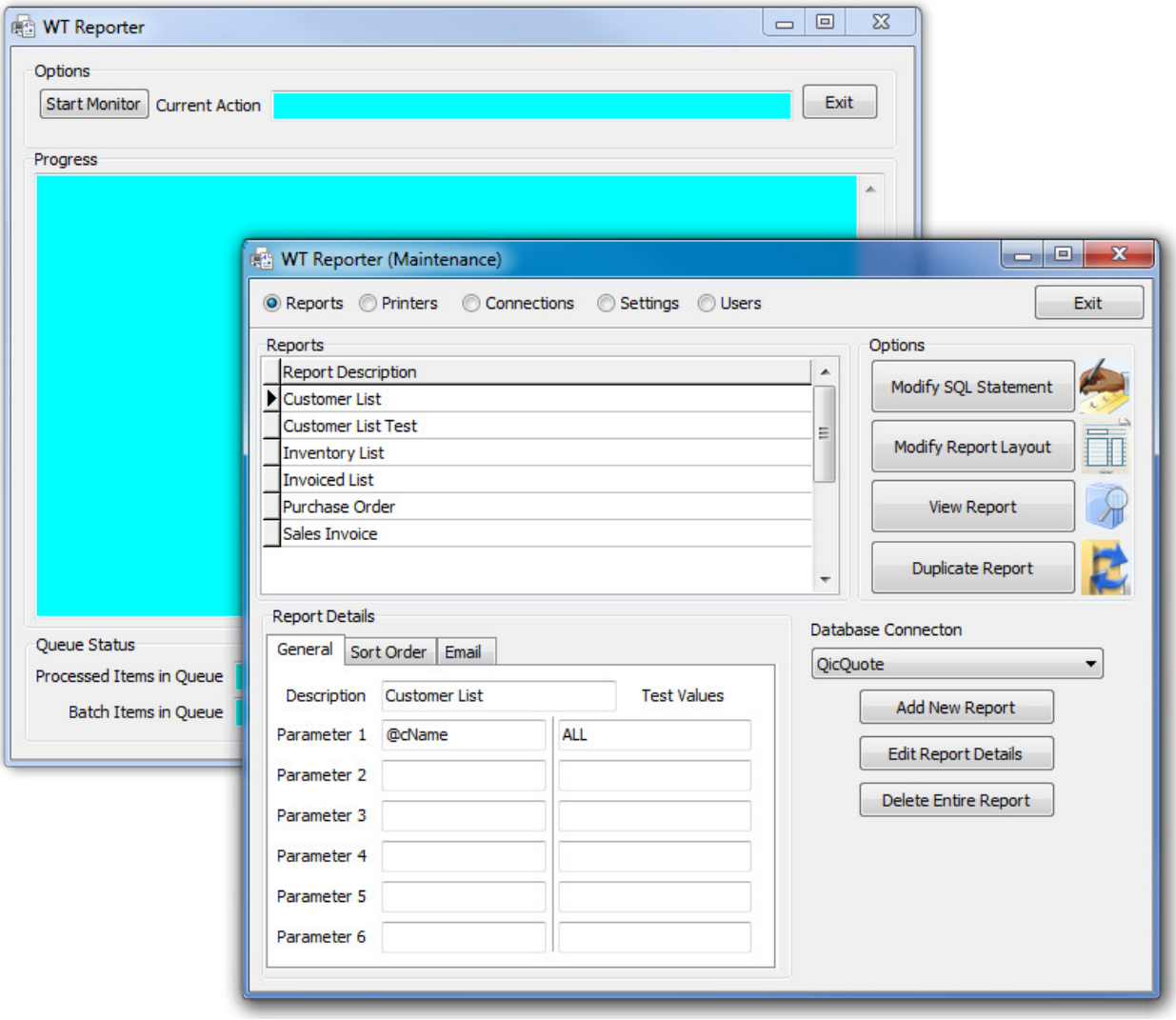
# *WT* Reporter

**Quick Reference Manual**




## Reporting from data made easy

***W**INTECH* **Computer Systems, Inc.**

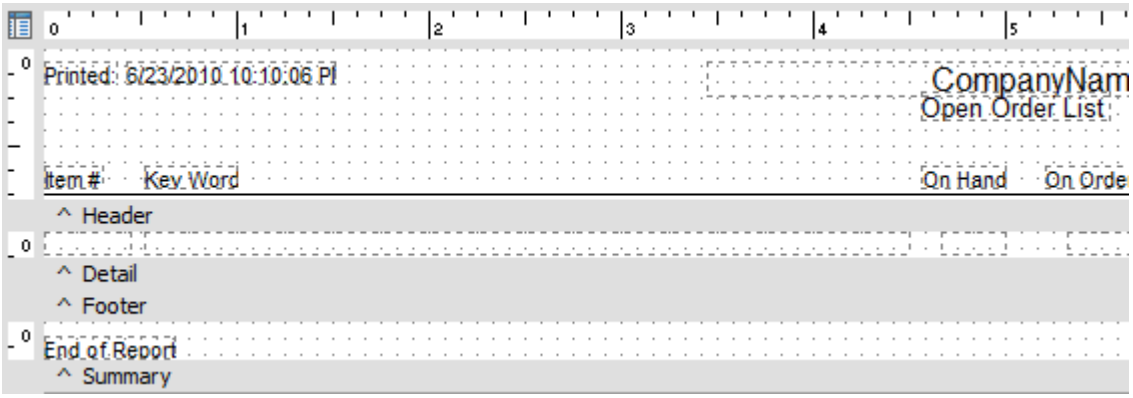
July 2010

# Contents

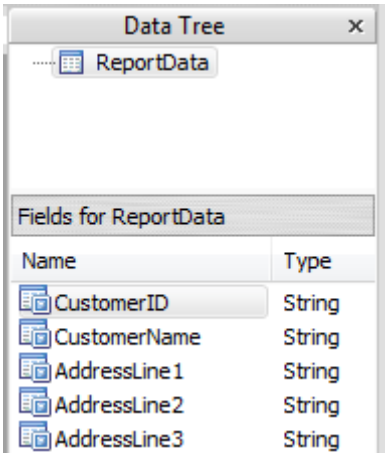
# Introduction

***WT*** **Reporter** is a simple way to get beautiful looking reports from your data the easy way. If you know how to write SQL commands to gather information from a database, then you will find this reporting engine easy to use and quick. Reports are printed directly to printers, PDF files or sent to email recipients as attachments. A simple command line interface is all that is needed to fire the program into action.

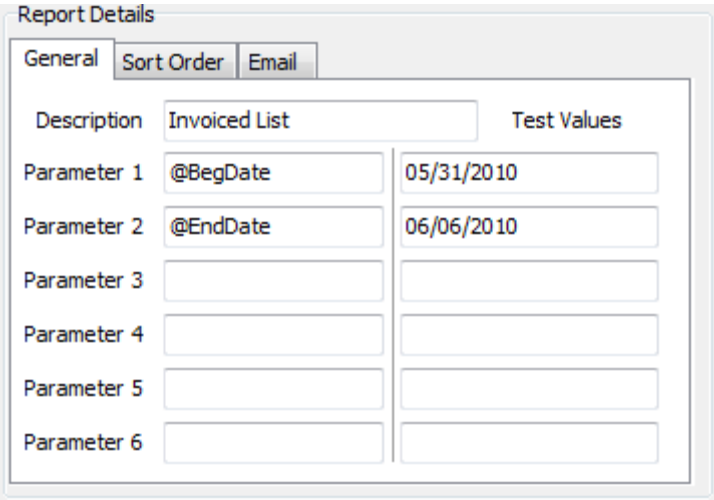
Reports are created using a common drag & drop method on a banded style report layout



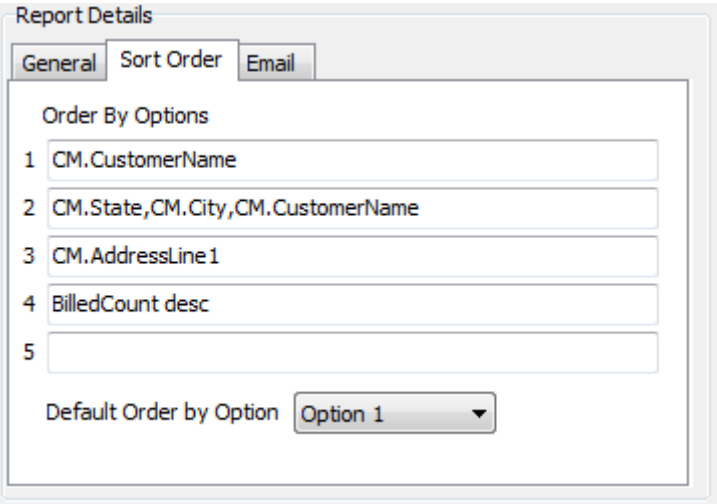

All the columns returned in the SQL select statement are available for dragging onto the report layout



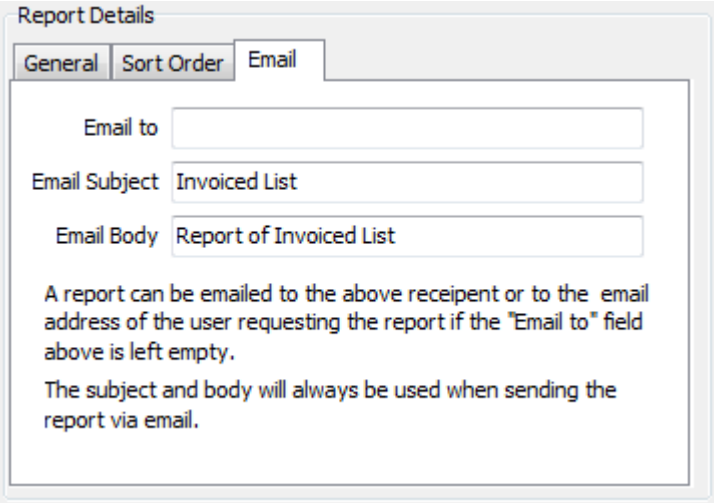

Each report can be passed up to six parameters



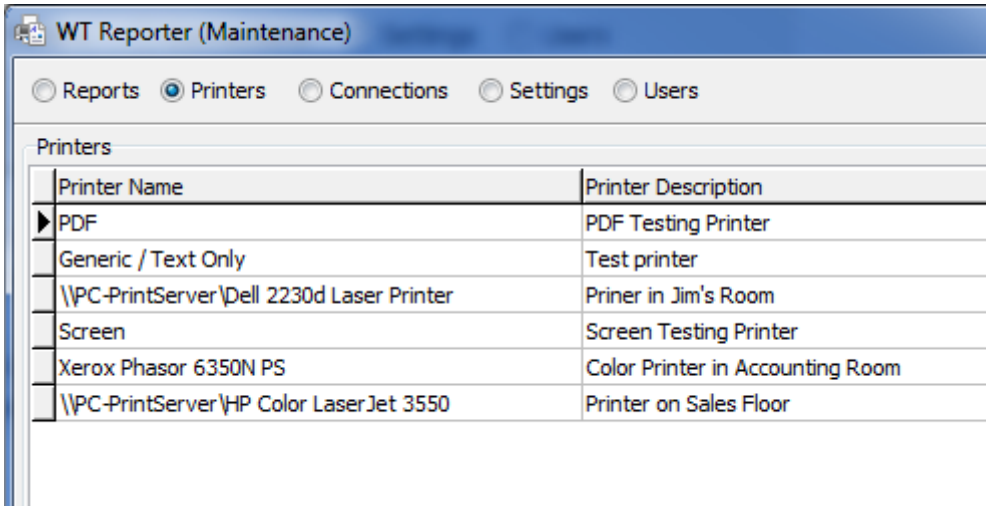

Reports can be sorted up to five different ways



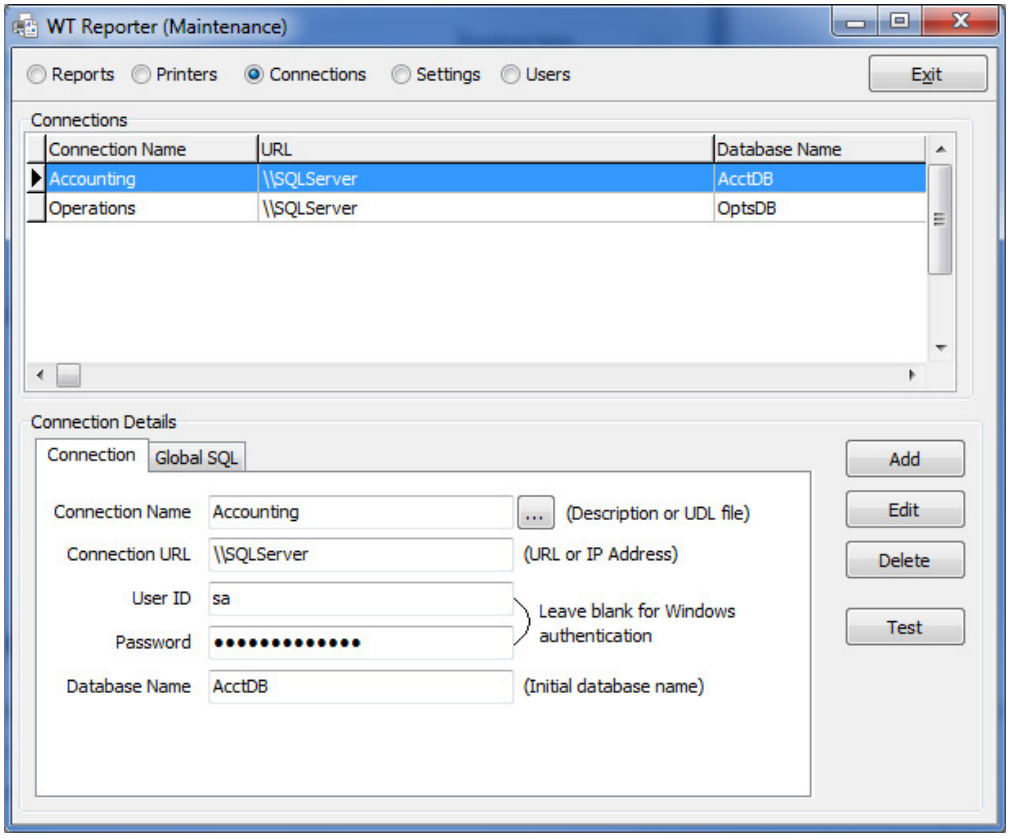

Reports can be emailed to a default email address or sent back to the requestors email address if desired.



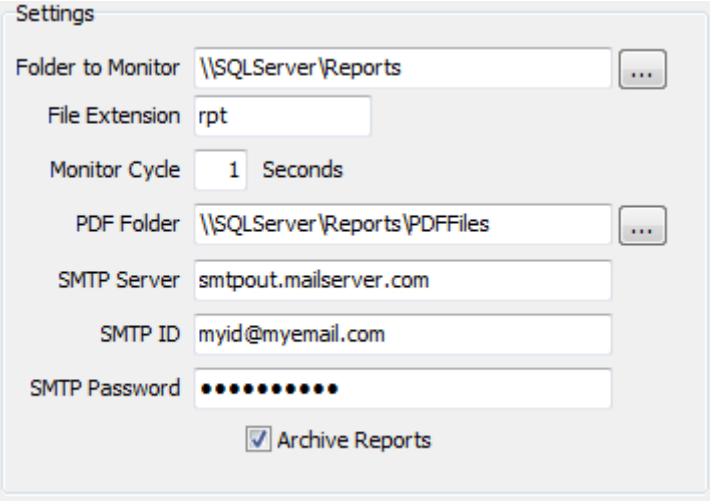

***WT*** **Reporter** can interface with all printers on your system



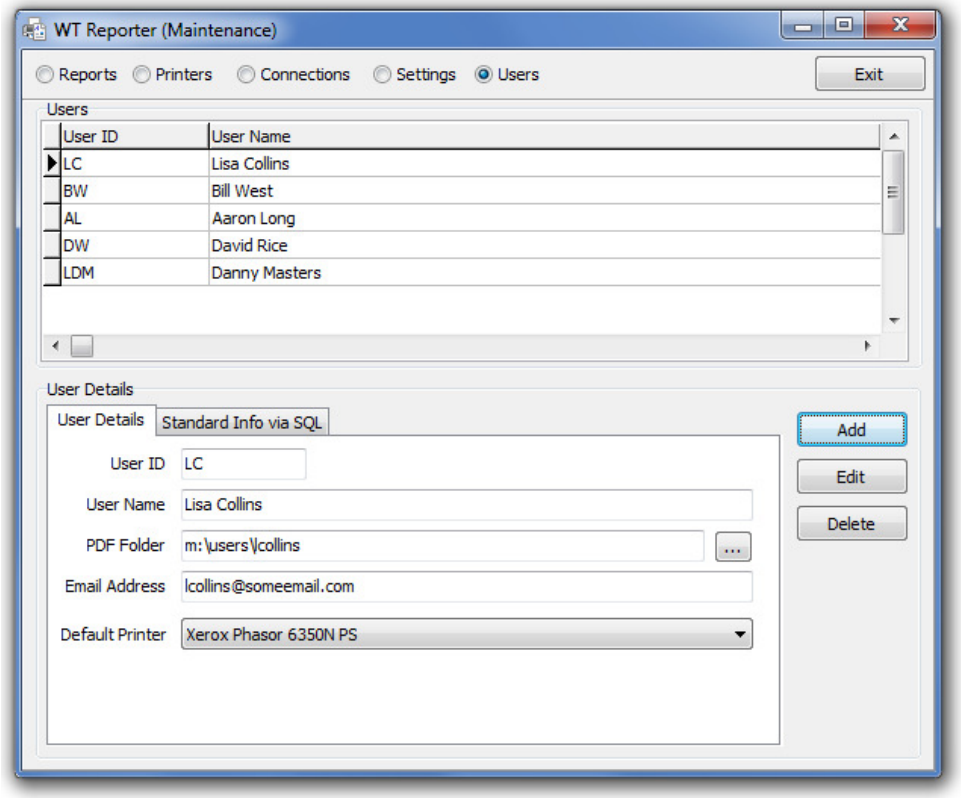

It can connect to your data in several ways using Windows integrated security, database login or simple UDL files.



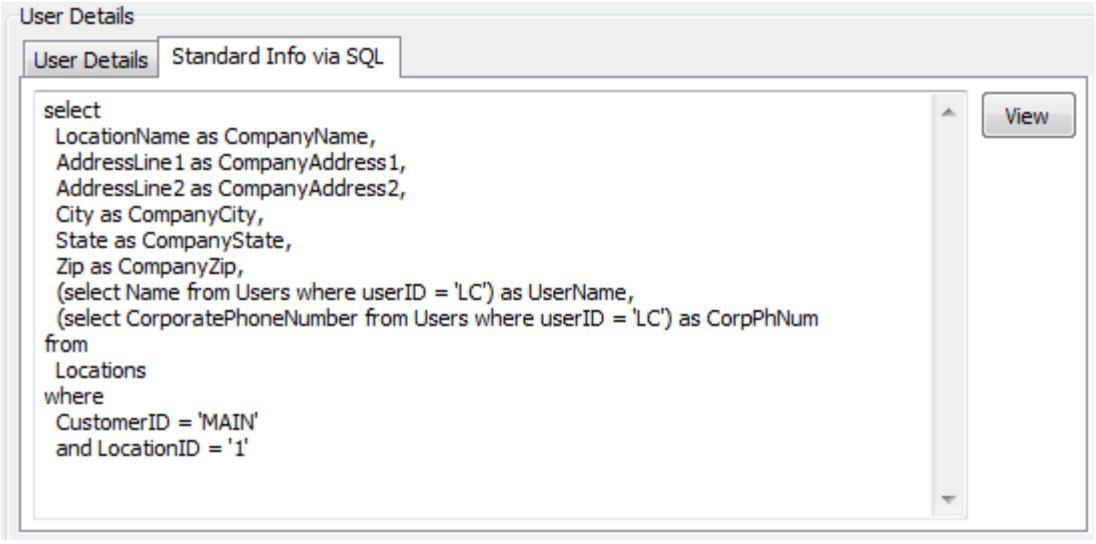
Global settings allow you to specify what folder to monitor and what file extension the command line interface will be looking for. You decide how often the program monitors for new report requests as well as the parameters to send mail to email accounts.



***WT* Reporter** knows users and some of their important default information so it can automatically produce reports to be placed into their user folder or sent to their email address. Each user requesting reports must have a user account set up.




Each user can also have their own information included in a report by utilizing a simple SQL statement that returns values such as the user name, department, phone number and other information that could be used on a report.



A simple command line interface is all that it takes to output the report. In its most basic sense, a text editor can be used to create this one-line text file, but most developers will want to use some sort of automation to create these files. For example, creating an icon to a batch file could generate the command line, or writing some VBA code in Word or Excel could be used with a simple interface. Actual development systems such as Visual Studio or Code Gear Delphi and many others could be used to interface to the report writer. You could even use a web site development environment to send commands to ***WT*** **Reporter** to create PDF files of reports, then have the web site push them back to the user to view. Another method would have the user requesting reports, and ***WT*** **Reporter** would email them to the users email account. There are lots of ways to request reports with ***WT*** **Reporter**. It's just a matter of picking which ones fit the task at hand.

```
/ID=LC /R=Customer List /P=default /C=1 /D=1 /P1=ALL /PDF=CustList
```

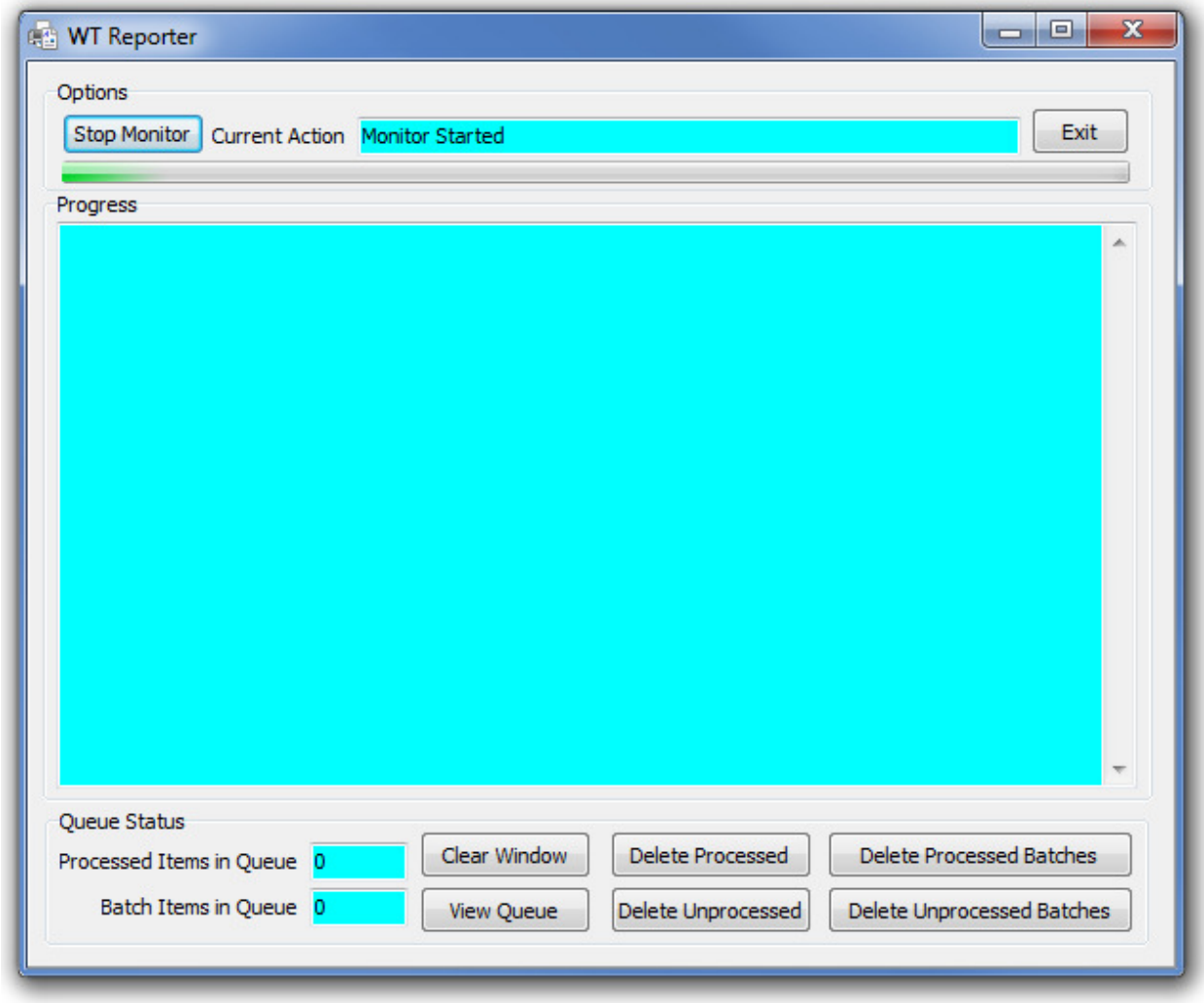
***WT*** **Reporter** can be set up to run multiple reports concurrently. This comes in handy when long running reports cause bottle necks in the report queue. Simply include the special /SHELL program switch in the command line text file and the report will run under another instance of ***WT*** **Reporter**. When the report is complete, the instance of the program will automatically exit. You can run all reports this way by setting up a special flag file in the program folder if you would like.

# *WT* Reporter – Quick Overview

***Wintech Computer Systems, Inc.***

***WT* Reporter** is a stand-alone reporting engine that will output reports that you design to printers, email accounts or PDF files based upon commands sent to it via text files. Simply put, the program monitors a file folder waiting for a file to appear with a specific file extension defined by the end user. When this file appears, the program will open it and follow the commands inside the file. This system requires all report data be contained in a SQL database, or a database that can be accessed through an ODBC driver that understands SQL commands.

The first section covers the ***WT* Reporter** print engine interface and the second section covers the command line interface that will put ***WT* Reporter** into action.



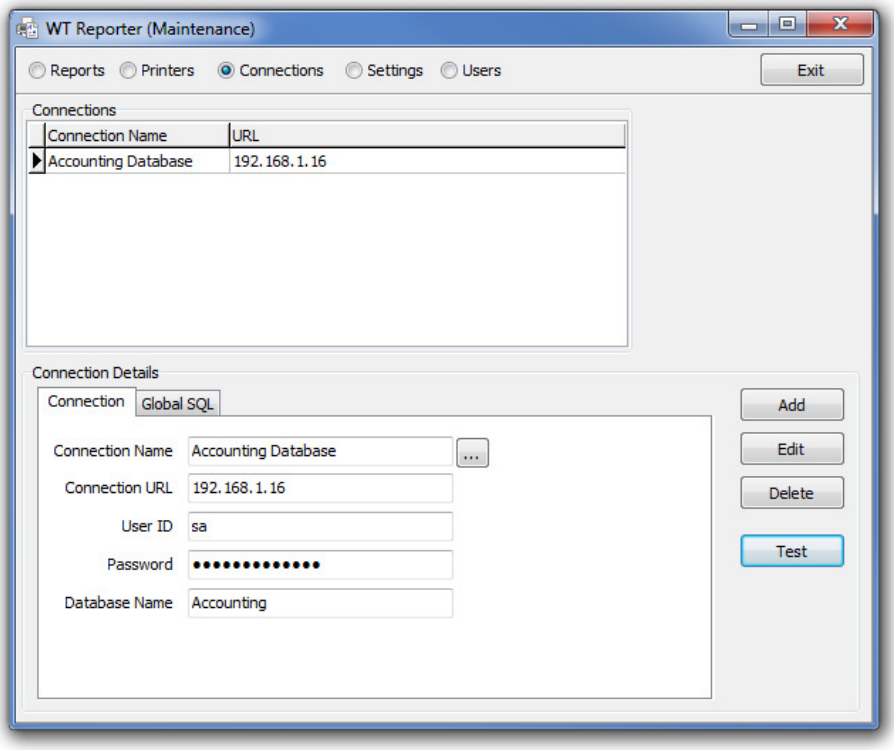
# Section 1 – *WT* Reporter maintenance interface

The program is divided into two applications. One application allows you to set up WT Reporter settings and create reports while the other application simply produces the requested reports. The maintenance application will be covered here and the report execution application will be covered in the next chapter. The maintenance application contains five sections:

**Printers** – Defines all printers that the program will be using to output reports
**Reports** – Defines all specific reports the print program contains
**Connections** – Defines all the SQL database connections for retrieving data necessary for report generation
**Settings** – Defines global settings
**Users** – Defines all users, their default printers and default PDF file folder

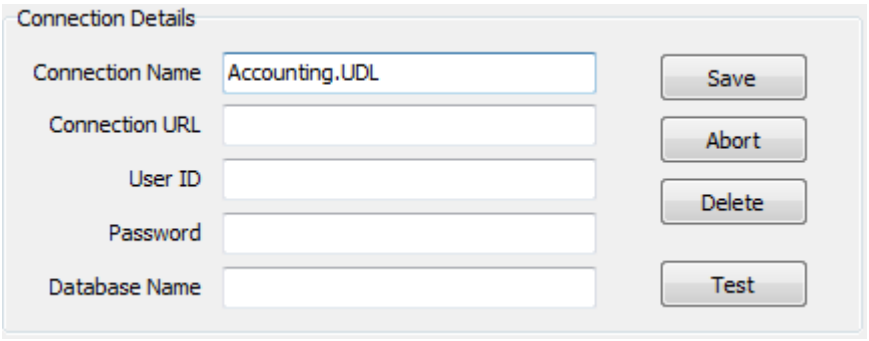
Below are screen shots of each section. These are not listed in the order shown above, but in an order necessary to get the system up and running properly.

## *Connections*




Before producing any reports, at least one database connection must be created. Currently, connections to Microsoft SQL server have been the only ones tested; however, other connections can be used by connecting to an ODBC driver using a UDL file. To add a new connection, click the **Add** button and enter a connection name, the URL to the SQL database server, a user ID, a password and the default database to use. Click the **Save** button, then click the **Test** button to make sure the connection works. Once a connection has been created, the program can access the necessary data to produce actual reports. If you would like to connect to a database via an ODBC driver, you can create a UDL file and place it in the

same folder as the program. Make sure the UDL file can connect to the desired database, then enter the actual UDL file name in the **Connection Name** field. For example, if your UDL file name is **Accounting.UDL**, the Connection Details box will look like this. Just make sure the **Accounting.UDL** file is located in the same folder as the program itself.




**Global SQL** tab

This tab allows you to create a single SQL statement that will contain global details that are available to any report. This feature is handy for setting up your company name, address and other details about your company that may be used on reports. The statement does not have to actually select data from an actual database. It can be something as simple as:

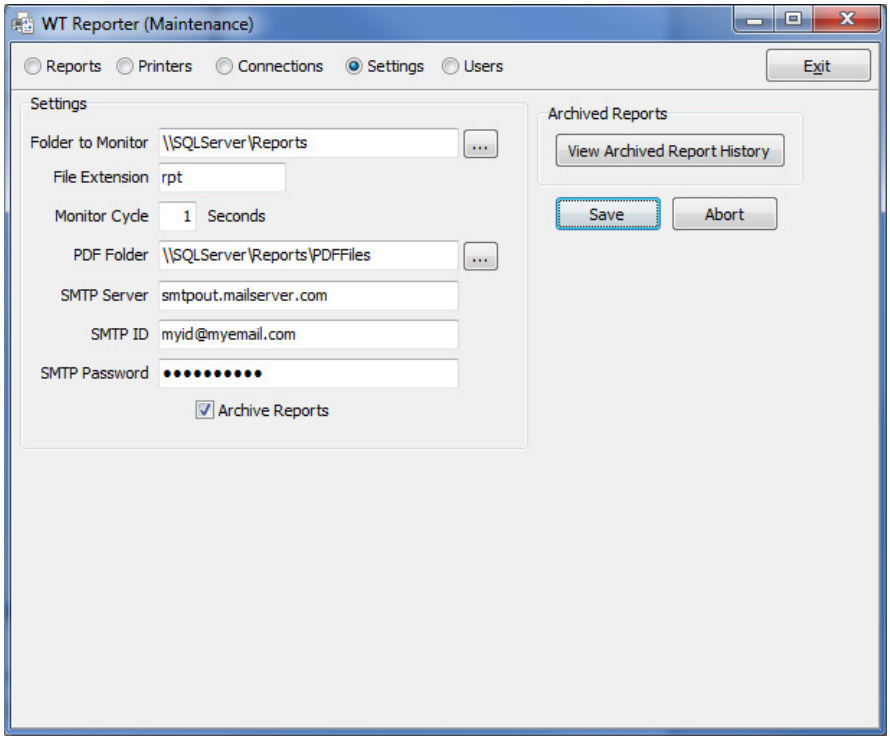
```
select
  'ABC Co, Inc.' as CompanyName,
  '123 Elm St' as Address1,
  'Denver, CO 81900' as Address2,
  '555-1212' as Phone
```

Or it could be selecting information from an actual database

```
select
  CompanyName as CompanyName,
  Address_1 as Address1,
  City + ', ' + State + ' ' + ZipCode as CSZ,
  MainPhone as Phone
from
  GlobalSettings
where
  CompID = 1
```

In either case, whatever data you want globally available for all reports that use this data connection, this is the place to put it. Just make sure only one row of data is returned with as many columns as you desire. There is also a feature to expose information to a report for each user that is set up in the system. Refer to the **Users** section below.

## *Settings*

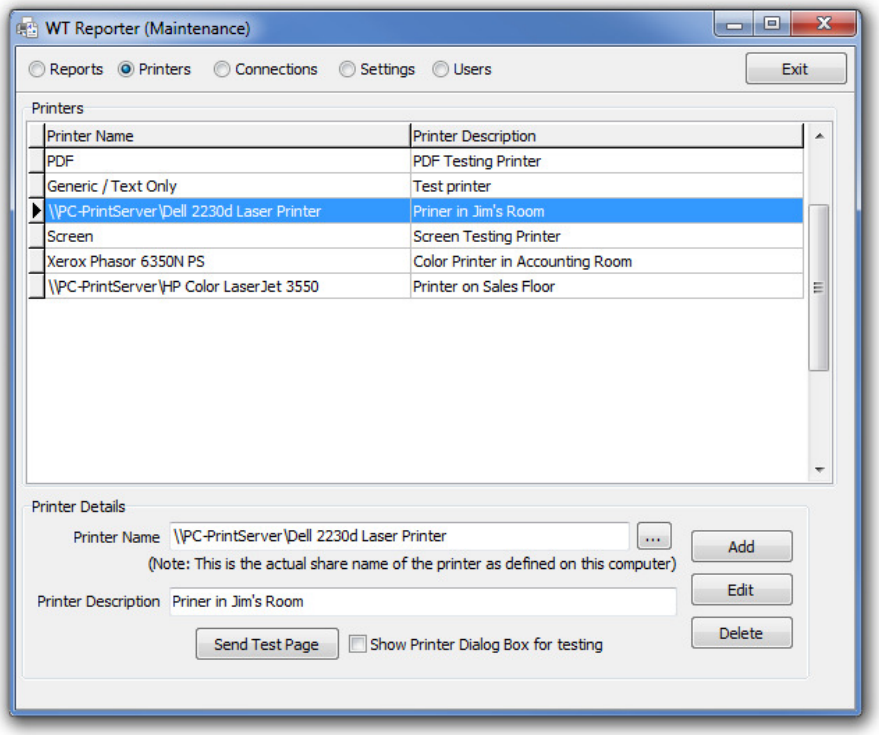



Entering some basic settings will be necessary before the system can start monitoring for print commands. Reports are generated by monitoring a specific folder on the network looking for text files with print commands in them. Enter the folder to monitor as well as the type of file to monitor. In the example above, the **\\SQLServer\Reports** folder is where the text files will be placed, and the file extension of the text files will be **rpt**. You can also set the interval at which the program will "look" for **.rpt** files. You should enter a value from between 1 and 5 seconds. If the special PDF printer is used (see **Printers** section below) for testing, you can specify the folder where they will be created. In this case, it will be in a folder called **PDFReports** below the **\\SQLServer\Reports** folder.

If reports are going to be emailed, you must fill out the SMTP information so ***WT*** **Reporter** can access a SMTP server. You will need the SMTP Server name, a user ID and password. Currently, no special connection features like SSL or custom port numbers are supported. This will be added in a future version. The port number used is the default port 25.

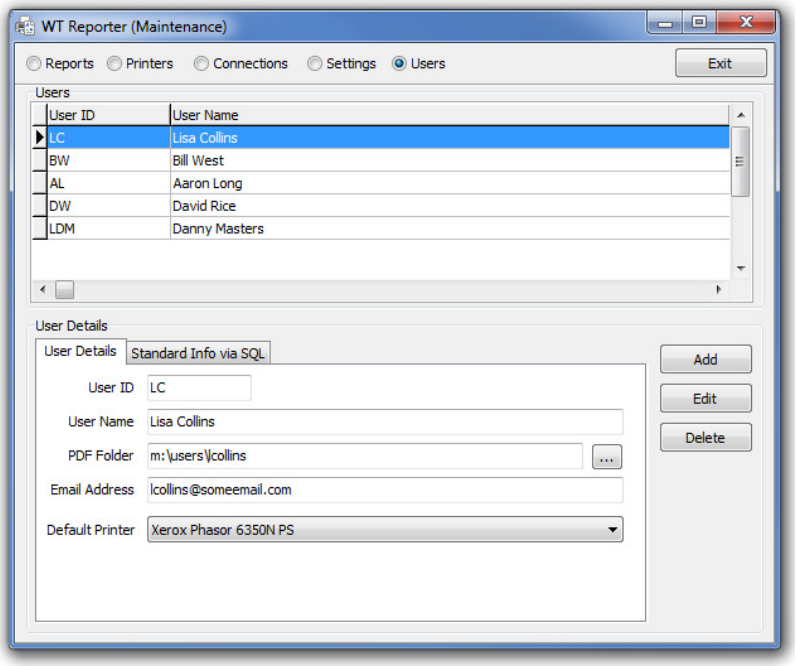
If you check the Archive Reports checkbox, each time a report is processed, a record of it will be added to the archive database. This is a record of what report was processed as well as when and who processed it. It does not contain the actual report content itself. To view the archive history, click the **View Archived Report History** button. Future versions will provide a feature to manage the archived report history.

## *Printers*



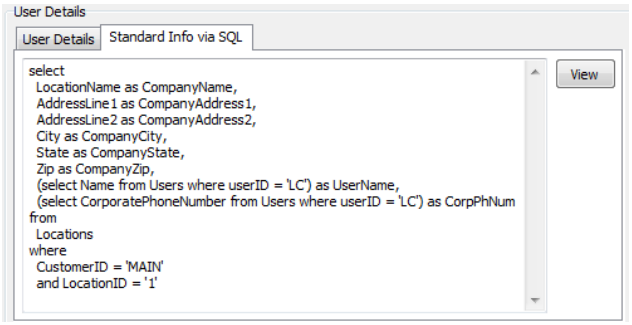

Each printer that receives reports can be defined in this window. The printer name must be the exact shared name of the printer on the network. The printer description will help you further define the printer. Two special printers can be set up, **PDF** and **Screen**. A command that selects the **Screen** printer will result in showing a report on the screen where the program is running from. This is for testing purposes only. A command that selects **PDF** will send a report to a PDF file. This is also for test purposes. Although PDFs can be created for any report, this is not how you would normally set this up.
See **Section – 3** for further details.

## *Users*




At least one user must be defined in the system. Go ahead and enter one user who will be sending commands to the program. Keep the user ID as short as possible but unique. It will be used in the printer command text file that is outlined in **Section - 3**. Enter a user name and a folder where PDF files will be placed if the print command file requests one. If you plan on sending reports via email, enter a valid email address from where the report will be sent. Again, **Section - 3** explains how to email reports via the command line interface. Last, select the default receiving printer for the user. Remember, printers must be added to the system before they can be selected in this screen.

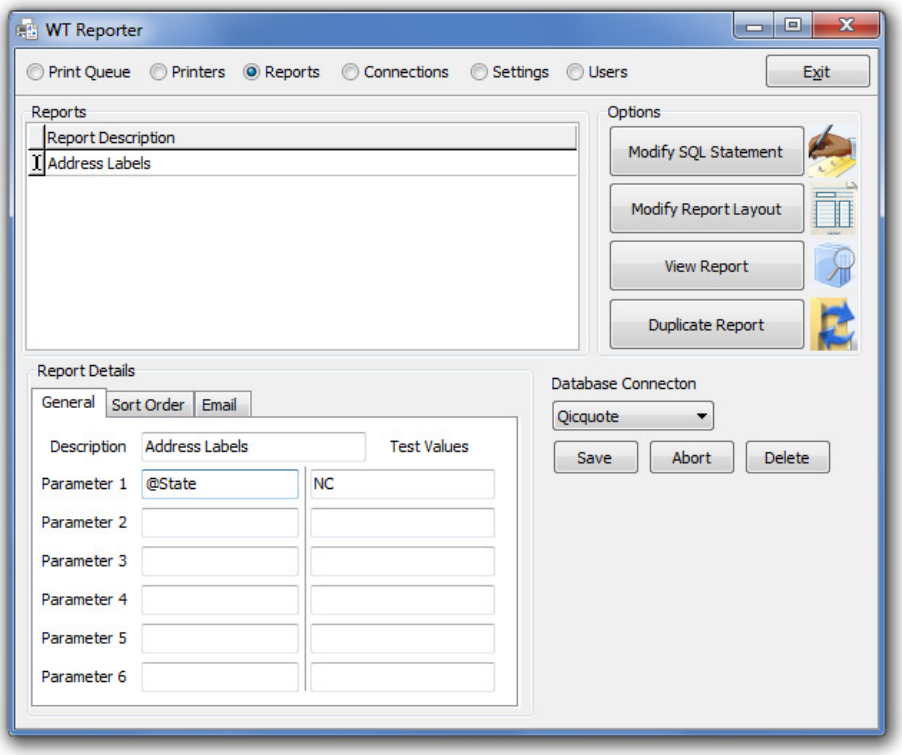
**Standard Info via SQL** tab




Use this tab to gather information about the user that can be placed onto a report. You can use any valid SQL statement that will return one row of data. It could be something like the one above of simply a select statement that returns raw data from text like the example below.

```
select
  'ABC Co, Inc.' as CompanyName,
  '123 Elm St' as Address1,
  'Denver, CO 81900' as Address2,
  '555-1212' as Phone
```

## *Reports*



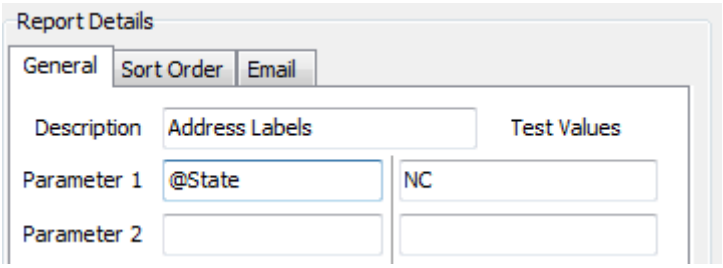

This is the heart of the system. This is where all the reports are created and maintained. Each report created relies upon a single SQL statement to retrieve all the necessary data for the entire report. Because of this, the SQL statement may become quite complex. It is required that you have a good knowledge of the SQL language and how to retrieve all the necessary data required producing the report you desire. It is advised that you use a tool that will allow you to create and test SQL statements that retrieve the desired data. After the statement is written and tested, it should then be copied into the SQL statement editor in this program. This program does not offer syntax checking, auto generation or highlighting as other tools such as Microsoft SQL Server Management Studio® does. It simply stores the statement and executes it when commanded to do so.

The 1st step in creating a new report is to click the **Add** button and give the report a name. This name must be unique. Once the name is given, you can create up to six parameters to limit the data that is returned via the where clause. The program allows you to create these parameters and give them values for testing purposes. In the example above, the **@State** parameter will be replaced with **NC** when the statement is tested or the test report is run. When a print command is created, the values from the text file are used.

Suppose a SQL statement looked like this:

```
Select
  CustName,
  Address1,
  City,
  State,
  Zip
from
  Customers
where
  State = '@State'
order by
  City,
  Address1
```

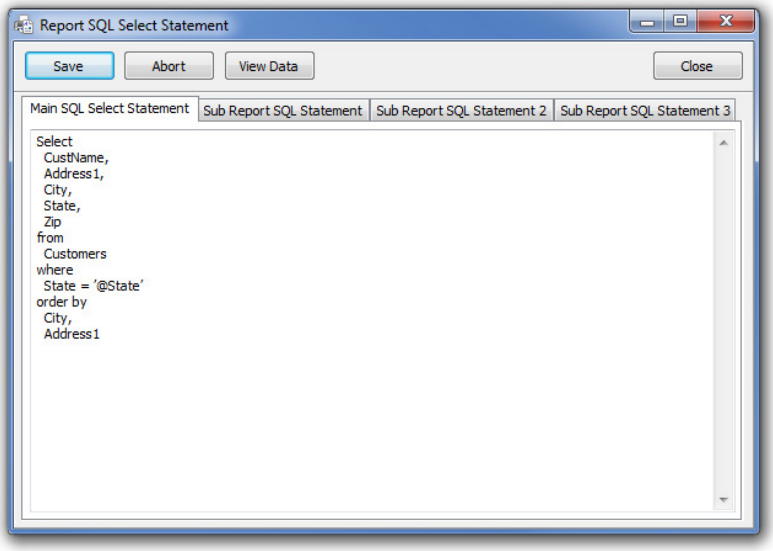
As you can see, a parameter called **@State** exists. Before the statement runs, the **@State** will be replaced with an actual state abbreviation such as **NC**. Below is an example of a report called **Address Labels** with a single parameter called **@State**.



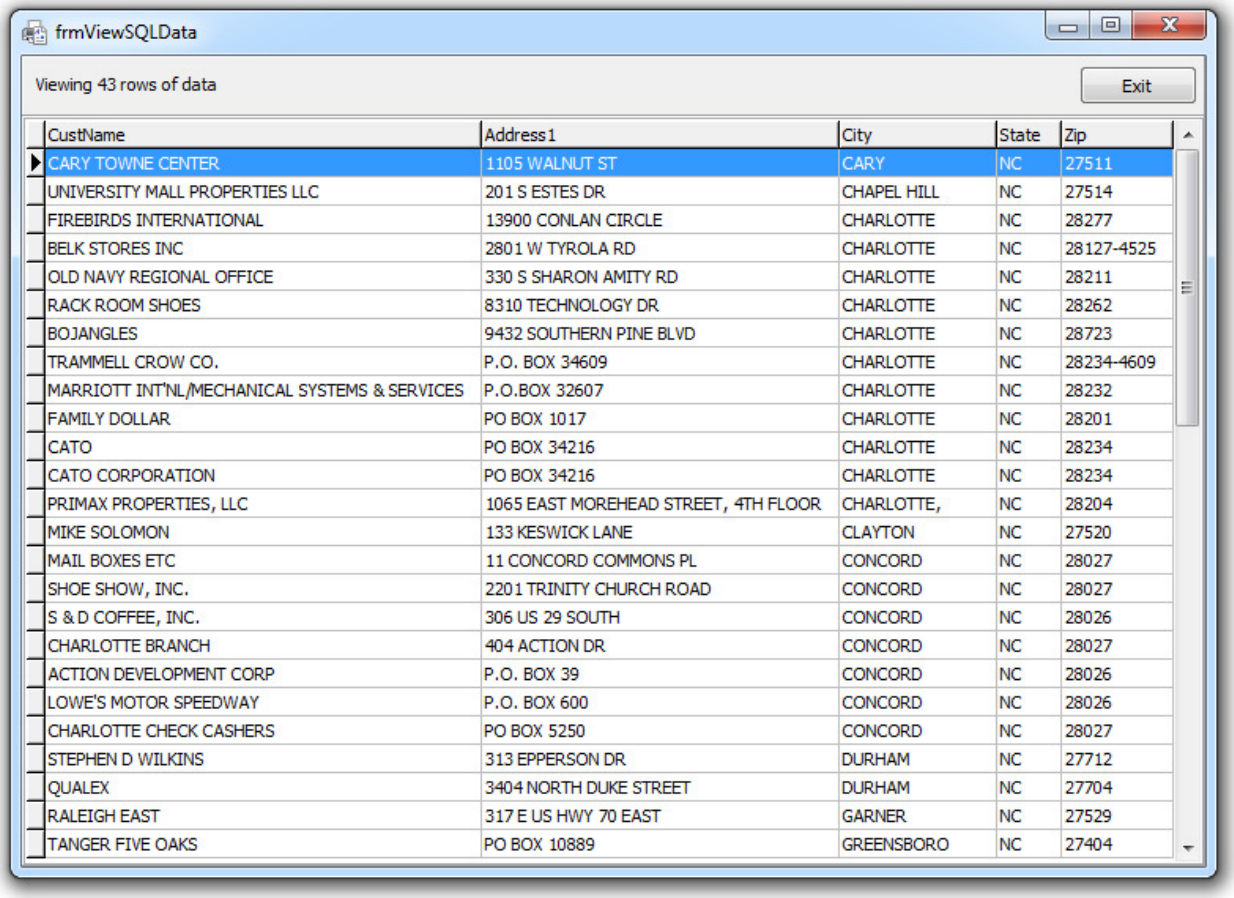

When the SQL statement example from above is executed in this screen, it will look like this:

```
Select
  CustName,
  Address1,
  City,
  State,
  Zip
from
  Customers
where
  State = 'NC'
order by
  City,
  Address1
```

Clicking the **Modify SQL Statement** button will display a window similar to this. Here is where the actual SQL statement will be written. Remember to write the statement with the parameter names and not the actual data. **Note**: It is not required to use any parameters; however, most report data will make use of them.



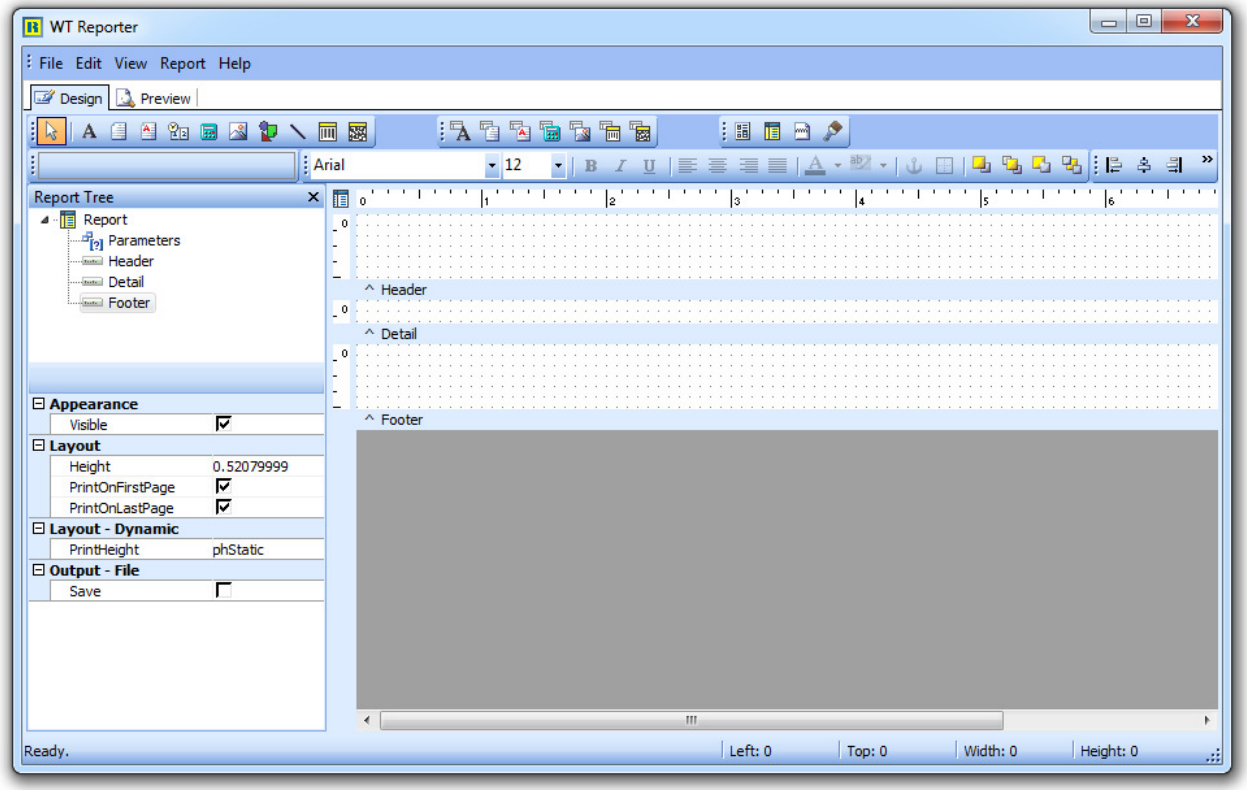

Once the statement is entered and you make sure you have a test value for each parameter, press the **View Data** button to display the results in a window like this. If there is an error in the SQL statement, a generic error message will appear. At this point, either determine the error or copy the SQL syntax back to a more qualified environment and try running it there to solve the errors.



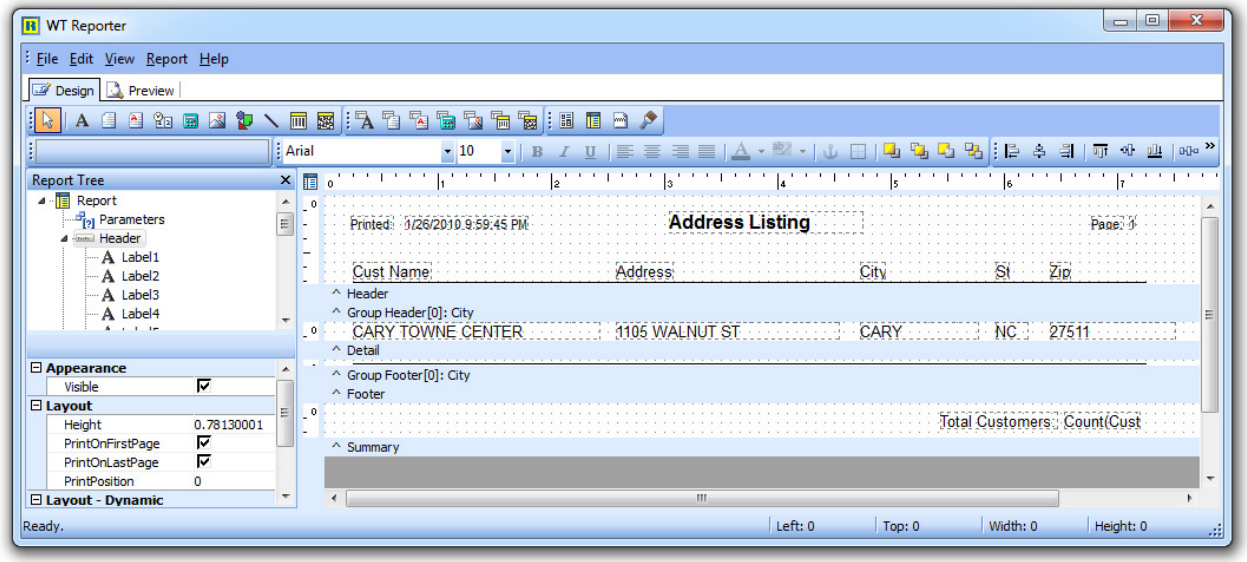

| CustName | Address1 | City | State | Zip |
|---|---|---|---|---|
| CARY TOWNE CENTER | 1105 WALNUT ST | CARY | NC | 27511 |
| UNIVERSITY MALL PROPERTIES LLC | 201 S ESTES DR | CHAPEL HILL | NC | 27514 |
| FIREBIRDS INTERNATIONAL | 13900 CONLAN CIRCLE | CHARLOTTE | NC | 28277 |
| BELK STORES INC | 2801 W TYROLA RD | CHARLOTTE | NC | 28127-4525 |
| OLD NAVY REGIONAL OFFICE | 330 S SHARON AMITY RD | CHARLOTTE | NC | 28211 |
| RACK ROOM SHOES | 8310 TECHNOLOGY DR | CHARLOTTE | NC | 28262 |
| BOJANGLES | 9432 SOUTHERN PINE BLVD | CHARLOTTE | NC | 28723 |
| TRAMMELL CROW CO. | P.O. BOX 34609 | CHARLOTTE | NC | 28234-4609 |
| MARRIOTT INT'NL/MECHANICAL SYSTEMS & SERVICES | P.O.BOX 32607 | CHARLOTTE | NC | 28232 |
| FAMILY DOLLAR | PO BOX 1017 | CHARLOTTE | NC | 28201 |
| CATO | PO BOX 34216 | CHARLOTTE | NC | 28234 |
| CATO CORPORATION | PO BOX 34216 | CHARLOTTE | NC | 28234 |
| PRIMAX PROPERTIES, LLC | 1065 EAST MOREHEAD STREET, 4TH FLOOR | CHARLOTTE, | NC | 28204 |
| MIKE SOLOMON | 133 KESWICK LANE | CLAYTON | NC | 27520 |
| MAIL BOXES ETC | 11 CONCORD COMMONS PL | CONCORD | NC | 28027 |
| SHOE SHOW, INC. | 2201 TRINITY CHURCH ROAD | CONCORD | NC | 28027 |
| S & D COFFEE, INC. | 306 US 29 SOUTH | CONCORD | NC | 28026 |
| CHARLOTTE BRANCH | 404 ACTION DR | CONCORD | NC | 28027 |
| ACTION DEVELOPMENT CORP | P.O. BOX 39 | CONCORD | NC | 28026 |
| LOWE'S MOTOR SPEEDWAY | P.O. BOX 600 | CONCORD | NC | 28026 |
| CHARLOTTE CHECK CASHERS | PO BOX 5250 | CONCORD | NC | 28027 |
| STEPHEN D WILKINS | 313 EPPERSON DR | DURHAM | NC | 27712 |
| QUALEX | 3404 NORTH DUKE STREET | DURHAM | NC | 27704 |
| RALEIGH EAST | 317 E US HWY 70 EAST | GARNER | NC | 27529 |
| TANGER FIVE OAKS | PO BOX 10889 | GREENSBORO | NC | 27404 |

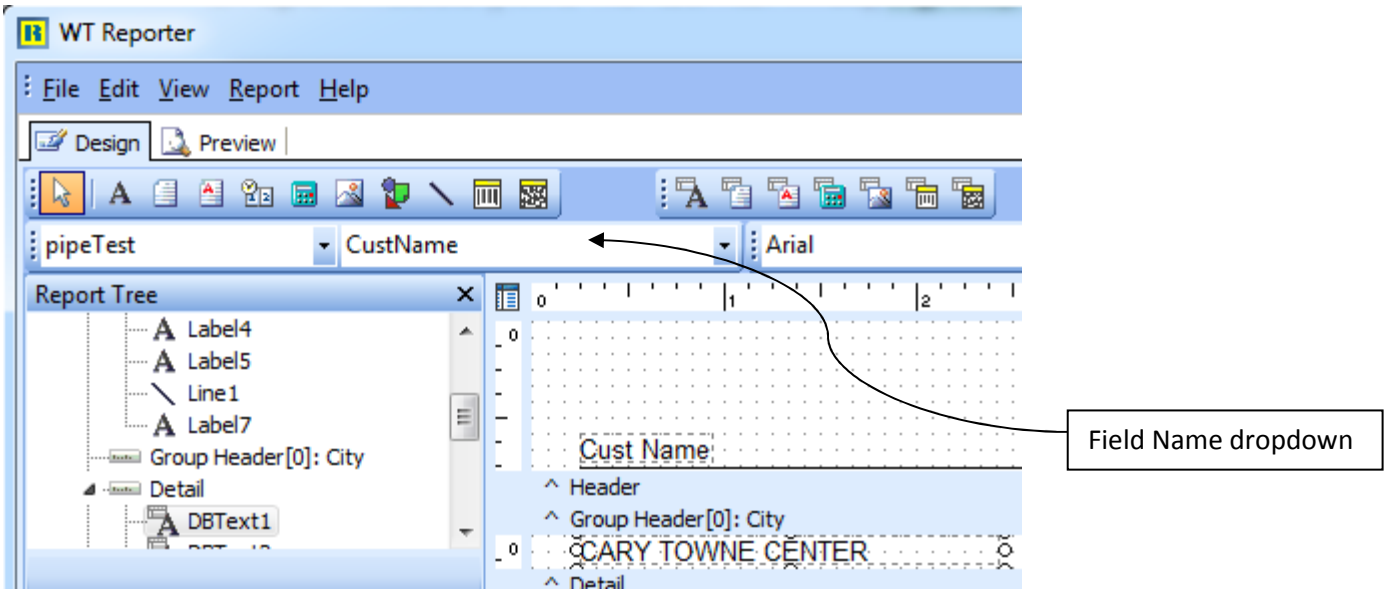
The upper left corner will display the number of data rows that are returned, in this case, 43.

Once you have the data you desire, it's time to start building the report. Save the SQL statement and click the **Modify Report Layout** button. The following empty report layout window will appear.




Begin by adding data fields, labels and shapes on the screen. This specific report contains five data fields, a report header, two horizontal lines, a Group Footer and a counting total in the lower right as well as a couple system variables for date and page number.



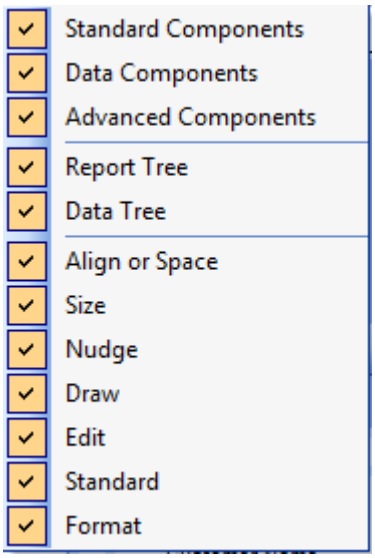



The most common toolbar button that will be used is  . This will place a data aware field on your report layout. Once the report field is shown, click the Fieldname dropdown, and select the desired field. In the example above, the fieldname dropdown is **CustName**. All the fieldnames in the dropdown are the names of the columns returned in the SQL statement. If you want meaningful names, you can use SQL syntax similar to:

```
Select
  CustName as CustomerName,
  Address1 as AddressLine1,
  Address2 as AddressLine2,
  City,
  State,
  Zip,
  CustBal as CustomerBalance
from
  Customers
```

## *Toolbars*

Each toolbar has a group of buttons or dropdowns that performs a specific task. Some are more important than others and some should never be used at all. This is a list of all toolbars that can be shown on the report layout screen. You have the option of showing or hiding these toolbars.




Standard Components 

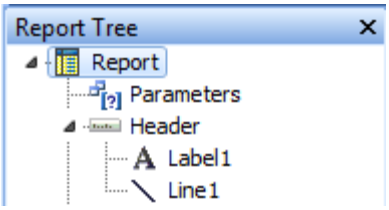
1) Selection Arrow
2) Text Label – Use this to add new static text labels to the report. Once added, you can type anything you desire in the Edit toolbar.
3) Memo – Use this to add a text memo (more than one line of text) to the report.
4) Rich Text – This is a memo field, but you can format the text using rich text formatting options such as bold, underline, font styles, etc.
5) System Variable – Use this to add the current date time and page numbering
6) Variable – Not used for anything
7) Image – Use this to insert an image such as a company logo
8) Shape – Use this to insert simple shapes such as boxes and circles
9) Line – This is used to insert horizontal, vertical or diagonal lines
10) Bar Code – Use this to insert a bar code
11) 2D Bar Code – This will insert a 2D barcode

Data Components 

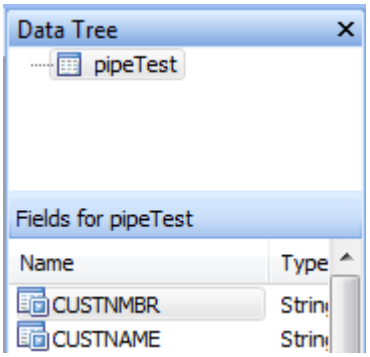
1) Database Text – Use this to insert a database field onto the report
2) Database Memo – Use this to add a text memo from a database column (more than one line of text) to the report.
3) Database Rich Text – This is a memo field, but you can format the text using rich text formatting options such as bold, underline, font styles, etc.
4) Database Variable – This can be used for creating database variables such as sum, average and total. One possible use is for totals and sub-totals of dollar amounts.
5) Database Image – Use this to insert an image that is stored in a database BLOB column
6) Database Bar Code – This will print a bar code using data from a database
7) Database 2D Bar Code – This will print a 2D bar code using data from a database

Advanced Components 

1) Report Region – This is a container that will group report controls together. The region can then be moved and all controls in it will move as well. It is useful for hiding or showing groups of controls at run time.
2) Sub Report – This will insert a sub-report into the current report. Currently, this feature should not be used because only one SQL statement is allowed. Sub-reports require additional SQL statements.
3) Page Break – Use this to insert a page break
4) Paint Box – This is used to set up a region on the report where direct drawing can be done at report generation time. It has no use with this system.

Report Tree 


This tree view displays all the controls that are placed on the report. Selecting any of these will place the control borders on the related control on the report. Use this to quickly identify each control.

Data Tree 


The data tree will display all the SQL columns returned in the SQL dataset. You can drag these on the report or use the Data Components toolbar.

Align or Space 

1) Align Left – Align a set of vertically selected controls to the horizontal left
2) Align Middle – Align a set of vertically selected controls to the horizontal center
3) Align Right – Align a set of vertically selected controls to the horizontal right
4) Align Top – Align a set of horizontally selected controls to the vertical top
5) Align Center – Align a set of horizontally selected controls to the vertical center
6) Align Bottom – Align a set of horizontally selected controls to the vertical bottom
7) Space Horizontally – Space a set of horizontally selected controls evenly
8) Space Vertically – Space a set of vertically selected controls evenly
9) Center Horizontally – Center selected controls horizontally in a report band
10) Center Vertically – Center selected controls vertically in a report band

Size 

1) Shrink Width to Smallest

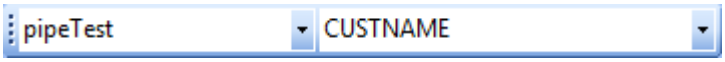
2) Grow Width to Largest
3) Shrink Height to Smallest
4) Grow Height to Largest

Nudge 

1) Nudge Up – Nudge selected controls up one pixel
2) Nudge Down – Nudge selected controls down one pixel
3) Nudge Left – Nudge selected controls left one pixel
4) Nudge Right – Nudge selected controls right one pixel

Draw 

1) Fill Color – Fill a control such as a box with a specific color
2) Line Color – Select a color for a line
3) Line Thickness – Change the thickness of a line or boarder
4) Line Style – Change the style of a line or boarder

Edit 


1) Data Pipeline – This should always be **pipeTest**
2) Data Field or Column Name – Select the desired field name of a data control

Standard 

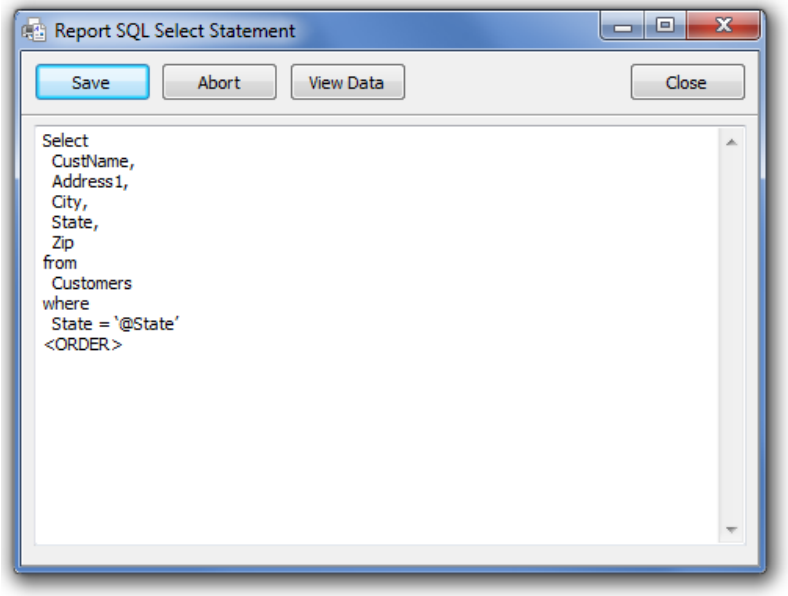
1) New – Create a new report. **Should never be used**.
2) Open – Open an existing report
3) Save – Save the current report
4) Page Setup – Use this to set up page margins orientation and other page aspects
5) Print – This is used to print a report
6) Print Preview – You can preview the report using this option
7) Cut – Use this for cutting a control from the report
8) Copy – Use this for copying a control
9) Paste – Use this for pasting the last copied control back to the report layout

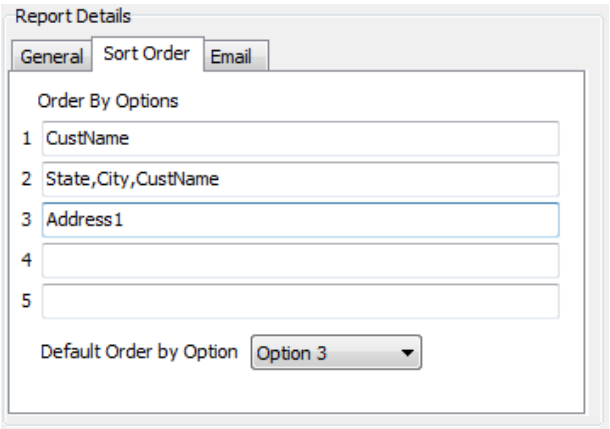
Format 


1) Font – Select the desired font
2) Font Size – Select the desired font size
3) Bold – Toggle bold on/off
4) Italic – Toggle Italic on/off
5) Underline – Toggle Underline on/off
6) Left Justify – Left justify a control
7) Center Justify – Center justify a control
8) Right Justify – Right justify a control
9) Font Color – Change the font color
10) Highlight color – Change the font highlight color

11) Anchors – Anchor a control to a position relative to a page border
12) Border – Create a border around a control
13) Bring Forward – Bring a control forward one level
14) Bring to Front – Bring a control to the front of all controls
15) Send Backward – Send a control back one level
16) Send to Back – Send a control to the very back

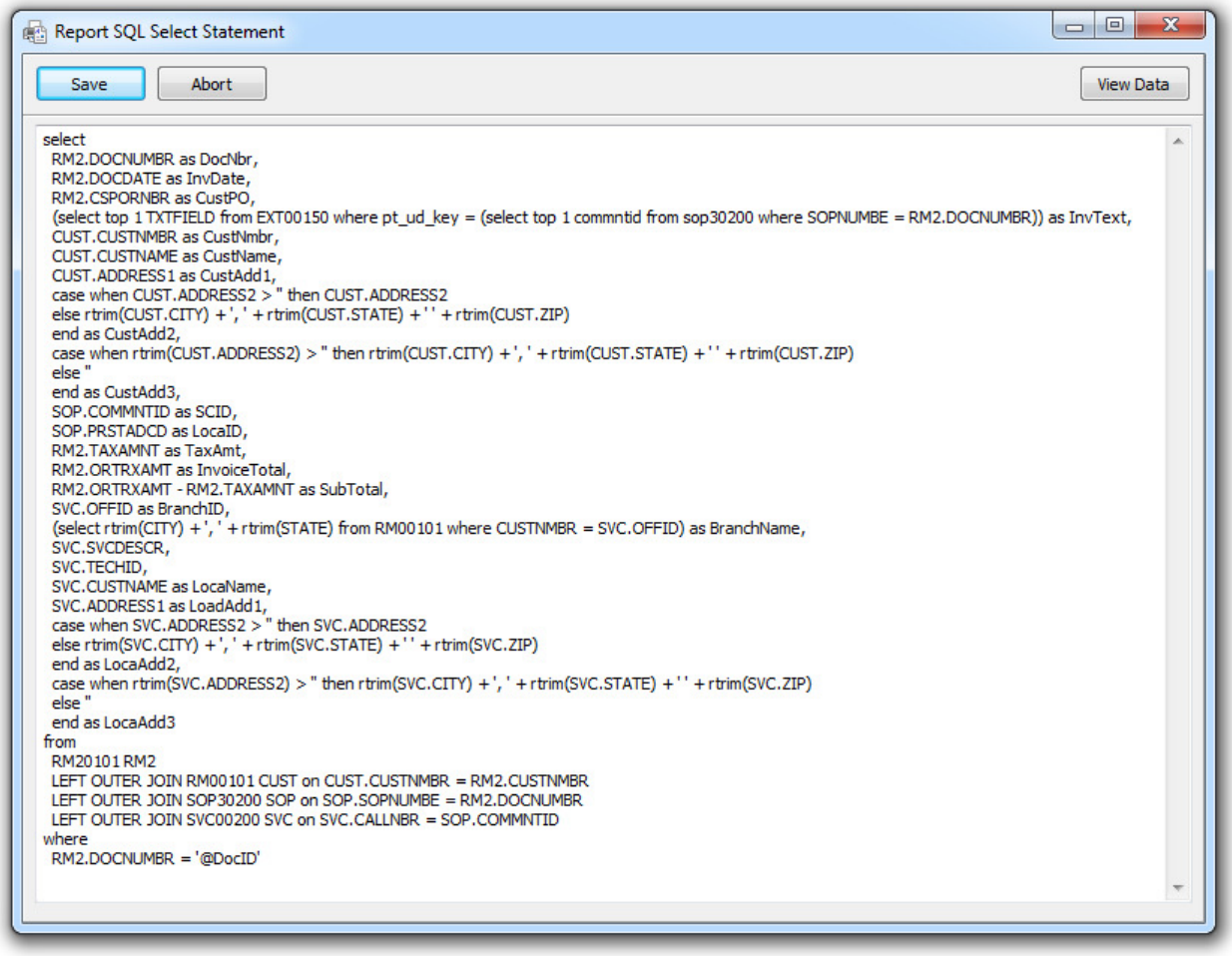
Another feature of an SQL statement is to allow for different sort orders to be used when running the report. If the SQL statement looked like this, the optional sort methods must be entered in the 2nd tab called Sort Order



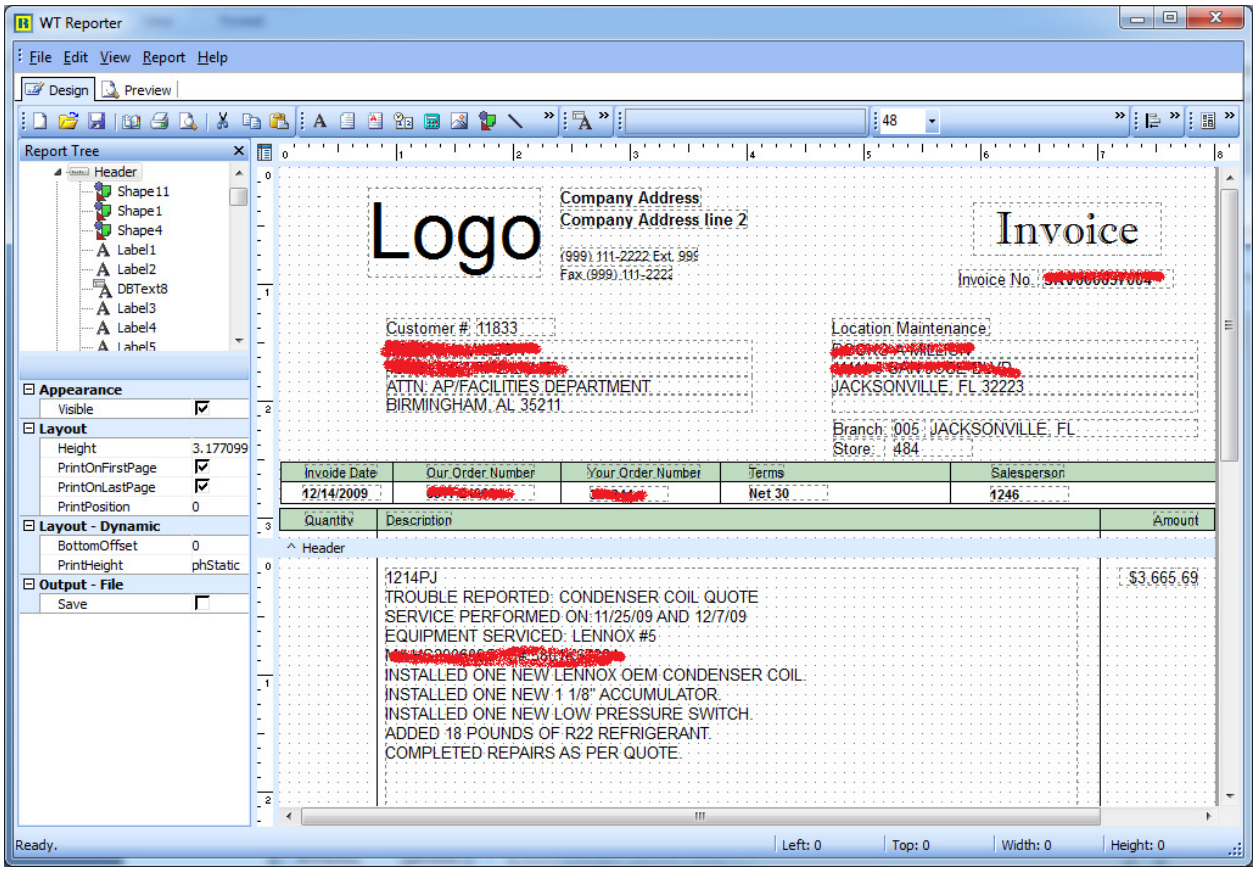




When the report is run, the **<ORDER>** section of the SQL statement is replaced by one of the three sort options identified in the above image. In this example, when the test report is run, the **<ORDER>** is replaced by **`order by Address1`**. When the production report is run, the **/S=n** option must be part of the parameter listing. In this case, it could be /S=1, /S=2, or /S=3.

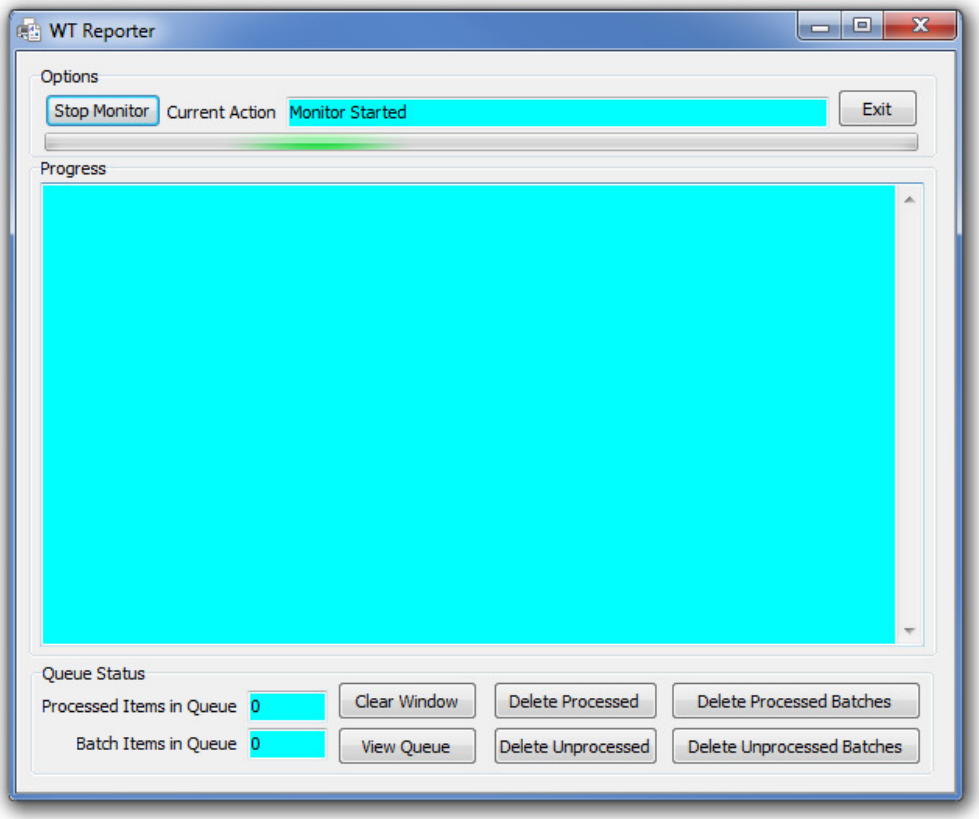
Here is another more complex SQL statement that would be used to print an invoice from an accounting SQL database.



Here is the report layout for the SQL statement



# Section 2 – Print Queue Application




This screen displays the current status of the printing functions. When a command appears, the Progress section shows the command that is being processed and its ultimate success or failure. This is where ***WT*** **Reporter** can be turned on and off.

The Queue status is updated every 10 seconds. You can view the contents of the print queue by clicking the View Queue button. You can also remove queue entries either before or after they have been processed by using the buttons in the lower right. Be careful when deleting queue entries that have not been processed. Once they are removed, they cannot be recalled.

There are two startup parameters that can be passed along when starting up the application.

/START – This will automatically start the print queue monitoring. It is the same as starting the program and clicking the Start Monitor button.

/EXIT***<n>*** -- This will make the program exit automatically after ***<n>*** number of cycles. A cycle represents the number you enter in the Monitor Cycle field in the Settings section.

**Example**: WTReporter.exe /START /EXIT30

This will start the print queue monitoring and exit after 30 cycles. Using these startup parameters can be handy for setting up ***WT*** **Reporter** from within Windows Task Scheduler to run reports at specific times.

# Section 3 – Command Line Interface

The command line interface consists of simple text files that ***WT*** **Reporter** opens and uses to perform printing and PDF file creation. Without the command line interface, **WT** Reporter would not be able to print or create any PDF files.

Normally, the printer text file would be created from within another program or user interface that would allow the end user to create them in a simple or transparent manner. Although this is the desired method, these text files can be created using Notepad or any other text editor. Just remember, it is not advisable to manually create these files from within the folder that ***WT*** **Reporter** is monitoring. If this is done, ***WT*** **Reporter** will attempt to process the file while it is being created! It is best to create the file in another folder, and then simply copy it to the folder being monitored.

The text file being created contains parameter label and parameter value sets. It is important to format the file like this for successful report generation. An example of a parameter set would be

**/ID=MNP** where **/ID=** is the parameter label and **MNP** is the parameter value. It is important not to place any spaces in between the label and the value. **/ID= MNP** is not valid.

**IMPORTANT**: The entire set of parameters must be on a single line with no line feeds unless there is another entire set of parameters on another line.

Below are all the parameters ***WT*** **Reporter** understands. The words inside the <> brackets are replaced with actual parameter values. Do not include the <> in the parameter value. /ID=<MNP> is not valid unless the value is actually "<MNP>" and not "MNP".

**/ID=*<userid>*** – This represents a user ID that equals the user ID that was set up in the ***WT*** **Reporter** interface. This is a required parameter and is necessary for the program to identify the users default printer and other information relevant to the user. This is case insensitive. Example: **/ID=MNP**.

**/R=*<report name>*** – This is the report to print. It must match one of the reports defined in the program under the reports section. Example: **/R=Invoice – Service**.

**/P=*<printer name>*** – This is the printer to send the report to. It can be one of three different value types. Type 1 is an actual printer name that is defined in the program under the Printers section. Type 2 is a reserved word called **default**. If, for example, **/P=default** is used, the program will print to the users default printer. Type 3 is a reserved word called **none**. If, for example, **/P=none** is used, the report will not be sent to the printer at all. This is normally used in conjunction with the **/PDF=** parameter when the desired result is to create a PDF file of the report without a hard copy.

**/S=<order>** – This represents the order in which a selected set of data will be sorted by. <order> will be replaced by an integer from 1 to 5 depending upon how many different sort orders are set up in the Reports section under the Sort Order tab.

**/C=*<copies>*** – This tells the program how many copies to print. This parameter can be omitted if you want only one copy printed. Example: **/C=2** would print two copies of the report. Note: This parameter is ignored when sending a report to a PDF file and not to the printer.

**/D=*<delete value>*** – This tells the program to delete the queue entry or to keep it. This parameter can be omitted if desired. If it is, the queue is automatically deleted after the report is processed. There are only two valid parameter values allowed. **/D=1** or **/D=0**. Setting it to one will remove the print queue entry. (default) Setting it to 0 will keep the print queue entry. (not recommended)

**/SHELL** – This tells the program to launch another instance of the program and process this specific report. This is handy if reports take a while to process. The multi-threading capability of the operating system is used to process reports instead of the program itself processing reports one by one. Note: If a file named SHELL.FLG (The file can be empty) is present in the program folder where WTReporter.exe is located, all reports will be run using this method.

**/P1=*<value>*** through **/P6=*<value>*** – These are parameters that are passed along to the SQL statement explained in the Reports section. As outlined in that section, these values replace the @<values> in the SQL statement. For example, if the SQL statement looks like this, the first parameter in the text file may look like this: **/P1=NC**.

```
Select
  CustName,
  Address1,
  City,
  State,
  Zip
from
  Customers
where
  State = '@State'
order by
  City,
  Address1
```

If NC was passed along in the text file, the SQL statement would transform into this before the data was actually retrieved from the database

```
Select
  CustName,
  Address1,
  City,
  State,
  Zip
from
  Customers
where
```

```
  State = 'NC'
order by
  City,
  Address1
```

Including these parameters in the text file are only required if you expect to use them in the SQL statement itself. If no values are passed in the text file and the SQL statement is expected, then the SQL statement will fail when it is executed.

**/PDF=*<filename>*** – This tells the program to create a PDF file of the report. This is an optional parameter. If no PDF file is required, simply do not include this in the text file. If it is used, the file will be created in the folder set up in the **Users** section of the program interface. Example: /PDF=Inv12234 would create a PDF file named Inv12234.PDF. If the PDF folder outlines in the Users section is **C:\MNP-PDF Reports**, the file would be found in **C:\MNP-PDF Reports** folder.

**/E=*<emailaddress>*** – This tells the system to send the report via email. ***<emailaddress>*** can be an actual email address or the word **DEFAULT**. When **DEFAULT** is used, the email address stored in the report definition will be used.

**/BATCH** and **/PRINTBATCH** – These are a paired set of parameters without values whose purpose is to hold processing of reports, then batch process them at a later time. If /BATCH is included in the text file, the report being requested is not printed or no PDF file is created at that time. It is simply put on hold until a special text file with just two parameters is sent.

In this example of a text file contents, **/ID=MNP /R=Invoice /P=default /P1=SRV12345 /BATCH**, the report would be queued but not actually run. Several of these style text files could be sent to ***WT*** **Reporter**, then at a later time, a text file with just two parameters can be sent to the program. This special text file would look like this **/ID=MNP /PRINTBATCH**. The program needs to know which user would like their batch reports processed. Of course, MNP is the user ID and would change according to which user's batch report need to be processed.

With all that said, let's see some examples:

```
/ID=MNP /R=Invoice - Service /P=default /C=1 /D=1 /P1=SRV000097233 /PDF=SRV000097233
```

This example will print the report called Invoice – Service and create a PDF file named SRV000097233.PDF. It will print one copy and delete the queue entry after it completed. A parameter of SRV000097233 is passed for the 1st parameter in the report setup. In this case, it's the invoice number.

```
/ID=MNP /R=Customer List /P=HP4015 /C=1 /D=0 /P1=NC /PDF=NC /BATCH
```

This example will print the report called **Customer List**. It will print one copy and delete the queue entry after it completed. A parameter of NC is passed for the 1st parameter in the report setup. In this case, it's the state that is wanted. When this one is processed, it will not print anything because it has the parameter /BATCH. Batch style entries like this can only be released from the queue with another command like `/ID=MNP /PRINTBATCH`.

Batch entries can contain multiple lines of information such as this. This specific set of lines will queue up 6 invoices to be printed at a later time when the `/ID=MNP /PRINTBATCH` text file is created in the folder being monitored.

```
/ID=MNP /R=Invoice - Service /P=default /C=1 /D=1 /P1=SRV000097233 /PDF=SRV000097233 /BATCH
/ID=MNP /R=Invoice - Service /P=default /C=1 /D=1 /P1=SRV000097137 /PDF=SRV000097137 /BATCH
/ID=MNP /R=Invoice - Service /P=default /C=1 /D=1 /P1=SRV000097155 /PDF=SRV000097155 /BATCH
/ID=MNP /R=Invoice - Service /P=default /C=1 /D=1 /P1=SRV000097227 /PDF=SRV000097227 /BATCH
/ID=MNP /R=Invoice - Service /P=default /C=1 /D=1 /P1=SRV000097209 /PDF=SRV000097209 /BATCH
/ID=MNP /R=Invoice - Service /P=default /C=1 /D=1 /P1=SRV000097222 /PDF=SRV000097222 /BATCH
```