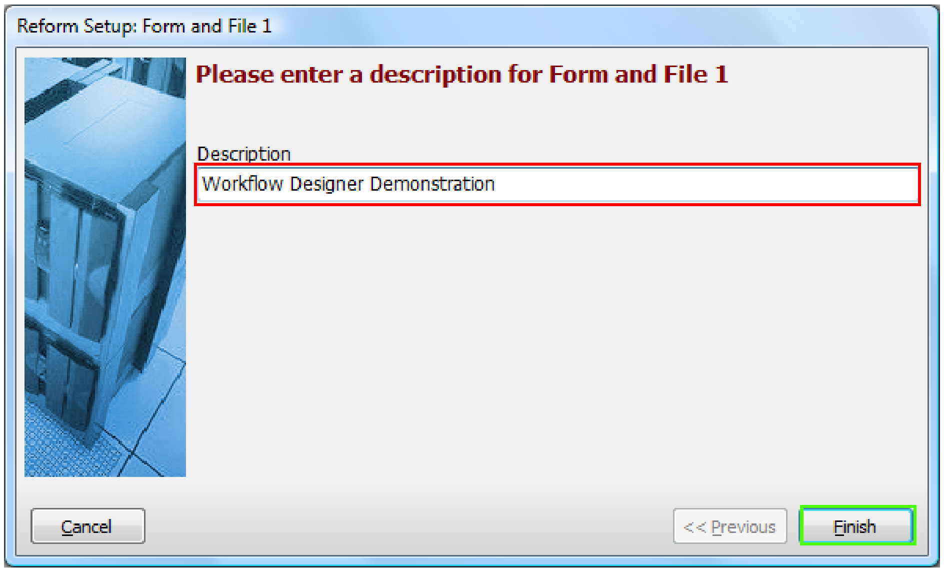


***Note:** You will now notice that your Inputs, Processes and Outputs have been populated by the various Plug-ins you have installed.

7. First, drag and drop a **Form and File** object onto the workflow by clicking and holding your mouse on **Form and File** and then moving your mouse over to the workflow and releasing the mouse button.



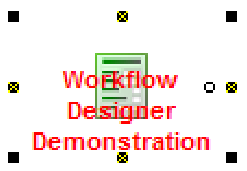
*You will be prompted with the following dialog:



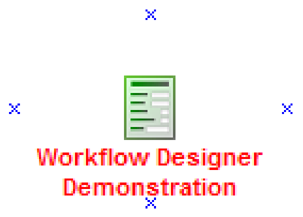
Please enter a description for Form and File 1

Description: Workflow Designer Demonstration

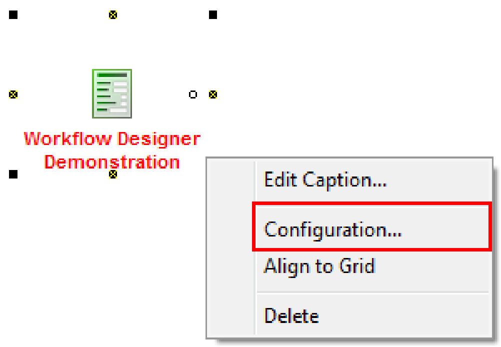
8. Enter a description for the object you just added. This is done by typing in the provided box. Type **"Workflow Designer Demonstration"** and click **Finish**.



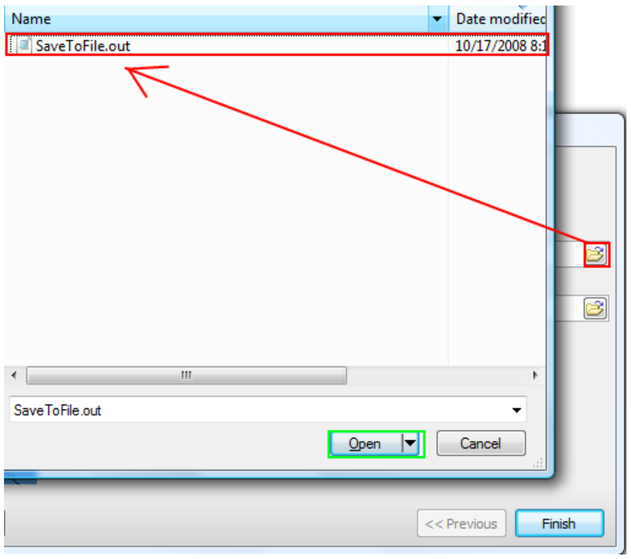
9. Since "Workflow Designer Demonstration" text does not fit properly into the object because of its length, we can resize the object to make it more readable. This is done by first select the object by clicking on it so that it looks like above and then using the small resize box on the bottom right to pull down and make the object larger. It should look something like this:



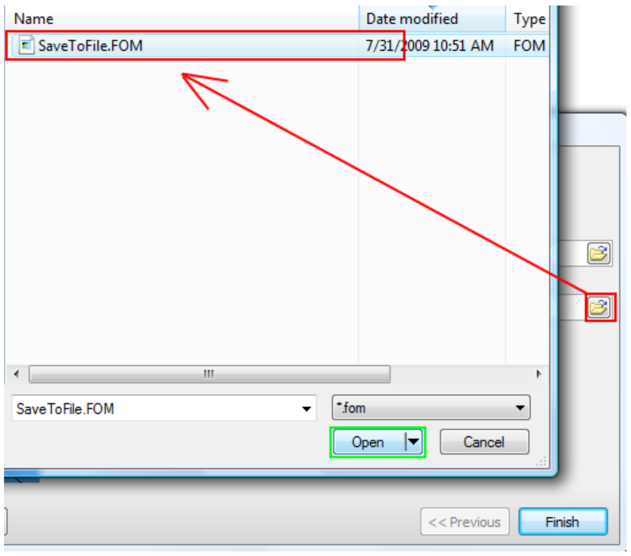
10. Now we are ready to configure the "Workflow Designer Demonstration" form and input file. To do so, right click on the newly added object and select **"Configuration…"** or simply double-click on it.



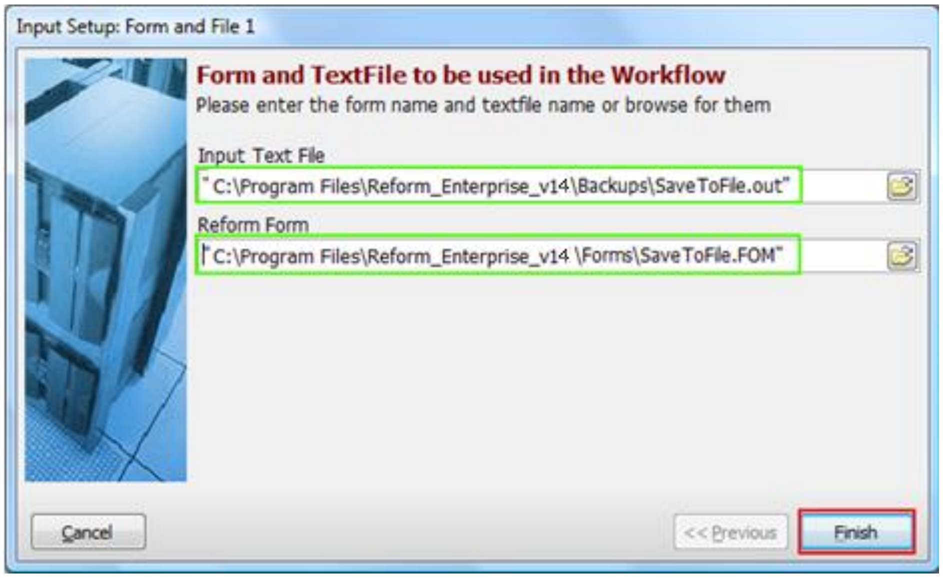
11. You are now presented with the Input Setup window. First, we want to select the TextFile or data stream that our form will be using. To do so, you can simply begin typing in the box, or you can browse to find it. We are interested in the **SaveToFile.out** file. Click the small folder icon to bring up a window which will let us choose that file. Select the file and click **Open**.



12. After choosing the text file, you are taken back to the Input Setup page so that you can now select a form. Click the small folder icon to bring up a window which will let us choose the desired **SaveToFile.FOM** form file. Select the FOM file and click **Open**.



13. Now that you have both your Input Text File and Reform Form selected, the setup of “Form and File” is complete. You can quickly double check that the correct files are selected and then click **Finish**.



Form and TextFile to be used in the Workflow

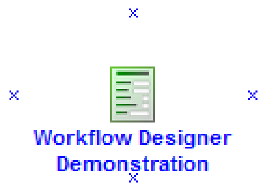
Please enter the form name and textfile name or browse for them

Input Text File: "C:\Program Files\Reform_Enterprise_v14\Backups\SaveToFile.out"

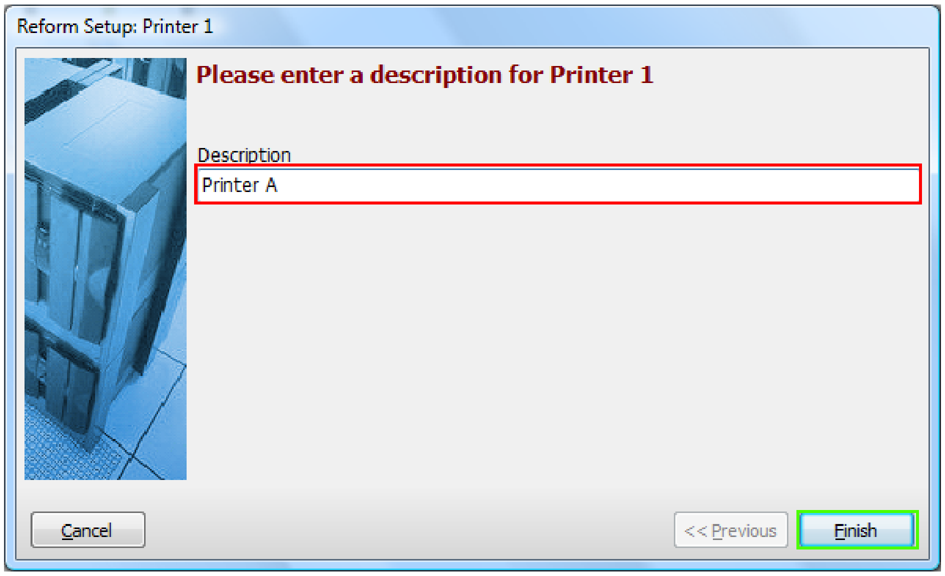
Reform Form: "C:\Program Files\Reform_Enterprise_v14 \Forms\SaveToFile.FOM"

14. You will also be prompted to add a page-break field to the forms. At this point, we want to select **No.**

Once configured, the "Workflow Designer Demonstration" text should have changed to a **blue** color signifying that setup has been run. Now your Input object is ready for an Output.



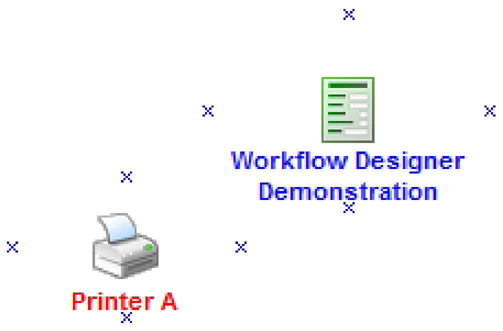
15. Next we want to add a printer to our workflow. We use the same drag and drop procedure as with the **Form and File** (see step 7). Take your mouse and drag the **"Printer"** object from under Outputs onto the workflow somewhere below your Input. Once you release your mouse, you will be prompted to describe **"Printer"**. Type "**Printer A**" in the box and then click **Finish**.



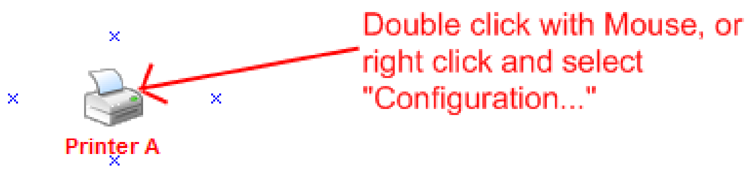
Please enter a description for Printer 1

Description: Printer A

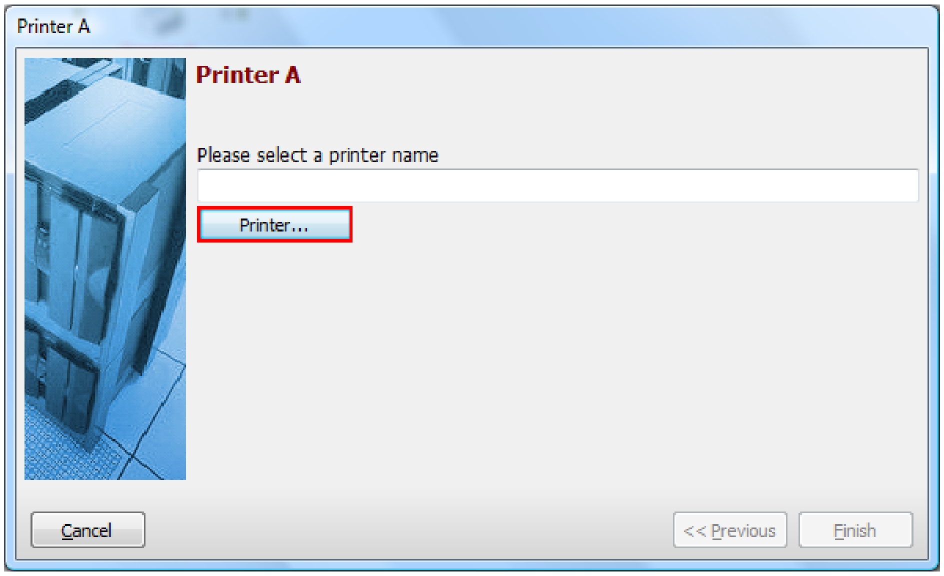
*Once you click **Finish**, you are taken back to your workflow. Your Printer will have a **Red** Caption because it is not yet configured. You can configure a Printer without making a connection with an Input or Process, whereas other outputs need to be connected before configuring them.



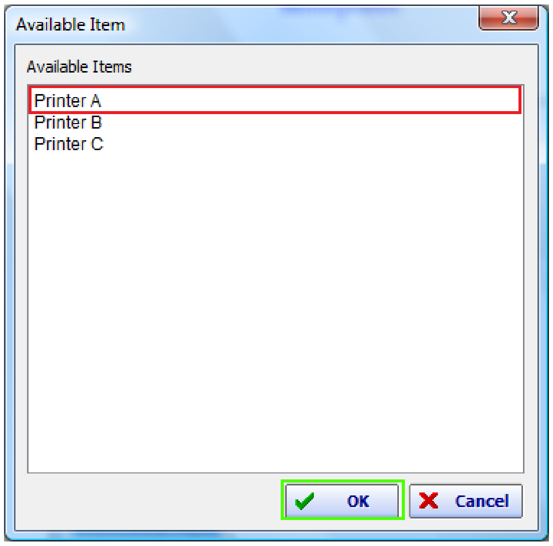
16. We will now configure "**Printer A**". Double click on the **"Printer A"** object to launch the setup screen. You can also right-click and select **"Configuration".**


Double click with Mouse, or right click and select "Configuration..."
Printer A

*You are presented with the following dialog:

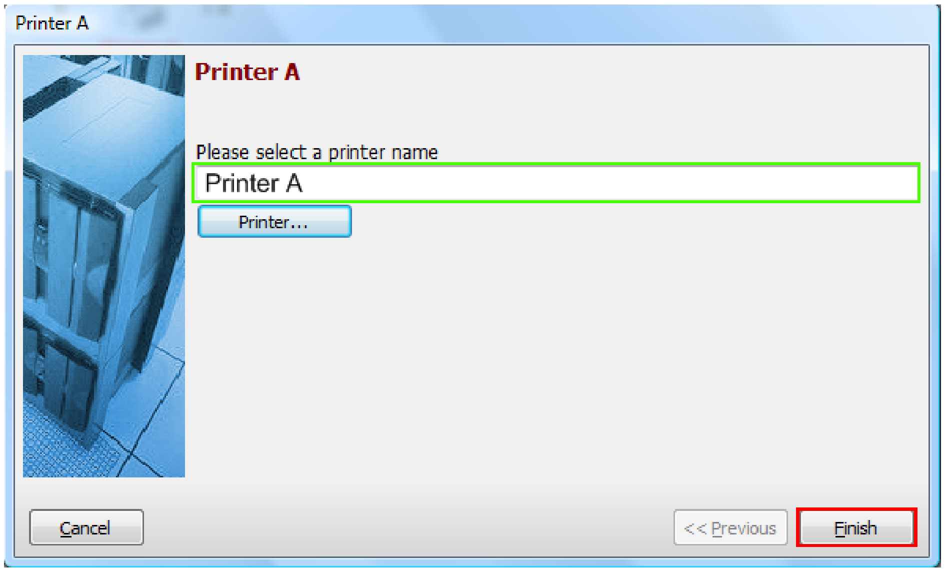


17. To select a printer, click the **“Printer…”** button. A list of printers will be available.

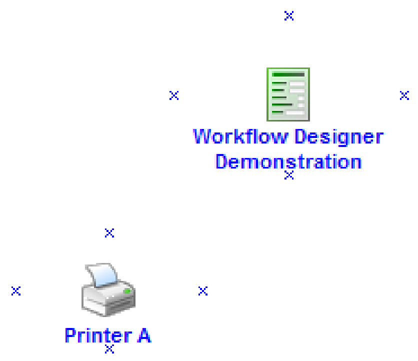


18. Select your desired Printer. For the demonstration, we selected **“Printer A”.** You may select any available printer. After you have your selection, click the **“OK”** button.

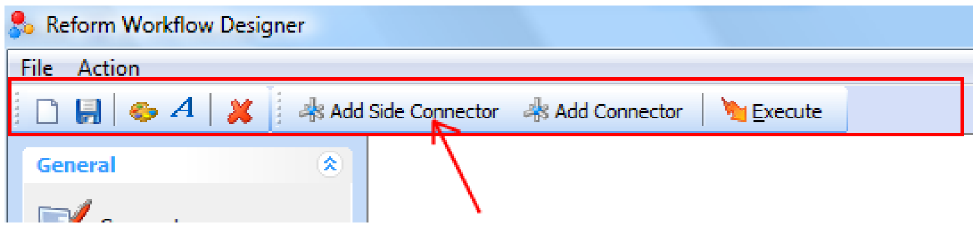
19. You are now taken back to your **“Printer A”** Setup dialog. You can confirm your selection in the box. If you need to make a change, simply hit the **“Printer…”** button once again. When verified, click **Finish.**



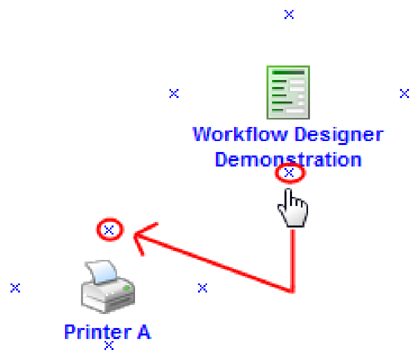
*This takes you back to your workflow. You can now see that the **"Printer A"** caption is **blue** since it has been configured.



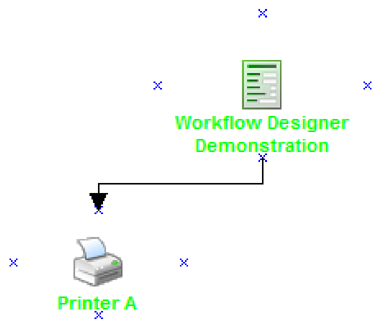
20. Now we are ready to connect the Input: **"Workflow Designer Demonstration"** to the Output: "**Printer A**". We need to choose either a **Connector** or a **Side Connector**; either can be used to connect the two. We will use a **Side Connector**. To use a connector on the screen, you first need to select it. To select, use your mouse to click the appropriate button in the menu bar on top.



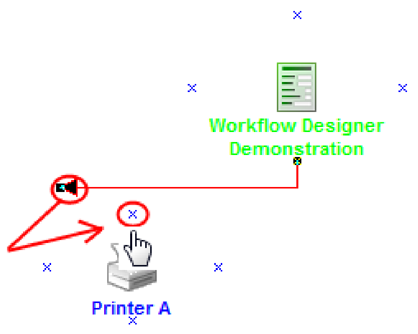
21. After you click **“Add Side Connector”**, you are ready to make the link between your two objects, “**Printer A**” and the “**Workflow Designer Demonstration**”. To do so, point your mouse at the small **x** right below “**Workflow Designer Demonstration**” and your mouse pointer will turn into a small hand as seen below. Once you see the hand, click and drag the mouse pointer to the small **x** right above the **Printer** object and your mouse pointer will once again turn into a hand icon. At that point, you will see a preview of the connector that is being made and you can release your mouse button. Don’t worry if you missed either end of the connector, you can always move it around.



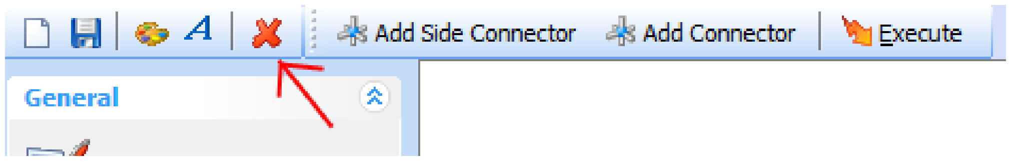
*If done correctly, both the captions will turn **green** as seen below. You can skip to step **19**.



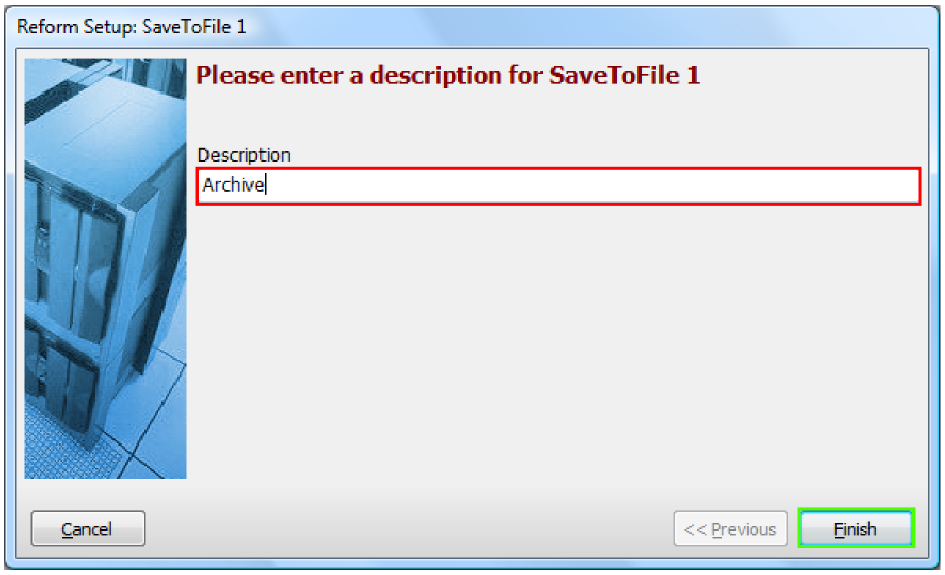
*If done incorrectly, you might get something like below. As you can see, the **Printer** caption does not turn green. You can still drag the end of the connector into “**Printer A”**. This is done by dragging the end of the arrow to the small **x** above the **Printer**. When you see the small hand icon, release your mouse button. Both are **green** again.



*** To delete an arrow or object from the form, you can use the Red X in the menu bar or right click on it and select Delete.**



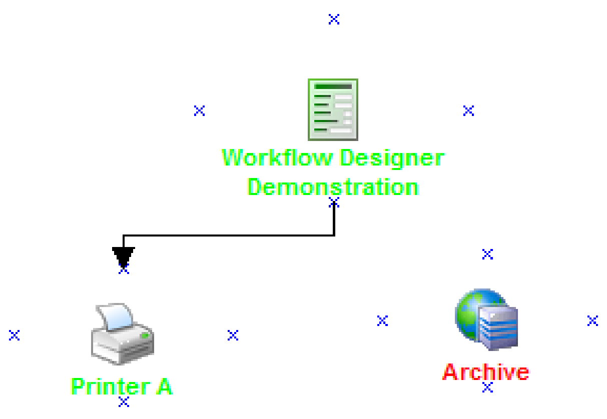
22. Now that our first connection is complete, our next task is to add a **“SaveToFile”** output onto our workflow. First we need to drag the **“SaveToFile”** object onto our workflow. This is done the same way as step 7. Click and hold your mouse over “**SaveToFile**” and drag it somewhere to the right of “**Printer A**”. Once you release your mouse button, you will be prompted to describe the **“SaveToFile”** output. Type “Archive” in the box and click **Finish**.


Reform Setup: SaveToFile 1

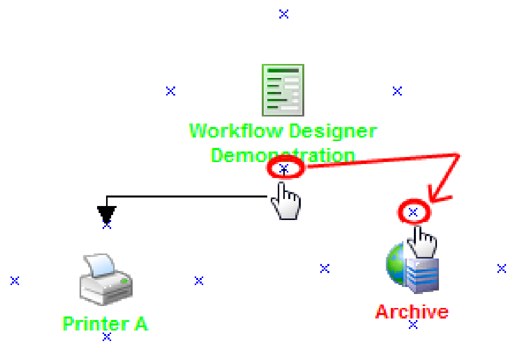
Please enter a description for SaveToFile 1

Description: Archive

***Once you click **Finish,** your workflow should look similar to below. You will notice a **red** caption for "**Archive**" because it is not yet configured.

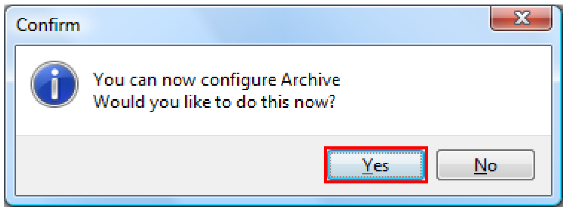


23. We need to configure "**Archive**". To do so, we first must make a connection between "**Workflow Designer Demonstration**" and "**Archive**". First choose "Add Side Connector" from the menu bar as in step **17**. Next, put your mouse over the small **x** below "Workflow Designer Demonstration" and you will notice a small hand icon. Then click and drag your mouse pointer to the small **x** on top of "**Archive**". You will once again notice a small hand icon and you can release your mouse.



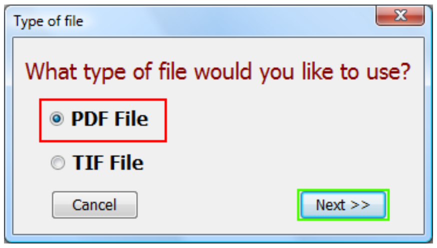
*If you do not make the connection on the first try, do not worry. Follow the second asterisk in step **18** to correct this.

24. If done correctly, you will be presented with a dialog box upon completing the connection. Click **Yes** to configure the Output you just connected.


Confirm

You can now configure Archive
Would you like to do this now?

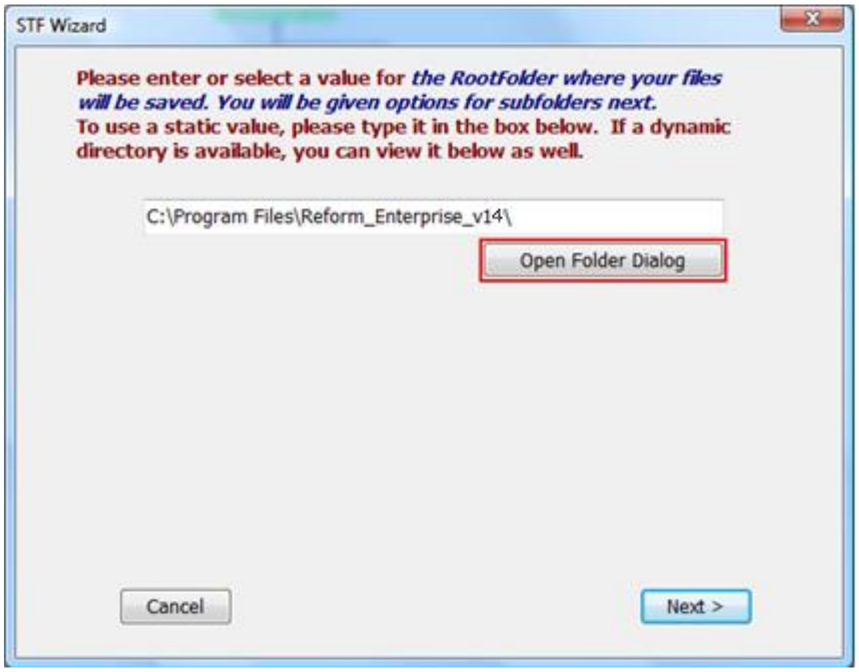
25. The setup for the **SaveToFile** output "**Archive**" begins. The first prompt is for the file type you want to save as. You can select either, but for demonstration, select PDF and click **Next >>**


Type of file

What type of file would you like to use?

[✓ PDF File]
TIF File

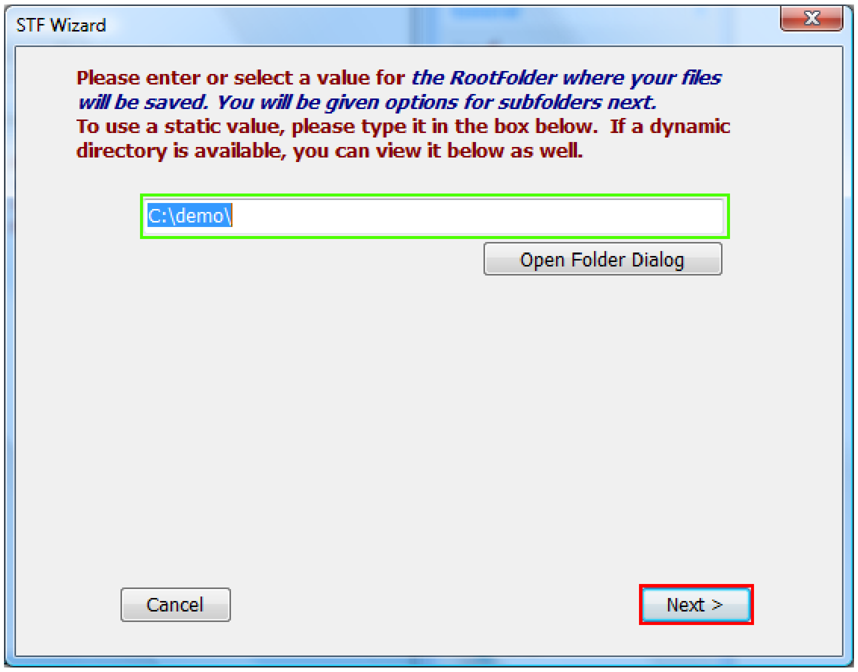
26. The setup wizard will now walk you through the configuration of your **SaveToFile** output. The first piece of information is the **Root Folder**. You should create a folder called **demo** on your C:\ drive for demonstration purposes (You can select a different directory if you prefer). Next, click **Open Folder Dialog** and navigate to your newly created folder: C:\demo. Click **OK** once selected. If a directory was available in your data stream, you would be able to use that as well. In our case, one is not available.


STF Wizard

Please enter or select a value for *the RootFolder where your files will be saved. You will be given options for subfolders next.*
To use a static value, please type it in the box below. If a dynamic directory is available, you can view it below as well.

C:\Program Files\Reform_Enterprise_v14\

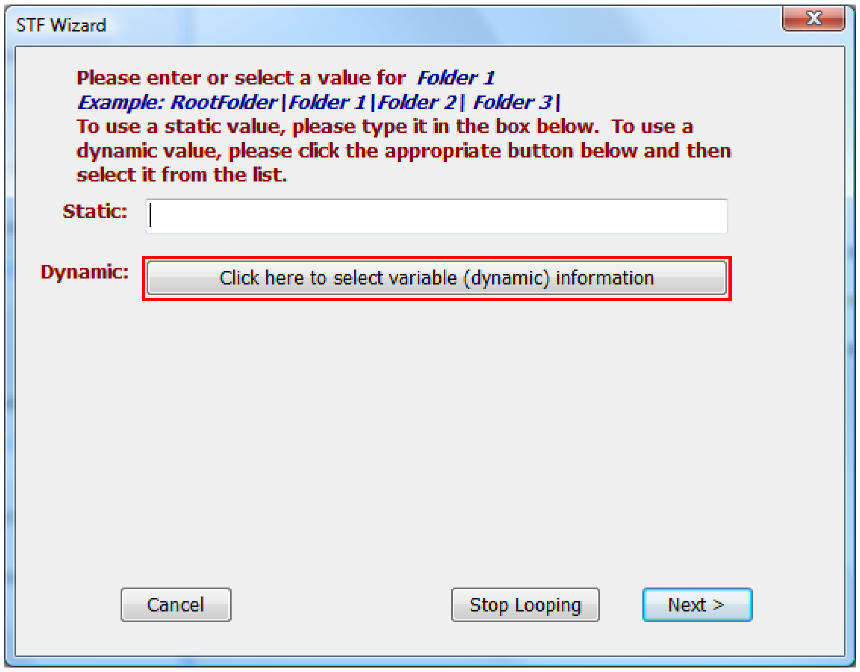
*After selecting the **demo** folder, your dialog should look like below. Verify the path in the box and click **Next.**


STF Wizard

Please enter or select a value for *the RootFolder where your files will be saved. You will be given options for subfolders next.*
To use a static value, please type it in the box below. If a dynamic directory is available, you can view it below as well.

C:\demo\

27. The next piece of information the wizard needs is the folder structure. Your root folder is **C:\demo\.** You can make subfolders in the **demo** either from static data (which you type in) or from variable data (available in the data stream). By default, you first have the option to enter static data. If you want to select variable data, you need to click the "**Click here to select variable information**" button.

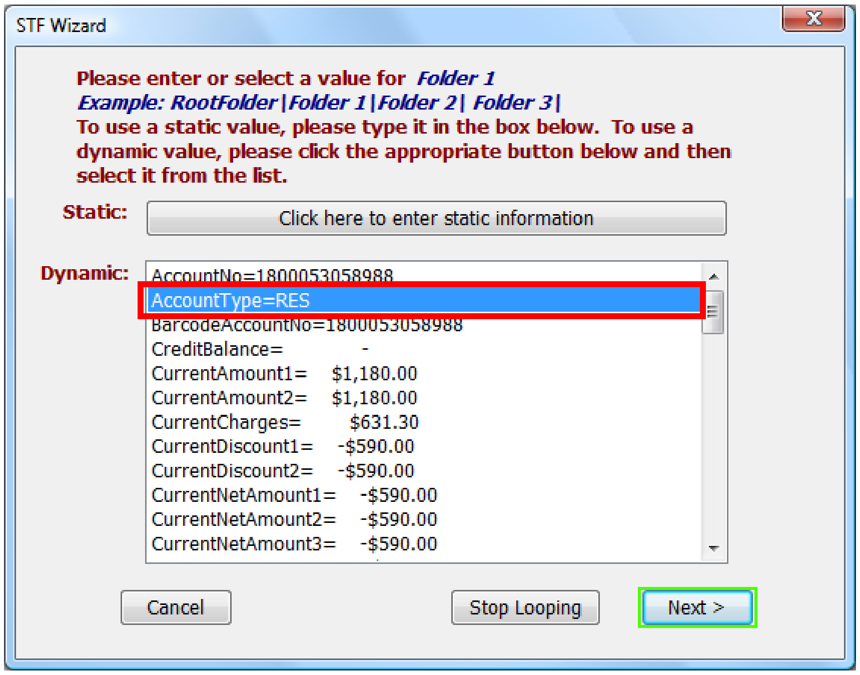

STF Wizard

Please enter or select a value for *Folder 1*
Example: RootFolder|Folder 1|Folder 2| Folder 3|
To use a static value, please type it in the box below. To use a dynamic value, please click the appropriate button below and then select it from the list.

Static:

Dynamic: Click here to select variable (dynamic) information

28. After clicking the above button, you will be presented with a new dialog which lets you select variable data. You can also go back to static data by clicking the appropriate button, but for the demonstration we want to choose variable data. We want **Folder 1** to be the **AccountType** so we select this from the list. Once selected, click **Next.**


STF Wizard

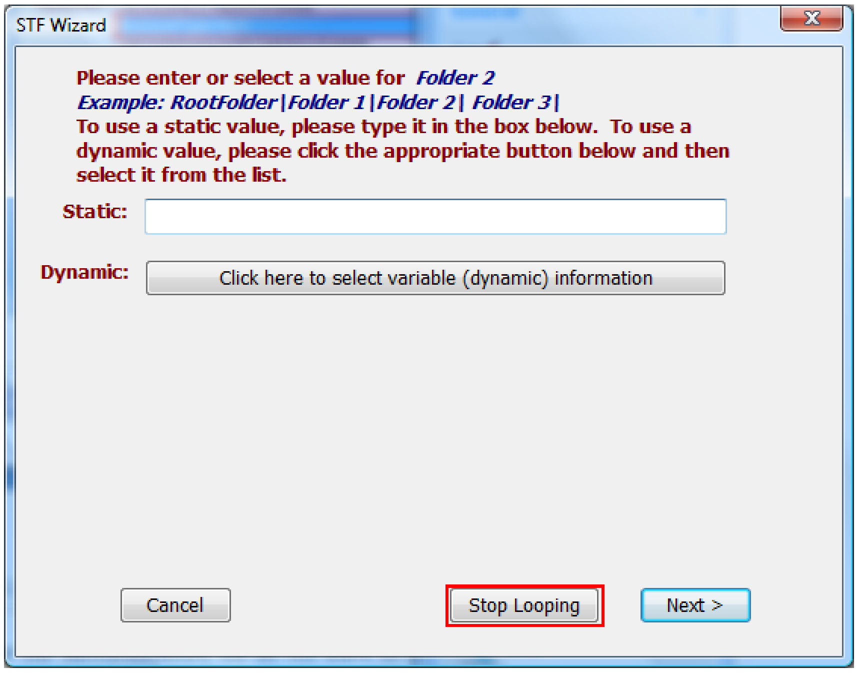
Please enter or select a value for *Folder 1*
Example: RootFolder|Folder 1|Folder 2| Folder 3|
To use a static value, please type it in the box below. To use a dynamic value, please click the appropriate button below and then select it from the list.

Static: Click here to enter static information

Dynamic:

```
AccountNo=1800053058988
AccountType=RES
BarcodeAccountNo=1800053058988
CreditBalance=          -
CurrentAmount1=    $1,180.00
CurrentAmount2=    $1,180.00
CurrentCharges=       $631.30
CurrentDiscount1=    -$590.00
CurrentDiscount2=    -$590.00
CurrentNetAmount1=    -$590.00
CurrentNetAmount2=    -$590.00
CurrentNetAmount3=    -$590.00
```

29. For our demonstration, we do not want to make any more subfolders, so click **Stop Looping** at the next dialog.



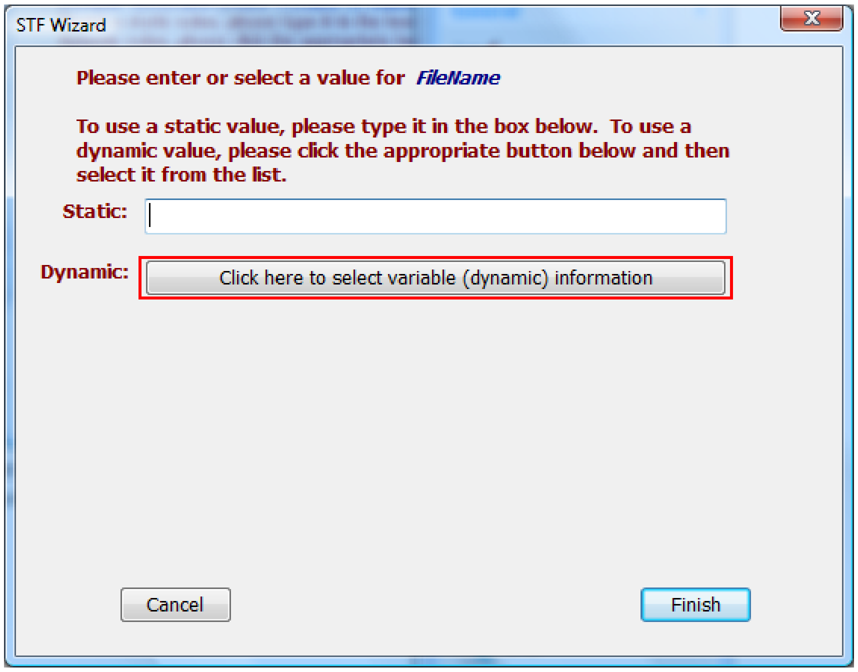
STF Wizard

Please enter or select a value for *Folder 2*
Example: RootFolder|Folder 1|Folder 2| Folder 3|
To use a static value, please type it in the box below. To use a dynamic value, please click the appropriate button below and then select it from the list.

Static:

Dynamic: Click here to select variable (dynamic) information

30. The final piece of information needed by the wizard is the filename. Once again we want this to be variable data from the data stream so we want to click the "**Click here to select variable information**" button at the following dialog.

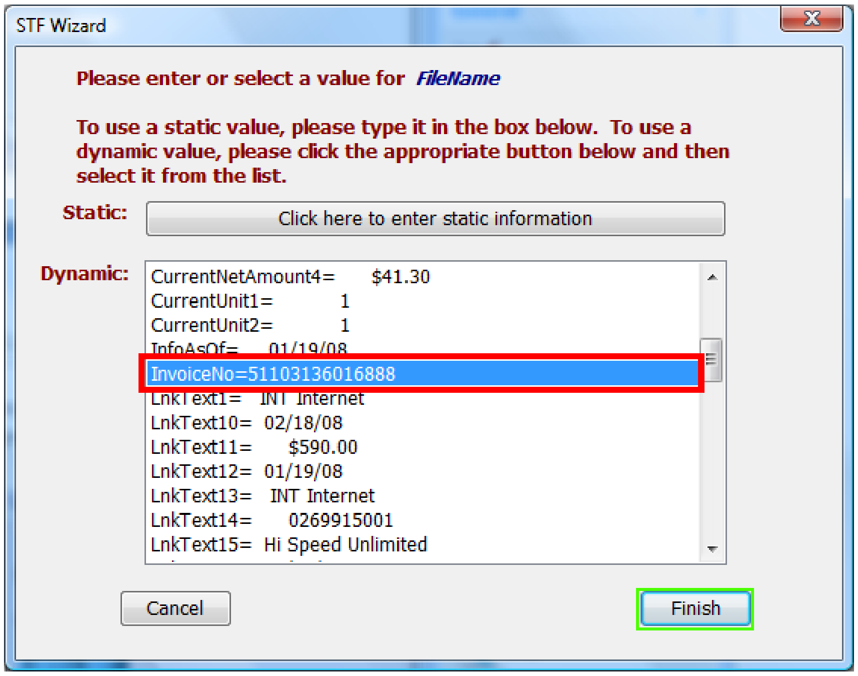

STF Wizard

Please enter or select a value for *FileName*

To use a static value, please type it in the box below. To use a dynamic value, please click the appropriate button below and then select it from the list.

Static:

31. Once clicked, you will be presented with a new dialog which allows you to select variable data. We want **FileName** to be **InvoiceNo.** Scroll down until you find this variable. Select the variable **InvoiceNo** and click **Finish.**

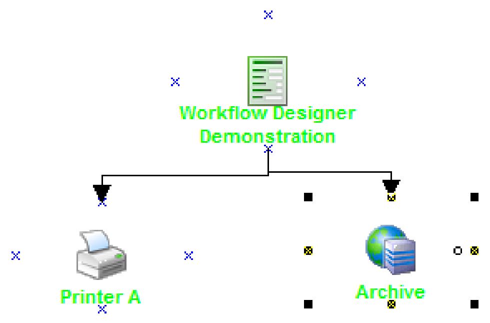

STF Wizard

Please enter or select a value for *FileName*

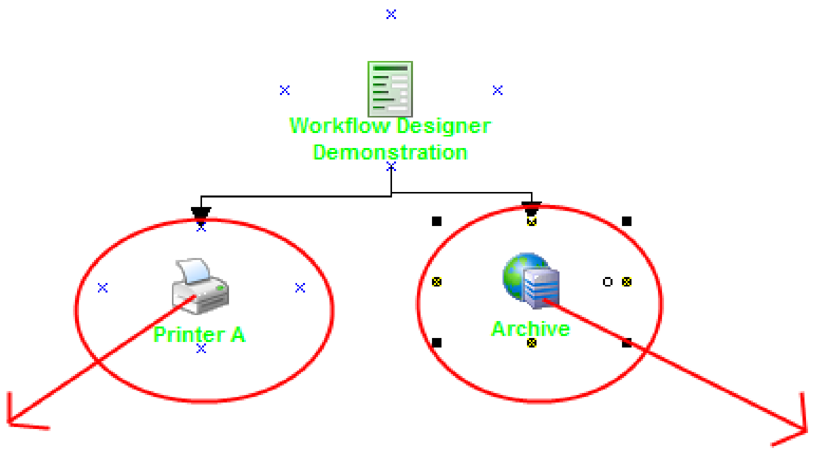
To use a static value, please type it in the box below. To use a dynamic value, please click the appropriate button below and then select it from the list.

```
CurrentNetAmount4=     $41.30
CurrentUnit1=          1
CurrentUnit2=          1
InfoAsOf=    01/19/08
InvoiceNo=51103136016888
LnkText1=  INT Internet
LnkText10=  02/18/08
LnkText11=     $590.00
LnkText12=  01/19/08
LnkText13=  INT Internet
LnkText14=     0269915001
LnkText15=  Hi Speed Unlimited
```

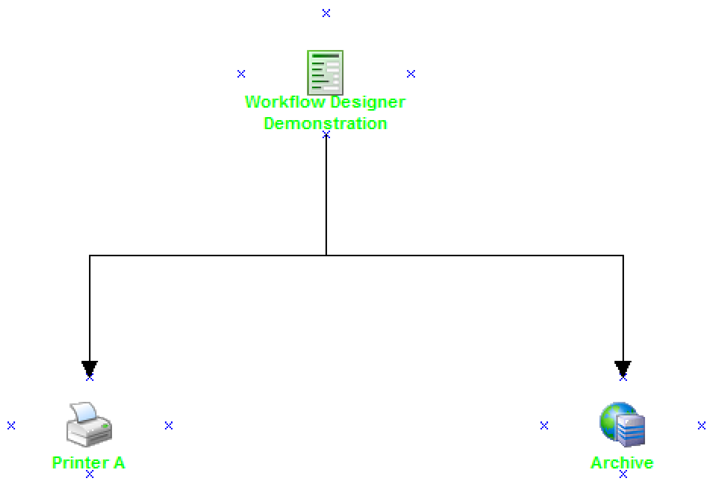
32. You have completed the configuration wizard for **SaveToFile** output "**Archive**". Your workflow should look something like below. Notice that the "**Archive**" caption is now **green** like the others.



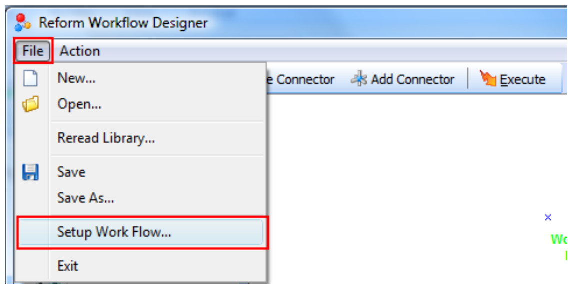
33. You can move around the objects on the workflow so your screen isn't as cluttered. To do so, simply click and hold your mouse over the objects and drag them to other portions of the screen. Your connectors will move along with the objects preserving the connections.



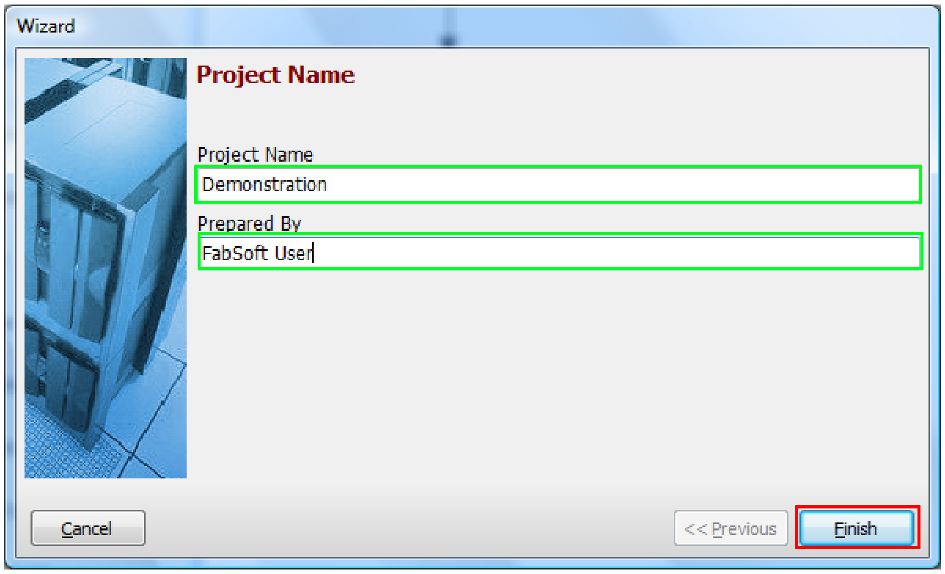
*Your final workflow design should look something like:



34. Since we are now complete, first thing we should do is author our project and save it. We will do this for the demonstration, but you simply could have skipped this and went to the execution stage. First, you should author the project. Go to **File** > **Setup Workflow**.

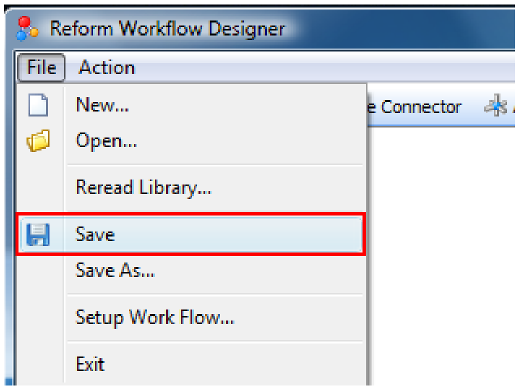


35. We are asked for the "**Project Name**" and "**Prepared By**". Type "**Demonstration**" for "Project Name" and "**FabSoft User**" for "Prepared By". Verify that your descriptions are correct and click **Finish**.

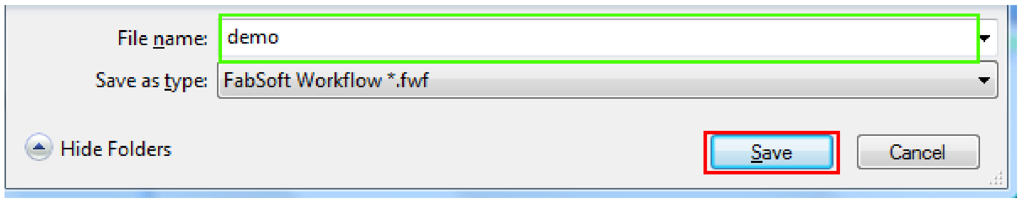


Project Name
Demonstration
Prepared By
FabSoft User

36. Now we will save the workflow so we can come back later to access it and make any needed changes. To do so, go to **File** > **Save.**



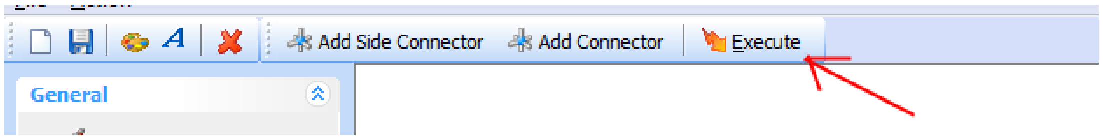
37. You may save your workflow file anywhere. For simplicity, we will save it in the **Workflow** folder of your default Workflow-Designer directory. The path should be **C:\Program Files\Reform...\Plug-ins\Workflow-Designer\**. Save it as **demo**.



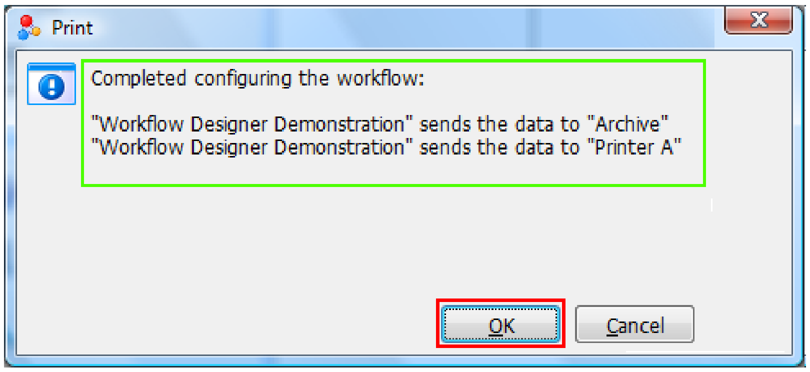
File name: demo
Save as type: FabSoft Workflow *.fwf

*You will notice that the filename will appear at the bottom left corner of the Workflow Designer.

38. Your workflow is now ready to be **Executed**. To do so, click the **Execute** button in the above menu bar.



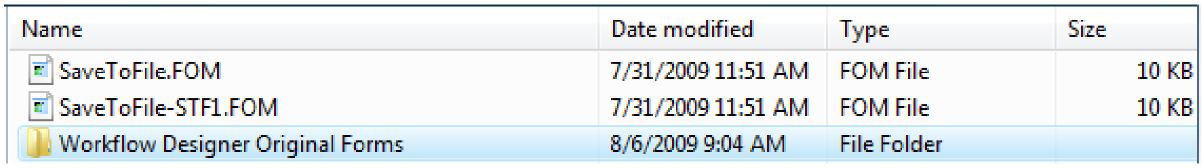
39. Execution speed will depend on each system but it should be fairly quick. Once complete, you will get a message box which displays what has taken place. Click **OK** once finished.



Completed configuring the workflow:

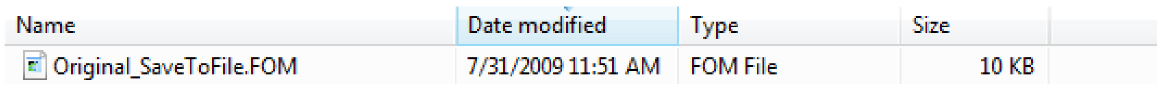
"Workflow Designer Demonstration" sends the data to "Archive"
"Workflow Designer Demonstration" sends the data to "Printer A"

40. To verify these results, navigate to your forms directory (usually **C:\Program Files\Reform… \Forms\).** You should notice three things:
    1) Your **SaveToFile.FOM** has been modified. This form now has all the properties needed to send your data to "**Printer A**".
    2) There is a new file called **SaveToFile-STF1.FOM**. This is the form which has all the properties to save your file to the previously specified location. There is also a form link specified in **SaveToFile.FOM** that links **SaveToFile.FOM** to **SaveToFile-STF1.FOM** so that both processes are carried out.



Name	Date modified	Type	Size
SaveToFile.FOM	7/31/2009 11:51 AM	FOM File	10 KB
SaveToFile-STF1.FOM	7/31/2009 11:51 AM	FOM File	10 KB
Workflow Designer Original Forms	8/6/2009 9:04 AM	File Folder	

3) Your **SaveToFile.FOM** has been backed up to the **"Workflow Designer Original Forms"** folder with the name **Original_SaveToFile.FOM**

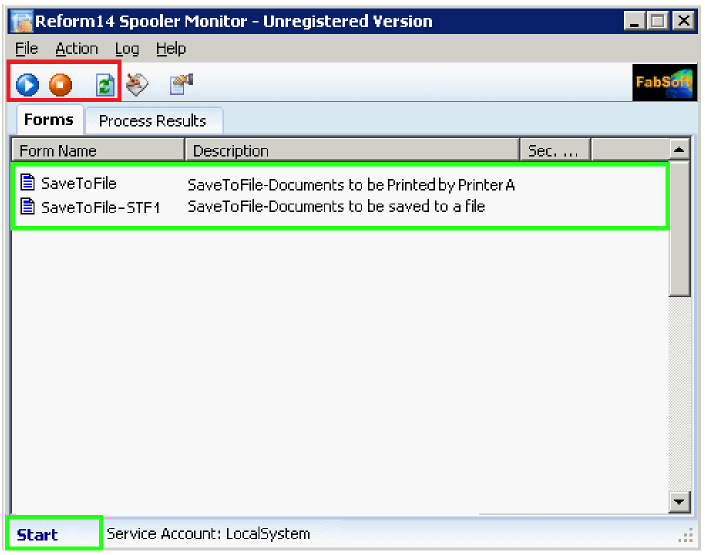


Name	Date modified	Type	Size
Original_SaveToFile.FOM	7/31/2009 11:51 AM	FOM File	10 KB

This is the form used throughout the form creation process so that each output device has a clean slate. If any changes need to be made to subsequent forms, they should be made to this original form as it is used as a template in the form creation process. Any changes made to this original form will be reflected the next time the workflow is executed. Another approach is to "cut" this file to your forms directory and rename it to SaveToFile.FOM overwriting any previous file. Then changes can be made to this form and it will be backed up in the same folder upon the next execution.

***Changes can be verified by opening each form in VDP Designer and viewing the properties and page setup.**

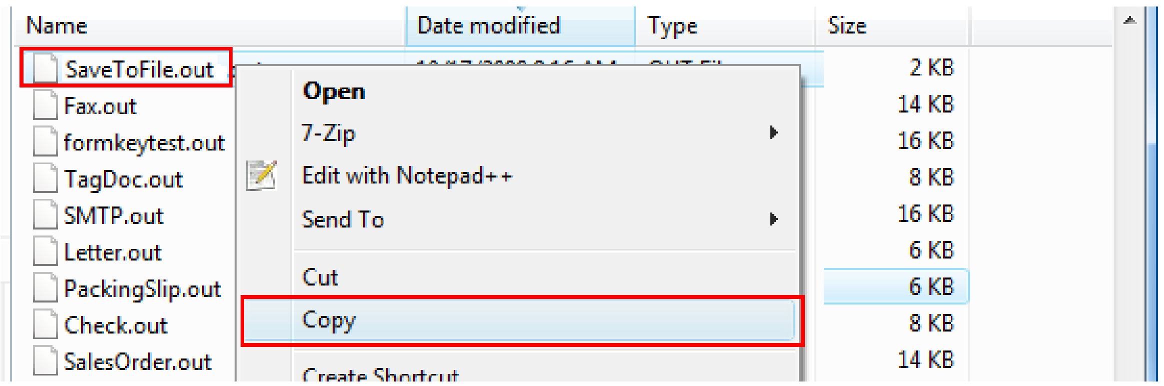
41. We want to send the sample data through our workflow now to verify the results. The Spooler Application should now be started by clicking **Start > Programs > Reform… > Spooler**. Once it is open, press the Start button. It should now say Start in the lower left corner of the Spooler window.
    a. If the Spooler was already running and started, it will need to be refreshed so that it recognizes the new forms that we just created with the wizard. The safest bet is to stop the spooler, refresh it and then start it again. Verify that your forms are listed after this procedure.



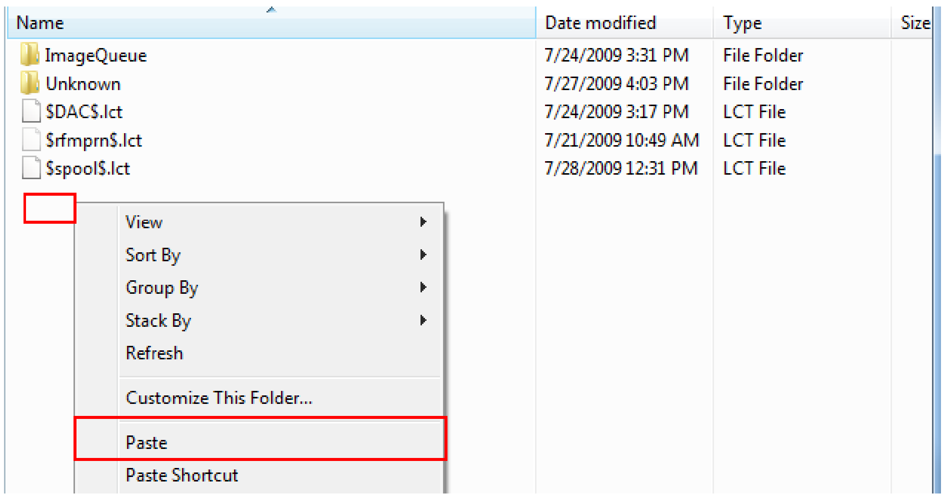
Form Name	Description	Sec. ...
SaveToFile	SaveToFile-Documents to be Printed by Printer A	
SaveToFile-STF1	SaveToFile-Documents to be saved to a file	

Start
Service Account: LocalSystem

42. Once the spooler has the correct forms, we can simulate a data stream by queuing up a file in the **Spooler** directory. This directory is usually located at: **C:\Program Files\Reform…\Spooler**. Navigate to your **Backups** location where your **SaveToFile.out** file is located (**C:\Program Files\Reform…\Backups**). Find the **SaveToFile.out** file and right-click on it and select **Copy**.

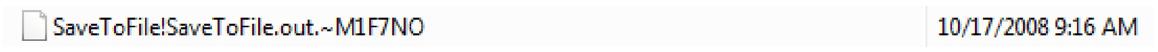


43. Next we want to paste that file in the spooler directory. Navigate to the **Spooler** directory. Right click in the white space and then select **Paste.**



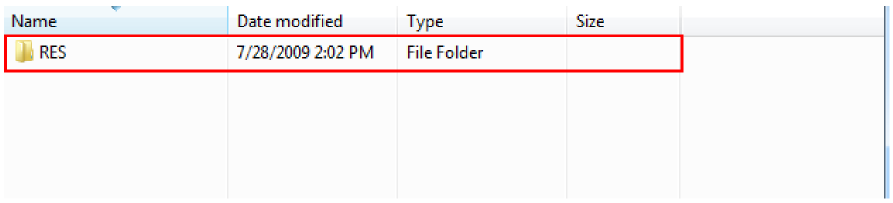
Name	Date modified	Type
ImageQueue	7/24/2009 3:31 PM	File Folder
Unknown	7/27/2009 4:03 PM	File Folder
DAC.lct	7/24/2009 3:17 PM	LCT File
$rfmprn$.lct	7/21/2009 10:49 AM	LCT File
$spool$.lct	7/28/2009 12:31 PM	LCT File

44. When the Spooler Application begins processing the file, you will notice the file name change to something similar to the image below. It shows that the file is being processed by the SaveTofile form.



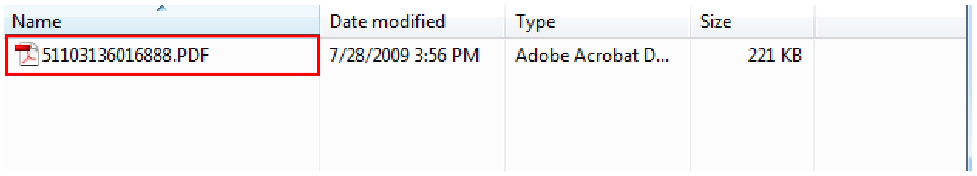
SaveToFile!SaveToFile.out.~M1F7NO 10/17/2008 9:16 AM

45. Our output (2 pages) should be in two locations: at the printer we had selected and a PDF in the directory we selected before (**C:\demo\**). Navigate to the **C:\demo\** folder. You will see the folder named RES because that is the subfolder structure we chose (**AccountType** was the variable). If you do not see the desired output, please make sure the service account for the Reform Spooler has enough permission to the selected directory. Please see the Reform Spooler section of this manual on how change the logon account.



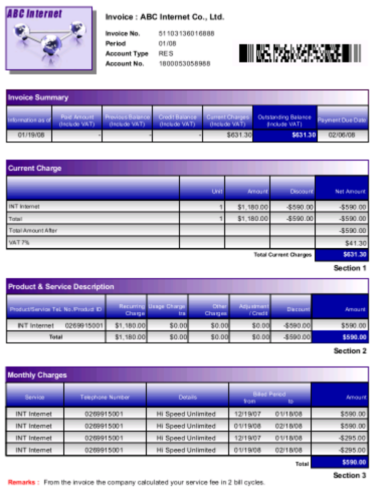
Name	Date modified	Type	Size
RES	7/28/2009 2:02 PM	File Folder	

46. Double click on **RES** and then you will see your 2 page PDF file. If we had turned on page-breaking in one of the previous prompts, and selected the **InvoiceNo** variable, then we would have ended up with 2 one-paged PDF files. Same would be the case with the Printer (queue would have been broken up into 2). Confirm the contents of this file by opening it and examining the data.



Name	Date modified	Type	Size
51103136016888.PDF	7/28/2009 3:56 PM	Adobe Acrobat D...	221 KB

47. Below is the PDF file that Reform has generated. At the same time, a page should print to the printer you selected in the wizard.



ABC Internet

Invoice : ABC Internet Co., Ltd.

Invoice No.	51103136016888
Period	01/08
Account Type	RES
Account No.	1800053058988

Invoice Summary

Information as of	Paid Amount (Include VAT)	Previous Balance (Include VAT)	Credit Balance (Include VAT)	Current Charges (Include VAT)	Outstanding Balance (Include VAT)	Payment Due Date
01/19/08	-	-	-	$631.30	$631.30	02/06/08

Current Charge

	Unit	Amount	Discount	Net Amount
INT Internet	1	$1,180.00	-$590.00	-$590.00
Total	1	$1,180.00	-$590.00	-$590.00
Total Amount After				-$590.00
VAT 7%				$41.30
			Total Current Charges	$631.30

Section 1

Product & Service Description

Product/Service	Tel. No./Product ID	Recurring Charge	Usage Charge tra	Other Charges	Adjustment / Credit	Discount	Amount
INT Internet	0269915001	$1,180.00	$0.00	$0.00	$0.00	-$590.00	$590.00
Total		$1,180.00	$0.00	$0.00	$0.00	-$590.00	$590.00

Section 2

Monthly Charges

Service	Telephone Number	Details	Billed Period from	to	Amount
INT Internet	0269915001	Hi Speed Unlimited	12/19/07	01/18/08	$590.00
INT Internet	0269915001	Hi Speed Unlimited	01/19/08	02/18/08	$590.00
INT Internet	0269915001	Hi Speed Unlimited	12/19/07	01/18/08	-$295.00
INT Internet	0269915001	Hi Speed Unlimited	01/19/08	02/18/08	-$295.00
				Total	$590.00

Section 3

Remarks : From the invoice the company calculated your service fee in 2 bill cycles.

48. Once the document is printed, and appears in the directory you selected, the demonstration is over. You can re-print the document by copying the file from the backup directory to the spooler directory again. You can also create another new form and run through the Workflow Designer again.

The Designer should *never* be used through terminal services (such as Windows Remote Desktop) or through a Citrix environment.

Advanced Features in Reform

Decision Maker

Decision Maker Introduction

Available in **Reform Enterprise/PDC** only.

The Decision Maker section presumes that you have an understanding of form overlay design. If you have not yet learned how to create and configure a form overlay, you must first learn how to use VDP Designer to design form overlays. This information can be found in the **VDP Designer** and **Output Devices** sections of this user manual.

Decision Maker is a powerful tool used to give you more flexibility and control over the output of a print-job based on information contained in the Information produced by your system. Using Decision Maker, it is possible to direct Reform to use different form overlays each time the Spooler encounters the same information file; this is done by supplying it with logical conditions and telling it what to do if the contents of the print-stream meet those conditions. Because of the open-endedness of the configuration process that goes into using Decision Maker, the complexity of the logical conditions that you supply it with is limitless, as are the destinations and configurations of your final output.

How Decision Maker Works

For Decision Maker to work, an intermediary form overlay must be created. This form overlay reads the contents of the print-stream and sends it to a script file that it pulls from the Reform…\SendToScripts\ directory. The script file contains the logical conditions and commands that manipulate the information in the text file. The information is then manipulated according to the instructions in the script file, resulting in any number of possible outcomes.

The most popular use of Decision Maker makes use of the ability to turn one print-stream into many ASCII/Text files of different names and write them back into the Reform…\Spooler\ directory to be paired with form overlays that match the names of the new files. This allows for a more dynamic output process than the use of Form Link. Form Link provides you with a static method of routing documents to different output devices that can only be changed by altering the form overlay's settings in VDP Designer. Decision Maker, on the other hand, allows you to create a number of forms that might or might not be used depending on conditions that you define during the design process. In this way, you don't have to manually change settings in a form overlay, and any number of form overlays may be used at any time, in any order.

To clarify, let's assume that your system sends a single file to the Spooler directory that contains a large number of purchase orders, and you want Decision Maker to determine the least expensive method of distributing them. The Decision Maker script contains instructions to analyze the text file and search it for an email address. If it doesn't find an email address, you've instructed it to look for a fax number. If it finds neither of those, then you've instructed it to print the purchase order in your mail room to be mailed to your vendor.

A more detailed example is as follows: Suppose your system generates an ASCII/Text file called "PurchaseOrders.out". When the file appears in the Spooler directory, the Spooler matches it with a form overlay called "PurchaseOrders.FOM"; this form overlay sends the information from the file into the Decision Maker script file, which searches the first purchase order for an email address. If it finds an email address for the first purchase order, it extracts all pages associated with that order and sends them back to the Spooler directory as a new file called "PurchaseOrderEmail.001". This new ASCII/Text file is matched with a form overlay of the same name(PurchaseOrderEmail.FOM), which has been configured to send it out via email. Decision Maker then continues for each consecutive order that it encounters in the text file and processes them individually, according to the same set of rules, generating new ASCII/Text files called "PurchaseOrderFax.001" or "PurchaseOrderPrint.001" if the necessary conditions are met. If it needs to generate many text files of the same name, it simply increments the file extension each time, generating files with extensions '.002', '.003' and so on.

In the hypothetical situation above, a limitless amount of purchase orders can be sent to as many customers as necessary, and Decision Maker will always make sure they go out via the least expensive distribution method. This minimizes the cost of sending the purchase orders, as well as the labor normally associated with their allocation, while maintaining a centralized system of control over their distribution.

How to Use Decision Maker

Decision Maker makes use of FabSoft's ShortCut technology. Some previous programming knowledge is required to write ShortCut scripts, and it is recommended that you have at least a rudimentary understanding of basic programming. You can use the sample script file provided by FabSoft as a template by which to configure Decision Maker for your purposes. The provided sample script is easy to understand for somebody with a basic understanding of programming fundamentals. FabSoft's trained Professional Service Consultants can also help you in creating scripts designed to meet your specific needs.

There are three steps to using Decision Maker. First, an intermediate form overlay must be created and set to call the Decision Maker script. Second, a script file must be created containing the logic necessary to manipulate the information received from the intermediate form overlay. Finally, all static form overlays associated with your needs must be created and configured to send to the right devices (these are regular Reform form overlays).

Overview of the Decision Maker Form Overlays

A Decision Maker form overlay contains no graphical formatting, but only special fields like Email-To, Fax, other custom fields, or page break fields which are mapped to the text layer to gather data from it. This is because this form overlay is not the final destination of the information in the associated text file, and is merely an intermediate step that sends the information to Decision Maker for logical destination routing. It also has a Decision Maker Send-To-Script as described in the previous section as the Send-To-Device.

Overview of the Decision Maker Script File

The following information contains samples of ShortCut script code and uses language associated with basic programming. It is recommended that you have a basic understanding of programming syntax and terminology in order to fully understand this section. All potentially confusing programming terminology employed in this section can be found in the glossary located at the back of this user manual. Please refer to the glossary if you find a term that you are not familiar with.

Open the DecisionMaker.fbs file, located in the Reform…\SendToScripts\ directory, in your ShortCut Script Editor (In the VDP Designer, go to **Options>Show Script Debugger** and then **File>Open**). Note the multicolored text- the red text signifies comments, and has no effect on the functionality of the script. You can place as many comments as you want in a script file without fear of causing any problems. It is recommended that you use comments to make notes for yourself in the script while learning how to use the script.

In the following pages, we will analyze the Decision Maker script. Remember, if you encounter terminology that you are unfamiliar with, please refer to the glossary.

The following is an overview of the commands, syntax and functionality of the script. The table below explores the script in the DecisionMaker.fbs script file line by line. The script is displayed in the gray areas, followed by a summary of what it does and how to use it in the white areas.

Section #

SCRIPT
SUMMARY / DESCRIPTION

Section 1

```
Include Homedir() + "\Scripts\Common.fbs"
Global SpoolerDirectory
SpoolerDirectory = GetSpoolerLocation()
```

You do not need to worry about making alterations to these lines. These lines store the Spooler directory path in a variable called **SpoolerDirectory**.

Section 2

```
FormName = Upper(FileName("Name",GetFormName()))
```

This line declares a variable called **FormName** and stores the name of the form overlay in the variable using the GetFormName() function. The **Upper** command causes the information to be stored in uppercase.

Section 3

```
Caseof
Case FormName = "PURCHASEORDER"
Gosub PurchaseOrderProcess
EndCase
```

These lines tell the program what to do if the information contained in the FormName variable equals "PURCHASEORDER". In basic English syntax, we are telling the program, "In a case where the form name is "PURCHASEORDER", execute the commands contained in the **PurchaseOrderProcess** subroutine."

Section 4

```
End
```

This line tells the program to stop. You may be confused by its location, as it is not located at the end of the file. This can be misleading; the reason for its location is quite simple. In the Caseof block above, the Gosub command tells the program to execute the code in the PurchaseOrderProcess subroutine. After the commands in the PurchaseOrderProcess subroutine are executed, the program then Returns control to the Gosub command and continues executing from that point. It then reaches the End statement, and the program stops. By the time the End command is reached, all code above and below it has been executed.

Section 5

```
PurchaseOrderProcess:

Caseof
'Put the case in order you want them to be process, example Email will be process before Fax.
Case Trim(_EmailTo) <> "" and Trim(_EmailTo) <> "empty@emailaddress.com"
CreateSpoolerFile(FormName + "Email")
Case Trim(_FaxNumber) <> "" and Trim(_FaxNumber) <> "EMPTY"
```

```
CreateSpoolerFile(FormName + "Fax")
Otherwise
CreateSpoolerFile(FormName + "Printer")
EndCase

Return
```

This is the PurchaseOrderProcess subroutine referenced in the "Caseof" block above the End command.

Everything contained between the lines "PurchaseOrderProcess:" and "Return" constitutes the commands that are executed when the subroutine is called. This subroutine contains a block of code that tells the program, in basic English syntax, "In cases where the EmailTo field in the DecisionMakerPO.FOM form overlay is not empty and does not contain the string "empty@emailaddress.com", append "Email" to the end of FormName and send it to the CreateSpoolerFile subroutine. If the EmailTo field is empty but the FaxNumber field is not (or contains the string "EMPTY", append "Fax" to the end of FormName and send it to the CreateSpoolerFile subroutine. Otherwise, append "Printer" to the end of FormName and send it to the CreateSpoolerFile subroutine."

Section 6

```
Sub CreateSpoolerFile(SpoolerFileName)
…
…
EndSub
```

The commands between the first and last lines of this block cause the new text file to be created and returned to the Spooler directory for processing. For our purposes, only the first and last lines of the Sub block need to be examined. The name of this Subroutine is CreateSpoolerFile, and it receives information from the command that called it. The information it receives is contained between the parentheses. In section 5, the command CreateSpoolerFile(FormName + "Email") appended the word ‘Email’ to the end of the form name and sent it to the CreateSpoolerFile subroutine. SpoolerFileName represents the received information. This is how the program knows what to call the file it places in the Spooler directory for processing.

This ends the summary of the provided script file and its contents. You can alter the script in this file to better serve your purposes. There are a few rules that must be followed in order for the script to function properly:

All references to Special Fields or Label Objects that you use to map information for use in a Decision Maker script must be notated in the code with an underscore followed by the object name. For instance, in the sample form overlay there is an object called ‘DMFormName’; in the Decision Maker script file, it is notated as ‘_DMFormName’.

Decision Maker only reads code written in the ShortCut scripting language. Make sure that your code employs the proper ShortCut syntax.

You may add as many lines as you want to the DecisionMaker.fbs file, and create logic for the use of many different intermediate form overlays by simply creating different subroutines for different form names.

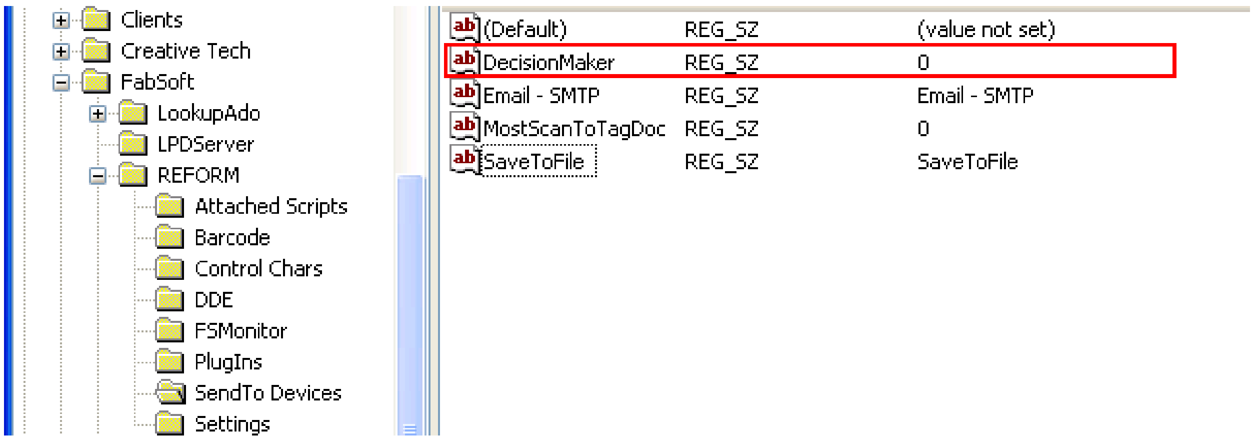
Alternately, you may save separate script files with different names and reference them as the Send-To-Device in your intermediate form overlays. To do so, you must make alterations to the Send-To-Devices in the registry. Please see **Adding Your Own Send-to-Device** section.

Chapter Summary

Having reached the end of this chapter, you should now have an understanding of the power and flexibility that you can achieve over your printed forms and their destinations through the use of Decision Maker. Decision Maker is capable of much more than we’ve had the opportunity to discuss within these pages; with a complete understanding of the ShortCut scripting language and the use of Decision Maker, the possibilities are endless. The time, labor and money saving benefits of using Decision Maker can be enormous. If you have any questions or comments regarding its use, we would be happy to hear from you. Feel free to call our Customer Service department at any time, and one of our trained professionals will be glad to hear your suggestions or answer any questions you may have.

Troubleshooting

Make sure the following entry exists in the registry location (**HKEY_LOCAL_MACHINE \SOFTWARE\FabSoft\REFORM…\SendTo Devices\)**



(Default)	REG_SZ	(value not set)
DecisionMaker	REG_SZ	0
Email - SMTP	REG_SZ	Email - SMTP
MostScanToTagDoc	REG_SZ	0
SaveToFile	REG_SZ	SaveToFile

Scripting

Page Process Script

Page Process Scripts are designed to visually alter the Design Layer before the form is printed. The script runs once for each page. For example, if a DataStream contains 10 pages the Page Script will run 10 times.

Design Layer can be altered by:

-Change object position, size

-Change object font type, size and color, justification

-Change object data

-Show and hide objects

-Change forms

To add or edit Page Scripting open VDP Designer >> File >> Form Properties >>Page-Process Script.

System variables that can be controlled in Page scripting are:

_PrinterProfile – can be used to change printer and printer setting. See section on Printer Profile.

Device Script

After a form is finished or if break fields are being used, the device script will get executed. For example if you are faxing 1000 page document and sectioning is being used and each page is for a different recipient the device script will get executed 1000 times. If sectioning is not being used the device script will only be executed once. Decision Maker is another powerful use of a device script.

Built-in system variables:

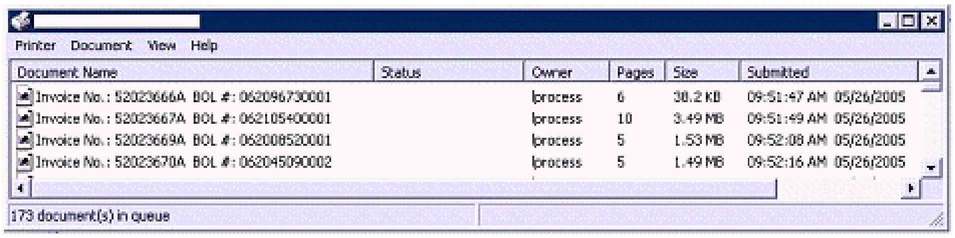
_JobStart – Contains the page # of the start of a section.

_JobEnd – Contains the page # of the end of the section.

_TextPage[<Page Number>] – The text content of the page, example _TextPage[4] would return the forth page of text.

_PageTotal – Contains the total # of pages in entire document. In the above example _PageTotal would equal 1000.

_PrinterTitle – Controls the name of the print job. For example _PrinterTitle="Invoice No.:52023666A BOL#:05209673001" would yield a print job title of "Invoice No.:52023666A BOL#:05209673001". The title of the print job will be displayed in the printer window.



Document Name	Status	Owner	Pages	Size	Submitted
Invoice No.: 52023666A BOL #: 062096730001		lprocess	6	38.2 KB	09:51:47 AM 05/26/2005
Invoice No.: 52023667A BOL #: 062105400001		lprocess	10	3.49 MB	09:51:49 AM 05/26/2005
Invoice No.: 52023669A BOL #: 062008520001		lprocess	5	1.53 MB	09:52:08 AM 05/26/2005
Invoice No.: 52023670A BOL #: 062045090002		lprocess	5	1.49 MB	09:52:16 AM 05/26/2005

173 document(s) in queue

Object information:

In order to retrieve the object information from the Text Layer user the object name and page # the syntax is:

_ObjectName[<Page Number>] for example _TotalAmount[5] would return the contents of the TotalAmount object on the fifth page.

For device scripts, if using an object that exists in the text layer, the special "Track Value" property should be turned on. To do so, select the object to display its attributes under Properties. Look for the "Track Value" option and change it to **True**.

In order to retrieve the object information from the Design Layer, simply use the object name without a page#. Since the information in the object never changes, there is no need for this extra parameter. The syntax is:

_ObjectName an example _IPAddress would return the contents of the IPAddress in the Design Layer.

The underscore tell the system that the information is being pulled from either the Text or Design Layer. A Trim statement should be used when retrieving object information this will make sure the data is clean and usable, for example:

TotalAmount = **Trim(_PageTotal[4])**

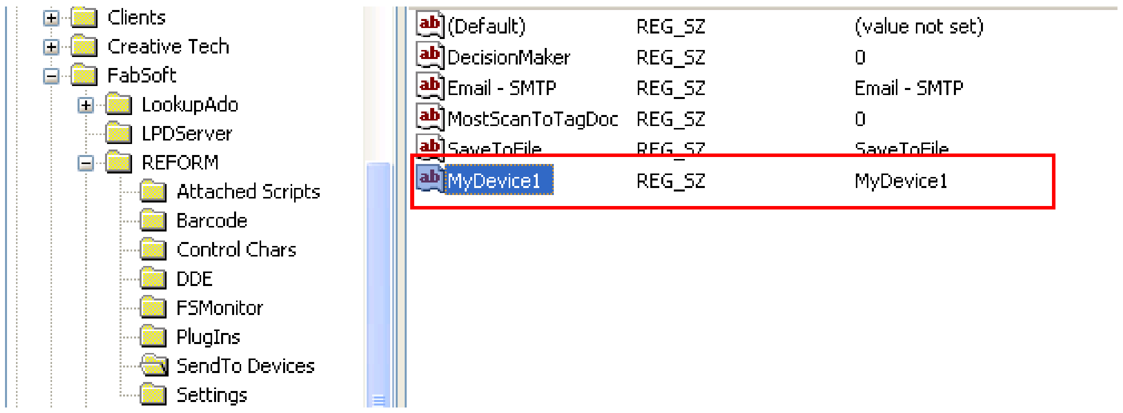
Adding Your Own Send-to-Device

Each send to device has a script associated with it, such as Email, Fax and SaveToFile. To add a new script you must edit the registry. This list is located at the following registry path: **HKEY_LOCAL_MACHINE\SOFTWARE\FabSoft\REFORM…\SendToDevices**

You will see a list of currently installed Send-To-Devices available in Reform. To add your own device, simply right-click in the whitespace below all the devices and select **New>String Value.** Name your device something meaningful, i.e. **MyDevice1**. You will also notice that some devices have a value of 0 and others have the same value as their key name. Enter the device name as the value if your device needs to produce a PDF or TIF file upon execution, otherwise enter 0. Two good examples are DecisionMaker and SaveToFile. Since Decision maker does not require an output file, it has a value of 0. SaveToFile, on the other hand, needs to create a TIF or PDF when it is processed so it can be saved. Depending on what you are going to be doing with your new **MyDevice1** send-to-device, double click on the key you just created and type in a value – either **0** or **MyDevice1**. If you set the registry value to the device name, an image (the type is specified in attachment setup) will be created in the

\Reform…\Queue directory for use inside your custom script. It is automatically deleted (cleaned up) once the script executes.

Your registry should look something like:



```
Clients
Creative Tech
FabSoft
    LookupAdo
    LPDServer
    REFORM
        Attached Scripts
        Barcode
        Control Chars
        DDE
        FSMonitor
        PlugIns
        SendTo Devices
        Settings
```

(Default)	REG_SZ	(value not set)
DecisionMaker	REG_SZ	0
Email - SMTP	REG_SZ	Email - SMTP
MostScanToTagDoc	REG_SZ	0
SaveToFile	REG_SZ	SaveToFile
MyDevice1	REG_SZ	MyDevice1

If you have VDP Designer open, close it and open it back up. Go to **File>Page Setup**. Select **SendTo…** and your newly created device will be available. You can also click the **Attachment Setup** button to specify the image type.

You also need to create a text file in the **\Reform…\SendToScripts** directory with an fbs extension, i.e. **MyDevice1.fbs**

Commands

BEEP

BEEP <type>

Generate a sound to your system. <type> corresponds to your window sound settings. The valid <type> are:

1 Windows Default sound
2 Windows Asterisk sound
3 Windows Exclamation
4 Windows Critical Stop
5 Windows Question

Example:

```
'-- Generate a windows Default sound
BEEP 1
```

CALL

CALL "<Script Name>"

This command will call other script and resume upon finish.

Example:

```
'-- wait for "Name" to appear at 1,1
'-- for 10 seconds
IF WAITDOS(1, 1, "Name", 10) = 1 THEN
CALL "c:\shortcut\scripts\amrscr.fbs"
PRINT "Back from called script"
ELSE
PRINT "Timeout Waiting for text!"
ENDIF
```

CASEOF..ENDCASE

```
CASEOF
CASE <condition>
<Your Codes>
OTHERWISE
<Your Codes>
ENDCASE
```

Executes the first statement block after CASEOF and before ENDCASE whose associated conditional statement evaluates to true.
CASE is used to execute a set of script commands based on the result of a logical condition. When CASEOF is executed, successive logical cases are evaluated; the results of the evaluations determine which set of commands is executed.
When the first met-condition CASE clause is encountered, the statements following it are executed. Execution of the statements continues until the next CASE or ENDCASE is reached. Execution then resumes with the first command following ENDCASE.
If a CASE clause does not meet the condition, the statements following it up to the next CASE clause are ignored.
One and only one CASE clause is executed - the first true CASE clause. Any succeeding true CASE clauses are ignored.
If all of the CASE clauses evaluate to false (.F.), OTHERWISE determines if any additional statements are executed:
If OTHERWISE is included, the statements following OTHERWISE are executed and execution skips to the first command
If OTHERWISE is omitted, execution skips to the first command following ENDCASE.
Comments can be placed after CASEOF and ENDCASE on the same line. The comments are ignored during program compilation and execution.
Example:

```
INPUT "What is your age?", Age
CASEOF
CASE Age < 20
PRINT "You are less than 20"
CASE Age >= 20 and Age < 40
PRINT "You are between 20 and 40"
```

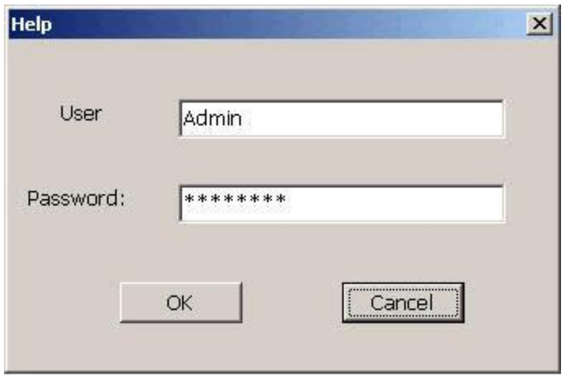
```
CASE Age >= 40 and Age < 60
PRINT "You are between 41 and 60"
CASE Age >= 60 and Age < 80
PRINT "You are between 61 and 80."
OTHERWISE
PRINT "You are over 80 years old.”
ENDCASE
```

DoDialog
DoDialog(<DialogString>)

Parameter:
<DialogString>
A string formatted similar to a Dialog script. This is mostly used for generating dialogs dynamically.

Example:

```
'-- This will display the dialog shown below
String ExampleDialog
  Dialog
  *FORM
    Dimension=525, 395, 341, 226
    Font=Tahoma, 10, , clBlack
    Color=clBtnFace
    Caption=Help
    FormPos=0
    TimeOut=0
    ActiveControl=
  *BUTT
    Name=BExit
    Dimension=204, 148, 75, 25
    Font=Tahoma, 10, , clBlack
    Caption=Cancel
    Default=0
    OnClicked=
  *BUTT
    Name=BOK
    Dimension=68, 148, 75, 25
    Font=Tahoma, 10, , clBlack
    Caption=OK
    Default=0
    OnClicked=
  *EDIT
    Name=TUser
    Dimension=104, 36, 200, 20
    Font=Tahoma, 10, , clBlack
    Value=Admin
    MaxLength=-1
    ReadOnly=0
    Password=0
    EditMask=
  *EDIT
    Name=TPass
    Dimension=104, 88, 200, 20
    Font=Tahoma, 10, , clBlack
    Value=password
    MaxLength=-1
    ReadOnly=0
    Password=1
    EditMask=
  *LABEL
    Name=LblUser
    Dimension=32, 36, 56, 20
    Font=Tahoma, 10, , clBlack
    Caption=User
    WordWrap=0
    TextAlign=0
  *LABEL
    Name=LblPass
    Dimension=12, 88, 76, 20
    Font=Tahoma, 10, , clBlack
    Caption=Password:
    WordWrap=0
    TextAlign=0
  EndDialog
EndString
DoDialog (ExampleDialog)
```



END

END

Signifies the end of the script. You can END the script at anywhere within your script. If there are subroutines in your script, make sure that the END command is include right before all the subroutine codes. This way the script will not try to execute those subroutines unintentionally.

Example:

```
count = 1
Again:
'-- Ask for a password
INPUT "Enter your password", password
'-- if password does not match JOHN then report error
IF UPPER(password) <> "JOHN" THEN
message = "Invalid Password"
GOSUB DisplayMessage
count = count + 1
'-- if too many attempt then we notify the user and
'-- end it here
IF count > 3 THEN
message = "Too many tries"
GOSUB DisplayMessage
END
ENDIF
GOTO Again
ENDIF
message = "Password verified."
GOSUB DisplayMessage
PRINT "Thank you!"
END
'-- Subroutine: Display Message on screen for 5 seconds
DisplayMessage:
PROMPT message, 5
RETURN
```

EXECUTE

EXECUTE "<Application Name>"

This command will execute other application and resume the next script line without waiting for the application to finish.

Example:

```
EXECUTE "c:\windows\notepad.exe"
FOCUS("Untitled - Notepad", 10, 0)
SEND "Hello Notepad"
```

FOR...ENDFOR

```
FOR <variable> = <begin value> TO <end value>
<your codes>
[EXITFOR]
ENDFOR
```

Executes the commands after FOR and before ENDFOR a specified number of times.

FOR.. ENDFOR executes a set of statements within a loop a specified number of times. A variable is used as a counter to specify how many times the statements inside the loop are executed.

Program statements after FOR are executed until ENDFOR is reached. The counter <variable> is then The counter is always incremented by 1. The counter is then compared with the final value <begin value>. If the counter is less than or equal to the final value <begin value>, the statements following the FOR clause are executed again. If the counter is greater than the final value <begin value>, program execution branches outside the FOR ... ENDFOR loop and continues with the first command following ENDFOR.

Caution

1. The values of <variable>, <begin value> and <end value> are read initially only. However, changing the value of the counter <variable> inside the loop affects the number of times the loop is executed.

2. Do not use GOTO to exit out of FOR.. ENDFOR loop, this may cause unexpected program execution. Use EXITFOR the exit out of FOR.. ENDFOR instead.

Example:

```
y = 0
FOR i = 1 TO 20
y = y + i
PROMPT y, 0.5
'-- Exit out if > 6
IF y > 6 THEN
EXITFOR
ENDIF
ENDFOR
PRINT "Done"
```

FUNC
FUNC <function name>(<parameters>)
<codes>
....
ENDFUNC
FUNC <function name> is a statement within a script file. It specifies the beginning of each function in a script file and defines the function name and parameters. Function names must begin with a letter or underscore and may contain any combination of letters, numbers and underscores.
The FUNC <function name> command line is followed by a series of commands that make up the function. You must include ENDFUNC as the last line of a function.
The difference between FUNC and SUB is that FUNC can return a value, but, SUB cannot. The result of a function is passed back by assigning RESULT with a value.
Example:

```
today = GetDate()
PRINT GetDate()
FUNC GetDate
'-- return todays date
RESULT = date(4)
ENDFUNC
```

To use a function, simply include the function name followed by its parameters into your codes (see example). Advance users may want to pass a parameter by reference. Passing by reference allows a function to modify the value of the parameter. To pass a parameter by reference, prefix the parameter name with &.
Example:

```
FUNC GetName(prompt, &name)
'-- name will be modified
INPUT prompt, name
ENDFUNC
```

Sub/Func Example

GLOBAL
GLOBAL <variable name>
Set the specified variable as global to make it visible throughout the program. Normally all variables are declared as private. As a result, they can only be seen by the program blocks they belong to. Keeping the variables private prevents overwriting the content of variables unintentionally. By declaring a variable as global, any blocks of codes can access the variable. Therefore, use global only if necessary to reduce programming errors.
Example 1: This example demonstrates how a private variable behaves.

```
x = 20
INPUT "Type in a number?", x
```

```
overwrite()
'-- will display the value entered
PRINT "x = " + x
SUB overwrite
x = 40
PRINT "x = " + x
ENDSUB
```

Example 2: This example demonstrates how a public variable behaves.

```
GLOBAL x
x = 20
INPUT "Type in a number?", x
overwrite()
'-- will display 40
PRINT "x = " + x
SUB overwrite
x = 40
PRINT "x = " + x
ENDSUB
```

GOTO

GOTO <Label Name>

Skip to the specified Label Name.

Example:

```
'-- Try to set the focus to an empty notepad window
IF FOCUS("Untitled - notepad", 10, 0) = 0 THEN
GOTO TimeoutError
ELSE
GOTO FindIt
ENDIF
TimeoutError:
PRINT "Cannot find Untitled - notepad"
END
FindIt:
PRINT "Find it"
END
```

Important Note:

You **should not** use GOTO to get out of a FOR..ENDFOR loop.

Example:

```
'-- Do not do this!!!!!!!!
FOR i = 1 TO 10
IF i = 5 THEN
GOTO OutOfLoop
```

```
ENDIF
ENDFOR
OutOfLoop:
PRINT "Getting out illegally"
END
```

IF..THEN / ELSE / ENDIF

Format #1

```
IF <condition> THEN
<Codes to execute if true>
ENDIF
```

Format #2

```
IF <condition> THEN
<Codes to execute if true>
ELSE
<Codes to execute if false>
ENDIF
```

Conditionally executes a set of commands based on the outcome of a logical expression. In this structured programming command, the <condition> is evaluated. If <condition> evaluates to true, any statements following IF and prior to ELSE or ENDIF (whichever occurs first) are executed. If <condition> is false and ELSE is included, any statements after ELSE and before ENDIF are executed.

If <condition> is false and ELSE isn't included, all statements between IF and ENDIF are ignored. In this case, script execution continues with the first command following ENDIF.

You can nest an IF ... ENDIF block within another IF ... ENDIF block.

Example:

```
'-- Let's say that I am 25
MyAge = 25
'-- Ask for name
INPUT "What is your name?", YourName
'-- Ask for age
INPUT "What is your age?", YourAge
'-- Find out if you are older, younger or the same.
'-- and display the appropriate message.
IF YourAge > MyAge THEN
PRINT "Your are Older than me, " + YourName
ELSE
IF YourAge < MyAge THEN
PRINT "Your are Younger than me, " + YourName
ELSE
PRINT "We are in the same age, " + YourName
ENDIF
```

ENDIF

INCLUDE
INCLUDE "<script file name>"
INCLUDE allows you to use subroutines or functions from an external script file. All the subroutines/functions that are used often should be grouped together in a script file. This way you can always have an access to those subroutines or functions just by using INCLUDE command. You can use as many INCLUDEs as needed.
When you execute a subroutine or a function, ShortCut searches the files specified by INCLUDE command if the subroutine or function isn't located in the currently executing script file.
Example:

```
library.fbs
FUNC GetDateTime
RESULT = Date(4) + " " + TIME(24)
ENDFUNC
SUB SayHello
PRINT "Hello, Stranger."
ENDSUB
main.fbs
INCLUDE "library.fbs"
SayHello()
PRINT "It is now: " + GetDateTime()
```

INPUT
INPUT "INPUT Prompt", <variable>
Prompt the user for an answer. The answer will be stored in the specified variable. Note that the variable name must not contain any white spaces.
Example:

```
'-- ask user for the customer name
INPUT "Customer Name ?", CustName
PRINT "Your customer name is "+ CustName
```

PAUSE
PAUSE <Number of Seconds>
PAUSE the script for a number of seconds. Fractional of second is allowed (make sure that it's in this format: 0.nn)
Example:

```
'-- Wait for 10 seconds
PAUSE 10
'-- Wait for 1/2 of second
PAUSE 0.5
```

PAUSE ,1
Appending “, 1” at the end of a pause statement uses a second method of pausing. In some environments the PAUSE command may not work if another software application modifies the systems clock. This method uses a different approach to pause the script and may help in this situation.
PAUSE <Number of Seconds>, 1
PAUSE the script for a number of seconds. Fractional of second is allowed (make sure that it's in this format: 0.nn)
Example:
'-- Wait for 10 seconds
PAUSE 10, 1
'-- Wait for 1/2 of second
PAUSE 0.5, 1

PRINT
PRINT "Display Message"
PRINT evaluate expressions and popup a window with the results. This is useful for notifying the user for an attention. Note that you can cascade the message with a combination of text and variables by using +.
Example:
'-- Display hello to john and display todays date
UserName = "John"
PRINT "Hello, " + UserName + ". Today is " + DATE(4)
Keep in mind that if a script is set to run as a service, Print statements will not prompt the user but will be written in the event log.

PROMPT
PROMPT "Display Message", <Number_of_Seconds>
PROMPT evaluate expressions and popup a window with the results for the specified number of seconds. This is useful for notifying the user for an attention. Note that you can cascade the message with a combination of text and variables by using +.
Example:
'-- Display hello to john and display todays date
'-- and wait 10 seconds for a respond from the user.
UserName = "John"
PROMPT "Hello, " + UserName + ". Today is " + DATE(4), 10
Keep in mind that if a script is set to run as a service, Prompt statements will be ignored.

RETURN
Returns the script control to a where the GOSUB was called.

Example:

```
Again:
'-- Ask for a password
INPUT "Enter your password", password
'-- if password does not match JOHN then report error
IF UPPER(password) <> "JOHN" THEN
message = "Invalid Password"
GOSUB DisplayMessage
GOTO Again
ENDIF
message = "Password verified."
GOSUB DisplayMessage
PRINT "Thank you!"
END
'-- Display Message on screen for 5 seconds
DisplayMessage:
PROMPT message, 5
RETURN
```

SUB

```
SUB <subroutine name>(<parameters>)
<codes>
.......
ENDSUB
```

SUB <subroutine name> is a statement within a script file. It specifies the beginning of each subroutine in a script file and defines the subroutine name and parameters. Subroutine names must begin with a letter or underscore and may contain any combination of letters, numbers and underscores.

The SUB <subroutine name> command line is followed by a series of commands that make up the subroutine. You must include ENDSUB as the last line of a subroutine.

The difference between FUNC and SUB is that FUNC can return a value, but, SUB cannot.

To use a subroutine, simply include the subroutine name followed by its parameters into your codes (see example). Advance users may want to pass a parameter by reference. To pass a parameter by reference, prefix the parameter name with &.

Example:

```
SUB GetName(prompt, &name)
'-- name will be modified
INPUT prompt, name
ENDSUB
```

WHILE..ENDWHILE

```
WHILE <condition>
```

<Codes to execute while condition is true>
ENDWHILE
Conditionally loops through a set of commands based on the outcome of a logical expression.
In this structured programming command, the <condition> is evaluated. If <condition> evaluates to true, any statements following WHILE and prior to ENDWHILE are executed.
If <condition> is false, all statements between WHILE and ENDWHILE are ignored. In this case, script execution continues with the first command following ENDWHILE.
Example:
'-- Loop until user enter a name
Name = ""
WHILE Name = ""
INPUT "Enter your name:", Name
ENDWHILE
PRINT "Thank you, " + Name

Functions

ALLCOMMANDS
ALLCOMMANDS
Displays all the available commands and functions
Example:
s = ALLCOMMANDS()
PRINT s

ASC
ASC(<char>)
Converts a character to an ASCII value.
Example:
´-- Get Letter A
s = ASC(65)
PRINT s

BOM(), EOM()
BOM format #1 (2-digit year)
<variable> = BOM(2)
BOM format #2 (4-digit year)
<variable> = BOM(4)
EOM format #1 (2-digit year)
<variable> = EOM(2)
EOM format #2 (4-digit year)
<variable> = EOM(4)
Returns the beginning/ending of the current month
This function is useful for month-closing automation for your accounting application.

BOM() returns the beginning date of the current month.
BOM(2) returns beginning date in this format: mm/dd/yy (example: 01/01/99)
BOM(4) returns beginning date in this format: mm/dd/yyyy (example: 01/01/1999)
EOM() returns the ending date of the current month.
EOM(2) returns ending date in this format: mm/dd/yy (example: 01/31/99)
EOM(4) returns ending date in this format: mm/dd/yyyy (example: 01/31/1999)
Example:

```
'-- display beginning date in 2-year digit
PRINT BOM(2)
'-- display beginning date in 4-year digit
PRINT BOM(4)
'-- display ending date in 2-year digit
PRINT EOM(2)
'-- display ending date in 4-year digit
PRINT EOM(4)
```

CHDIR()
CHDIR("<directory>")
Change the current directory to specified <directory>.
Return:
0 = if fails to change the directory
1 = if successful
Example:

```
'-- Change directory to c:\windows
IF CHDIR("c:\windows") = 1 THEN
PRINT "CHDIR is executeted successfully"
ENDIF
```

Converting CSV to a Reform usable format
This function will read in a variable or a file and break up each line of the CSV file into on page per each line and each field will be placed on it own line.

Example:
CSV File:

```
"FabSoft","Reform","","Version 10"
```

Converted format:

```
FabSoft
Reform

Version 10
```

**End of example

Usage: ConvertCSV(InVariable, OutFile)
or
Result = ConvertCSV(InVariable)
-Added: ConvertCSVFile() function
Usage: ConvertCSVFile(InFile, OutFile, PageBreakChar)
or
Result = ConvertCSVFile(InFile)

CURDIR
<variable> = CURDIR()
Retrieve the current directory/folder.
Example:
'-- Retrieve the current directory and display it
cdir = CURDIR()
PRINT "The current directory is: " + cdir

Directory Variables
Returns the location of the specific directory.
x = reformdir("SPOOLER") – returns the path of the spooler folder
x = reformdir("FORMS") – returns the path of all the Reform forms
x = reformdir("BACKUPS") – returns the path of where the print stream backups are stored
x = reformdir("IMAGES") - returns the path of the images folder
x = reformdir("SCRIPTS") – returns the path of where the scripts are stored
x = reformdir("QUEUE") – returns the path of the queue folder
x = reformdir("SENDTO") – returns the path of where the device scripts are stored

ERRORLOG
ERRORLOG("Log_File_Name")
By default, Shortcut runs the script verbose mode. When there is an error, Shortcut will popup a window with an error message. By specifying an error log file, Shortcut automatically runs in a silent mode. Instead of popping up a mesage when there is an error, Shortcut silently records the error to the log file. This is useful for running an unattended script. If you would like to turn off the logging (and run in verbose mode) in a middle of the script, you can do this by passing an empty string ("") to ERRORLOG function (example: ERRORLOG("")).
Example:
'-- Run Shortcut silently when checking mail
ERRORLOG("c:\shortcut\logs\errors.txt")
POP3SETUP("mail.host.com", "john", "56021")
lFrom = ""
lSubject = ""

```
'-- Check for the first mail
i = POP3FIRSTHEADER(lFrom, lSubject)
POP3DISCONNECT()
'-- Run Shortcut in Verbose mode
ERRORLOG("")
'-- if there is a mail
IF i > 0 THEN
PRINT "You've got mail!"
ENDIF
```

EXITFOR

EXITFOR

Use EXITFOR to break out of FOR...ENDFOR loop.

Example:

```
y = 0
FOR i = 1 TO 20
y = y + i
PROMPT y, 0.5
'-- Exit out if y > 6
IF y > 6 THEN
EXITFOR
ENDIF
ENDFOR
PRINT "Done"
```

EXITWHILE

EXITWHILE

Use EXITWHILE to break out of WHILE...ENDWHILE loop.

Example:

```
i = 0
WHILE i < 100
PROMPT y, 0.5
'-- Exit out if i > 50
IF i > 50 THEN
EXITWHILE
ENDIF
i = i + 1
ENDWHILE
PRINT "Done"
```

FEEDBACK

<variable> = FEEDBACK()

FEEDBACK() reports the confirmation status from INPUT, PRINT or PROMPT command. Include this function immediately after INPUT, PRINT or PROMPT. The value returned from FEEDBACK() reflects the value of the most recent INPUT, PRINT or PROMPT command.
Note: Normally Shortcut Script ends when a user press Cancel from INPUT, PRINT , PROMPT commands. To make the script continues even when a user press Cancel, you must set ENDONCANCEL = 0 (see _ENDONCANCEL)
Returns:
0 if user presses Cancel
1 if user presses Ok to exit out of INPUT, PRINT, PROMPT commands.
Example:

```
'-- Tell the script to continue even when Cancel button is pressed from PRINT command.
_ENDONCANCEL = 0
PRINT "Hello, World!"
'-- the script still continues
IF FEEDBACK() = 1 THEN
PRINT "You pressed Ok"
ELSE
PRINT "You pressed Cancel"
ENDIF
```

FILECOPY
Copy a file. Returns True/False
x = FILECOPY(<"source">, <"destination">)

FILEDELETE
Delete a file
X= FILEDELETE("File_Name")

FILEEXISTS
Check an Existance of a file. Returns True/False
x = FILEEXISTS(<"filename">)

FILEEXISTS("|File_Name")
FileInfo
Returns information about the specified file
X= FileInfo(“<Type>”, “FileName Path”)
Type= Age, Size, Version

File Name Variables
Returns the file name of the specific file
x = REFORMVAR("SPOOLFILE")
x = REFORMVAR("FORMNAME")

FileVersion
Returns the version number of the EXE/DLL Specified
X= FileVersion("C:\Program Files\Reform…\ReformENT.exe")

FTPCONNECT
FTPCONNECT(<Server>, [User], [Password], [Proxy], [Port])
Establish a connection to an FTP server.
Parameters:
<Server> Name of an FTP Server (Ex. "ftp.nowhere.com")
[User] User Name (optional, leave blank for anonymous logon)
[Password] Password (optional, leave blank for anonymous logon)
[Proxy] Specify the proxy server if used (optional)
[Port] Specify the FTP port (optional, default is 21)
Returns:
True or 1 If successful
False or 0 If fail

FTPDISCONNECT
FTPDISCONNECT()
Disconnect from an FTP server.
FTPEXEC
FTPEXEC(<command>, <param 1>, [param 2])
FTPEXEC provide a way to interact with an FTP server. There are many different commands that you can execute. <file 1> and <file 2> may or may not need, depending on what ftp commands you are executing. This function also returns different result depending on the command executed.
Parameters:
<command> is an ftp command you would like to execute. The followings are the available commands you can use:
"local" Specify the target directory on your computer. <param 1> specifies the local directory. This is necessary for download and upload files to the server.
"remote" Specify the target directory on the server. <param 1> specifies the remote directory. This is necessary for download and upload files to the server.
"get" Retrieve/Download a file (or files) from the Server. <param 1> specifies a file to download. Wildcards are allowed for downloading a group of similar-name files. Example: "text.txt" or "text*.txt"
"delete" Delete a file from the server. <param 1> specifies the file to be deleted.
"put" Send/Upload a file (or files) from the Server. <param 1> specifies a file to upload. Wildcards are allowed for uploading a group of similar-name files. Example: "text.txt" or "text*.txt"
"mkdir" Make a directory on the server. <param 1> specifies the directory name to create.

"rmdir" Remove a directory from the server. <param 1> specifies the directory name to remove.
"rename" Rename a file. <param 1> specifies the old name. <param 2> specifies the new name.
"find" Find a file or group of similar-name files. <param 1> specifies the file to find. Wildcards are allowed for finding a group of similar-name files. Example: "text.txt" or "text*.txt". Once the command is executed, you can retrieve the find result by using "found" command. The function returns number of found files which can be used for iterate through the found files.
"found" Retrieve the files found as a result of "find" command. <param 1> specifies the file index to retrieve. 1 = first file.

GETDIR
GETDIR("[path/directory]")
Popup a directory selection dialog box for user to choose a directory. [path/directory] is optional. If [path/directory] is not included, it defaults to the current directory.
Parameters:
[path] optional parameter that specifies the default path.
Returns:
If User presses:
OK Returns path selected with "\" padded. (example: "c:\windows\")
Cancel Returns ""

Example:
```
s = GETDIR()
PRINT "You have selected "+ s
```

GETFORMNAME
<variable> = GETFORMNAME()
Retrieves the form name that Lanier PDC currently processing. This function is useful when a script is used by multiple forms. The script can take different actions based on the form that is currently being processed.
Example:
```
s = GETFORMNAME()
CASEOF
CASE s = "INVOICE1.FOM":
HIDEFIELD("Label_100")
SHOWFIELD("Label_200")
CASE s = "INVOICE2.FOM":
SHOWFIELD("Label_100")
HIDEFIELD("Label_200")
ENDCASE
```

SHOWFIELD/HIDEFIELD
(used only in Shortcut for Lanier PDC)
HIDEFIELD("_LanierPDC_Field_Name")
SHOWFIELD("_LanierPDC_Field_Name")

HIDEFIELD and SHOWFIELD are used to hide or show a Lanier PDC Objects on the Design Layer. Each Lanier PDC Object name must be preceed with an underscore symbol ("_"). This is the distince a Lanier PDC object from a regular variable.
Example:

```
s = GETFORMNAME()
CASEOF
CASE s = "INVOICE1.FOM":
HIDEFIELD("Label_100")
SHOWFIELD("Label_200")
CASE s = "INVOICE2.FOM":
SHOWFIELD("Label_100")
HIDEFIELD("Label_200")
ENDCASE
```

GETPRINTERINFO(<"PrinterName">, <"Info">)
Retrieve information about the printer.
<"Info"> = Property that you want to retrieve
Returns:
If successful, value of the specified property, if fails, empty string
Available <"Info">
ServerName
PrinterName
ShareName
PortName
DriverName
Comment
location
SepFile
PrintProcessor
Datatype
Parameters
Attribute
Priority
DefaultPriority
StartTime
UntilTime
Status

Jobs
AveragePPM
DeviceName
SpecVersion
DriverVersion
Size
DriverExtra
Fields
Orientation
PaperSize
PaperLength
PaperWidth
Scale
Copies
DefaultSource
PrintQuality
Color
Duplex
YResolution
TTOption
Collate
FormName
LogPixels
BitsPerPel
PelsWidth
PelsHeight
DisplayFlags
DisplayFrequency
ICMMethod
ICMIntent
MediaType
DitherType

GETPRINTJOBS(<"PrinterName">, <OutPages>, <OutStatus>)
Get the print job info from a printer

Returns:

True/False.
Also passes back OutPages and OutStatus

GETPRINTERSTATUS(<"PrinterName">)
Get the printer status whether it is online, offline, error, etc.
Returns:
"ready" = Printer is ready
"na" = Printer is not available
"offline" = Printer is offline
"paused"
"printing"
"busy"
"notoner"
"tonerlow"
"paperproblem"
"paperout"
"paperjam"
"unknownerror"

BITOP(<Value1>, <"Operation">, <Value2>)
Performs bitwise manipulation of 2 values

HOMEDIR()
<variable> = HOMEDIR()
Retrieve the directory where Reform is executed from. The returned value will always contain \ at the end.
Example:
'-- Retrieve the directory and display it
shortcutdir = HOMEDIR()
PRINT "The home directory is: " + shortcutdir

LOADVARS
LOADVARS("<filename>", "[spec]")
Load variables from file that is created with SAVEVARS. It allows you to load selected group of variables. See About Spec. This function is useful for loading / saving settings for later use.
Parameters:
<filename> specifies the variable file (example: "saved.var")
[spec] optional parameter for qualifying variables (example: "gUSR_").
About Spec:
Spec is not case sensitive. It is used for qualifying variables. Only qualify variables that the leftmost part of the name matches the spec will be loaded.
Returns:
False (0) Fails
True (1) Success.
Example:

```
gUSR_Name = "John"
gUSR_AGE = 27
gALL_Price = 100
gALL_Discount = 10

'-- only save the variables that begin with "gUSR"
SAVEVARS("c:\temp\usr.var", "gUSR")
'-- Clear the variables
gUSR_Name = ""
gUSR_AGE = 0
gALL_Price = 0
gALL_Discount = 0
'-- now load the variables that begin with "gUSR" back
LOADVARS("c:\temp\usr.var", "gUSR")
'-- gALL_Price, gALL_Discount will still be 0
PRINT gUSR_Name, gUSR_Age, gALL_Price, gALL_Discount
```

LPR()

LPR(<"Host">, <"Queue">, <"FileName">) = True/False

Sends the queue file to specified host to a specific queue. The function will return True if it is sent successfully and false if there is an error.

MACRO..ENDMACRO

```
MACRO "<destination>"
<Keystrokes>
ENDMACRO
```

MACRO allows you to send series of keystrokes at once. This is useful if you need to send a large set of keystrokes or a paragraph of text to a window. The keystrokes will be sent exactly as appear within MACRO..ENDMACRO (including spaces). To send a content of a text variable along with the keystrokes, enclose the variable name with "|" (vertical bar). The special keystrokes can also be sent. (See SEND command for more information)

<destination> is the destination window title you want to send the macro to. However, you don't need to include <destination> if you want to send the keystrokes to the current destination window previously set by FOCUS command.

Example

```
Name = "John"
MACRO "Untitled - Notepad"
Hello |Name|, how are you?
{tab}This is a demonstration of how a large text
can be sent to an application without having to
use multiple "SEND" command.
```

Thank you, |Name|.
ENDMACRO

MKDIR
MKDIR("<directory>")
Create a new directory/folder specified by <directory>.
Return:
0 = if fails to create the directory
1 = if successful
Example:
'-- Create a new directory
IF MKDIR("c:\testdirectory") = 0 THEN
PRINT "Error creating the directory"
ENDIF
NOERROR
NOERROR(<option>)
Turn on/off, error notification. This is useful if you want the script to run without any interruption regardless of errors that may occur. NOERROR() is only in effect until the script ends. When the script ends. NOERROR() is reset back to default (0)
Parameters:
<option> 0 = display error notification and halt the script.
1 = ignore error notification and continue.
Example:
'—tell it to ignore error
NOERROR(1)
x = 0
y = 2
'—Create Divided by 0 error
z = y/x
'—tell it to display the error
NOERROR(0)

OSTYPE
<variable> = OSTYPE()
Retrieve the current Operation System type.
Returns:
1 = Windows 95/98
2 = Windows NT
3 = Windows 32
Example:
'-- Retrieve the current OS
x = OSTYPE()

```
IF x = 1 THEN
PRINT "Windows 95/98"
ELSE
IF x = 2 THEN
PRINT "Windows NT"
ELSE
IF x = 3 THEN
PRINT "Windows 32"
ENDIF
ENDIF
ENDIF
```

RANDOM

RANDOM(<range>)

Generates random numbers within a specified range. Random returns a random number within the range 0 <= X < Range

Parameters:

<range> specify range to generate a random number. (0 <= X < Range**)**

Example:

```
PRINT RANDOM(100)
```

REFORMVAR

REFORMVAR(<parameter>)

This function can retrieve information about your form and the current print stream. **NOTE** This function will only work correctly if it is used in a Page-Process script!

Parameters:

"PAGENO" – this will retrieve the current page number of the print stream that Reform is processing.

"PAGES" – this will retrieve the total number of pages in the print stream.

"Spoolfile" – this will retrieve the filename of the current print stream that Reform is processing.

"Totalpages" – this will retrieve the total number of pages in the print stream.

"Formname" – this will retrieve the filename of the form that is open.

"Currentpage" – this will retrieve the text of the current page.

"Spooler" – this will retrieve the path to the Spooler folder.

RESULT

RESULT = <value>

RESULT is used exclusively within a function. It's a special variable that is used to return a value from a function.

Example:

```
today = GetDate()
```

PRINT today

FUNC GetDate
'-- return todays date
RESULT = date(4)
ENDFUNC
Sub/Func Example

ROUND
ROUND(<number>)
Returns the value of <number> rounded to the nearest whole number. For example: 1.5 » 2.0, 1.2 » 1.0
Parameters:
<number> number to round up.
Example:
PRINT ROUND(1.2), ROUND(1.5), ROUND(1.7)

SAVEFORMVARS/LOADFORMVARS
SAVEFORMVARS("[Optional_FileName]", "[Optional_Field_Spec]")
LOADFORMVARS("[Optional_FileName]", "[Optional_Field_Spec]")
SAVEFORMVARS saves the entire content of Reform Fields to a File. This function provides an easy way to memorize the content for later retrieval using LOADFORMVARS.
There are 2 optional parameters:
[Optional_FileName] = Specifies the file name to save to. If left blank, the Form Name with an extension ".mem" will be used.
[Opitional_Field_Spec] = Only the fields that begin with the Field Spec will be save/load
Example:
s = GETFORMNAME()
CASEOF
'-- If this is Invoice1 then we save all the labels to file
'-- for later retrieval
CASE s = "INVOICE1.FOM":
SAVEFORMVARS("default.mem", "_Label")
'-- If this is Invoice2 then we load all the labels from file
CASE s = "INVOICE2.FOM":
LOADFORMVARS("default.mem", "_Label")
ENDCASE

SAVEVARS
SAVEVARS("<filename>", "[spec]")

Save specific variables to a file that can later be loaded using LOADVARS. It allows you to save selected group of variables. See About Spec. This function is useful for loading / saving settings for later use.
Parameters:
<filename> specifies the variable file (example: "saved.var")
[spec] optional parameter for qualifying variables (example: "gUSR_").
About Spec:
Spec is not case sensitive. It is used for qualifying variables. Only qualify variables that the leftmost part of the name matches the spec will be saved.
Returns:
False (0) Fails
True (1) Success.
Example:

```
gUSR_Name = "John"
gUSR_AGE = 27
gALL_Price = 100
gALL_Discount = 10

'-- only save the variables that begin with "gUSR"
SAVEVARS("c:\temp\usr.var", "gUSR")
'-- Clear the variables
gUSR_Name = ""
gUSR_AGE = 0
gALL_Price = 0
gALL_Discount = 0
'-- now load the variables that begin with "gUSR" back
LOADVARS("c:\temp\usr.var", "gUSR")
'-- gALL_Price, gALL_Discount will still be 0
PRINT gUSR_Name, gUSR_Age, gALL_Price, gALL_Discount
```

SetPDFProp
SetPDFProp(<"Properties">, <Value>)

Description
Set various properties of the created PDF document. This function only works within a Page-Process Script. This will override the settings set via attachment setup.

Parameters:
<"Properties">
"title"
Title of the PDF (Max 50 characters)
"author"

Name of the author of the PDF (Max 50 characters)
"subject"
Subject (Max 50 characters)
"keywords"
List of keywords separated by "," (Max 200 characters
"embedfont"
(True/False) embed the font into the PDF document.
"userpass"
Set the user password (Max 20 characters. Setting this property automatically turns "encrypt" to True)
"ownerpass"
Set the owner password (Max 20 characters. Setting this property automatically turns "encrypt" to True)
"encrypt"
(True/False) Explicitly turn on the encryption. It will initially turn off the "printing", "changing", "copying", and "form" flags. You must explicitly turn these options on.
"128bit"
(True/False) Specify 128 bit encryption, otherwise it will use 40 bit encryption if "encrypt" is True. (only in effect if encrypt is True)
"printing"
(True/False) Enables printing (only in effect if encrypt is True)
"changing"
(True/False)Enable modification to the PDF document (only in effect if encrypt is True)
"copying"
(True/False)Enable copying of the content in the PDF document (only in effect if encrypt is True)
"form"
(True/False)Enable form fill out in the PDF document (only in effect if encrypt is True)

<Value>
Can be a string or True/False depending on the specified property.

Example:

```
'-- The PDF can not be opened unless the user has the user password.
'-- encrypt is automatically True because we set a user password.
'-- They can copy, print, and fillout the form

SetPDFProp("userpass", "canread")
SetPDFProp("form" True)
SetPDFProp("printing", True)
SetPDFProp("copying", True)
SetPDFProp("128bit", True)
```

SMTPDISCONNECT
SMTPDISCONNECT()
Disconnect from SMTP server. This function should be called when all the SMTP transactions are completed.

SMTPSEND
SMTPSEND(<"From">, <"To">, <"Subject">, <"Content">, ["Attach Files"])

Sends an E-mail. Be sure to include all the appropriate parameters. The parameters are explained below:
Parameters:
<"From"> e-mail address of a sender
<"To"> e-mail address of a receiver
<"Subject"> Subject header
<"Content"> Text Content. Text content can be as long as you like (see example for creating a multiple-line text).
["Attach Files"] (optional) List of attached file name(s) to be sent. If you need to send more than one file then separate each file name with ";". (example: "File1.txt;File2.txt;File3.txt")
Returns:
0 = if it fails to send the mail
1 = if successful.

SMTPSETUP
SMTPSETUP("Host", [Port])
Setup SMTP for sending E-mail. The setting remains in effect until the script ends or another SMTPSETUP is executed. Port is an optional variable. By default, Port is set to 25.
UNIQUEFILE
UNIQUEFILE(<"FileName">, [IncrementFileExtension (True/False)], [StartingNumber])
Description:
This function returns the unique file name. If the file name already exists it auto increment the filename or file extension portion (depending on the "IncrementFileExtension" flag). If the Starting Number is passed, it uses the Starting Number as the starting number for the auto incrementing.

Here is the example:
lFileName = "c:\temp\test.out"
'-- Increment file name (default)
PRINT UNIQUEFILE(lFileName)
'-- Increment file name with starting number PRINT UNIQUEFILE(lFileName, False, 100)
'-- Increment file extension
PRINT UNIQUEFILE(lFileName, True)

'-- Increment file extension with starting number PRINT UNIQUEFILE(lFileName, True, 100)

VARBYNAME
<variable> = VARBYNAME("<Variable Name>")
To retrieve the content of a variable by its name as a literal string. <Variable Name> is a string representation of the variable name you want to retrieve.
Example:

```
FirstName = "John"
'-- Display the name
PRINT FirstName
'-- Display the name by using VARBYNAME
PRINT VARBYNAME("FirstName")
```

WAITDOS (DOS Only)
<variable> = WAITDOS(Row, Col, "Text", <Timeout in seconds>)

Wait for a DOS text to appear at the specified Row and Column for the specified amount of time. The valid Row ranges from 1 to 25. The valid Column ranges from 1 to 80. If the text appears within the time, WAITDOS returns 1, otherwise, it returns 0. This command only works with DOS Application.
Example:
Example #1:

```
'-- Wait for the word Name: to appear for 10 seconds
x = WAITDOS(1, 1, "Name", 10)
IF x = 0 THEN
PRINT "Timeout waiting for text!"
ELSE
PRINT "Found Name"
ENDIF
```

Example #2:

```
'-- Wait for the word Name: to appear for 10 seconds
IF WAITDOS(1, 1, "Name", 10) = 0 THEN
PRINT "Timeout waiting for text!"
ELSE
PRINT "Found Name"
ENDIF
```

The example above waits for "Name" to appear at Row 0 and Column 0 for 10 Seconds. If the text does not appear within 10 seconds, report the error.
Note: use DOS capturing tool to help you getting the exact row and col from the DOS screen.

WAITWIN (Windows App. Only)
<variable> = WAITWIN("Window Title, "Text", <Timeout>)

Wait for a text to appear within the specified window for the specified amount of time (in seconds). If the text appears within the time, WAITWIN returns 1, otherwise, it returns 0. This command only works with windows Application.

Example:

The example below waits for "Name" to appear at Row 0 and Column 0 for 10 Seconds. If the text does not appear within 10 seconds, report the error.

Example #1:

```
'-- Wait for the word Name: to appear from within the
'-- window "Windoze" for 10 seconds
x = WAITWIN("Windoze", "Name", 10)
IF x = 0 THEN
PRINT "Timeout waiting for text!"
ELSE
PRINT "Found Name"
ENDIF
```

Example #2:

```
'-- Wait for the word Name: to appear from within the
'-- window "Windoze" for 10 seconds
IF WAITWIN("Windoze", "Name", 10) = 0 THEN
PRINT "Timeout waiting for text!"
ELSE
PRINT "Found Name"
ENDIF
```

Running Scripts as Services

To run a script as a service, the script must first be compiled to an executable using the included ShortRun.exe program. To do this, we are going to assume you have a script called ServiceSample.fbs located in your scripts folder. Creating the script executable and installing the service is all done via the Windows command prompt.

1. Go to **File>Run**. Enter cmd and press Enter.
2. Type the following line, and press Enter:

```
cd “C:\Program Files\Reform…\Scripts”
```

3. You are now in the same directory as your script, as well as the ShortRun.exe application. We need to run the ShortRun.exe application and provide it with parameters to compile the script file into an executable that can be run as a service or a standalone application.
4. Type the following line, and press Enter:

```
ShortRun.exe –e “ServiceSample.fbs” “ServiceSample.exe”
```

5. You will notice that ServiceSample.exe was created inside your \Reform\Scripts directory. Now, you must run the executable through the command prompt and supply it with the /install parameter.

6. Type the following line, and press Enter:

ServiceSample.exe /install

7. A message box will pop up after the script is installed as a service.

To Uninstall a Service:

1. Go to File>Run. Enter cmd and press Enter.
2. Type the following line, and press Enter:

cd “C:\Program Files\Reform…\Scripts”

3. We will assume your script application is named ServiceSample.exe. Enter the following line, and press Enter:

ServiceSample.exe /uninstall

4. A message box will pop up after the service is uninstalled.

Reference

For reference purposes, the parameters for the ShortRun executable to create an application from a script are follows:

ShortRun.exe –e “<name of script file>” “<name of executable file>”

Once you have built an executable from a script file, you can pass it the parameters **/install** or **/uninstall** to install or uninstall the service, such as in the example below:

ScriptEXE.exe /install

ScriptEXE.exe /uninstall

Inbound Connectivity

Client Capture Utilities

Because the Reform Spooler monitors one directory for text/ASCII or other file types to appear, your application needs a way to place these files in the spooler directory. Reform comes with capture utilities to make this possible by seamlessly linking to all operating systems and applications.

Some applications may already have a seamless link to Reform. Consult your software manufacturer for information on existing seamless links.

Reform has many ways to link to your existing system:

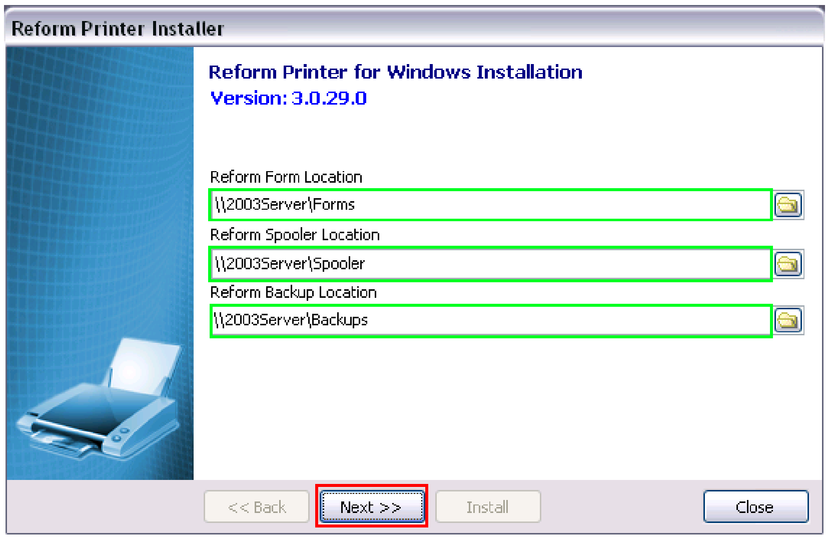
1. Using the included Reform Printer Driver. Users can print directly from their applications. The Reform Printer Driver can capture data from any Windows-type application.
2. Using the included LPR/LPD for connections to AS/400, UNIX, Linux, AIX and other systems.

3. Capturing print jobs from your workstation's LPT ports.
4. Capturing print jobs from the DOS operating system.
5. Reading from a common text file or temporary print file.
6. Capturing data from a serial or parallel port.
7. Capturing data from a telnet connection.
8. Capturing PCL script.
9. Capturing Postscript.
10. Capturing PDFs.

Reform Printer Installation

Server-Side Installation

1. To start the installation of the Reform Print Driver, Click **Start > Programs > Reform… > Install Windows Printer Driver.**
2. Be sure that the directories are correct. These 3 directories are created and automatically shared during the Reform installation.
3. Click **Next**.

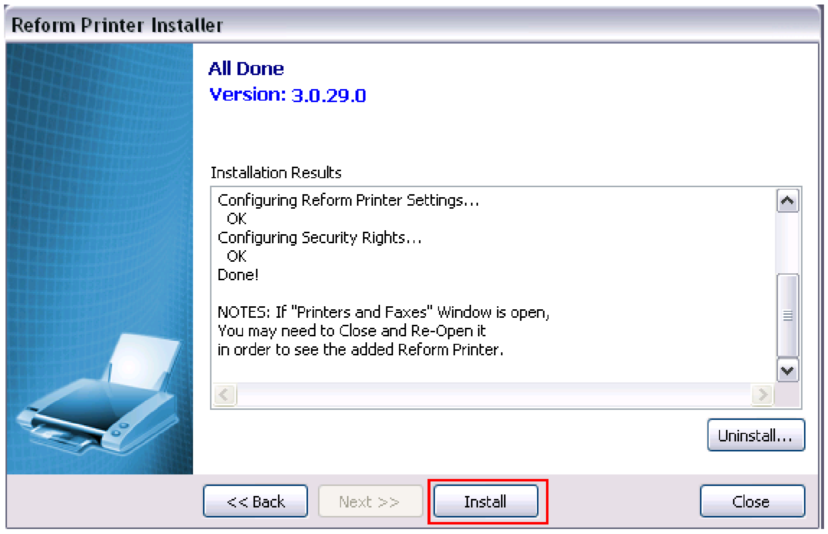


Reform Printer Installer

Reform Printer for Windows Installation
Version: 3.0.29.0

Reform Form Location: \\2003Server\Forms
Reform Spooler Location: \\2003Server\Spooler
Reform Backup Location: \\2003Server\Backups

4. Click **Install**

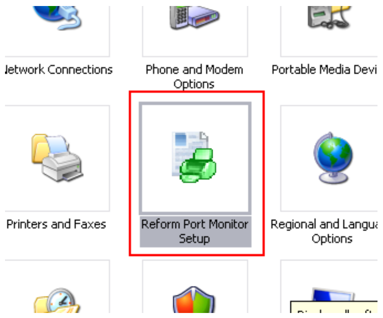

Reform Printer Installer

All Done
Version: 3.0.29.0

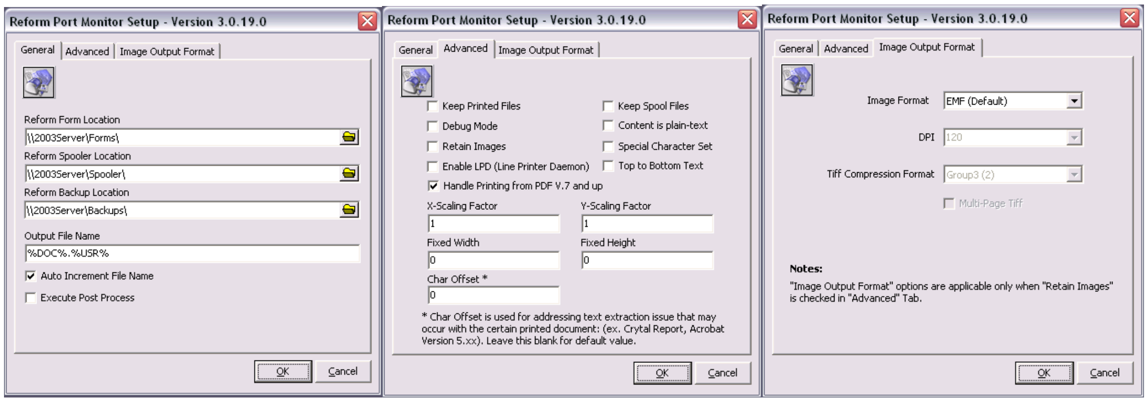
Installation Results

```
Configuring Reform Printer Settings...
  OK
Configuring Security Rights...
  OK
Done!

NOTES: If "Printers and Faxes" Window is open,
You may need to Close and Re-Open it
in order to see the added Reform Printer.
```

5. Once the installation has completed, you can close the installer.
6. Click **Start > Control Panel**



7. Open **Reform Port Monitor Setup**
8. You can then modify advanced features for the Reform print driver.


Reform Port Monitor Setup - Version 3.0.19.0

General | Advanced | Image Output Format

Reform Form Location: \\2003Server\Forms\
Reform Spooler Location: \\2003Server\Spooler\
Reform Backup Location: \\2003Server\Backups\
Output File Name: %DOC%.%USR%
[✓] Auto Increment File Name
[ ] Execute Post Process

OK | Cancel

Reform Port Monitor Setup - Version 3.0.19.0

General | Advanced | Image Output Format

[ ] Keep Printed Files
[ ] Keep Spool Files
[ ] Debug Mode
[ ] Content is plain-text
[ ] Retain Images
[ ] Special Character Set
[ ] Enable LPD (Line Printer Daemon)
[ ] Top to Bottom Text
[✓] Handle Printing from PDF V.7 and up

X-Scaling Factor: 1
Y-Scaling Factor: 1
Fixed Width: 0
Fixed Height: 0
Char Offset *: 0

* Char Offset is used for addressing text extraction issue that may occur with the certain printed document: (ex. Crytal Report, Acrobat Version 5.xx). Leave this blank for default value.

OK | Cancel

Reform Port Monitor Setup - Version 3.0.19.0

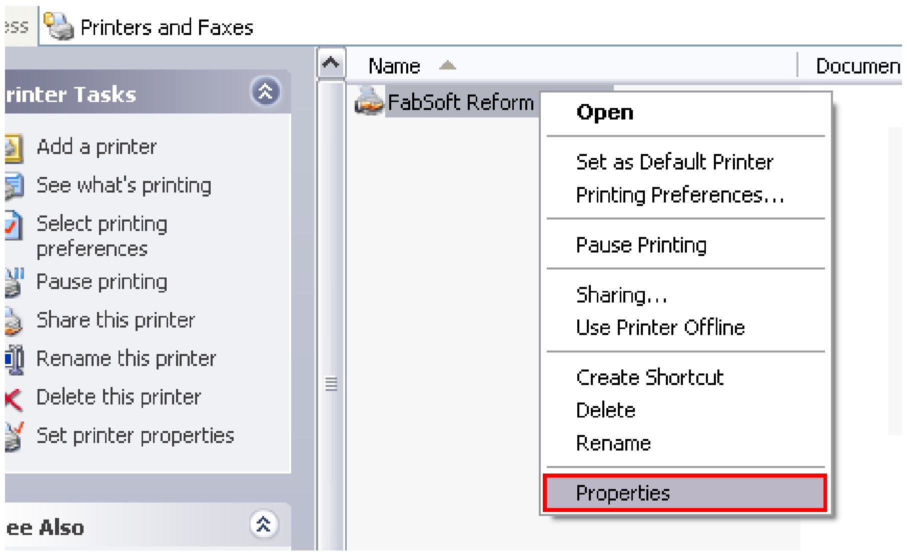
General | Advanced | Image Output Format

Image Format: EMF (Default)
DPI: 120
Tiff Compression Format: Group3 (2)
[ ] Multi-Page Tiff

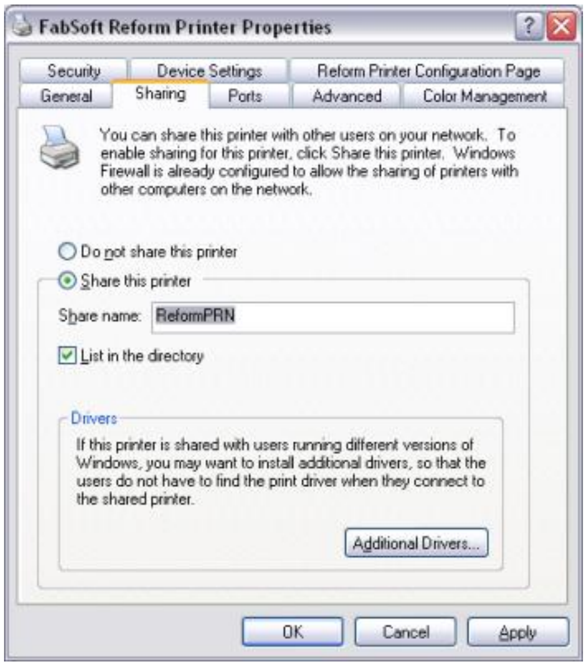
Notes:
"Image Output Format" options are applicable only when "Retain Images" is checked in "Advanced" Tab.

OK | Cancel

9. Click **OK** to exit the setup.
10. Open your Printers folder. To do this, click **Start > Control Panel**
11. Open **Printers and Faxes**
12. You should see your list of printers including the FabSoft Reform Printer.
13. Right click the **FabSoft Reform Printer** and click **Properties**



14. In the Sharing tab, be sure that the printer is shared.


FabSoft Reform Printer Properties
Security
Device Settings
Reform Printer Configuration Page
General
Sharing
Ports
Advanced
Color Management

You can share this printer with other users on your network. To enable sharing for this printer, click Share this printer. Windows Firewall is already configured to allow the sharing of printers with other computers on the network.

Do not share this printer
Share this printer [✓]

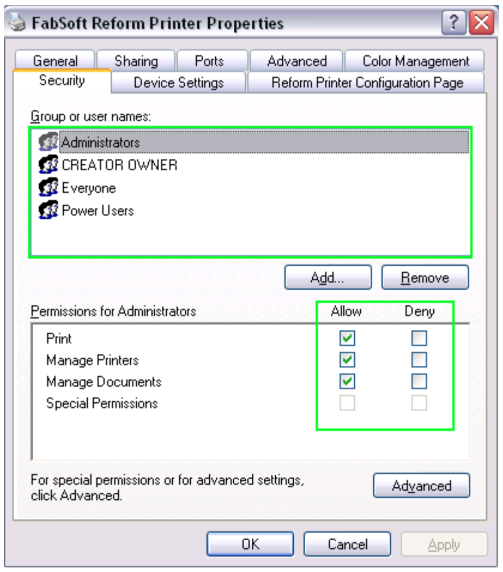
Share name: RefomPRN

[✓] List in the directory

Drivers

If this printer is shared with users running different versions of Windows, you may want to install additional drivers, so that the users do not have to find the print driver when they connect to the shared printer.

15. On the Security tab, be sure that the windows user or group that will be using the driver is given appropriate permission.



Group or user names:
Administrators
CREATOR OWNER
Everyone
Power Users

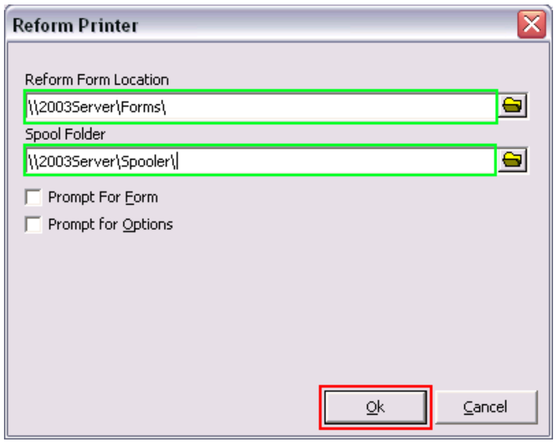
Permissions for Administrators

	Allow	Deny
Print	✓	
Manage Printers	✓	
Manage Documents	✓	
Special Permissions		

For special permissions or for advanced settings, click Advanced.

16. In the **Reform Printer Configuration Page** tab, Click **Configure…**

17. Fill in the Form Location and Spooler Folders. If you want to use the computer's name, these should be: \\ServerName\Forms\ and \\ServerName\Spooler\ or if you want to use the IP address (If the IP Address was 192.168.1.123): \\192.168.1.123\Forms\ and \\192.168.1.123\Spooler\

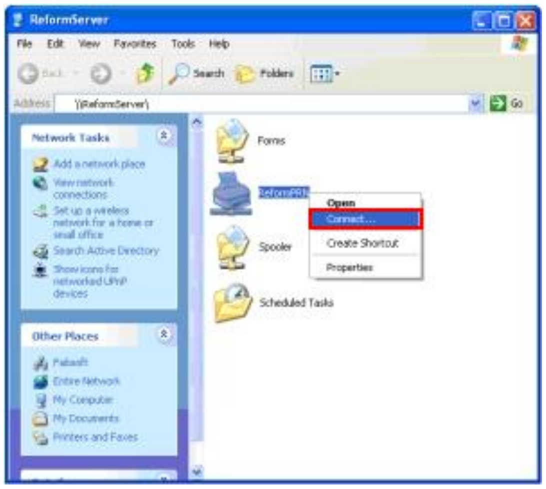


Reform Form Location: \\2003Server\Forms\
Spool Folder: \\2003Server\Spooler\
Prompt For Form
Prompt for Options

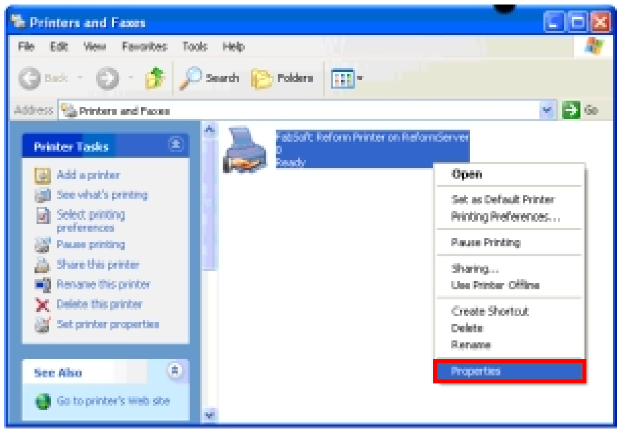
18. Click **OK** to close the configuration window.
19. Click **OK** to close the printer properties.
20. The server side of the installation is now complete.

Client-Side Installation

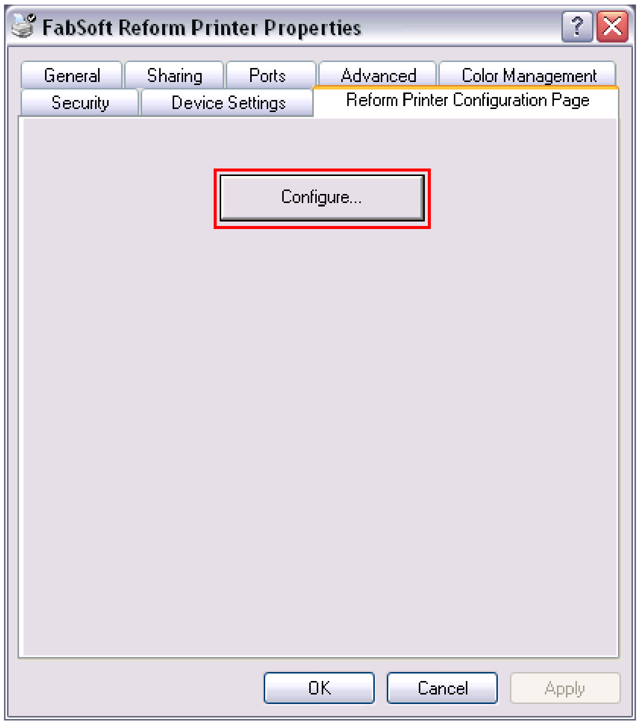
1. Once the Server Side Installation is complete, the print driver can be connected to by the client's pc.
2. Browse to the server using the server's name or IP address. Sample: \\ReformServer\ or (If the Server's IP address was 192.168.1.123) \\192.168.1.123\
3. You should then see the **ReformPRN** printer in the list as a shared printer.
4. Select and right click on the **ReformPRN**, and click **Connect**.



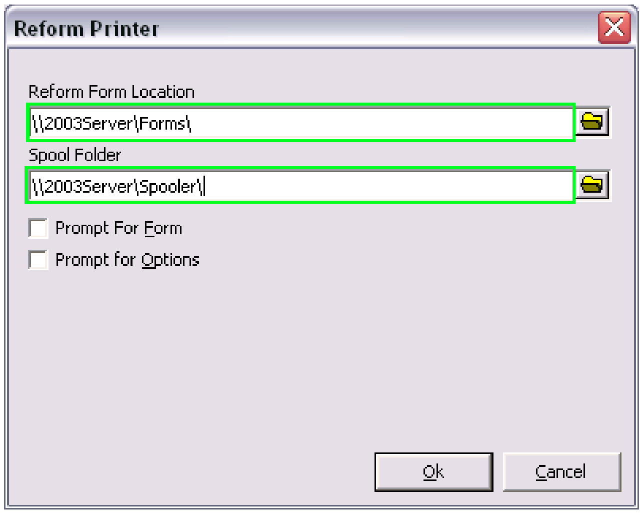
5. After it completes the installation, open your Printers folder by clicking on **Start > Control Panel.**
6. Open **Printers** and Faxes
7. Select and right click on the **Reform Printer**, and click on **Properties**.



8. In the Reform Printer Configuration Page tab, click **Configure…**



9. Fill in the Form Location and Spooler Folders. These should be \\ServerName\Forms\ and \\ServerName\Spooler\ or if you want to use the IP address (If the IP Address was 192.168.1.123): \\192.168.1.123\Forms\ and \\192.168.1.123\Spooler\


Reform Printer

Reform Form Location: \\2003Server\Forms\

Spool Folder: \\2003Server\Spooler\

☐ Prompt For Form

☐ Prompt for Options

***Extra Options:**

Prompt for Form

Checking this box causes a form prompt to appear any time a user prints, listing every available form overlay. This is useful when no automatic form associations have been set.

Prompt for Options

Checking this box will enable the Advanced Printer Driver features. Advance Printer Driver options will enable display of custom screens used to capture input by the user. This option is only used if the Reform Printer Driver is also used on end user workstations.

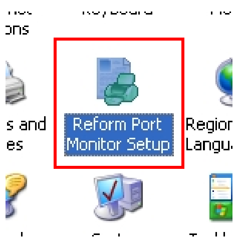
10. Click **OK** to close the configuration window.
11. Click **OK** to close the printer properties.
12. The Print Driver is now ready to use.

Reform Printer Driver

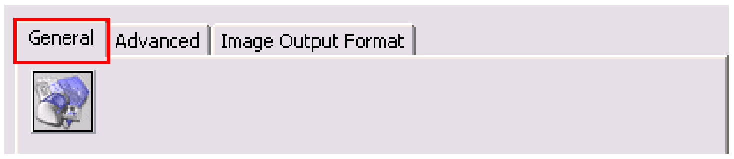
Reform Port Monitor Configuration

The Reform Printer configuration utility can be found in the Windows control panel. Double-click the **Reform Printer Setup** icon to open the utility.

The configuration utility is a program used to alter different aspects of the way Reform functions; it contains many settings that can be adjusted to suit the needs or configuration of a system or network. The utility contains three tabs; **General, Advanced, and Image Output Format**. Descriptions of the controls and properties contained in each tab window are discussed in detail below.



The General Tab

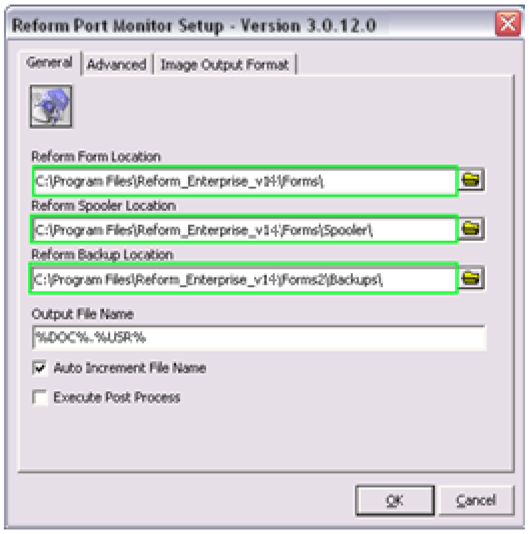


Path designation fields

These fields define the locations of the directories used by Reform to search for and store form overlays, spooler files and backups. If you change the locations of these folders in your system, you must change the path in its corresponding field here to properly reflect the directory's location on your system. If you do not, Reform will malfunction.

Reform Form Location

Defines the location of the directory in which all form overlays are stored. C:\Program Files\Reform…\Forms\ is the default.



Reform Form Location: C:\Program Files\Reform_Enterprise_v14\Forms\
Reform Spooler Location: C:\Program Files\Reform_Enterprise_v14\Forms\Spooler\
Reform Backup Location: C:\Program Files\Reform_Enterprise_v14\Forms2\Backups\
Output File Name: %DOC%.%USR%
[✓] Auto Increment File Name
[ ] Execute Post Process

Reform Spool Location

Defines the directory in which spooler files will be directed to. The Reform Spooler monitors this directory for files ASCII/Text files. C:\Program Files\Reform…\Spooler\ is the default.

Reform Backup Location

Defines the location of the backup directory. All spooler files will be backed up to this directory if the Keep Spool Files option under the Advance tab is checked. C:\Program Files\Reform…\Backups\ is the default.

Checkboxes

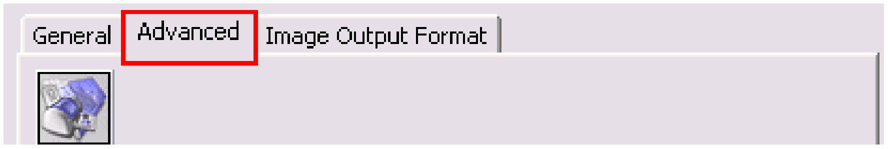
Auto Increment File Name

Checking this box causes the Reform Printer to increment the filename extension when it finds a spooler file of the same name in the Spooler directory.

Execute Post Process

Checking this box will enable the Advanced Printer Driver features on the client side. Advance Printer Driver options will enable display of custom screens used to capture input by the user.

The Advanced Tab



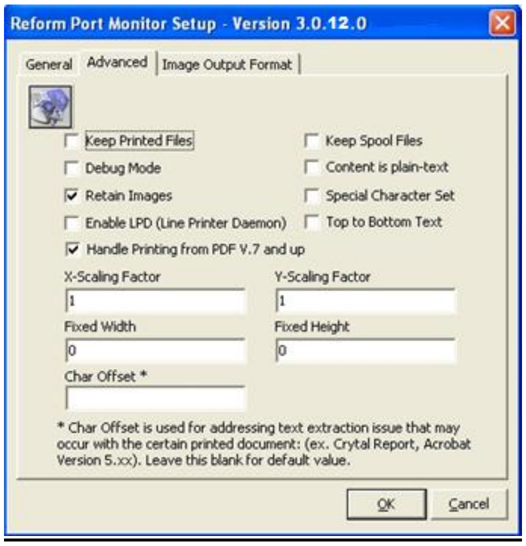
Checkboxes

Keep Printed Files

Checking this box creates a backup of the EMF file in the Backups directory.

Keep Spool Files

Checking this box causes the Reform Printer to make backups of the SPL files (located in C:\Windows\System32\Spool\Printers) that it processes. These backups are stored in the Backups directory. While this is a useful function for purposes of debugging and form design, leaving it active on systems that process a lot of print jobs can cause the directory to get very large and eventually consume a great deal of disk space.


Reform Port Monitor Setup - Version 3.0.12.0

General | Advanced | Image Output Format

☐ Keep Printed Files ☐ Keep Spool Files
☐ Debug Mode ☐ Content is plain-text
[✓ Retain Images] ☐ Special Character Set
☐ Enable LPD (Line Printer Daemon) ☐ Top to Bottom Text
[✓ Handle Printing from PDF V.7 and up]

X-Scaling Factor: 1
Y-Scaling Factor: 1
Fixed Width: 0
Fixed Height: 0
Char Offset *:

* Char Offset is used for addressing text extraction issue that may occur with the certain printed document: (ex. Crytal Report, Acrobat Version 5.xx). Leave this blank for default value.

OK | Cancel

Debug Mode

Checking this option causes the debug window in the Reform spooler to make checks and display messages relating to form associations.

Content is Plain Text

This option causes the Reform Printer to interpret an incoming print stream as plain text, regardless of any specific text formatting it contains. Enabling this option is used to correct text alignment problems in spooler files in certain situations.

Retain Images

This option causes Reform to save graphical images of the files processed by the Reform Printer in the \Spooler\ImageQueue directory. The images are saved as EMF files. This option is useful if you want to maintain the graphical formatting of your print jobs in your form overlays.

Special Character Set

Some printed documents may contain a shifted-character set. This option will fix the shifted-character set. If the document is read without this feature, the text may not be readable.

Enable LPD* (Line Printer Daemon)

This option is used if your data stream comes from a non-Windows based application (Unix, AS/400, AIX, etc.) This option switches the Reform print processor from EMF 1.006 to RAW.

Top to Bottom Text

This option is used with Retain Images. Text is broken by line (vertically) instead of horizontally (which is the standard method). This option is useful when Reform is capturing data from host systems that use formatting (such as Word, Crystal Reports, PDFs and PCL).

Handle Printing from PDF v7 and up

Use this option when capturing data stream or images from PDFs printed in Adobe Acrobat or Acrobat Reader 7.0 and up.

Size and Scaling adjustment fields

X and Y Scaling Factor

Adjust these values if the spooler files sent from the Reform Printer contain text that is not properly spaced. X adjusts horizontal scaling and Y adjusts vertical scaling.

Fixed Width and Height

These fields cause all text information received from the Reform Printer to conform to the same width and/or height settings.

Image Output Format Tab

On the Image Output Format tab, you can choose the file extension for the image files that are generated when you have Retain Images turned on. These images will be saved in ReformDir\Spooler\ImageQueue. The recommended extension is EMF.

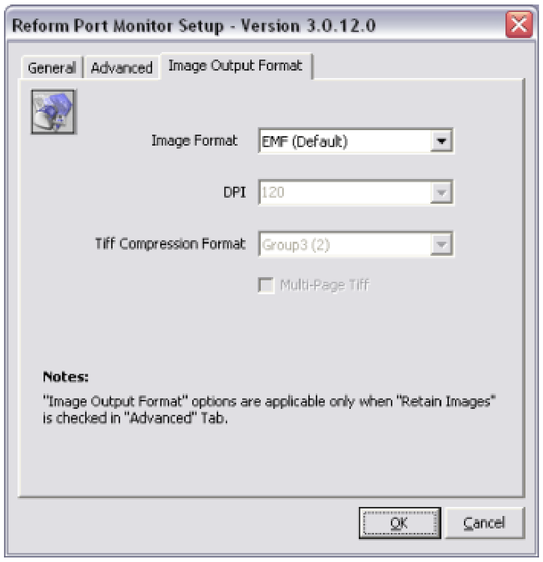

Reform Port Monitor Setup - Version 3.0.12.0

Image Format: EMF (Default)

DPI: 120

Tiff Compression Format: Group3 (2)

Multi-Page Tiff

Notes:
"Image Output Format" options are applicable only when "Retain Images" is checked in "Advanced" Tab.

Generic Text File

Printing to a Generic Text File

Some form systems, such as accounting systems and others, are capable of writing to files at printing time. To use this option, the user would select from their system the option to print a file, and then specify a file name. This file name should be the same as the name of the Reform form that has been pre-defined in the Reform editor, and the location should be the same as the Spooler directory defined in Reform Setup. It is recommended to increment or make the file extension unique.

Reading from a Common Text File or Temporary Print File

Some form systems, such as accounting systems and others, create print files in a temporary or defined directory. Many software manufacturers make an interface to Reform. Please consult your form system manufacturer for such interface information. The setup window can be customized to fit many types of output text files.

Serial, Parallel, and TCP/IP Ports

Serial or Parallel Ports

Reform Port Monitor

Port Monitor is a program used to capture data from a computer’s Serial or TCP/IP connection and redirect it to another computer. It is used primarily for the purpose of capturing the customer’s print stream without having direct access to the customer’s operating system. You can find Reform Port Monitor program in the \Reform…\Misc directory. If you need to run multiple copies of the Reform Port Monitor create a separate directory for each then copy the PortMonitor.exe into each directory.

Parallel to Serial:

You will need one Parallel to Serial converter - This is a device used to create a connection between parallel and serial ports. In our tests, we used a converter made by Black Box called a Serial-Parallel Converter VI. Their part number is PI130A-R2, and it is available for purchase at www.blackbox.com.

One DB25 Male-to-DB25 Male Parallel Cable – This is a standard printer cable. These are also available from Black Box. Their part number is BC00705 and it is also available for purchase at their website.

One DB25 Male-to-DB25 Male (or DB9 Female) Serial Modem Cable – Any generic Serial cable capable of connecting to your computer’s serial port and the converter. Note: Black Box’s converter accepts a DB25 male connection.

Serial to Serial Port:

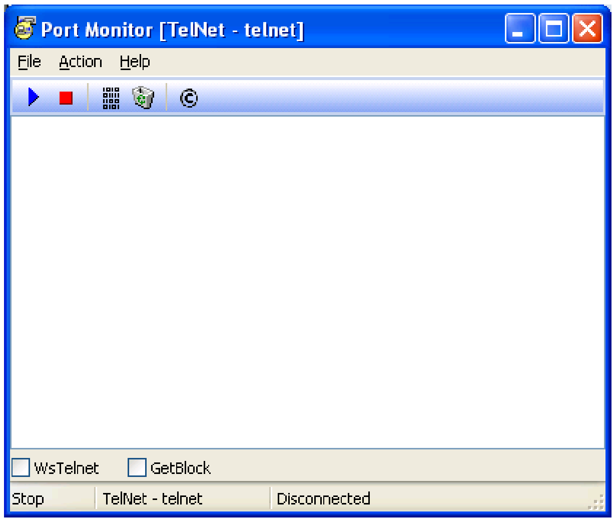
Just a Serial cable and a null connector, just connect it between the two computer or serial blocks. Follow the same steps as the Parallel Port except without the Parallel to Serial converter.

Setting up Port Monitor

Port Monitor resides in Reform’s Miscellaneous folder. To start Port Monitor, simply double click on the file called **ReformPortMon.exe**. When Port Monitor starts up, it will display a window and a menu bar across the top, as shown below:

Debug Results window

Setup menu

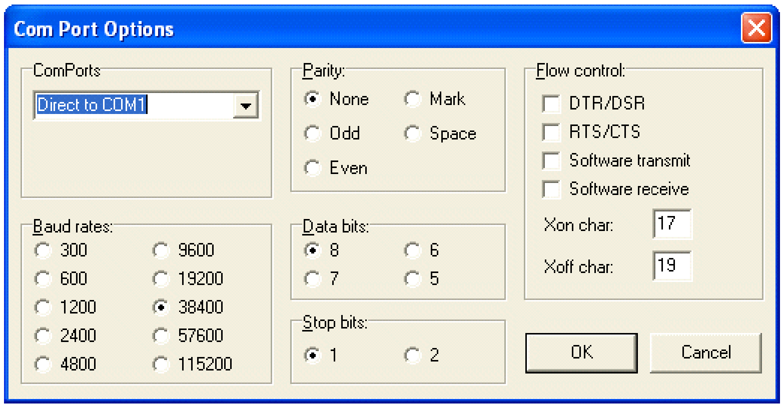


First, go into the Setup menu by clicking on **File > Setup.** The Setup menu contains three tabs, labeled General Settings, Telnet Settings and Serial Settings.

Under General Settings, you will need to define a filename for the text file that will contain the data once the transfer operation has been performed. This can be done by typing the desired filename into the box labeled Save to File. Also make sure the Show Debug Results box is checked if you wish to monitor the data transfer operation in the Debug Results Window.

In the Active Device drop-down menu, you must select either Serial or Telnet, depending on the type of connection you are establishing. The instructions in this document are geared toward using Port Monitor in Serial mode.

Next, click on the **Serial Settings** tab. It will display a button labeled Configure Serial Device. Click this button and you will be taken to the Com Port Options menu shown below.



ComPorts: Direct to COM1

Parity: [✓ None] Mark Odd Space Even

Flow control: DTR/DSR RTS/CTS Software transmit Software receive

Xon char: 17

Xoff char: 19

Baud rates: 300 9600 600 19200 1200 [✓ 38400] 2400 57600 4800 115200

Data bits: [✓ 8] 6 7 5

Stop bits: [✓ 1] 2

First, select a Com Port from the Com Ports dropdown box. This should be the Com Port of your computer (the machine that will receive the data). The Baud rates, Parity, Data bits and Stop bits settings must be set to the same configuration as the Serial Parallel Converter. To find out which settings to use, consult the converter's manual. Using the Black Box Serial-Parallel Converter described above, you can obtain this information by examining the DIP-switches on the back of the converter and comparing them to a chart in the user's manual. The settings shown in the diagram above were the settings we used in our tests, and they are the default settings for the Black Box Serial-Parallel Converter.

When you have finished defining the Com Port Options, click **OK**. You will be taken back to the Setup menu. Click **OK** again and you will be taken back to Port Monitor's main interface. Restart Port Monitor to insure that any changes you made to the Com Port or General settings take effect.

Port Monitor should now be ready to receive data from the host machine.

Connecting to the Host Computer

Before running Port Monitor, you must physically connect the host computer to the receiving computer. To facilitate this, do the following:

Connect the Serial Cable to the receiving computer, then to the Serial side of the Serial-Parallel Converter. Then, connect one end of the Parallel Cable to the Parallel side of the Serial-Parallel Converter. The other end of the Parallel Cable must be plugged into the host computer’s Parallel port. See the diagram below for details. If Port Monitor is running, you will now be ready to receive text data from the host computer.

Using Port Monitor

For Port Monitor to receive the data from the host computer, you must press the Start Monitoring button, on the menu bar (indicated by a blue arrow). This causes Port Monitor to begin monitoring the Com Port for incoming data. When the data is received, Port Monitor will automatically create the file you specified in General Settings in its designated folder.

Testing the Connection

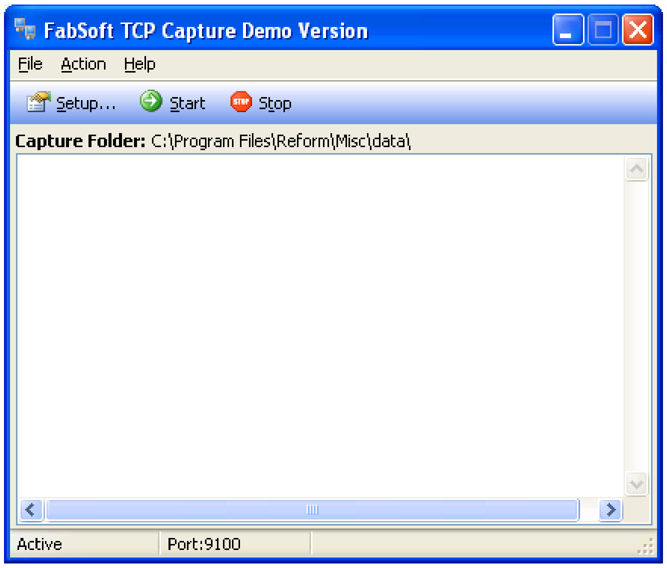
It is recommended that you test the Port Monitor connection prior to using it in the field. To set up a test run of the program you will need two computers; one to simulate the Host computer, and another to simulate the Receiving computer. Both machines should be running the Windows Operating System. You will also need the required hardware to make the connection.

Make sure a generic / text only printer driver is installed on the Host computer. This can be verified by going into the Windows Control Panel and opening the Printers and Faxes folder. If there is no generic / text only driver installed, it can be installed by clicking **Add a Printer**. When Windows prompts you for the type of printer being used, select local printer attached to this computer. When prompted to install the printer software, select Generic from the manufacturers list and Generic/Text Only from the list of available printer drivers.

When the printer driver has been installed, the connection can be tested by accessing the driver properties and clicking **Print Test Page**. Before printing a test page, make sure that Port Monitor is running and the message Connected is displayed in the information box on the lower-right of its main interface. This will send data from the Host computer into the Parallel connection. If everything is working correctly, a data file containing the test page text should appear in Port Monitor’s Data folder.

TCP/IP (Port 9100)

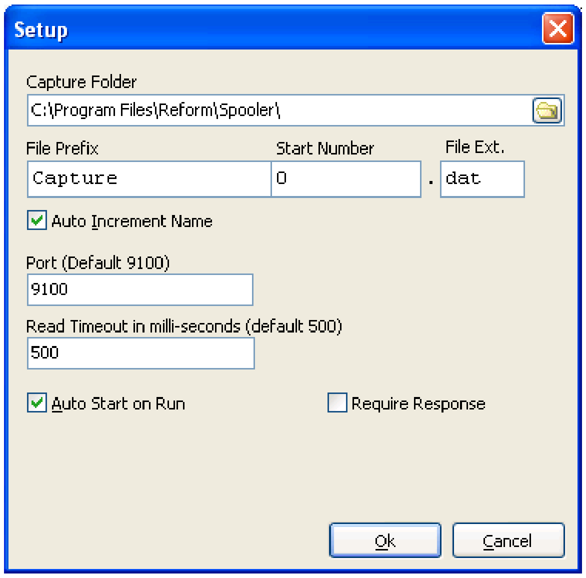
To capture a data streams being sent from a host application to the printer via a standard TCP/IP connection (by default port 9100 is used), the Reform 9100 Port Capture Utility should be used on the Reform Server. In this scenario, the Host system must first be configured to print the Server running the Capture Utility. Typically, the host system will be directed to the Reform Server by specifying the IP address of the server in the print settings. Once the host system is configured, The 9100 capture utility should be started on the Server. 9100 Port Capture Utility resides in Reform’s Miscellaneous folder. To start Port Monitor, simply double click on the file called **9100PortCapture.exe**.



Capture Folder: C:\Program Files\Reform\Misc\data\

Active | Port:9100

Once the application is started, go to File > Settings on the menu. Here, you can configure the directory that the data will be saved to. Typically the files should be saved to Reform's Spooler folder.



Capture Folder: C:\Program Files\Reform\Spooler\
File Prefix: Capture
Start Number: 0
File Ext.: dat
[✓] Auto Increment Name
Port (Default 9100): 9100
Read Timeout in milli-seconds (default 500): 500
[✓] Auto Start on Run
[ ] Require Response

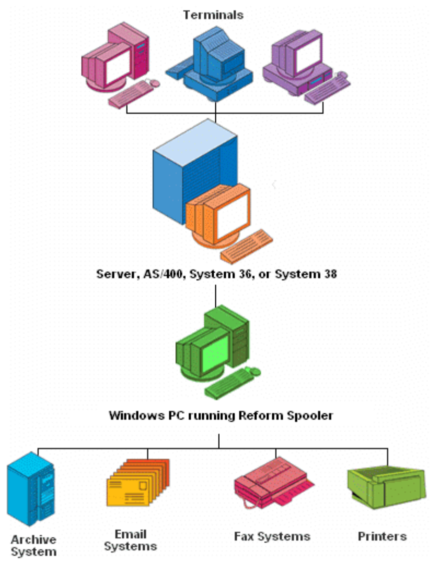
AS/400, System 36, System 38

Configuring AS/400

The AS/400 capture method is referred to as **Inbound Connectivity**. Inbound Connectivity is the process of redirecting an application's print stream into a file so the Reform Spooler can automatically process the information.

Methods used to accomplish Inbound Connectivity

The first method uses AS/400 native functions and is called Line Printer Requester and Line Printer Daemon or LPR/LPD for short. The AS/400 printer Output Queue is redirected to a remote printing device LPR. This printing device is the Reform Printer driver that resides on a Windows computer LPD. This method uses TCP/IP to connect the AS/400 and the Windows computer.


Terminals
Server, AS/400, System 36, or System 38
Windows PC running Reform Spooler
Archive System
Email Systems
Fax Systems
Printers

Create AS/400 Output Queue

To Create an AS/400 Output Queue, type **CRTOUTQ** in the AS/400 command prompt and then press **F4**, **F10**. Once the Create Output Queue screen appears, fill it in from the example below. The Output queue can be any name; in this example we used **reformprn**, but you should use a name the means something to the solution.

```
Create Output Queue (CRTOUTQ)
Type choices, press Enter.

Output queue . . . . . . . . . . . > TESTPRT04_ Name
 Library . . . . . . . . . . > QUSRSYS___  Name, *CURLIB
Maximum spooled file size: _
 Number of pages. . . . . . *NONE_____ Number, *NONE
 Starting time . . . . . . .  __________ Time
 Ending time . . . . . . . .  __________ Time
  + for more values _
Order of files on queue. . . . . . *FIFO_____ *FIFO, *JOBNBR
Remote System. . . . . . . . . . .  *NONE________________________________
```

____
_
User defined option. . . . . *NONE Option, *NONE
 + for more values __

Remove Printer Codes

In order to remove printer codes you must first edit the WSCST object.

To Retrieve the WSCST object type in **RTVWSCST** at the AS/400 command prompt.

```
Retrieve WSCST source (RTVWSCST)
Type choices, press Enter.
Device type. . . . . . . . . . . >  *TRANSFORM *TRANSFORM, 3101, 3151
Manufacturer type and model . . >  *WSCSTNONE__
Source member. . . . . . . . . . >  TEXTONLY___ Name
Source file. . . . . . . . . . . >  QTXTSRC____ Name
 Library . . . . . . . . . >  QUSRSYS Name, *CURLIB, *LIBL
Text ‘description’ . . . . . . .  *BLANK_____________________________________
```

Press **Enter** to end this section.

Type **strseu srcfile(qusrsys/qtxtsrc) srcmbr(textonly)** at the AS/400 command prompt. This will open up the WSCST object in a text editor.

```
Columns …………: 1 71  Edit   QUSRSYS/QTXTSRC
SEU
_______________________________________________________________________
TEXTONLY
 ******************* Beginning of data *********************
0000.01 :WSCST DEVCLASS=TRANSFORM
0000.02
0000.03  :TRNSFRMTBL.
0000.04  :INITPRT
0000.05   DATA =’00’X.
0000.06  :SPACE
0000.07   DATA =’20’X.
0000.08  :CARRTN
0000.09   DATA =’0D’X.
0000.10  :FORMFEED
0000.11   DATA =’0C’X.
0000.12  :LINEFEED
0000.13   DATA =’0A’X.
0000.14 :EWSCST.
 ************************** End of data ********************
```

Remove the lines for INITPRT, SPACE and CARRTN. When finish your results should look like:

```
Columns ...........: 1 71  Edit   QUSRSYS/QTXTSRC
SEU⍰ ____________________________________________________________ TEXTONLY
******************* Beginning of data *******************
0000.01 :WSCST DEVCLASS=TRANSFORM
0000.02
0000.03  :TRNSFRMTBL.
0000.10  :FORMFEED
0000.11   DATA ='0C'X.
0000.12  :LINEFEED
0000.13   DATA ='0A'X.
0000.14 :EWSCST.
************************** End of data ******************
```

To save the changes to the WSCST object, type **CRTWSCST** at the AS/400 command prompt.

```
Create WSCST (CRTWSCST)
Type choices, press Enter.

Workstation customizing object > TEXTONLY__ Name
Library. . . . . . . . . . >  QUSRSYS Name, *CURLIB
Source member . . . . . . . . . .  *WSCST__  Name, *WSCST
Text 'description'. . . . . . . . *SRCMBRTXT___________________


Additional Parameters
Source file . . . . . . . . . . . > QTXTSRC__ Name
Library. . . . . . . . . . > QUSRSYS Name, *CURLIB
Authority . . . . . . . . . . . .  *LIBCRTAUT__ Name, *LIBCRTAUT, *CHANGE
Text 'description'. . . . . . . . *SRCMBRTXT___________________
```

Configure AS/400 Output Queue

To Change an AS/400 Output Queue, type **CHGOUTQ** at the AS/400 command prompt. Change the Output Queue setting to direct the Output Queue stream to a Network IP.

```
Create Output Queue (CRTOUTQ)
Create Output Queue (CRTOUTQ)

Type choices, press Enter.

Output queue. . . . . . . . . . > TESTPRT04__ Name
Library. . . . . . . . . >  QUSRSYS__ Name, *CURLIB
```

Maximum spooled file size: _
Number of pages. . . . . *NONE______ Number, *NONE
Starting time. . . . . . ___________ Time
Ending time. . . . . . . ___________ Time
+ for more values _
Order of files on queue . . . . *FIFO______ *FIFO, *JOBNBR
Remote system . . . . *INTNETADR_____________________
____
____
____
Remote printer queue. . . . . . > 'REFORMPRN'_________________
____

Press the **Page-Down** button to view the next screen.

This screen allows you to specify the IP address of the computer that will be running the Reform Spooler. In the example below, the Reform Spooler is running on a computer whose IP address is 10.0.0.233.

Change Output Queue (CHGOUTQ)

Type choices, press Enter.

Writers to autostart. . . . . 1_________ 1-10, *SAME, *NONE
Queue for writer messages . . . . . QSYSOPR___ Name, *SAME
Library. . . . . *LIBL___ Name, *LIBL, *CURLIB
Connection type . . . . *IP_______ *SAME, *SNA, *IP, *IPX
Destination type. . . . . *OTHER____ *SAME, *OS400, *OS400V2
Host print transform. . . . . *YES______ *SAME, *YES, *NO
User data transform . . . . *SAME_____ Name, *SAME, *NONE
Library. . . . . ________ Name, *LIBL, *CURLIB
Manufacturer type and model . . . . *WSCSTNONE__________
Workstation customizing object. . . TEXTONLY__ Name, *SAME, *NONE
Library. . . . . *LIBL___ Name, *LIBL, *CURLIB
Image Configuration . . . . *NONE__________ *SAME, *NONE, *IMGA01
Internet address. . . . . '10.0.0.233_____'
VM/MVS class. . . . . *SAME *SAME, A, B, C, D, E, F, G
Forms Control Buffer. . . . . *SAME____ Character value, *NONE

More…

Press the **Page-Down** button to view the next screen.

Change Output Queue (CHGOUTQ)

Type choices, press Enter.

Destination options. . . . . ’XAIX XAUTOO’___________________________
____

Print separator page . . . . *NO_______ *SAME, *YES, *NO
User defined option. . . . . *NONE_____ Option, *SAME, *NONE
+ for more values __________
User defined object:
Object. . . . . *NONE_____ Name, *SAME, *NONE
Library. . . . . ___________ Name, *LIBL, *CURLIB
Object type . . . . _______ *DTAARA, *DTAQ, *FILE…
User driver program . . . . *NONE_____ Name, *SAME, *NONE
Library . . . . ___________ Name, *LIBL, *CURLIB
Spooled file ASP. . . . . *SYSTEM___ *SAME, *SYSTEM, *OUTQASP
Text ‘description’. . . . . *BLANK________________________________

More…

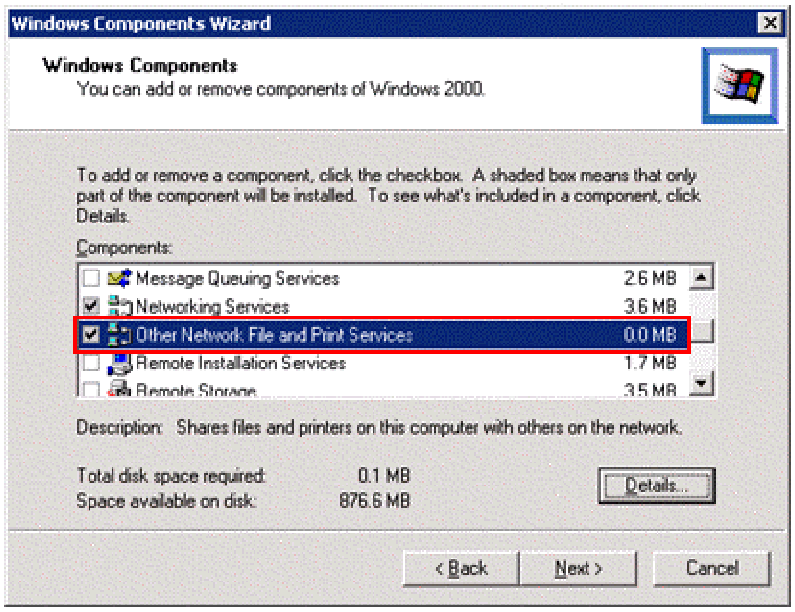
Press the **Page-Down** button to view the next screen.
Change Output Queue (CHGOUTQ)
Type choices, press Enter.

Additional Parameters

Display any file. . . . . *NO_________ *SAME, *NO, *YES, *OWNER
Job separators. . . . . 0___________ 0-9, *SAME, *MSG
Operator controlled . . . . *YES________ *SAME, *YES, *NO
Data queue. . . . . *NONE_______ Name, *SAME, *NONE
Library. . . . . ____________ Name, *LIBL, *CURLIB
Authority to check. . . . . *OWNER______ *SAME, *OWNER, *DTAAUT

LPD Windows 2000/XP/2003

To configure LPD Windows 2000, go to **Control Panel >> Add/Remove Programs >> Add/Remove Window Components >> Other Network File and Print Services >> Printer Services for UNIX**. Select **Other Network File and Print Services**, as shown below.


Windows Components Wizard

Windows Components
You can add or remove components of Windows 2000.

To add or remove a component, click the checkbox. A shaded box means that only part of the component will be installed. To see what's included in a component, click Details.

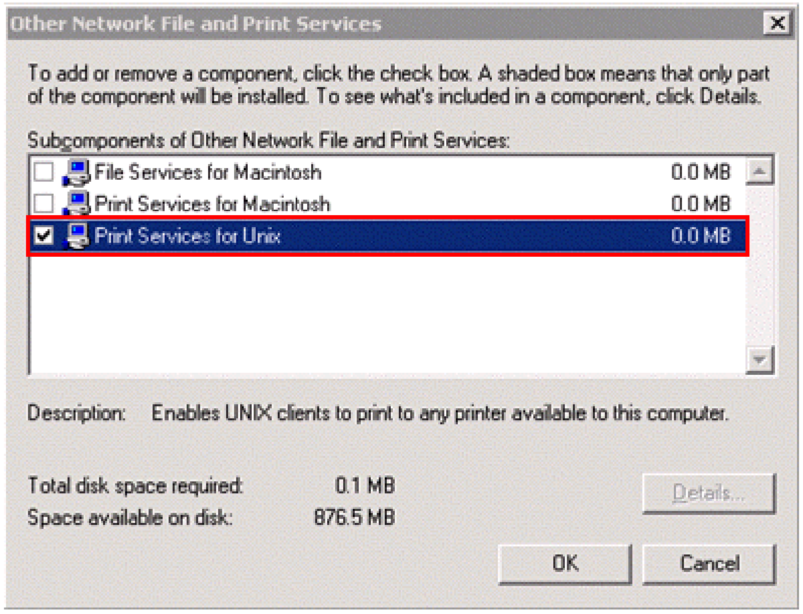
Components:

☐	Message Queuing Services	2.6 MB
✓	Networking Services	3.6 MB
✓	Other Network File and Print Services	0.0 MB
☐	Remote Installation Services	1.7 MB
☐	Remote Storage	3.5 MB

Description: Shares files and printers on this computer with others on the network.

Total disk space required: 0.1 MB
Space available on disk: 876.6 MB

Select **Print Services for Unix**.


Other Network File and Print Services

To add or remove a component, click the check box. A shaded box means that only part of the component will be installed. To see what's included in a component, click Details.

Subcomponents of Other Network File and Print Services:

☐	File Services for Macintosh	0.0 MB
☐	Print Services for Macintosh	0.0 MB
✓	Print Services for Unix	0.0 MB

Description: Enables UNIX clients to print to any printer available to this computer.

Total disk space required: 0.1 MB
Space available on disk: 876.5 MB

Select **OK** to install service.

LPR/LPD Reference Information

The LPD service listens on TCP/IP port 515.

Unix and Linux Operating Systems

Configuring Unix & Linux

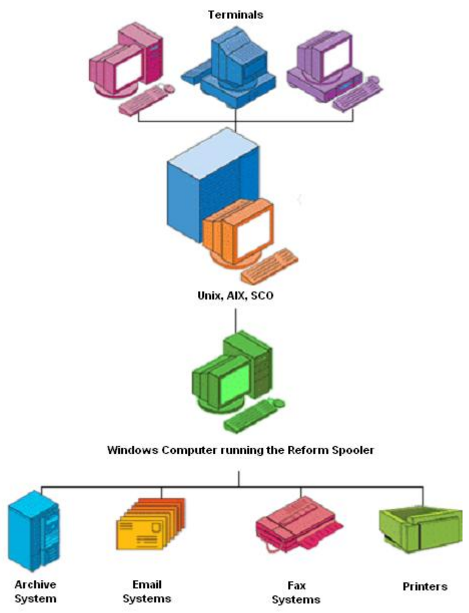
The Unix and Linux capture method is referred to as **Inbound Connectivity**. Inbound Connectivity is the process of redirecting an application’s print stream into a file so that the Reform Spooler can automatically process the information.

Methods used to accomplish Inbound Connectivity

The first method uses Unix/Linux native functions, and is called Line Printer Requester and Line Printer Daemon, or LPR/LPD for short. The Unix/Linux printer Output Queue is redirected to a remote printing device LPR. This printing device is the Reform Printer driver that resides on a Windows computer LPD. This method uses TCP/IP to connect the Unix or Linux machine to the Windows computer.

The second method uses file transfer to allow the Reform Spooler to process your print streams. First, create a directory on the UNIX file server and call it "ReformSpooler" or something to that effect. This directory will serve as a shared directory between the UNIX machines and the Windows machine running the Reform Spooler. In Reform setup, change the **Spooler Directory Location** to the shared spooler location on the UNIX file server or the Windows machine. The UNIX applications will need to redirect their printouts to this shared spooler directory in order for Reform to detect and process the output files. There are many utilities on the market to make the connection between the UNIX and Windows computers; the two that we recommend are:

Samba – www.samba.org
FacetWin by FacetCorp – www.facetcorp.com



Setting up a Unix LPR Printer

SCO Unix

Setup steps

1. Go to **scoadmin >> Printers >> Printer Manager**.
2. Go to **Printer >> Add Remote >> Unix**.

The Host name is the IP address of the windows computer running the Reform Spooler.

The Printer name is **reformprn**.

Your Unix screen should look like the following, though the Host IP address may differ:

```
---------------------------- Add Remote UNIX Printer ----------------------------------------
Host 192.168.0.3________________________________________| [ Select... ]
Printer reformprn _____________________________________| [ Select... ]
[ ] Use extended remote printing protocol
[ OK ] [ Cancel ] [ Help ]
```

Note: In some systems like AIX you cannot use an IP address for the host address, you must add a name to the Host file, this host file is usually located in the /etc directory. For example in the HOST file you have:

REFORM 192.168.0.3

When you add your printer queue you would use the name REFORM instead of the IP address.

Setting up a Linux LPR Printer

Linux Redhat 7.2 KDE

Go to System >> Printer Manager, and adjust the following settings:

First Dialog:

Queue Name: ReformPrinter

Type: Unix Printer LPD

Second Dialog:

Server: <Windows IP #>

Queue: reformprn

Note: the Windows IP # is typically the IP address of the windows computer running the Reform Spooler.

Third Dialog:

Printer: Text Only Printer

Other Linux system can edit the printcap file directly. The queue name in the printcap file located in the /etc/printcap directory should look like the following:

```
ReformPrinter:\
:sh:\
:ml=0:\
:mx=0:\
:sd=/var/spool/lpd/ReformPrinter:\
:af=/var/spool/lpd/ReformPrinter/ReformPrinter.acct:\
:rm=192.168.0.3:\
:rp=reformprn:\
:lpd_bounce=true:\
:if=/usr/share/printconf/util/mf_wrapper:
```

Note: Change the rm ip # to reflect your Windows IP address.

AIX LPR Setup instructions

# vi /etc/hosts

In vi add an entry for Reform <IP address of windows machine that is running Reform>

Note to insert a new line move down to the location and press [ESC] Shift i

To quit and save press [ESC] : then wq

Vi command:

Text Input Mode

Text input mode can be used to enter and delete new text in your file. Below is a table showing the various key sequences to place you into text input mode as well as those to return to command mode and perform operation while in text input mode. When you are in text input mode, everything you type will be placed into your document until you press the [ESC] key.

```
Key Action Key Action
----------------------------------------------------------
a Insert after cursor A Append to end of line
i Insert before cursor I Insert at beginning of line
o Insert line below cursor O Insert one line above cursor
[CTRL][J] Move down one line [CTRL][W] Move back one word
[ENTER] Add a new line [Backspace] Move back one character
kill Delete line (set by stty) [CTRL][H] Move back one character
[CTRL][I] Insert a tab [CTRL][T] Move to next tab setting
[ESC] Returns to command mode [CTRL][V] Quote next character
```

# smit

System Management

Move cursor to desired item and press Enter.

Software Installation and Maintenance
Software License Management
Devices
ystem Storage Management (Physical and Logical Storage)
Security & Users
Communications Applications and Services
> Print Spooling **
Problem Determination
Performance & Resource Scheduling
System Environments
Processes and Subsystems
Remote Customer Support and Services
Applications
Cluster Systems Management
Using SMIT <information only>

Move cursor to desired item and press Enter.

AIX Print Mode Only:

Start a Print Job
Manage Print Jobs
List All Print Queues
Manage Print Queues
> Add a Print Queue **
Add and Additional Printer to an Existing Print Queue
Change / Show Print Queue Characteristics
Change / Show Printer Connection Characteristics
Remove a Print Queue
Manage Print Server
Programming Tools

AIX and System V Print Mode:

Change / Show Current Print Subsystem

Move cursor to desired item and press Enter.

AIX Print Mode Only:
Start a Print Job
Manage Print Jobs
List All Print Queues
Manage Print Queues
> Add a Print Queue **
Add and Additional Printer to an Existing Print Queue

Type of Remote Printing

Move cursor to desired item and press Enter.

Standard Processing **
Standard with NFS access to server print queue attributes
Local filtering before sending to print server

Add a Standard Remote Print Queue
Type or select values in entry fields.
Press Enter AFTER making all desired changes.
[Entry Fields]

1. Name of QUEUE to add [reformprn]
2. HOSTNAME of remote server [reform]
3. Name of QUEUE on remote server [reformprn]
4. Type of print spooler on remote server AIX Version 3 or 4
   Backend TIME OUT period <minutes>
   Send control file first? []
   To turn on debugging, specify output file pathname no
   DESCRIPTION of printer on remote server [reform]

Quick test:
At the AIX prompt:

lp –d reform /etc/hosts

you should see a file appear in the \Reform…\Spooler directory
Note make sure the Reform Spooler is off or in the stop position.

Trouble Shooting:

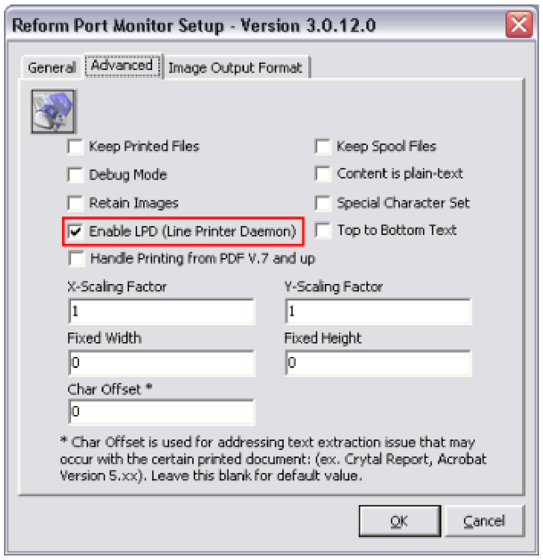
On the Reform windows machine go to DOS prompt:
Type ipconfig
Note this should return the IP address that you specified in the AIX host file.

Windows Type NETSTAT –NA
Note you should see a port 515 that is in the listening state, if not the TCP/IP Print windows service may not be running. To check windows service go to Control Panel >> Administration Tools >> Services.

AIX prompt type lpstat –preformprn to see that status of the job

AIX prompt ping the windows machine by typing:
ping reform
Note if ping fails the /etc/hosts table may have the wrong IP number, the IP number should be the same as the ipconfig results from the windows machine.

Configuring the Reform Printer
Note: Before proceeding, you must install the Reform Printer on the machine running Reform. See the section titled 'Reform Printer Driver' in this manual for Reform Printer installation instructions.
Once the Reform Printer has been installed, it must be configured to accept print jobs. To do so, follow the steps below:
Open your Windows Control Panel on the machine running Reform (**Start >> Control Panel**), and double-click **Reform Port Monitor Setup**.
Click the **Advanced** tab, and make sure the box labeled 'Enable LPD (Line Printer Daemon)' is checked, as illustrated below.


Reform Port Monitor Setup - Version 3.0.12.0

- [ ] Keep Printed Files
- [ ] Keep Spool Files
- [ ] Debug Mode
- [ ] Content is plain-text
- [ ] Retain Images
- [ ] Special Character Set
- [x] Enable LPD (Line Printer Daemon)
- [ ] Top to Bottom Text
- [ ] Handle Printing from PDF V.7 and up

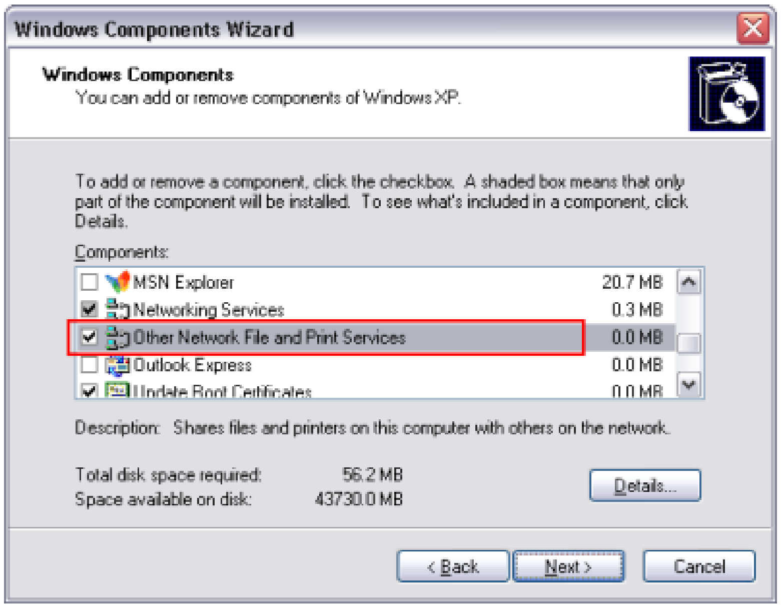
X-Scaling Factor: 1
Y-Scaling Factor: 1
Fixed Width: 0
Fixed Height: 0
Char Offset *: 0

* Char Offset is used for addressing text extraction issue that may occur with the certain printed document: (ex. Crytal Report, Acrobat Version 5.xx). Leave this blank for default value.

Click **OK** to exit the Reform Port Monitor Setup.

LPD Windows 2000/XP/2003

To configure LPD Windows 2000, go to **Control Panel >> Add/Remove Programs >> Add/Remove Window Components >> Other Network File and Print Services >> Printer Services for UNIX**. Select **Other Network File and Print Services**, as shown below.


Windows Components Wizard

Windows Components
You can add or remove components of Windows XP.

To add or remove a component, click the checkbox. A shaded box means that only part of the component will be installed. To see what's included in a component, click Details.

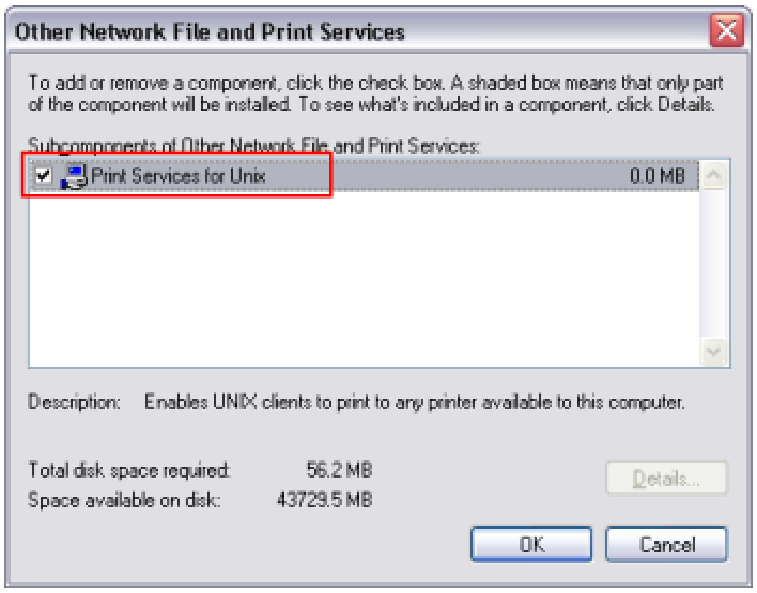
Components:

☐ MSN Explorer	20.7 MB
☑ Networking Services	0.3 MB
☑ Other Network File and Print Services	0.0 MB
☐ Outlook Express	0.0 MB
☑ Update Root Certificates	0.0 MB

Description: Shares files and printers on this computer with others on the network.

Total disk space required: 56.2 MB
Space available on disk: 43730.0 MB

Click the **Details** button Select **Print Services for Unix**.


Other Network File and Print Services

To add or remove a component, click the check box. A shaded box means that only part of the component will be installed. To see what's included in a component, click Details.

Subcomponents of Other Network File and Print Services:

✓ Print Services for Unix 0.0 MB

Description: Enables UNIX clients to print to any printer available to this computer.

Total disk space required: 56.2 MB
Space available on disk: 43729.5 MB

Select **OK** to install service.

LPR/LPD Reference Information

The LPD service listens on TCP/IP port 515.

Maintenance

There are certain directories that must be maintained after you start using Reform. Depending on the options you have configured in the Reform Spooler, the Spooler may create backups of all processed files into the Backups directory. If the Retain Images option is used with the Reform Printer driver, images of processed documents will be stored in the Spooler\ImageQueue directory. As a system processes more and more jobs, the size of these directories can grow very quickly. Overall system performance can be affected by the large amounts of space that the files may take up. Obsolete files should be deleted according to your needs and file retention specifications.

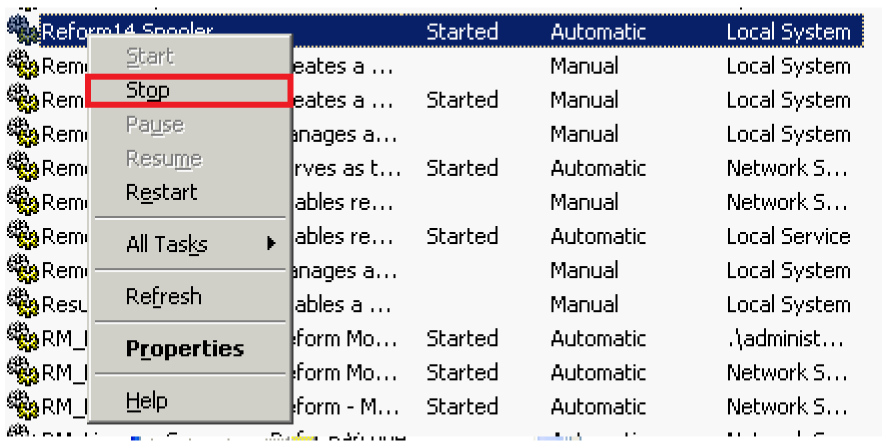
Please see the Health Monitor and Maintenance Utility section for more information and instructions for monitoring and maintaining your server. It is recommended to monitor the following directories and SQL tables for growth:

C:\Program Files\Reform…\Spooler\ImageQueue
C:\Program Files\Reform…\Backups

Troubleshooting

Q: *The Reform Spooler suddenly stops*.

A: If the Spooler is stopping or restarting erratically, close the Spooler application. Bring up your system's Services panel and find **Reform Spooler Service**. Refresh the Services list. Right-click the entry for Reform Spooler Service. Hit **Stop**. Open up the Spooler application [**Start > Programs > Reform... > Spooler**], and after it loads, click **Run**. The Reform Spooler Service should start up.



Reform14 Spooler		Started	Automatic	Local System
Rem	eates a ...		Manual	Local System
Rem	eates a ...	Started	Manual	Local System
Rem	anages a...		Manual	Local System
Rem	rves as t...	Started	Automatic	Network S...
Rem	ables re...		Manual	Network S...
Rem	ables re...	Started	Automatic	Local Service
Rem	anages a...		Manual	Local System
Resu	ables a ...		Manual	Local System
RM_	form Mo...	Started	Automatic	.\administ...
RM_	form Mo...	Started	Automatic	Network S...
RM_	form - M...	Started	Automatic	Network S...

If this does not work, try right-clicking the service and hitting **Restart**, and then re-launching the Spooler Application.

Alternatively, for information on what might be causing your Spooler to randomly turn off, you can check in the Event Log. To open the Event Log, open the Control Panel, go to Administrative Tools, open Event Viewer, select **Application** on the left pane, and look for errors with the source **reformEnt**.

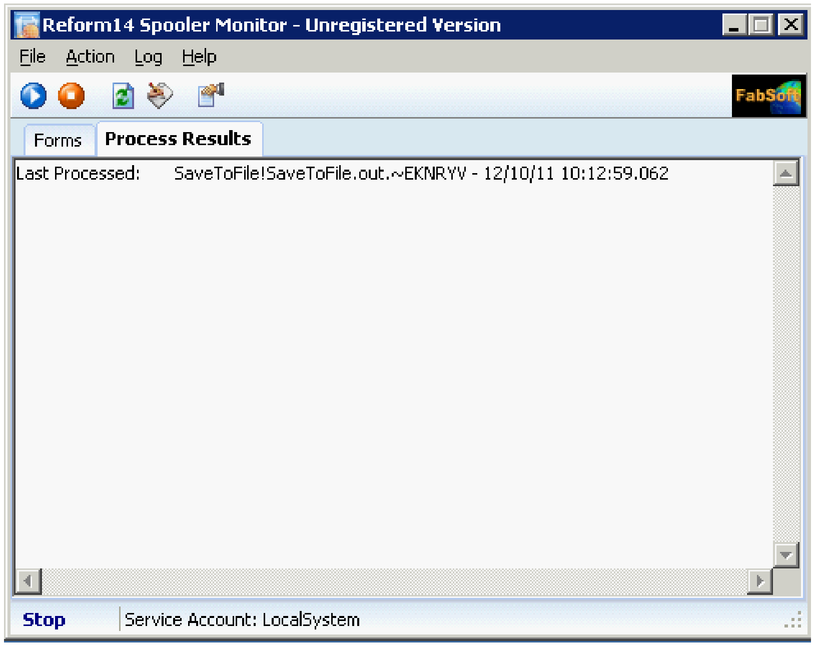
Q: *The Reform Spooler Service will not start.*

A: Be sure that the Spooler Service is being run as an administrator. To check this, open up the Services panel on the server computer. Look for **Reform Spooler Service**. First, refresh the Services panel to make sure that the service is definitely not running. Next, right-click on the Reform Spooler Service item and click **Properties**. Go to the **Logon** tab, select **This Account**, and enter an account that is an administrator on the server machine. It will need to run with full access because it needs to monitor folders and create, write to, rename, and move around files.

Q: *The Reform Spooler service will not start even though it is set to use an administrative account*.

A: If you have created a new administrator account, you must log in to the Reform server **at least once** so that the appropriate permissions can be applied to the new account. If you are setting the service to run under an account that you have not yet logged in under, the service will be unable to start correctly because it has not been configured by Windows yet to have full administrative access to the machine.

Q: *The Reform Spooler log states that it could not find any forms to match the files to*.
A: This is caused by the service account not having sufficient permission to the spooler file. If the detailed log in the Reform Spooler is enabled, it will say that the file is locked because it does not have permission to read from the file.

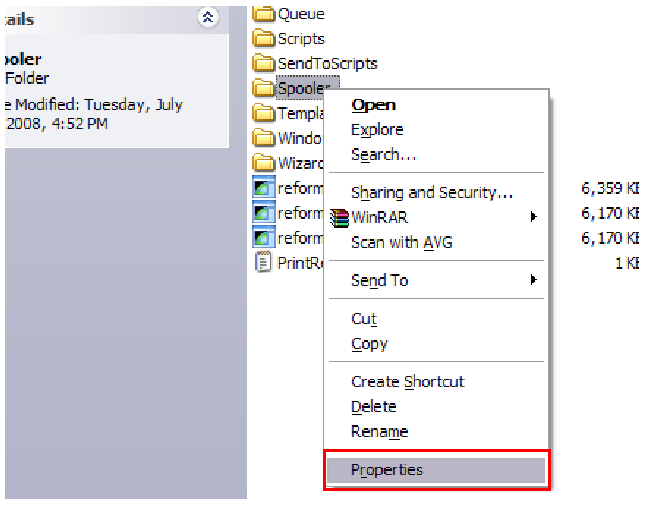


```
Last Processed:     SaveToFile!SaveToFile.out.~EKNRYV - 12/10/11 10:12:59.062
```

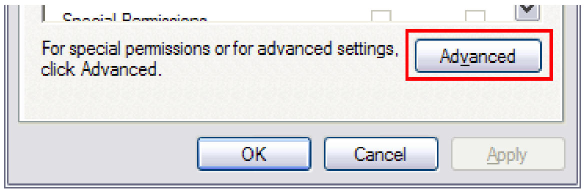
Service Account: LocalSystem

To correct this, you need to repair the security permissions on the Spooler folder.

1. Open a Windows Explorer window and navigate to **C:\Program Files\Reform...\**

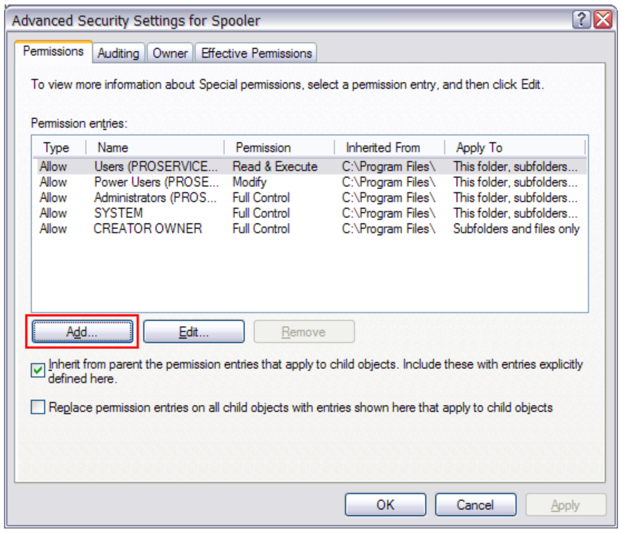


1. Right-click on the **Spooler** folder and click **Properties**.
2. Select the Security tab and click the **Advanced** button on the bottom of the window.



For special permissions or for advanced settings, click Advanced.

3. Click the **Add** button to add a new permission entry.



Advanced Security Settings for Spooler

To view more information about Special permissions, select a permission entry, and then click Edit.

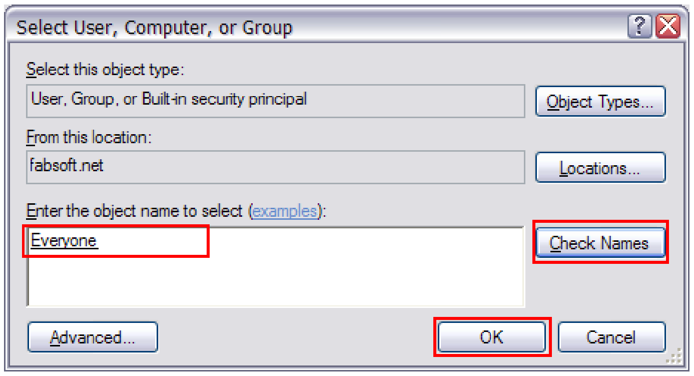
Permission entries:

Type	Name	Permission	Inherited From	Apply To
Allow	Users (PROSERVICE...	Read & Execute	C:\Program Files\	This folder, subfolders...
Allow	Power Users (PROSE...	Modify	C:\Program Files\	This folder, subfolders...
Allow	Administrators (PROS...	Full Control	C:\Program Files\	This folder, subfolders...
Allow	SYSTEM	Full Control	C:\Program Files\	This folder, subfolders...
Allow	CREATOR OWNER	Full Control	C:\Program Files\	Subfolders and files only

[✓] Inherit from parent the permission entries that apply to child objects. Include these with entries explicitly defined here.

[ ] Replace permission entries on all child objects with entries shown here that apply to child objects

4. Enter **Everyone** in the box and click **Check Names**. Then click **OK** to create the new permission entry.



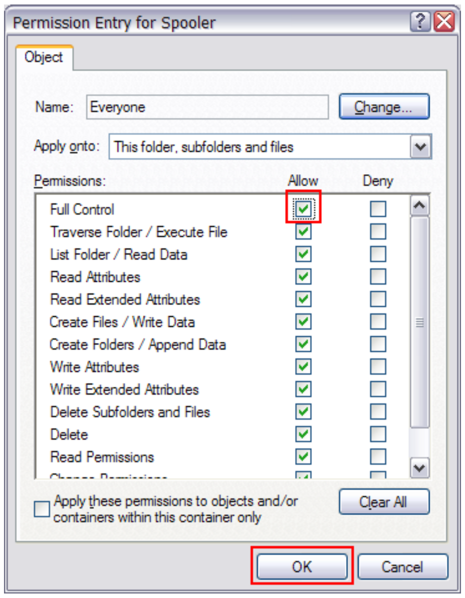
Select User, Computer, or Group

Select this object type:
User, Group, or Built-in security principal

From this location:
fabsoft.net

Enter the object name to select (examples):
Everyone

5. Select the checkbox for **Allow Full Control**. It will automatically check the rest of the boxes for you. Click **OK**.


Permission Entry for Spooler

Object

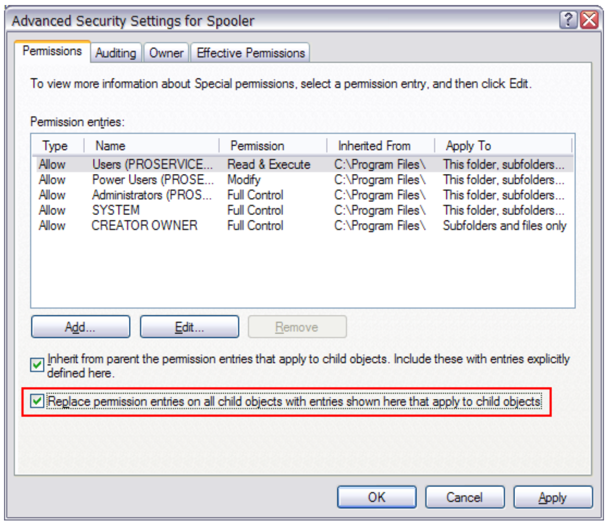
Name: Everyone

Apply onto: This folder, subfolders and files

Permissions:	Allow	Deny
Full Control	✓	
Traverse Folder / Execute File	✓	
List Folder / Read Data	✓	
Read Attributes	✓	
Read Extended Attributes	✓	
Create Files / Write Data	✓	
Create Folders / Append Data	✓	
Write Attributes	✓	
Write Extended Attributes	✓	
Delete Subfolders and Files	✓	
Delete	✓	
Read Permissions	✓	

☐ Apply these permissions to objects and/or containers within this container only

6. Check the box **Replace permission entries on all child objects with entries shown here that apply to child objects**. Then click **Apply**, and click **OK**.


Advanced Security Settings for Spooler

Permissions | Auditing | Owner | Effective Permissions

To view more information about Special permissions, select a permission entry, and then click Edit.

Permission entries:

Type	Name	Permission	Inherited From	Apply To
Allow	Users (PROSERVICE...	Read & Execute	C:\Program Files\	This folder, subfolders...
Allow	Power Users (PROSE...	Modify	C:\Program Files\	This folder, subfolders...
Allow	Administrators (PROS...	Full Control	C:\Program Files\	This folder, subfolders...
Allow	SYSTEM	Full Control	C:\Program Files\	This folder, subfolders...
Allow	CREATOR OWNER	Full Control	C:\Program Files\	Subfolders and files only

✓ Inherit from parent the permission entries that apply to child objects. Include these with entries explicitly defined here.

✓ Replace permission entries on all child objects with entries shown here that apply to child objects

The permissions of the folder should now be set correctly.

Index

ALLCOMMANDS 71
Area 11
  Mapping 11
ASC 72
Background Image 26
Barcode Object............. 39
BEEP 60
BOM 72
Call................................ 60
CASE 61
CASEOF......................... 61
CHDIR 72
Company-Lookup Field. 21
Converting CSV to a Lanier PDC usable format........................ 73
CURDIR 73
Decision Maker Introduction................ 46
Design Layer 7
Design Tips 12
Device Script 52
Duplex Form Name 26
Edit Menu..................... 16
ELSE 67
Email Fields 22
END............................... 63
ENDCASE 61
ENDFOR........................ 64
ENDFUNC 65
ENDIF............................ 67
ENDMACRO.................. 81
ENDSUB........................ 71
ENDWHILE.................... 71
ERRORLOG.................... 74
Execute......................... 64
EXITFOR........................ 74
EXITWHILE.................... 75
Fax Fields...................... 21
FEEDBACK..................... 75
FOR............................... 64
Form Description 26
Form Key Field.............. 20
Form Link works 27
Form Password............. 26
Form Properties 26
FTPCONNECT................ 76
FTPDISCONNECT 76
FTPEXEC........................ 76
FUNC 65
GETDIR 77
GETFORMNAME........... 78
GLOBAL 66
Goto 67
HIDEFIELD..................... 78
HOMEDIR 80
IF.................................. 67
Image Linking 43
Image Object................ 40
INCLUDE 68
Information 6
INPUT 69, 81
Keyboard Shortcuts....... 13
Link Form Name 26
LOADFORMVARS.......... 84
LOADVARS.................... 81
MACRO......................... 81
Mapping 11
  Area 11
MKDIR 82
NOERROR 82
Object Alignment 41
Object Basics 30
Options Menu 17, 23
OSTYPE 83
OTHERWISE 61
Page Script 52
Page Setup 45
Paragraph Object 35
PAUSE........................... 69
Pre-Process Script 26
Print.............................. 69
Printer Profile............... 44
PROMPT 70
RANDOM 83
Reform Directory Variables.................... 73
Reform File Name Variables.................... 76
RESULT 84
Retain Images............... 11
RETURN 70
Rich Paragraph Object .. 36
ROUND 84
Save 15
Save As 15
SAVEFORMVARS 84
SAVEVARS..................... 85
Shape Object................ 38
SHOWFIELD 78
SMTPDISCONNECT........ 87
SMTPSEND 87
SMTPSETUP 87
Special Fields Menu 19
SUB 71
SUB..ENDSUB 65
Techniques 12
Text Layer....................... 9
THEN............................. 67

TO 64
UNIQUEFILE................... 87
VARBYNAME 88
WAITDOS....................... 88
WAITWIN....................... 89
WHILE 71

Reform Issue Ticket

*User's Name: ______________________________________ *User's Login Name: _________________________________

*User's Email Address: ______________________________ User's Building: _____________________________________

User's Department: _________________________________ Support Contact's Name: ______________________________

Support Contact's Email: ____________________________ Support Contact's Phone: _____________________________

***Severity of Issue (Please check one):**

- ☐ 0 - Test environment (non-production)
- ☐ 1 - Users are affected, but able to continue business
- ☐ 2 - Users are affected and it is affecting business, other functionality is working properly
- ☐ 3 - The software is not functioning properly at all – system outage

Number of Users Affected: ______________ List Each Users Login Name ___________________________________

____

____

***Description of Issue:**

____

____

____

____

***Observations about the issue:**

Frequency of Issue: Once / Random / Every Attempt / Other (Please Describe): ___________________________________

____

First Occurrence (Date/Time): ______________________ Latest Occurrence (Date/Time): _________________________

*Issue is occurring at (Fill in all fields for Reform Server):

- <u>Reform Server(s)</u>

  Reform Server IP(s): ____

  Reform Version: ________________________________ Service Account: _________________________________

  Is the Reform Server licensed? yes / no System ID: _______________________________________

  Operating System: ______________________________ Anti-Virus Software: ______________________________

*** Additional Information**

*Have any changes been made to the Reform Server or Environment? yes / no If yes, What? When? By Who?

____

____

____

*What Plug-ins are being used? ____

____

*Is the service account a local administrator? yes / no

*Does the Reform Spooler service start? yes / no If no, what is the message that appears? ____________________

____

*When logging onto Reform Server are you logging in as the service account? yes / no

*What is the default printer when logged in as the service account on Reform Server? ___________________________

*Have any new printers been added or printer drivers been updated recently? yes / no If yes, Please explain

____

____

Is Remote Desktop being used to connect to the server? yes / no If yes, Is printer mapping disabled in the client settings tab of the RDP-TCP properties? yes / no

*Have any changes been made to the Forms? yes / no If yes, please explain _______________________________

____

____

What type of system is being used to print from? (Please check one)

☐ - AS400	☐ - Windows
☐ - UNIX	☐ - Other _______________________

*Have any changes or updates been applied to the antivirus software used on the Reform Server? yes / no

If yes, What? When? ____

____

*Have any changes been made to the device scripts or page process scripts? yes / no If yes, What? When? _______

____

____

*** Fields are required**