SimpleSoftware

Simple solutions for document management

SimpleCoversheet 3.0 MANUAL

www.SimpleIndex.com

Revised January, 2010

Table of Contents

1 Introduction

SimpleCoversheet is an easy, automated tool for printing barcode cover sheets, and with the advent of recent versions, not just cover sheets but sheets of sticky labels and any other document of any shape, size or layout. Cover sheets can be used for a variety of scanning applications. The possibilities include:

- Fully automate the indexing for your scanned documents using barcode technology with accuracy of 99.9% or more
- Anyone in your office can fill out and print cover sheets with a single, low-cost site license
- Incredibly easy user interface
- Use with digital copiers and MFPs to automatically file scanned documents from anywhere in the office
- Centralized scanning operations can receive documents with the index values pre-coded by the people that created them
- Print generic separator sheets that can be used to indicate breaks between multi-page documents
- Users can select from a list of values from a database or text file
- Perform database merges to automatically print many barcode cover sheets at once, or many barcodes on a single sheet
- Use the command line interface to automatically print cover sheets from a custom application

By default, SimpleCoversheet uses a Code 128 barcode. However, there are also many other barcode types to choose from under the 'Barcode Settings.' Be sure to specify code 128, or the other barcode type that you're using, in SimpleIndex or any other application you use to read barcodes printed with SimpleCoversheet. Code 128 is used by default because it supports a large character set, including special characters. However, sometimes there is more information than space on the page. If you have a large amount of information that you want to put into barcode format, and have limited space available, we suggest using the PDF417 barcode type, which is a 2D barcode that can fit more information into a limited area.

2 SimpleCoversheet

SimpleCoversheet provides an interface for designing, populating and printing barcode document separator sheets, sheets of sticky labels (2.2), and any other type of document that contains barcodes. These cover sheets and other barcodes allow documents to be automatically indexed after capture. The scanning process is most efficient whenever such cover sheets can be printed by the person generating the document.

The SimpleCoversheet license allows you to register SimpleCoversheet on as many computers at your location as you wish. Companies with multiple offices must obtain a SimpleCoversheet license for each location they wish to use it in.

2.1 Interface

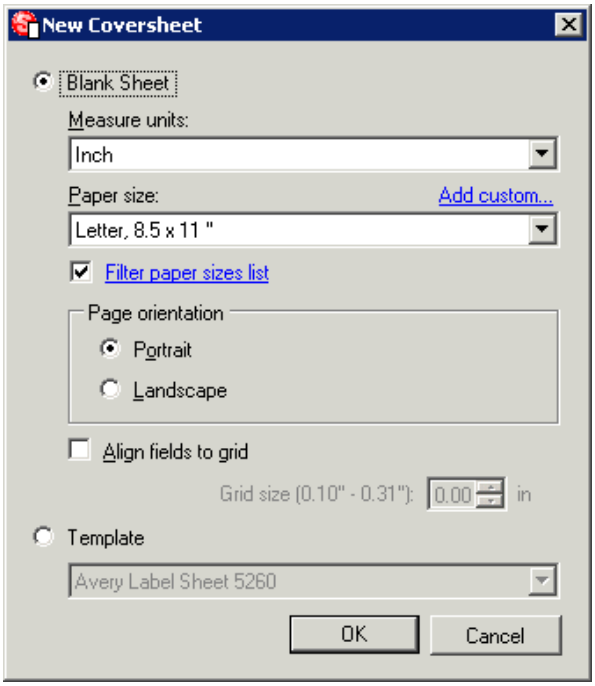
SimpleCoversheet works via two modes; designer mode and user mode. Cover sheet templates are laid out and linked to data sources in design mode, and, once designed, are ready for user mode. In user mode the appropriate databases and resource files are accessed by the program, according to your configuration, in order to compose the cover sheets. You can then browse through the available cover sheets and print what you need. You may switch between design and user modes via the toolbar, or via the *Mode* menu.

2.1.1 Design Mode

In design mode you are presented with a blank cover sheet, and a number of elements that you can put on the page. The placement and configuration of these elements will determine how your cover sheet looks and behaves in user mode. On the leftmost pane of the design mode interface is a list of elements you can place on to a cover sheet. In the middle pane is your cover sheet as viewed in design mode, and on the rightmost pane the properties for the selected element or cover sheet are displayed. The selected element can be changed using the drop down list of element names above the list of properties or by selecting an element on the cover sheet with the pointer. The properties can also be arranged by category or alphabetically via buttons below the drop down list.

2.1.2 The Cover Sheet

When SimpleCoversheet first opens up you are presented with a blank window and have access to the cover sheet toolbar and menu. In order to create a new cover sheet either click *New* on the toolbar, or select *New* from the *File* menu. On the *New Coversheet* dialog you may choose whether to create your cover sheet from a blank sheet, or from a template (2.2). If you create the cover sheet from a template, the size, measurements, and such will be chosen for you from the template. If you decide to create a new cover sheet from a blank sheet, you will need to know what size paper you intend to print the cover sheets. You are also able to configure the measurement units setting and the grid properties for the new cover sheet.



The measurement units control how the rulers are displayed, as well as the (X, Y) coordinates that determine the placement and size of elements placed on the cover sheet. There is also a list of paper sizes to choose from that can be filtered based on different standards and locations. If you can't find a size that you need, click the *Add custom…* link in order to add a custom size to the list. Next you can decide if you want to have SimpleCoversheet help you lay out the fields on your cover sheet by aligning them to a grid. For instance, if *Align fields to grid* is checked and the grid size is 0.10 inches, then any time you place an element on the cover sheet, its location will be moved to the nearest increment of 0.10 inches. This helps to align and evenly space the elements on the page and

will make your cover sheet more visually appealing and easier to read. However, if you decide not to turn on the *Align fields to grid* option then when you draw an element on the page it will stay exactly where you put it. The grid options are also available in the list of cover sheet properties in design mode, and can be changed later if desired. Once you have everything setup as desired, click *OK* to have the program generate your new, blank, cover sheet. On the right side of the interface, properties of the cover sheet will be listed. This is where you can alter the grid settings, as well as the background color or image and the default font settings for elements placed on the cover sheet. The cover sheet will change to reflect an altered property when you hit [Enter] on the keyboard, or the property field loses focus. Note that the *Measurement units*, *Page orientation*, and *Paper size* properties can only be changed when creating a new cover sheet.

2.1.3 Elements

Elements are the visible and invisible parts that make up your cover sheet, and how they are configured governs how your cover sheet is displayed in user mode (2.1.5). The list of elements on the left most pane of the interface, when selected, will change how the cursor functions and will allow you to draw the selected element on the cover sheet, starting the process of adding the element to the cover sheet. After drawing a new element, a wizard will be displayed that will allow you to configure the placed element. This is true for all elements except the *Pointer* which allows you to select elements that are already on the cover sheet, and to adjust their position.

You can also insert barcode (2.1.3.5) and label(2.1.3.4) elements by using the toolbar buttons labeled *Insert Barcode* and *Insert Label*. In addition to inserting barcodes and labels from the toolbar, there are also toolbar buttons that will allow you to align elements, center them on a page, and change which elements appear on top of or behind others. In order to use this toolbar, just select the elements on the page using the pointer, and then press the appropriate toolbar button. Be aware that when selecting a single element, the *align to top*, *bottom*, *left* and *right* buttons will not be available as they require multiple elements to align against. To select multiple elements at the same time, hold down [Ctrl] while clicking on each individual element. Once multiple elements have been selected you can

then align them in accordance with last element selected via the toolbar buttons.

2.1.3.1 Element Hierarchy

Certain elements, namely the labels and barcodes, can depend upon other elements for their values and configuration. The hierarchy of the elements and their dependencies is as follows:

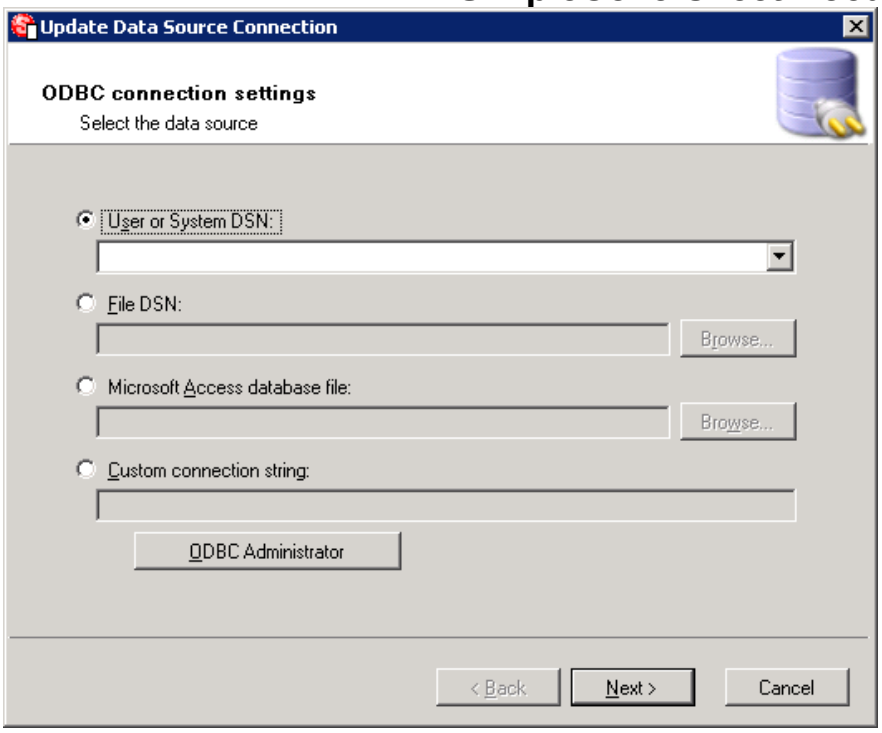
Barcode
    Data Source
        Data Source Connection
Label
    Data Source
        Data Source Connection
Text

Note that it isn't necessary to have data source connection(2.1.3.2) and data source(2.1.3.3) elements in order to create labels and barcodes, but if you want the barcodes and labels to have values from a database then you will need to create some of these elements.

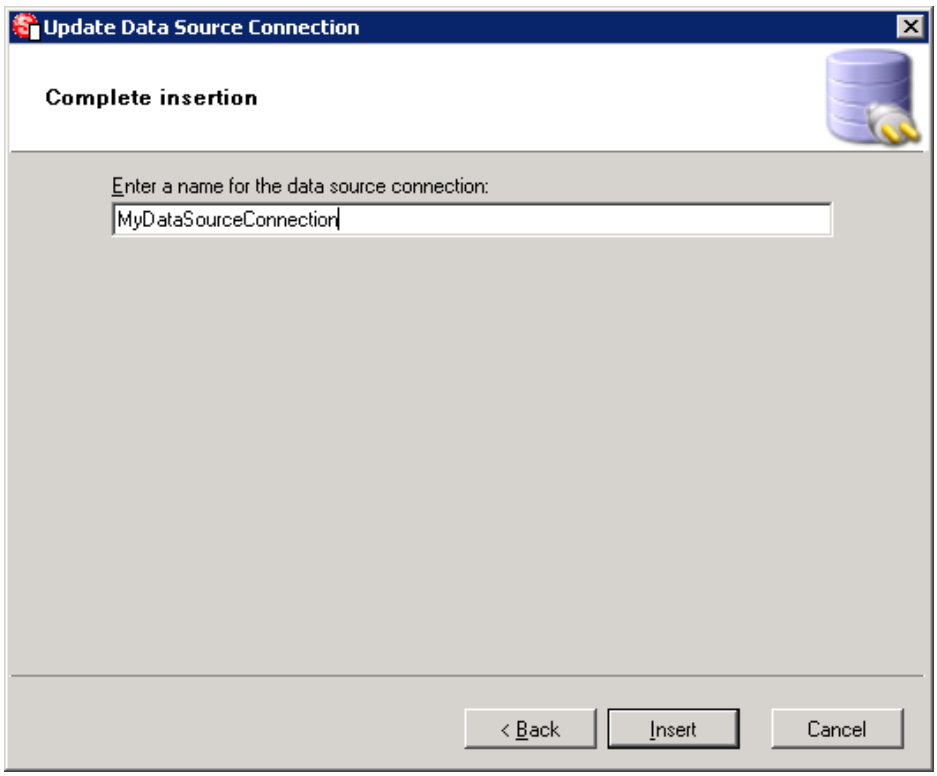
2.1.3.2 Data Source Connections

Data source connections tell the program how to connect to an ODBC data source that you wish to retrieve values from. They are created, like all the other elements, by drawing them on to the cover sheet. However, unlike some of the other elements, they are not directly visible on the cover sheet; they will appear on the strip below the cover sheet display.

After drawing the data source connection on to the cover sheet you will be presented with the data source connection configuration wizard. This wizard will also be displayed if you created a new cover sheet from a template(2.2). You have the options of using DSN connections(User, System, or File), Access database files, or a custom connection string. If you need to create a new DSN for your data source connection, click the *ODBC Administrator* button and it will appear and allow you to create and configure one.

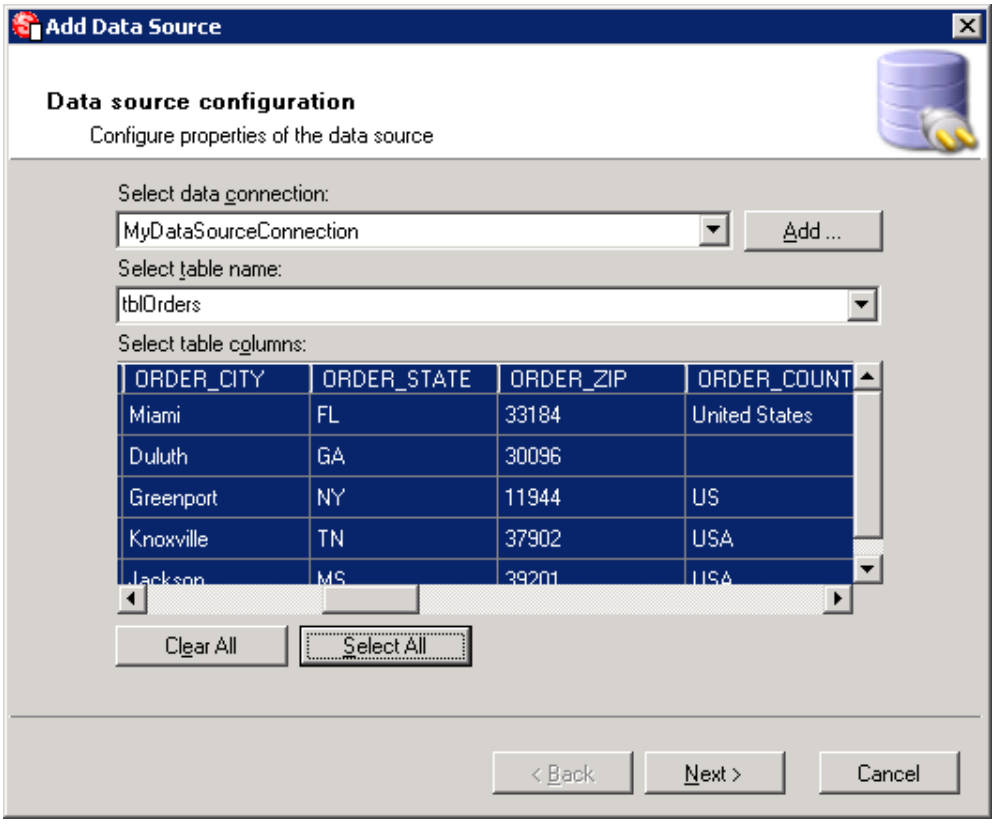


After selecting where the data will come from you can optionally enter a login and password to be used with the DSN. And finally, if not configuring a template, you can give the data source connection a name that you can remember in order to reference it when creating a data source element.



2.1.3.3 Data Sources

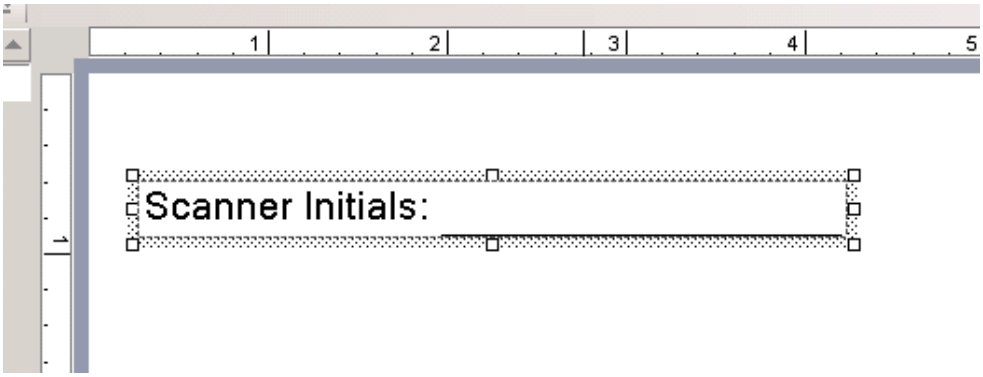
Data sources are dependent on data source connections. The first step in setting up a data source in the data source configuration wizard is selecting one of these connections. Note that if you are creating a new cover sheet from a template(2.2), the data source connection will be automatically selected. So if at this point you have not created a data source connection you may do so by clicking the *Add...* button next to the drop down of data source connections, which will bring up the data source connection configuration wizard.



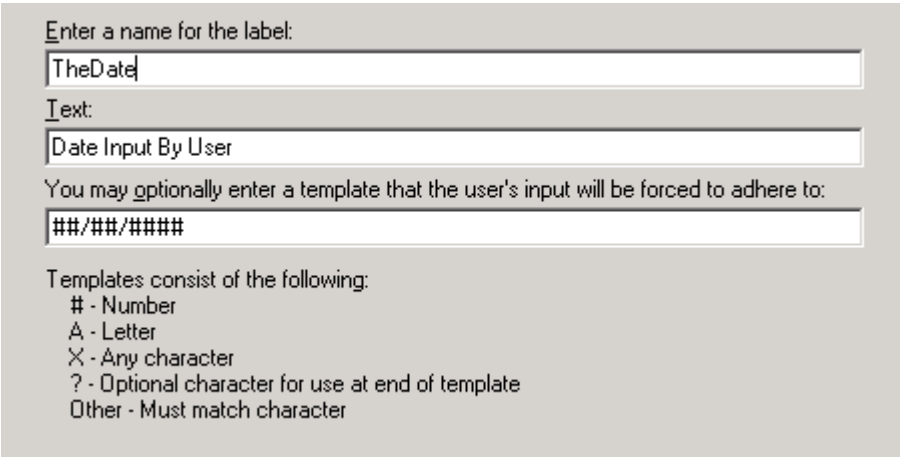
Once you've selected a data source connection, the drop down of tables available from the connection will be populated, allowing you to select one. After selecting a table, the grid display below will show the contents of the selected table. At this point you will be able to select which columns in the table you would like to make available to use as values for labels and barcode elements. Finally, if not configuring a template, you can give the data source a memorable, descriptive name so that you can easily reference it when configuring labels and barcodes.

2.1.3.4 Labels

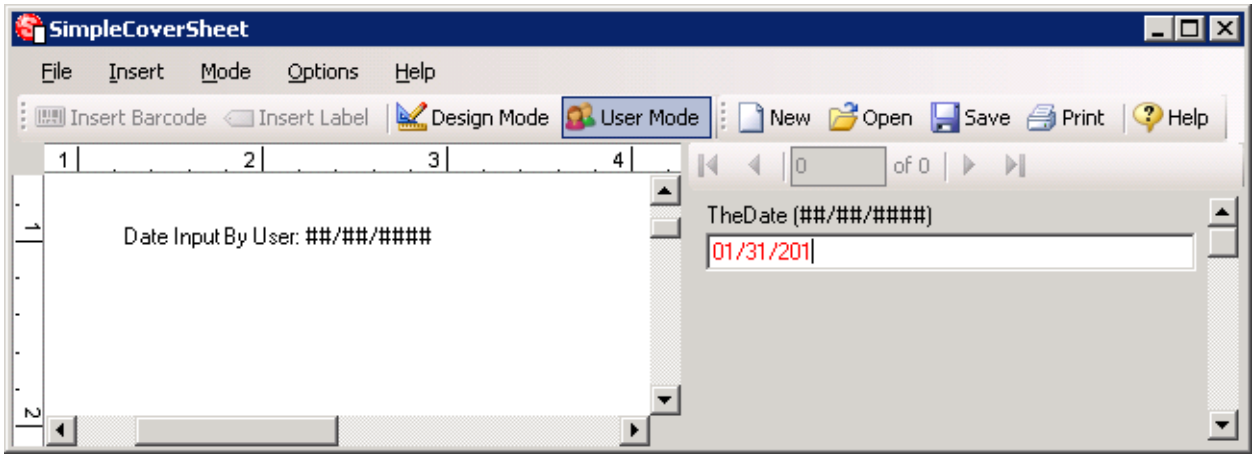
Labels are elements of textual information on the cover sheet that can be handwritten after printing, input by the user before printing, or selected or populated from data sources or text files. This will be evident to you after drawing a label on to the cover sheet as you will first have to decide how the label should behave when the label configuration wizard appears. The *handwritten after printing* option is one of the most basic and will create labels that can be filled in by hand after the cover sheet has been printed.



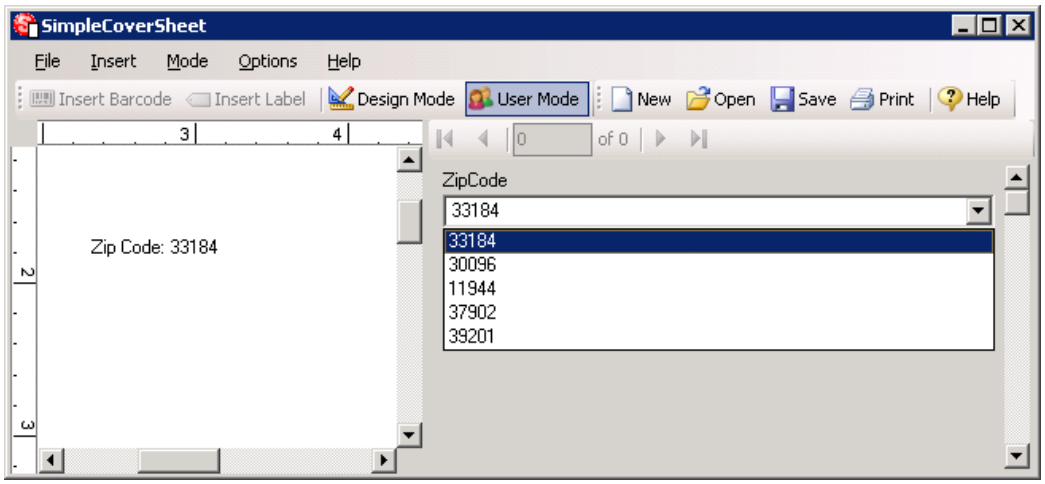
This is useful, for example, when wanting to do something such as having the person who scans a batch initial the cover sheet. The *input by the user before printing* option is similar except that the value has to be entered by the user in user mode before printing the cover sheet. If the values fit a certain format that should adhered to, you can create a template for the format that will dictate valid and invalid values during user mode(See Templates Section for more information on templates). If an invalid value is present the program will not allow the cover sheet to be printed. So a label configured in the following manner:



Ends up behaving like this in Design Mode(once the final zero is entered for 2010 the field will turn black and be accepted for printing):



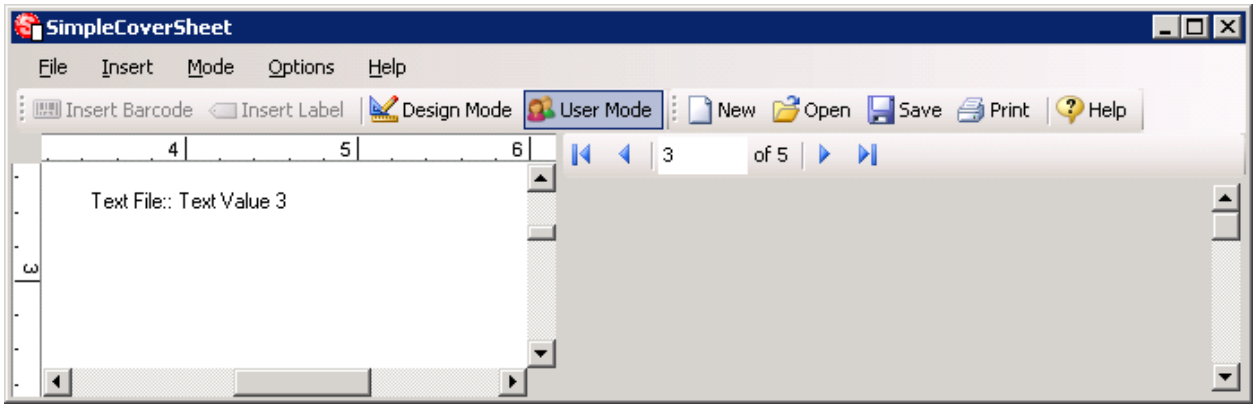
The remaining options for how a label's value will be generated are either by selection or population from text files or columns in a data source. The selection types are another way of restricting the values available in user mode. For these options, during user mode, a drop down will be shown containing all the values from the specified text file or column in a data source.



If you want to restrict the user to only being able to use values from the list then check the option labeled *Only allow values from list*. Otherwise the list is basically a list of suggestions, allowing the user to select options, but also to type in a value if necessary. If you choose to get your values from a text file, all you need to do is specify the location of the text file. The program assumes there is one option per line in the file. When using a data source, you will need to select the data source and then the column of values to use. If at the point when you need to choose a data source, you

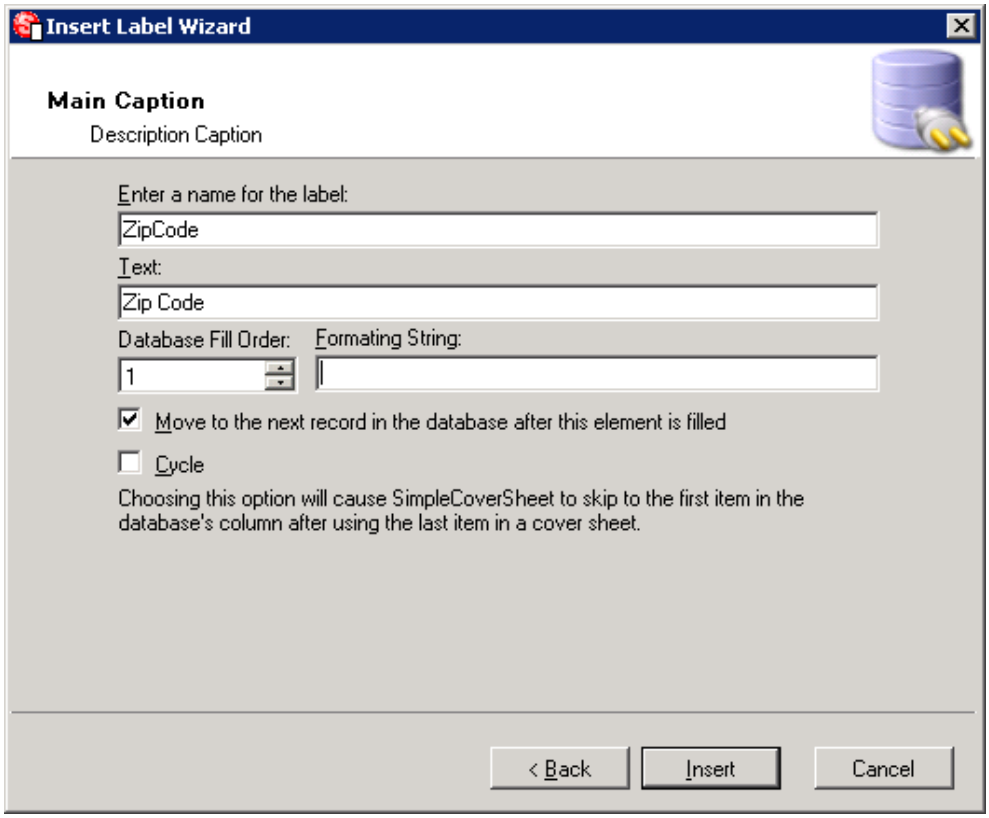
have not created one already, just click the *Add…* button next to the drop down of available data sources. This will bring up the data source configuration wizard.

The *populated from a text file* option is fairly similar to the setup for the *selected from a text file* option. The difference is in the behavior of the cover sheet in user mode. When a field is populated from a text file, multiple pages are created, where the different values in the file appear each on a separate cover sheet page. So if you have 5 values in your text file, then you would end up with 5 cover sheet pages, each one having one of the 5 values.



The number of cover sheet pages in user mode can change depending on how many labels you have and how many values each of them have. The *cycle* option will cause there to be an unlimited number of cover sheets if there is no other element with a limited number of values. With the *cycle* option enabled the 5 values in a text file or data source column will be repeated over and over again, every 5 pages, ad infinitum, unless there is another element to limit the number of pages.

When *populating from a data source*, the configuration is similar to the configuration when *selecting from a data source*, but the behavior in user mode is more similar to the *populated from a text file* option. Some of the properties that can be set at the end of *populated from a data source* configuration are different as well, and are exclusive to this option. These are the *Move to next record after element is filled* and *Database Fill Order* properties. By default, these will be set automatically, and you can change the automatic behavior via the *Options* > *Database Fill Order* menu option. The *Database Fill Order* property allows you to determine what order elements receive the values from the same data source and column. The values are positive integers (1, 2, 3, etc…) and dictate the order, where the



smaller values get populated first. The *Move to next record after element is filled* property tells the program to skip to the next record in the data source after populating the value for the element in question. So if you had a page of labels(or barcodes) to be printed, and wanted to have a different value for each one then you would want to check the *Move to next record after element is filled* property for each individual element, or select *One Record Per Element* from the *Options* menu under *Database Fill Order*. Alternatively, if you were creating barcode separator sheets where you only need values from one record per page, then you only need to check the *Move to next record after element is filled* property on the last label on the page, or select *One Record Per Page* from the *Options* menu under *Database Fill Order* to have it done automatically. For more information, see section 2.1.6.

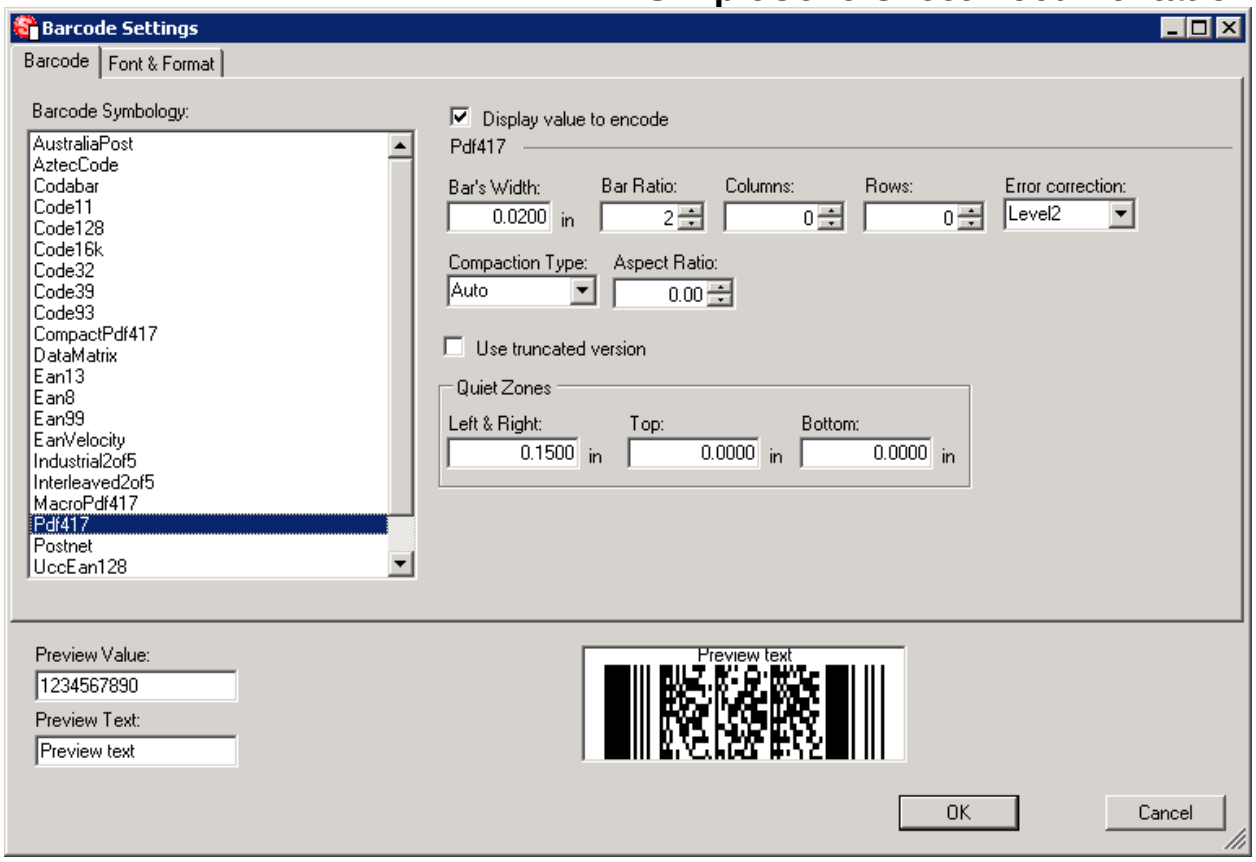
2.1.3.5 Barcodes

Barcodes, when it comes to configuration, are nearly identical to labels. Their behavior in user mode is nearly identical as well, except that the values, instead of being printed as text, are encoded into a barcode. The only real difference in configuration is the *handwritten after printing* option

is removed in exchange for the *static* option. The *static* option creates a barcode with a fixed value that does not change from page to page. These are useful when needing to create a barcode separator sheet that is only used for separation and the value isn't read.

After you are done with the configuration wizard, you are presented with the configured barcode on the page in design mode. At this point you may change how the barcode appears. By default, a code 128 barcode is used because it supports a larger character set than other barcode types. However, you may use other barcode types if you so desire. For instance, sometimes the amount of information you want to have encoded creates a barcode that is too large for the page. In this case we recommend using a two dimensional barcode such as the PDF417 type as it will allow you to fit a lot of information into a smaller space. To do this, right-click on the barcode that has been placed on the cover sheet and choose *Barcode settings*.

This will bring up the barcode configurator. On the left hand side of the barcode configurator is the list of available barcode symbologies that the program supports and that can be read by SimpleIndex. To change the barcode type, just select one of the options in the list. This will also display a page of options for that individual barcode symbology. For the most part, you can leave the options at their default values. In addition to changing the barcode type or symbology, you may also change the format and location of the text that accompanies it, the colors, and the orientation. You may also choose whether or not to display a textual version of the value that's encoded. Once you are done configuring, click OK to have your changes applied to the cover sheet being designed.



When using SimpleCoverSheet in conjunction with SimpleIndex, it is important to properly set certain barcode settings so that SimpleIndex will read them properly. Firstly, it is recommended to use a code128 type barcode as this type of barcode supports a full character set and fits well on a page. Code39 is a commonly used type of barcode, and when using code39 it is important to note that this type of barcode only supports upper case letters and digits. You can check the *Extended* option on the code39 configurator page to include lower case letters, but be aware that you must explicitly set SimpleIndex to read this type of barcode. Otherwise, any lowercase letters, such as ‘a’, will be read as ‘A+’ which is how the lower case letters are represented internally by the barcode type. Lastly, do not use a checksum when using code39 type barcodes as SimpleIndex does not process the checksum and consequently reads it as an extra character.

2.1.3.6 Text

Text elements are fairly basic. They essentially allow you to place any static text that you may need on to the cover sheet. You can use them to create something such as a cover sheet title. Once you have drawn a text element on to the cover sheet, you may change the font settings via the properties pane on the right side of the design mode interface.

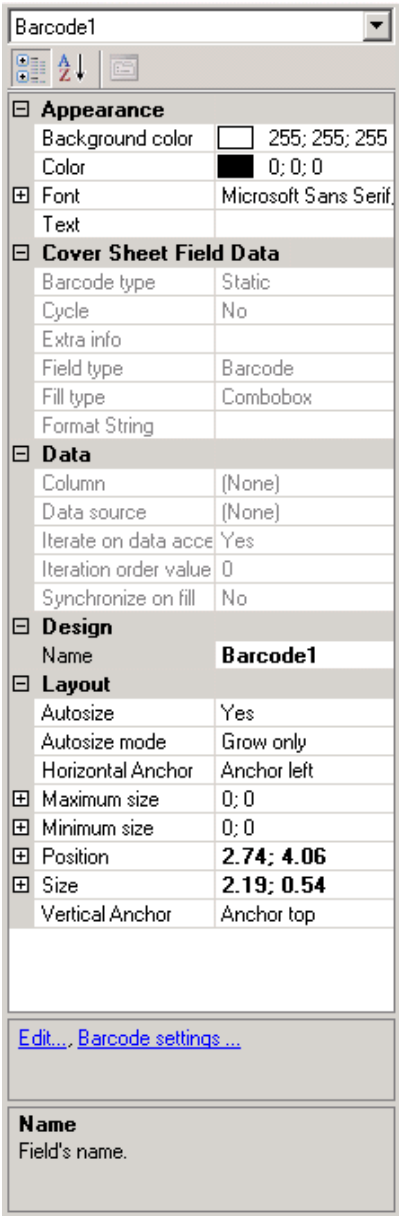
2.1.4 Element Properties

In addition to the properties that are set during element configuration via the wizards, there are other properties that are available to be changed via the properties pane in design mode.

2.1.4.1 Layout Properties

The following properties are valid for most elements.

Autosize – Determines whether or not the element should change size based on its contents in user mode. If some values for a label or barcode are longer or shorter than the previous values or designed size, then this allows the element to change size. Turn this off if you would like the text in an element to wrap.



Autosize mode – Specifies whether the element should only grow to accommodate large values, or if it should also shrink when a smaller value is being used.

Horizontal Anchor – Specifies in which direction the element should grow and shrink when the horizontal size is hanged by the autosize option. Anchoring the element on the left side will cause it to grow to the right. Anchoring the element on the center will cause it to grow to the left and right while remaining centered. Anchoring the element on the right side will cause it to grow to the left.

Maximum size – Specifies the maximum size to grow to when autosize is enabled. A value of '0; 0' means that there is no maximum size.

Minimum size – Specifies the minimum size to shrink to when autosize is enabled. A value of '0; 0' means that there is no minimum size.

Position and Size – These values can be changed via the properties pane, but it is much easier to just resize the element or drag it to a new position with the pointer.

Vertical Anchor – Specifies in which direction the element should grow and shrink when the vertical size is changed by the autosize option. Anchoring the element on the top will cause it to grow ownward. Anchoring the element on the center will cause it to grow both up and down while remaining centered. Anchoring the element on the bottom will cause it to grow upward.

2.1.4.2 Appearance Properties

Background color – Changes the background color of the selected element or cover sheet.

Border style – Changes whether or not the element has a border, and what type of border. A *Fixed Single* border is a small, thin, flat border. A *Fixed 3D* border is a beveled border like on windows style forms and buttons. Only applies to labels and text.

Character(s) separating the text and value – By default this is a colon and then a space, but can be changed or removed entirely. Only applies to labels.

Color – Allows for the color to be changed.

Font – Allows for the font to be changed.

Text – Allows the text to be changed for elements that are static.

Text Alignment – Determines how the text should be aligned inside the borders of the element.

2.1.4.3 Cover Sheet Field Data Properties

These are only for display and cannot be changed via the properties pane. They give basic configuration information about the currently selected element. To change these properties you must either create a new element or right-click on the element and choose *Edit* so that the configuration wizard starts and the changes can be made there.

2.1.4.4 Data Properties

Column – Specifies which column to get values from in the Data Source.

Data Source – Specifies which data source to get values from.

Move Next Record After Fill – Specifies whether or not to skip to the next record after populating the value for the element in question.

Database Fill Order – Specifies the order in which the element should be populated in regards to other elements with the same data source. This can only be changed via the configuration wizards.

Synchronize on Fill – Only available for elements populated by a data source, or selected from a data source. When two or more elements with the same data source both have *Synchronize on Fill* enabled, the fill functionality changes to synchronize the two or more elements. So selecting or populating a value from a record in a data source for one element, would cause all the synchronized elements' values to change to their corresponding column values of the same record.

2.1.4.5 Design Properties

The only design property is the *Name* property which is an internal name used by elements to reference each other. For instance, a data source with the name of 'BindingSource1' is referenced by a label or barcode whose values are populated by it.

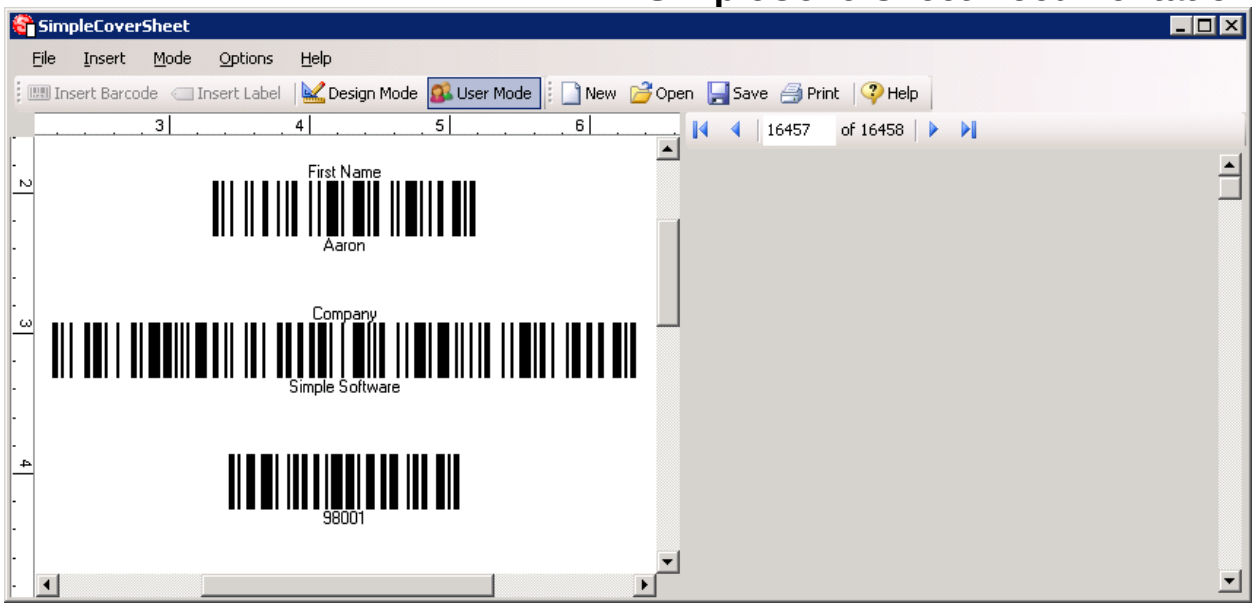
2.1.4.6 Miscellaenous Properties

Use Verifier – This is used for elements where the value will be input by

hand. It adds another level of verification when inputting values by requiring the value to be input twice. When setting this option to *True*, it causes SimpleCoverSheet to create a second input box in User Mode. If this second box's value is not the same as the first box's value then you will not be able to print. The following figure shows what it would look like in User Mode if an element named *RequiresVerification* is on the page whose *Use Verifier* option is set to true.

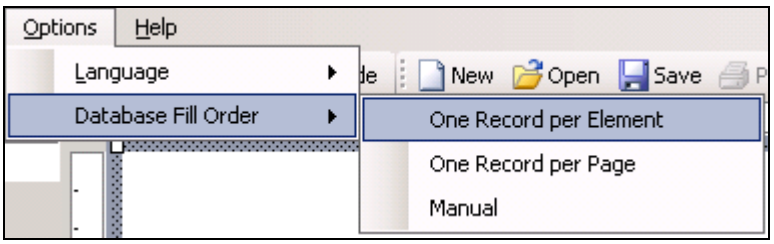
2.1.5 User Mode

To enter User Mode, either click *User Mode* on the toolbar, or select it on the *Mode* menu. User mode is where the designed cover sheets are put together by the program and displayed. On the left side of the interface in user mode the current cover sheet is displayed. On the right side are all the options for changing the values of elements on the cover sheet(s), as well as page navigation options. Any element that is configured to be selected or entered will have a drop down list on this pane that allows you to select or enter a value. Values displayed in red are invalid values and the program will not allow you to print the cover sheet until a valid value has been entered. Above the available values are the page navigation controls. If any of your elements are populated from text files or data sources then it's likely that you will have multiple pages available to you, as the program creates a page for each available value. Once you have entered valid values and have the cover sheets looking how you want them, you can print them via the *Print* command under the *File* menu. At that point, you may also choose if you want to print one page, a range of pages, or all the pages at once.



2.1.6 Database Fill Order Options

The *Database Fill Order* and *Move Next Record After Fill* are used to control how data is pulled from a data source and populated onto the page. The *Database Fill Order* property is a number, and when SimpleCoverSheet populates elements with data, it goes by the Database Fill Order numbers, starting at the lowest number and going to the highest number. The *Move Next Record After Fill* option tells SimpleCoverSheet to skip to the next record in the data source after the element's value has been populated. With the release of 3.0, these options can be set automatically by using the *Options > Database Fill Order* menu option. The available options are *One Record per Element*, *One Record per Page*, and *Manual*. The automatic options will automatically setup the *Database Fill Order* numbers by numbering elements from top to bottom and left to right.



The *One Record Per Element* option should be used in cases where each element on the page has a value from a different record and sets the *Move Next Record After Fill* property to true for each element on the page. For instance, if you were printing a page of sticky labels, this would be the

option to select as each barcode would contain a different value, each from a different record in the database.

The *One Record Per Page* option should be used in cases where the values from a single record will populate different elements on the page, and you want each page to have values from a different record. Selecting this option causes SimpleCoverSheet to set the *Move Next Record After Fill* option to true for the bottom right most element on the page, but false for the rest of the elements.

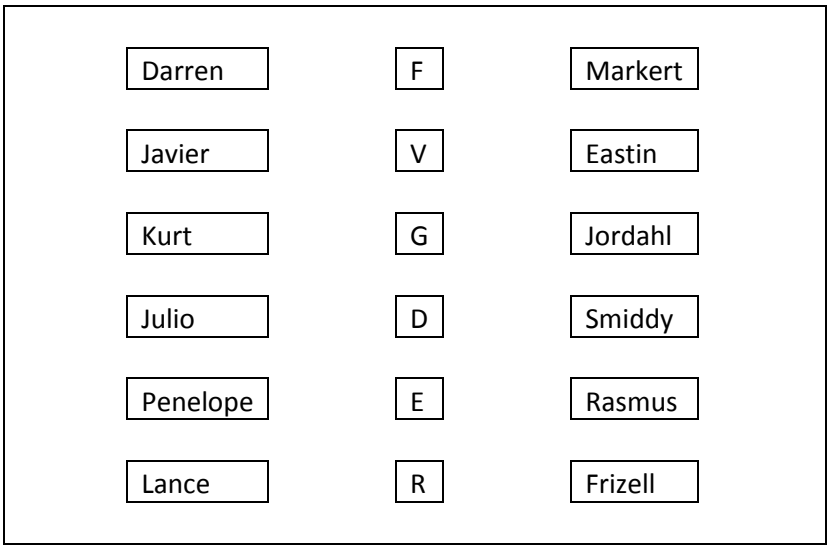
The *Manual* option allows you to manually set the *Database Fill Order* and *Move Next Record After Fill* by selecting an element on the design surface and modifying its properties in the properties pane. This should be used in cases where the automatic options are not capable of populating the data as you need. Suppose you had a coversheet with 6 rows of information and 3 elements on each row, and each row represents a different record in your database. In order to accomplish the required data population behavior, you would need to number the *Database Fill Order* from top to bottom, left to right, and every third element you would need to set the *Move Next Record After Fill* to true, and the rest to false. The following figures will illustrate this. Assume you have the following data:

First Name	MI	Last Name
Darren	F	Markert
Javier	V	Eastin
Kurt	G	Jordahl
Julio	D	Smiddy
Penelope	E	Rasmus
Lance	R	Frizell

In order to get one full name per row, the elements would need to be setup as follows(assume each square is a label or barcode element):

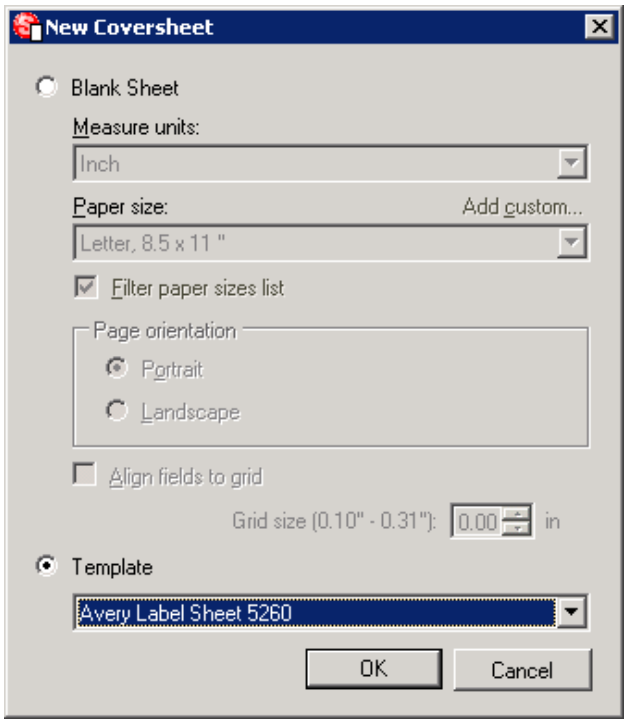
First Name Row 1 Move Next Record After Fill FALSE Database Fill Order 1	Middle Initial Row 1 Move Next Record After Fill FALSE Database Fill Order 2	Last Name Row 1 Move Next Record After Fill TRUE Database Fill Order 3
First Name Row 2 Move Next Record After Fill FALSE Database Fill Order 4	Middle Initial Row 2 Move Next Record After Fill FALSE Database Fill Order 5	Last Name Row 2 Move Next Record After Fill TRUE Database Fill Order 6
First Name Row 3 Move Next Record After Fill FALSE Database Fill Order 7	Middle Initial Row 3 Move Next Record After Fill FALSE Database Fill Order 8	Last Name Row 3 Move Next Record After Fill TRUE Database Fill Order 9
First Name Row 4 Move Next Record After Fill FALSE Database Fill Order 10	Middle Initial Row 4 Move Next Record After Fill FALSE Database Fill Order 11	Last Name Row 4 Move Next Record After Fill TRUE Database Fill Order 12
First Name Row 5 Move Next Record After Fill FALSE Database Fill Order 13	Middle Initial Row 5 Move Next Record After Fill FALSE Database Fill Order 14	Last Name Row 5 Move Next Record After Fill TRUE Database Fill Order 15
First Name Row 6 Move Next Record After Fill FALSE Database Fill Order 16	Middle Initial Row 7 Move Next Record After Fill FALSE Database Fill Order 17	Last Name Row 8 Move Next Record After Fill TRUE Database Fill Order 18

This would create a cover sheet that looks similar to the following:



2.2 Avery Label Templates

SimpleCoversheet allows for the easy creation of sheets of barcodes on Avery Labels via a template system. A template is basically a saved cover sheet configuration with empty values that are configured by the user directly after creating a new cover sheet. This saves you time in that all you're required to do is configure the database that the values will be pulled from. To begin creating a sheet of Avery Labels, choose to create a new cover sheet from the file menu or the toolbar.

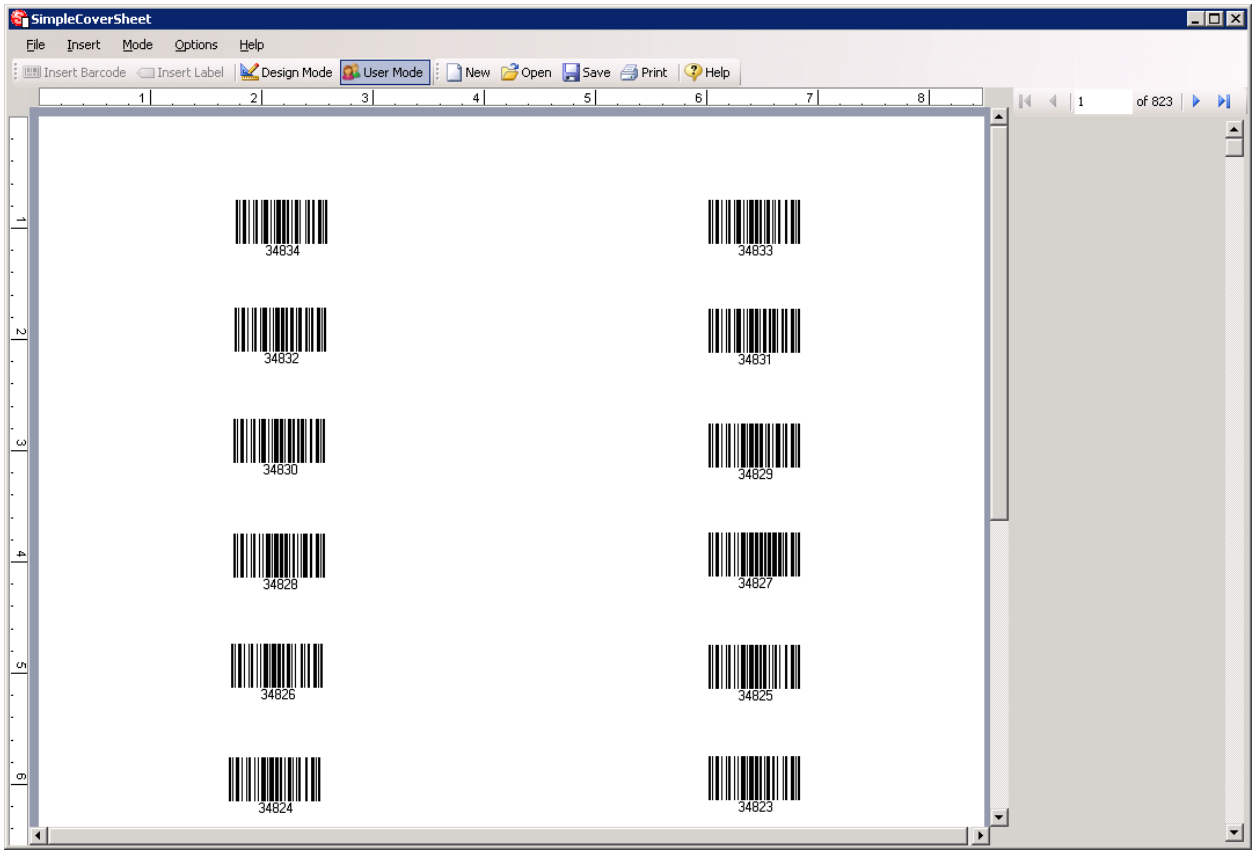


2.2.1 Template Configuration

The template configuration is fairly straight forward if you have previously created any cover sheets. When creating a new cover sheet from a template you will be presented with the configuration wizards for the elements in the template. Once you have finished configuring the template elements the program will switch to user mode(2.1.5) and allow you to print your configured cover sheets.

Each sheet of barcodes created with the Avery Label templates, get their values from a data source. So in order to create a sheet of barcoded sticky labels you must configure the elements in the template with a data source containing the values to be used on the barcodes. The first element to configure via the wizard in an Avery Label template will be the data source

connection (2.1.3.2). This determines the location of the data, whether it be from a database server such as an SQL server, a file such as an access database, an excel spreadsheet, or any other ODBC compatible connection. After configuring the data source connection, you will need to configure the data source (2.1.3.3) via the configuration wizard. When configuring a data source you will need to select a table in the database, and then you will need to select the columns of data that you wish to make available as values for the barcodes. Lastly, in the final part of the configuration wizard, you will configure the barcodes (2.1.3.5) by selecting the column of data you wish to use as values. Now, if everything has been properly configured, the program will switch to user mode (2.1.5) and will display a sheet of barcoded sticky labels encoded with the values from your database. At this point you are now ready to print your newly created cover sheet. Also note that in order to keep from having to reconfigure your cover sheet again, be sure to save it.



2.3 Command Line Interface

SimpleCoversheet can be called from the command line to automatically print coversheets and optionally hide the interface. One can also add filters to database queries, and modify parameters stored in SQL statements by a configuration. Lastly, one can group elements from separate binding sources to create line item type functionality. Usage is as follows:

SimpleCoversheet.exe [File Name]
SimpleCoversheet.exe [http://www.simpleindex.com/coversheet.xcs]
SimpleCoversheet.exe [File Name] [/print] [/hide]
SimpleCoversheet.exe [File Name] [/filter:field_name=value] [/param:@parameter_name=value] [/groupby:field_name]

Note:

To include a space in a value use quotations to surround it. For different data types when using the filter and param switches you need to specify the proper enclosing characters; such as single quotes(‘) for string literals.

Options:

/print Automatically print the cover sheets from the specified configuration.
/hide Hides the interface when /print is specified.
/filter:field_name=value
Adds a “WHERE field_name = value” clause to any SQL statements used in the configuration.
/param:@parameter_name=value
Replaces *@parameter_name* with *value* in any SQL statement used in the configuration.
/groupby:field_name
Groups a second binding sources values to a repeated value in a first binding source

Examples:

```
SimpleCoversheet.exe BarcodeSheet.xcs /print /hide
```

This command line will open the configuration name barcodesheet.xcs and then print all its pages without showing the SimpleCoversheet interface.

```
SimpleCoversheet.exe AddressLabels.xcs /filter:CITY=Knoxville
```

This command will add a 'WHERE CITY = Knoxville' clause to any SQL statement utilized in the configuration. Assume there is a data source connection in *AddressLabels.xcs*, and there is a binding source that utilizes a table named tblCustomers that has all kinds of information necessary for printing address labels. Adding the clause specified above will cause SimpleCoversheet to only retrieve records from the table where the *CITY* field is equal to *Knoxville*.

```
SimpleCoversheet.exe Expirations.xcs /param:@expire_date=2010-09-10
```

Assume Expirations.xcs is bound to a query or view that has parameters, specifically a parameter named *@expire_date*, and that the query looks something like this:

```
SELECT * FROM tblMaintenance WHERE EXPIRATION_DATE = '@expire_date'
```

Prior to retrieving the records from the database, SimpleCoversheet will replace the @expire_date in the select statement with the value specified on the command line so it ends up looking like this:

```
SELECT * FROM tblMaintenance WHERE EXPIRATION_DATE = '2010-09-10'
```

```
SimpleCoversheet.exe LineItems.xcs /groupby:ORDER_NUMBER
```

This command line specifies that SimpleCoversheet should group all records in a second database with a repeated value in the *ORDER_NUMBER* column in relation with each unique *ORDER_NUMBER* column in a first database. Assume there is a binding source mapped to a table named *tblOrders* with a record containing the value *467D-2839* in the *ORDER_NUMBER* column, and a second binding source mapped to a second table named tblOrderItems with 5 records containing *467D-2839* in the *ORDER_NUMBER* column. When SimpleCoversheet retrieves the values from the second binding source and table it will retrieve the records grouped by the value of the *ORDER_NUMBER* column in the first binding source.

3 Contacting Simple Software

SimpleIndex is a product of Simple Software from ScanStore. Simple Software and ScanStore are divisions of Meta Enterprises, LLC.

SimpleIndex Website: http://www.simpleindex.com

Mailing Address:
PO Box 548
Knoxville, TN 37901-0548

Physical Address:
500 West Summit Hill Drive, Suite 302
Knoxville, TN 37902

Sales/General E-Mail: info@SimpleIndex.com

Tech Support E-Mail: support@SimpleIndex.com

Phone Number: 877-355-4141