

TymeTrax Software Report Designer User Guide



Table of Contents

Click the 'Table of Contents' link at the bottom of any page to return here.

The Report Designer

Overview.....	Page 3
Adding Fields from the Parent Table	Page 5
Creating Calculated Fields for Reports	Page 6
Query Builder: The Non-Techy Way to Write Data Queries.....	Page 7
How the Company Table Links to Other Data.....	Page 8
Building Report Data Queries.....	Page 9
The Report Layout Designer	Page 11
The Designer Toolbars	Page 12
Other Options in the Layout Designer	Page 14
The Layout Designer Grid	Page 15
Using Context Menus	Page 16
Adding Company Logos & Other Info	Page 17
Using Custom Scripts to Fine-Tune Reports	Page 19

Tutorial - Building A Report Design from Scratch

Tutorial Overview	Page 21
The Report Query: Choosing the Data To Display	Page 23
The Layout: Start at the Top... The Page Header	Page 25
Automate Logos & Company Info with Custom Scripts	Page 28
Group Header & Detail Sections: Adding Data Fields & Formatting	Page 31
Adding Footer & Page Breaks	Page 33
Viewing the Results	Page 34

Additional Information

TymeTrax Recommended System & User Requirements	Page 36
Contact Information	Page 37



Report Designer Overview

Unlike many applications that require a third-party or additional application for creating reports, all TymeTrax applications have a built-in Report Designer that gives you complete control over building report designs.

The Report Designer stores all designs in the same database as your data, which means that all changes are available to all users without manually having to distribute separate templates. Active reports are available to users from the Print menu on each Toolbar.

Many report templates are included in the Sample database and when you create a new database, the sample reports are copied to the new database automatically. Report designs can also be imported and exported between two databases using the Utilities, Import/Export functions on the Toolbar.

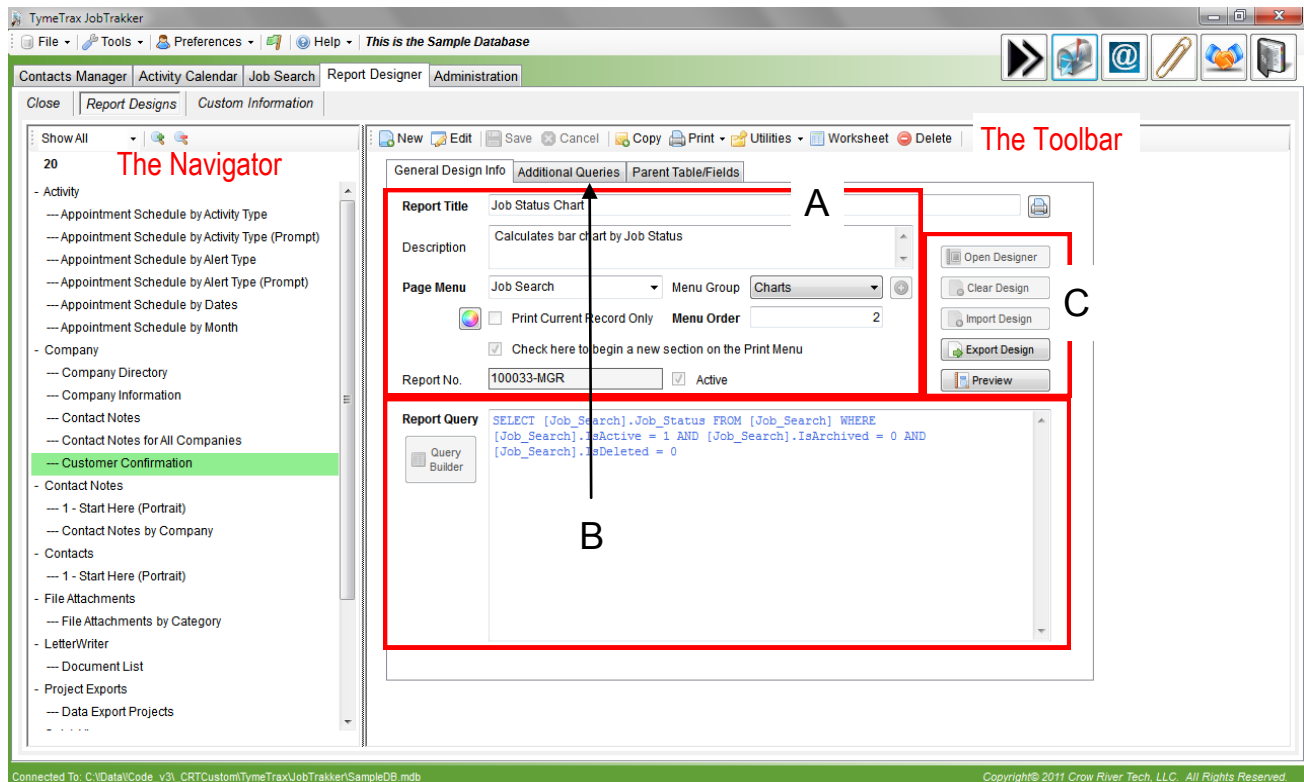
The Report Designs Tab

The Report Designer interface works exactly like all other Departments in the system, with the Navigator on the left that organizes reports by the Department name, and the Toolbar on the form that has the same functions as other pages in the system.

The Custom Information Tab

The “Custom Information” tab lets you define your company logo and other images and all business information in one place. Each report can be designed to use that information when the report is generated through the use of custom scripts. This makes it easy to make changes to your logo or address info without having to edit each report when the images or information changes. See Page 17 for screenprints and how-tos.

Please note that this guide uses TymeTrax CRM application for demonstration purposes; however the functionality is the same for all.



Each Report Design has three essential parts: A) The fields that determine where and how the Report will be displayed (Report Title, Page Menu, Menu Order, Print Current Record Only); B) the queries that determine where the data is coming from; and C) the Report Design (template) that's created in the Layout Designer.



The Report Designer Overview

The Report Designer interface looks like this:

NOTE: The Active Checkbox - When this is checked the print option is put on the "Page Menu's" print menu. To remove a print option, uncheck this box. This is checked by default when creating a new record, but unchecking it gives you the opportunity to test your print design from this tab before other users will see it on their print menus.

The screenshot shows the Report Designer interface with the following callouts:

- Report Title:** Job Status Chart. Callout: "Each report needs a unique title; the title is displayed in the Print menu."
- Description:** Calculates bar chart by Job Status. Callout: "Enter a short description of what this report displays. Users view this information from a Quick View called 'Company Directory - Print Menu Options' on the Utilities Toolbar to help them choose which report they want to run."
- Page Menu:** Job Search. Callout: "Defines a sub-category on the Print menu. Use this value to group reports within a Page Menu."
- Print Current Record Only:** Unchecked. Callout: "This is the Page the print option will be displayed on."
- Menu Order:** 2. Callout: "Defines order on Print menu."
- Active:** Checked. Callout: "Uses data from current record only."
- Report No.:** 100033-MGR. Callout: "Unique reference number for Report."
- Report Query:** SELECT [Job_Search].Job_Status FROM [Job_Search] WHERE [Job_Search].IsActive = 1 AND [Job_Search].IsArchived = 0 AND [Job_Search].IsDeleted = 0. Callout: "The 'Report Query' field stores the main SQL query to get the data you want to report on. Use the 'Formula Query' feature to use standard SQL language to create calculated and/or formatted virtual fields that will appear in the Data Field list in the Report Designer just as 'real' fields do. The 'Appended Query' field stores GROUP BY or HAVING scripts used by complex queries) and the Results Query is used to bring back a value stored in the 'QueryResultsText' property for further calculations (using Scripts). For more examples, see Page 6, 'Additional Query Scripts - Manipulating Field Data On-The-Fly'."
- Buttons (A):** Open Designer, Clear Design, Import Design, Export Design, Preview. Callout: "A - The Buttons"
- Color Button (B):** A color selection button. Callout: "B - Color Button"

A - The Buttons

- **Open Designer** - opens the Layout Designer interface; you must have a Report Query defined before you can open the designer.
- **Clear Design** - deletes the current attached design, which is an attached .xml document stored in a binary field. There is no "undelete" so be careful!
- **Import Design** - You can use the 'Export Design' button on another report to export the design (as an .xml document), then click this button to Import it into this design.
- **Export Design** - Click this to export the current design as an .xml document; which can then be Imported (see above) to another report design.
- **Preview Report** - This displays see the results of the current design, even if you haven't saved your changes. Use this feature to see if your changes are correct BEFORE saving them.

B - Color Button

- Click this to assign a color to all Reports in the Navigator that are 'Print Current Record Only' reports. This gives you an easy way to identify one for editing later on.

The Report Designer Overview (continued)

Parent Table/Fields

The example at the right shows the Parent Table & Fields for the Contacts or Contact Notes tables. If you create a new report and choose either of those “Page Menu” tables, the fields above will populate with the values from the Company table, which is the Parent table to both Contacts and Contact Notes.

Why is the Company table a ‘parent’ to Contacts and/or Contact Notes? The Company table has a one-to-many relationship with both the Contacts and Contact Notes tables. In other words, there can be many Contacts and/or many Contact Notes associated to each Company record, making the Company record the ‘parent’ of the related Contacts/Contact Notes.

When creating reports, you can link two related tables together in Query Builder, so you can read data from both of them in a related format. For example, you may want to display a report of Company records with all associated Contacts, as the report in the Sample database called “Company Directory” does.

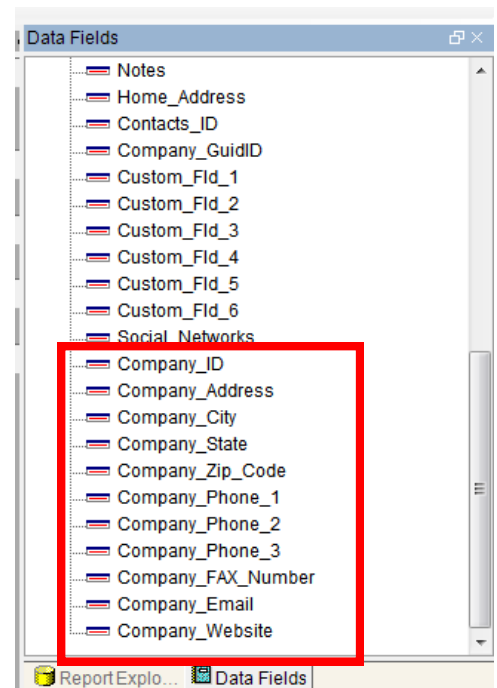
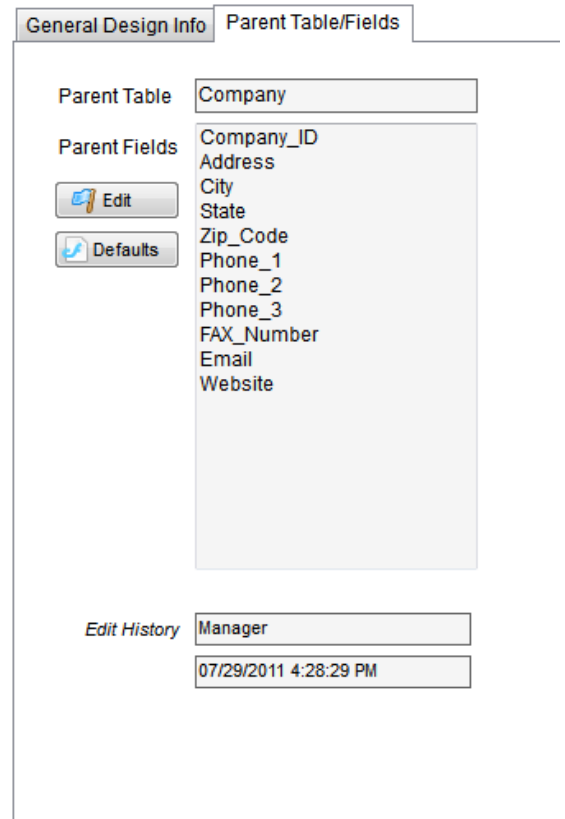
But what if you want to link Contacts and Activity records (from the Calendar) in a report that includes some of the Company information (name, etc.). Since you can only link two tables together in Query Builder, you would not be able to print the Company data with it. The fix to that is that the Contacts table “knows” that the Company table is the parent, and includes some of the Company fields in every report.

If a table has a parent table, the default parent fields are displayed in the list, and you don’t have to do anything to have them available for a report. Use the “Edit” button to add other fields from the parent table if you need them. If you’ve added or removed fields, you can click the ‘Defaults’ button to reset the parent fields to the default values.

When displayed in the Report Designer, the parent fields will be named with the Parent table name, an underscore and then the original field name. For example, the ‘Address’ field (above) will be renamed “Company_Address” to distinguish it as field from the Company table.

The picture at the right shows the default Parent fields for the Contacts table in the Data Fields list in the Report Designer:

Default Parent tables/fields are determined programmatically, and cannot be added if no parent table exists.



Additional Query Scripts - Manipulating Field Data On-The-Fly

Using Complex Queries

The **Formula Query** field can be used to manipulate data from the database to create calculated fields that can be added to reports just like standard fields. Reports in the Sample database that use this feature are:

- Activity - Appointment Schedule By Dates
- Activity - Appointment Schedule By Month
 - ◇ Both of these use this formula to pull the full Month name (January, February, etc.) and 4-digit year values from the **Start_Date** field

```
FORMAT(Start_Date,'MMMM') AS ActivityMonth, FORMAT(Start_Date,'YYYY') AS ActivityYear
```

- Company - Company Directory
- Company - Company Information
 - ◇ Both of these use the following script to concatenate the names and address fields into separate values:

```
[Company].First_Name + ' ' + [Company].Last_Name AS FullName, [Company].City + ', ' +  
[Company].State + ' ' + [Company].Zip_Code + ' - ' + [Company].Country AS CityStateZipCountry
```

In the Report Designer we use an “IF” statement to get a resulting value based on conditions:

```
IIF([Company].Company_Name <> '', [Company].Company_Name, [Company].Last_Name + ', ' + [Company].First_Name  
+ ' ' + [Company].Middle_Name) AS FullName
```

In every table is a field called “Editor”, which is the unique (GuidID) value of the logged in user. You can pull the full User Name into any report (report field will be “LastEditor”) by adding this text to the **Formula Query** script:

```
(SELECT User_Name FROM Security WHERE GuidID = <tablename>.Editor) AS LastEditor
```

Appended Query Scripts

This field stores additional scripts that will be appended to the data query AFTER the Additional Query has been inserted. For example, to summarize data in a report, use an Aggregate Function (Count, Sum, etc.) in the SELECT query then add a “GROUP BY <fieldname>” here, where <fieldname> is the “categorizing” field. An example of this is in the Toolbar / Quick Views (in the sample database). Click on “Company Directory/ Companies By State” record, then click the “Run” button on the right side of the Title to get the results below. As you can see, the results are grouped by the State field, and display only State and CompanyCtr, which is a calculated value.

The screenshot shows the Report Designer interface with three query fields:

- Formula Query** (Enter queries that create calculated values):
`COUNT(Editor) AS Status_Count`
- Appended Query** (Enter GROUP BY, ORDER BY, HAVING or any other clauses that can be appended to the final query):
`GROUP BY [Job_Search].Job_Status ORDER BY [Job_Search].Job_Status`
- Results Query** (Enter a query that returns a results in the first row, first column of the datatable):
`SELECT COUNT(Job_Status) AS RESULT1 FROM [Job_Search] WHERE [Job_Search].IsActive = 1
AND [Job_Search].IsArchived = 0 AND [Job_Search].IsDeleted = 0`

Results Query

This is something you may use rarely, but offers a way to pull an aggregate (Count, Sum, etc) value using the same WHERE statement as the main Report query that can be used in the C# Scripting feature for calculations.

How To Create These Scripts

These scripts are traditional SQL scripts, so anyone with SQL or T-SQL experience will be able to write these easily to manipulate the report query. If you're interested in learning this skill, there are many online resources that teach the basics of the SQL language. One of our favorites is: <http://www.w3schools.com/sql/default.asp>, which also has other techy tutorials you might find interesting.

As part of your license subscription to TymeTrax products, each license has a free hour of support each year, which you can use to have custom scripts designed by simply sending us an email at customersupport@tymetrax.com. Finally, check our FAQ and Downloads sites for free scripts; we'll continue posting new ideas as they come in.



Using Query Builder

Overview

Query Builder is a tool to create database queries for reading data from the database, and it's used in the Report Designer, the Quick View and Data Export forms, and also from the System Administrator's tool, Script Manager.

It provides an easy "point-and-click" way for non-techy users to select, filter and order data. If you are a skilled SQL programmer, you could write the Select statements manually, but Query Builder offers a few more options that are best implemented through its interface.

First, Some Background...

Each record in each table in TymeTrax databases has a field called "GuidID", which is the unique identifier for the record. This key is created when the record is created, and is never changed throughout the life of the record (until it's hard-deleted). Users never have access to change a key, so the data is 'safe' from user changes.

It's this "GuidID" field that determines what records are the same during Replication and how table data can be linked based on parent/child relationships.

Query Builder lets you create JOINS between two tables, using the GuidID "key" in the Primary/parent table, and a matching value in a child table in a field that is named in this format: ParentTableName_GuidID. This is consistent throughout the system; if you find a table that has a field ending in "_GuidID", it's a linked table to the table name before the "_GuidID".

For example, the Contacts and Contact_Notes tables are both linked/child tables of the Company table. Both Contacts and Contact_Notes have a field called "Company_GuidID", which stores the GuidID value of the parent (Company) record. The Company in the sample database called "Carts and More, LLC" is the parent of the Contacts record for Rachel Blake; if you looked in the database, the Company GuidID field value and the Contacts' Company_GuidID field value will be the same.

The Company table is linked to many other tables in TymeTrax software products. See the table on the next page for the workflow of how Company records are used.



Using Query Builder (Continued)

Company Table Links

Use the table below to link the Company information to other tables in TymeTrax applications. The parent field in the Company table is always "GuidID"; the linked field in the child table is always "Company_GuidID". All linked drop-down lists display the "Company_ID" field value, which is the unique identifier in the Company table, but ONLY the Company's GuidID field value is stored in the child table.

NOTE: *Not all TymeTrax applications have the Departments listed below.*

Department	Child Table Name	How It's Used
Company & Contacts	Contacts	Each contact (person) belongs to the Company record.
Company & Contacts	Contact_Notes	Each note is related to a specific Company record.
Activity Calendar	Activity	Each calendar entry is linked to a Company record; entries may be linked to the Company designated as "Our Company" or to the Company which participates in the Activity.
Expense Reporting	Expense_Report	Expenses MAY be related to a specific company, but it's not required. If a Company is designated "Our Company", it will be the default value.
Inventory	Purchase_Order	The Vendor drop-down list is populated from the Company table where the Contact Type includes "Vendor".
Invoices / Shipping	Invoice	The Customer drop-down list is populated from the Company table where the Contact Type includes "Customer" or "Lead".
Invoices / Shipping	Shipping	The Shipper drop-down list is populated from the Company table where the Contact Type includes "Shipper" or "Broker".
Sales / Shipping	Sales_Order	The Customer drop-down list is populated from the Company table where the Contact Type includes "Customer" or "Lead".
Sales / Shipping	Shipping	The Shipper drop-down list is populated from the Company table where the Contact Type includes "Shipper" or "Broker".
Payroll	Payroll	The Employee drop-down list is populated from the Company table where the Contact Type includes "Employee" or "Former Employee".

Other Department tables also have links to their own child records and can be linked in reporting to display related data. The "Invoice" table's detail records are stored in a table called "Invoice_Items". The "Sales_Order" table's detail records are stored in "Sales_Order_Items". Typically child table names start with the names of their parent table ("Payroll_Time_Card" is a child table of "Payroll"), although there are some exceptions such as the Accounts table is the parent table of the General_Journal table.

If you choose a table name from the "Choose Linked Table" list that cannot be linked to the parent table, a prompt will tell you so, and you'll need to select another.



Using Query Builder (Continued)

Building A Query

For our report, we'll use the following steps to build the query to read the data needed for it.

Choosing the Tables

Click the "Query Bldr" button on the Report to open the dialog. Select "Company" from the Primary table list, and "Contacts" from the Linked table list. The default Link style (the first option) is the correct one. Note the JOIN script next to the Custom button and how it automatically adds the link between the Company's GuidID field and the Contacts table's Company_GuidID field.

Choose Table(s) and Fields

Choose Primary Table: Company

Choose Linked Table: Contacts


Choose How to Link Tables:

- Include ALL Primary table records even if there are no matches in the linked table(s)
- Include ONLY records where there are matching values in the linked tables
- Include linked table records even if there are no matching Primary table records

LEFT JOIN

Custom: LEFT JOIN [Contacts] ON [Company].GuidID=[Contacts].Company_GuidID

Choosing the Fields

Click the  button next to the Company table to show the fields. By default, all fields are selected. You can leave this as it is (it will work just fine); however if you're building a report that will read thousands of records, you can reduce the amount of time it takes to run the report by excluding some or all fields that you don't need in the report.

Select Fields

- Our_Company
- Our_Division
- Distance
- Create_Date
- Rate_1
- Rate_2
- Rate_3
- Rate_4
- Custom_Fld_1_CBO
- Custom_Fld_2_CBO
- Custom_Fld_3_CBO
- Custom_Fld_4
- Custom_Fld_5
- Custom_Fld_6
- Custom_Fld_7
- Custom_Fld_8
- Custom_Fld_9
- Custom_Fld_10
- Product_Security

For this report to exclude some of the unneeded fields you can scroll to the bottom of this list and uncheck all fields AFTER the "Distance" field. There are other fields that are not needed, but won't hurt anything to leave them in.

Click "OK" on the toolbar above the list to save your changes. You can always add fields back into the report simply by checking them if you expand your report at a later time.

Do the same with the Contacts table; you can uncheck the custom fields at the bottom of the list if you like.

Select Fields

OK Cancel All None

- GuidID
- EditDate
- Editor
- IsActive
- IsDeleted
- IsEdited
- IsArchived
- Company_Name
- First_Name
- Middle_Name
- Last_Name
- Address
- City
- State
- Zip_Code
- Company_ID
- County
- Country
- Website
- Email
- Phone_1
- Phone_2
- Phone_3
- FAX_Number
- Classification_Code
- Needs_Attention
- Contact_Types
- BillingGuidID
- Comments
- Company_Links

Move Up Move Down

NOTE: When building reports in the Report Designer you do not need to worry about the order of the fields in the list, which are by default, in the same order as they appear in the Worksheets. However, if you are using Query Builder for reading data for Data Exports or Quick Views (on the Toolbar), you can reorder the results by using the "Move Up" or "Move Down" buttons to move the selected field in the list. This is especially helpful for the Quick Views queries.

Using Query Builder (Continued)

Choosing Which Records to Use: the WHERE Clause

The next section “Search for data WHERE” lets you define what data to read for the report. By default, on every new Report query, the line to exclude Archived data is added (`[Company].IsArchived = 0` in the example below):

In some reports, such as financial history in Invoices or Sales Orders, you may want to include archived data in order to get a complete history in the report. If so, just click on the line and then click “Remove” to remove that part of the WHERE clause.

Field Name	Operator	Value
[Company].IsArchived	=	0
[Company].Contact_Types	LIKE	'%Customer%'

The second line above tells us what type of Company records to use. To duplicate this, choose “[Company].Contact_Types” from the Field Name list. Note that all Primary table fields are listed first in alphabetical order, then the Linked table fields are next in alphabetical order.

We want all Company records that have the “Customer” Contact Type checked. Since the Contact Types field may contain any number of values, in any combination, we cannot look for records that equal (=) “Customer”, but must look for records where the Contact Types field contains “Customer”.

To that purpose, from the Operator list, choose “Contains” and then type “Customer” (no quotes) in the “Value” field. Click “Add” to add the text to the list. It’s displayed as `[Company].Contact_Types LIKE '%Customer%'`. This is the SQL syntax databases need to understand the ‘contains’ concept.

Choosing “AND” or “OR”

If all conditions in the query must be met, use the “AND” concatenator between conditions. If one condition must be met OR another condition must be met, use the “OR”. The Order of the conditions does NOT matter if all of them must be true to return the data. However, if you’re using “OR” statements, then it’s important to group all of them in a logical order.

Field Name	Direction
[Company].Company_ID	ASC
[Contacts].Last_Name	ASC
[Contacts].First_Name	ASC
[Contacts].Middle_Name	ASC

Determining the Sort Order

The last thing to add to the Query is how to sort or Order the data. If you are going to use Group Headers/Footers in your report to categorize the Detail sections, you must have the data sorted using the same values in the query.

Choose the field from the “Field Name” list and then choose whether to order the data in Ascending (ASC) order or Descending (DESC) order by choosing the Direction from the second drop-down list. Click “Add” to add your choices to the list.

You can add as many levels of sorting to the list as you need so that the data displays in the expected order. In the example above for our Tutorial Report, the data will be sorted first by Company’s Company_ID field, then by the Contact’s Last_Name, First_Name and Middle_Name values.

You can re-order your selections by clicking on a line, then clicking the up or down arrow buttons on the right to move it. Click “OK” to validate your selections; Query Builder closes and the query is displayed in the DataQuery field on the Reports page.

Using the Layout Designer

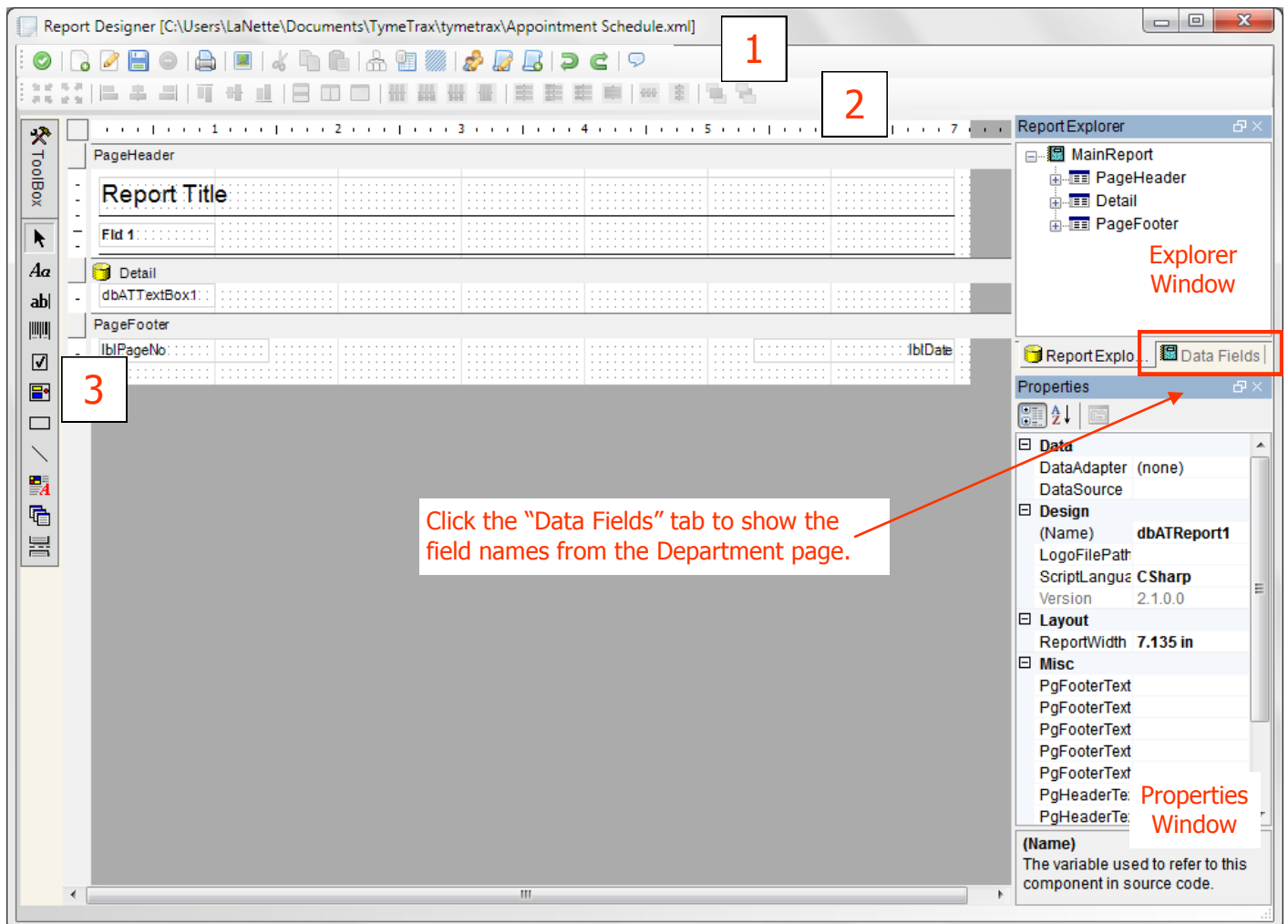
Overview

The Layout Designer is where you decide where on each printed page each field value will print. You can copy an existing template and modify it, or you can start from scratch.

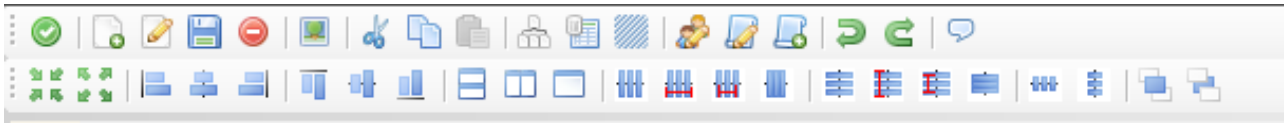
Click the "Open Designer" button on the Report Designs form to open the Layout page. **PLEASE NOTE:** You must have a query defined in the Report Query field (using the Query Builder) before you can open the layout window, as the DataQuery determines which fields will be available to arrange. Use the Query Builder to point and click to create a query that includes all of the fields you want on each report.

The Toolbars

There are three toolbars on the Designer; the Main toolbar (1) which manages properties common to the entire report, such as Page Layout (size, margins, etc.) and Custom Scripts. The Layout toolbar (2) is used to format and arrange the labels, fields and other objects you place on the layout page (PageHeader, Detail, Footer); and the Controls Toolbar (3) on the left is used to choose which types of controls (Labels, TextBoxes (Data fields), Lines, etc.) are placed on the layout page.



The Layout Designer Toolbars



1
2

The Main Toolbar (1) - Buttons in this order:

Close - Closes the Designer; you'll be prompted to save your changes if you've started a template or edited one.

New - Clears the layout page and creates a new template.

Open - Displays the Browse dialog to let you open an existing Report Template which you may have exported using the "Export Design button. NOTE: All templates are saved with an .xml extension when exported.

Save - Functions as a "Save As..." letting you decide whether you want to overwrite an existing template or change the name to save it as a new one.

Delete - Deletes all selected controls on the page.

PgSetup - Opens the Page Setup dialog for changing Paper and Printer settings. **NOTE:** Custom settings are saved with each report template.

Cut, Copy, Paste - Standard Windows actions.

Explorer - Shows the Explorer window if closed (see diagram on Page 14)

Properties - Shows the Properties window if closed (see diagram on Page 14)

Grid - Toggles the grid-lines on/off on the layout surface (see diagram on page 5)

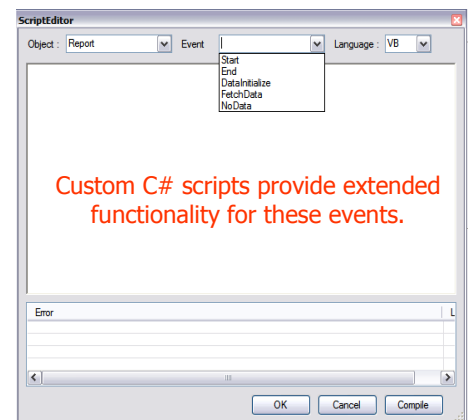
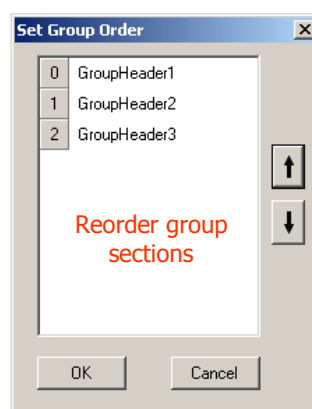
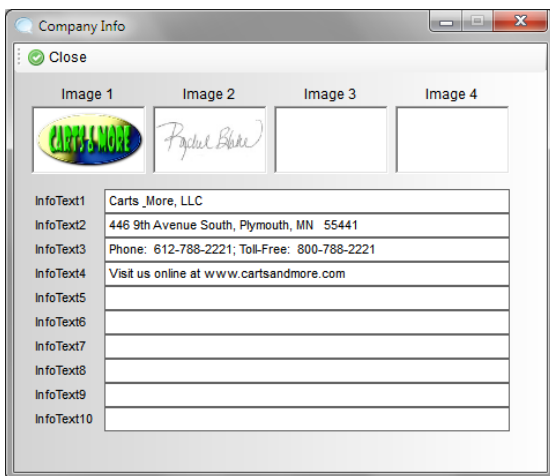
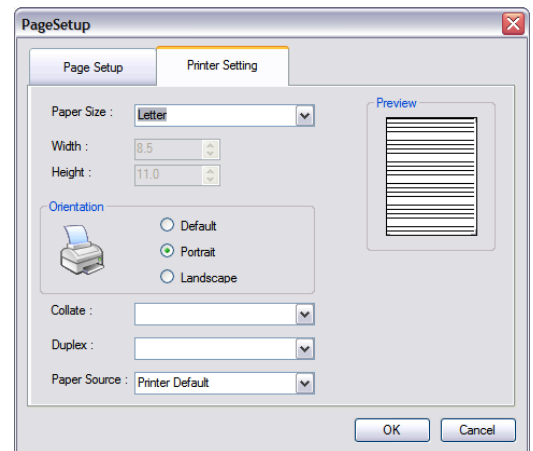
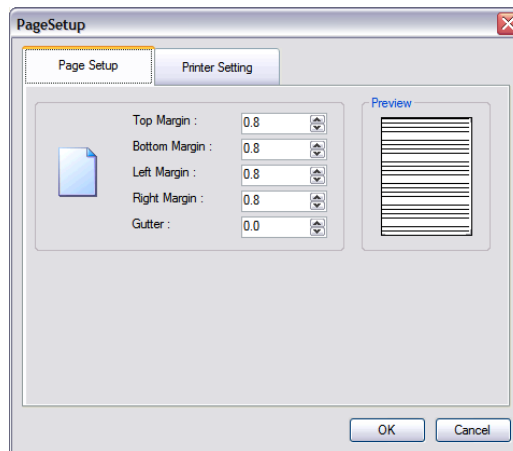
Reorder - Opens the Group Reorder form for reordering the group sections

Script - Opens the scripting window for custom scripts (see Page 19)

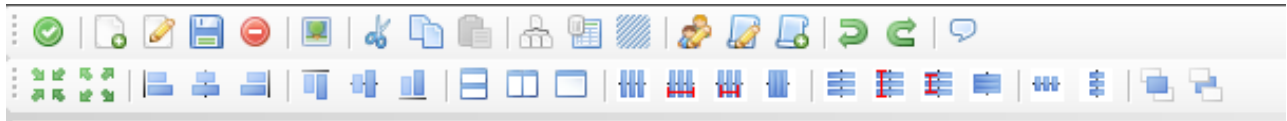
Add Default Scripts - Automatically adds the script needed to auto-number pages, add the date to the bottom of footer area and pull the Company Images & Text info into the Page Header section.

Undo & Redo - Restores the layout before the last change, or reapplies the last change.

Show Company Info - Displays the Company Images & Text info defined on the Custom Information tab.



The Layout Designer Toolbars (continued)



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

1

2

The Layout Toolbar (2)

1. Undo last action
2. Redo last action
3. Align left edges of selected controls
4. Align centers of selected controls
5. Align right edges of selected controls
6. Align top edges of selected controls
7. Align middles of selected controls
8. Align bottom edges of selected controls
9. Make selected controls equal width
10. Make selected controls equal height
11. Make selected controls equal size (height & width)
12. Space equally horizontally
13. Increase horizontal spacing
14. Decrease horizontal spacing
15. Remove horizontal spacing
16. Space equally vertically
17. Increase vertical spacing
18. Decrease vertical spacing
19. Remove vertical spacing
20. Center horizontally
21. Center vertically
22. Bring selected control(s) to front
23. Send selected control(s) to back

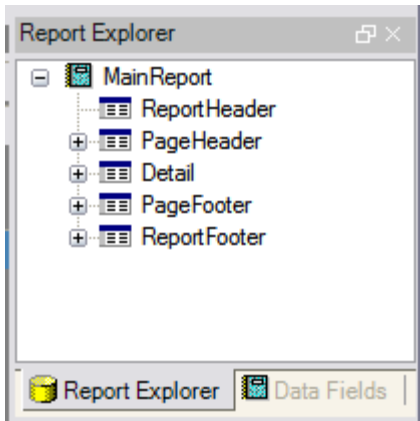


3

The Controls Toolbar (3)

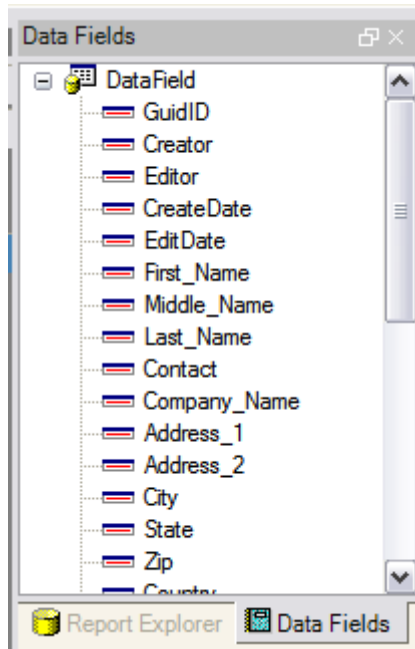
1. **Pointer** - Used to select controls or sections of the report
2. **Label** - Used to add a new Label control.
3. **Textbox** - Used to add a new Textbox, bound to a database field or unbound
4. **Bar Code** - Used to add a barcode control.
5. **Checkbox** - Used to add a new CheckBox, bound to a database field or unbound
6. **Picture** - Used to add a new PictureBox, bound to a database field or unbound loaded from a file.
7. **Shape** - Used to add new Shape control. (Rectangle, circle or square).
8. **Line** - Used to add new Line control.
9. **RichText** - Used to add new RichText control.
10. **Subreport** - Used to add new SubReport control.
11. **PageBreak** - Used to add new PageBreak control.

Other Layout Designer Options



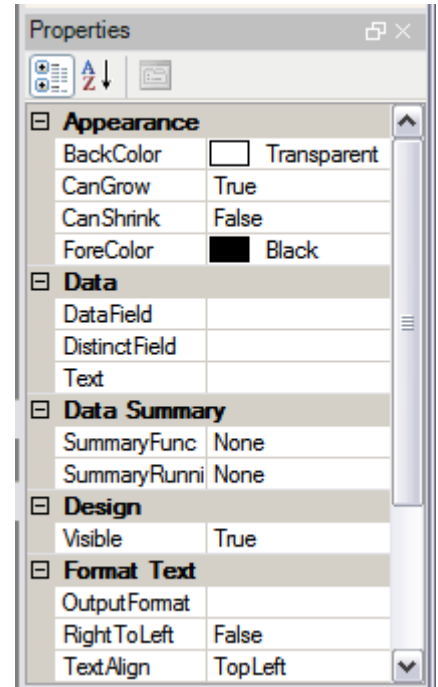
Report Explorer

Report explorer gives a tree view representation of all the current report components. This control is used to view and select the current report components.



Data Fields

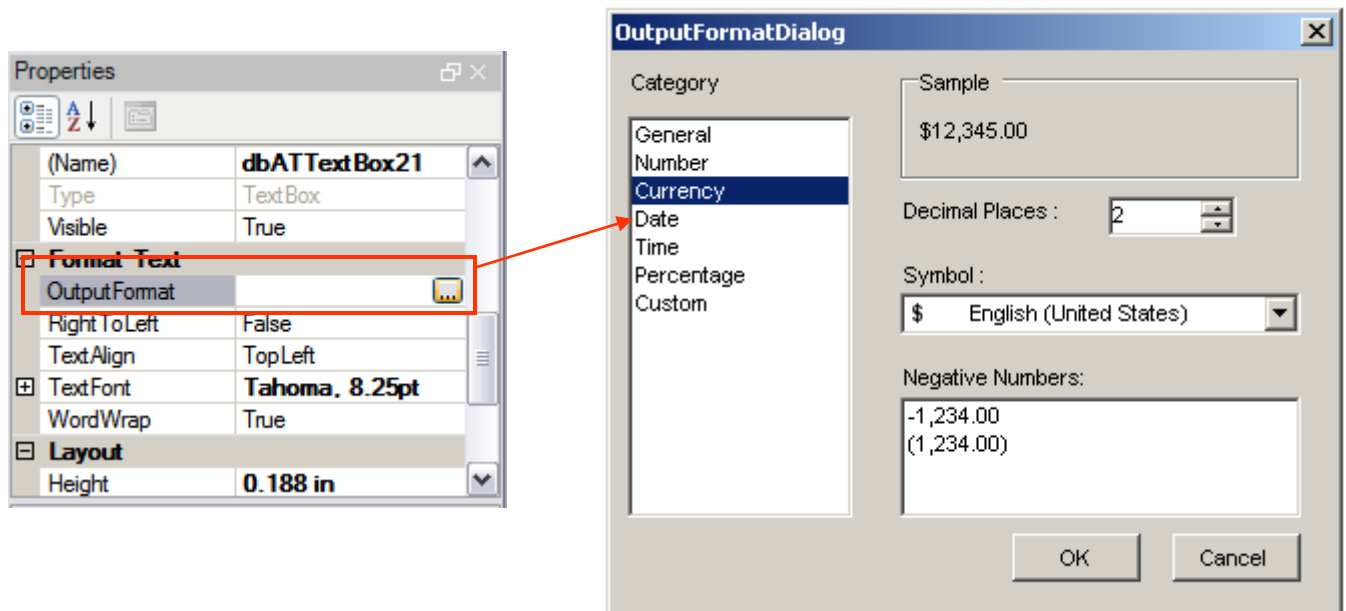
The Data Field gives a tree view representation of all the fields from the current report data source. Dragging and dropping the data field on to the designer surface will add a new data bound TextBox control to the corresponding section.



The Property Grid

The property grid is used for viewing and changing editable properties for a control. The properties change depending on what type of control is selected.

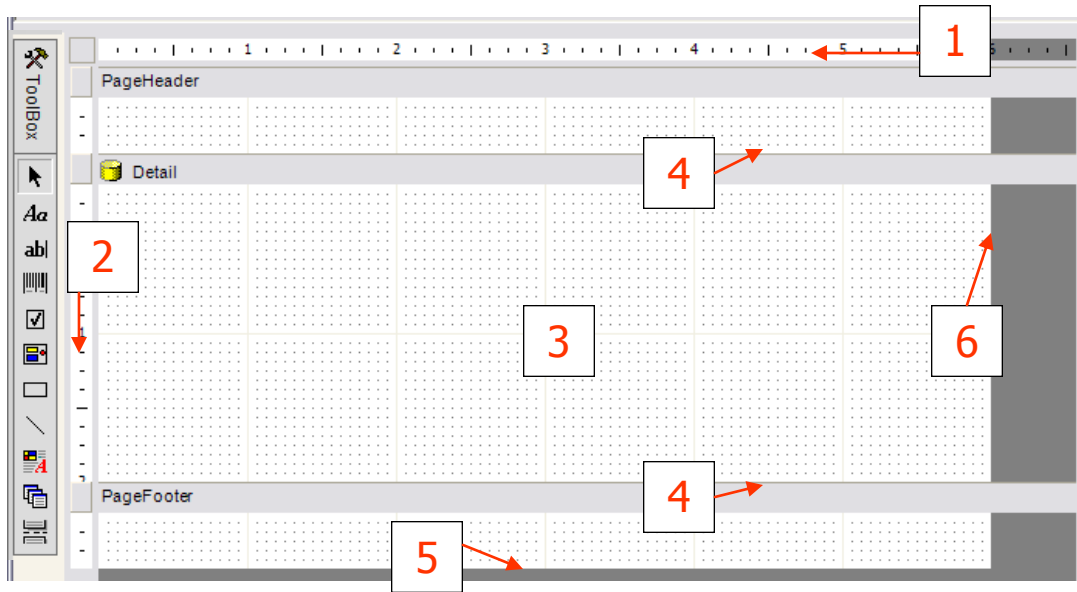
If a textbox that will display data is selected you can set the format by clicking on the [...] button on the "Output Format" property. This opens a dialog that provides options for formatting data as numbers, currency, dates, time, percentage and custom formats.



Layout Designer Grid

The Report Designer surface is made up of the following base components.

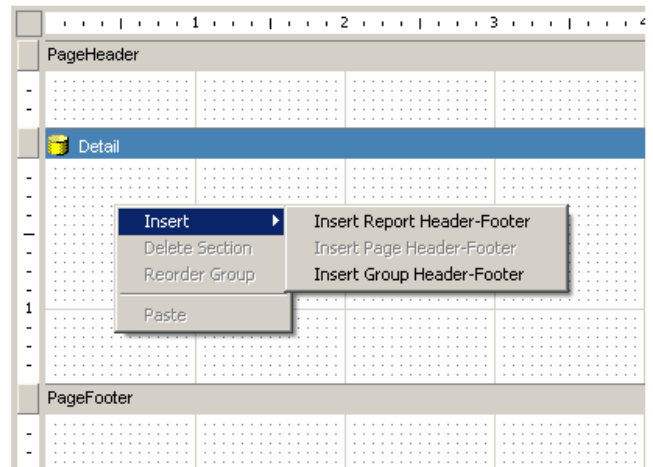
- ◇ **PageHeader** section - This section can be used to print column headers, page numbers, page titles, or any information that needs to be printed once at the top of each page.
- ◇ **Detail** section - This section is the body of the report that prints once for each record in the data source.
- ◇ **PageFooter** section - This section can be used to print page totals, page numbers or any other information that needs to be printed once at the bottom of each page.



1. Horizontal ruler in inches for measurements on X-Axis.
2. Vertical ruler in inches for measurements on Y-Axis.
3. Section surface on which report controls are designed.
4. Section seizer (Height) component used to change the corresponding section height. Placing the mouse over this component, click and drag to change the section height.
5. Section seizer (Height) used to change the corresponding section height. Placing the mouse over this component, click and drag to change the section height.
6. Report seizer (Report Width) used to change the Report width. Placing the mouse over this component, click and drag to change the Report width.

The image above only shows the PageHeader, Detail and PageFooter sections. However, you can also add Report Header-Footer sections, and multiple Group Header-Footer sections to group or categorize data. Just right-click in the section BELOW where you want to put a Group section and choose "Insert Group Header-Footer" to insert it.

Group Header-Footer sections will create a new group each time the concatenated value of the fields in that section changes. For example, if you're printing Contact Notes and you want to Categorize by the Company_ID field, put the Company_ID field in the Group Header and all related Contact Notes fields in the Detail section. NOTE: you should always put the Group Header fields in the "Order By" section in Query Builder so the DataQuery reads the data in the same order to match the layout design.

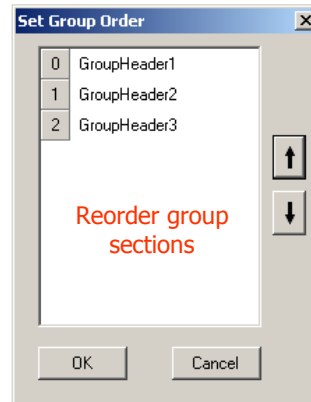
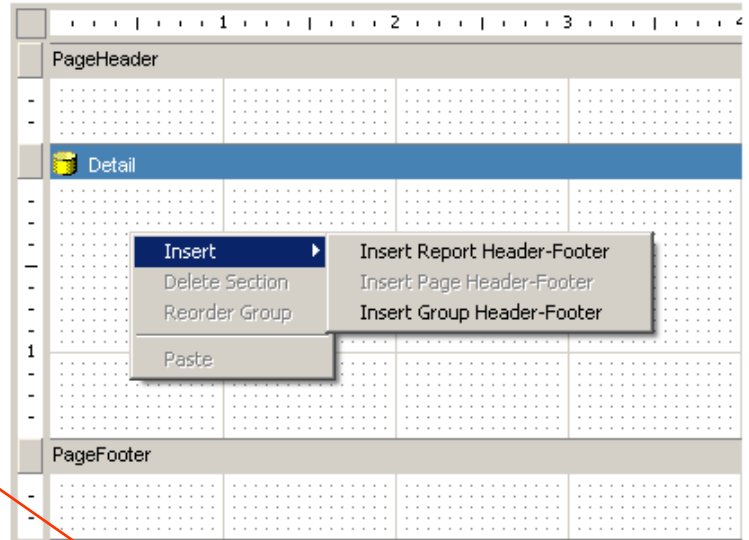


Context Menus

Context Menus for the Report Designer

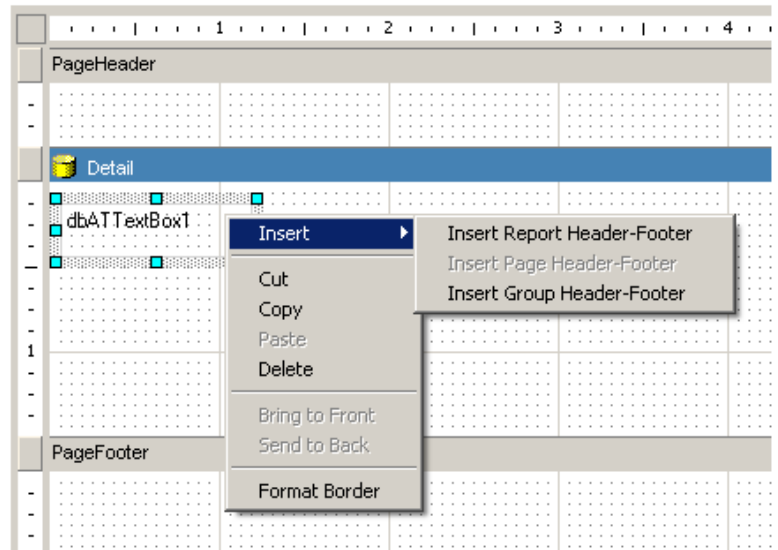
There are two context menus for designer surface. Right-click on the Detail section to activate the Section Context Menu:

1. **Insert** : Activates the insert submenu.
2. **Insert Report Header-Footer** : Inserts Report Header and Report Footer to the Designer.
3. **Insert Page Header-Footer** : Inserts Page Header and Page Footer to the Designer.
4. **Insert Group Header-Footer** : Inserts Group Header and Group Footer to the Designer.
5. **Delete Section** : Deletes the Header and Footer of selected section. (You can not delete Detail section.)
6. **Reorder Group** : This will open up the Group Reorder form for reordering the group sections.
7. **Paste** : This will paste one of the Controls current copied in the clipboard to the selected section.



Right click on a Control to activate the Control Context Menu:

1. **Insert** : Same function as section context menu.
2. **Cut** : This command first copies the selected Control objects to clipboard and then deletes the selected control.
3. **Copy** : This command copies the Control objects to the clipboard.
4. **Paste** : This command pastes the Control objects from the clipboard to the current section.
5. **Delete** : This command deletes the selected Control objects.
6. **Bring To Front** : Bring the primary selected Control object to front of the selected controls.
7. **Send To Back** : Send the primary selected Control to back of the selected controls.
8. **Format Border** : This command opens up Format Border Form for formatting Control object border.



Adding Company Logo & Other Information to Reports

Many times you will want to include your Company logo, address, phones, etc., on reports or forms that you send out. Examples of this would be Invoices, Payroll stubs, Sales Quotes & Orders, or Customer Service/Warranty documents (depending on what version of TymeTrax you have).





You could certainly add an image field to each of your reports, and labels to enter your Company name, address, etc., into on each report, but if at some point that information changes, you would need to update each report design to edit everything.

The “Custom Information” tab in the Report Designer makes it much easier to add and maintain your Company information so you can store that info in one place and have each report simply use it when it prints.

Please Note: This feature uses custom scripts to place this company information on each report. See the help topic “Creating Custom Scripts” on Page 20 for details on using this feature.

Report & LetterWriter Images **Custom Text**

Click Browse to choose an image of file type .bmp, .jpg, or .gif. Right-click on an image to copy it to Windows clipboard.

Image 1 <i>Suggestion: Add your Company Logo here to display on Report Headers</i>	<input type="button" value="Browse..."/> <input type="button" value="Clear"/> <input type="button" value="Delete"/>		Add 'picImage1' PictureBox on Report
Image 2 <i>Suggestion: Add a scanned image of your Signature for signing letters, etc.</i>	<input type="button" value="Browse..."/> <input type="button" value="Clear"/> <input type="button" value="Delete"/>		Add 'picImage2' PictureBox on Report
Image 3 <i>Any additional image for a Report or LetterWriter document</i>	<input type="button" value="Browse..."/> <input type="button" value="Clear"/> <input type="button" value="Delete"/>		Add 'picImage3' PictureBox on Report
Image 4 <i>Any additional image for a Report or LetterWriter document</i>	<input type="button" value="Browse..."/> <input type="button" value="Clear"/> <input type="button" value="Delete"/>		Add 'picImage4' PictureBox on Report

Logos, Signatures and Other Images

Adding company images does not have to be in any particular order, however, you might find it easier to remember what's what when designing reports if you use the first field, “Image 1” for your most common company logo and the second field for a Signature image (if you need a signature). Otherwise you can add images in any order you wish.

To add your logo, click the ‘Browse’ button and choose the image file, which must be of type Bitmap (.bmp), JPEG (.jpg) or GIF (.gif). As with all images, it's best to resize the image BEFORE you add it to reduce the amount of space it will take up both in the database and in each report. NOTE: The image field on reports will automatically “Zoom” it to fit the space, but if you use the “[Image1]” tag to add the logo when building LetterWriter documents (see the CRM/LetterWriter help topic), it will be displayed at its actual size. So if you attach it here at the size you would want it to actually be printed, you'll get much better results when using it in LetterWriter documents.

Please note the text in green at the right side of each picture box, which tells you what the name of the Picture control on each report (and in each script) needs to be in order to display the image.

Adding Company Logo & Other Information to Reports (Continued)

Custom Company Text

You can store up to 10 different text phrases for reports on the “Custom Text” tab. You can put anything you want in each line, and you do not need to use all of them. However, you will need to add a label in your report design named the same as the “Label on Report” notation next to each line of text, and make sure your script is named the same as well (the default scripts ARE named the same as displayed below, but if you edit the scripts, be careful not to change the names).

It’s easier to remember what data is in which line if you start with the most important text, such as your Company Name, address-city-zip, phones, etc., as displayed below.

	Label on Report:
Enter your Company name	lblInfoText1
Enter address, city, state, zip	lblInfoText2
Enter phone/fax numbers	lblInfoText3
Enter website / email info	lblInfoText4
Enter any additional info	lblInfoText5
Enter any additional info	lblInfoText6
Enter any additional info	lblInfoText7
Enter any additional info	lblInfoText8
Enter any additional info	lblInfoText9
Enter any additional info	lblInfoText10

NOTE: The LetterWriter tags are the same as the Report Label names, without the 'lbl' prefix (InfoText1, InfoText2...)

All users can see what values are stored in these two tabs by clicking the “View Company Info” button, which is available on the LetterWriter page and also on the Layout Designer toolbar.

Users can right-click on any image or textbox to copy the value to the Windows clipboard.

When viewing this info from the Layout Designer, the dialog must be closed before you can return to the Layout page. When viewing this from any other button in the system, the dialog can be minimized and reopened from the Windows taskbar at any time.

Image 1	Image 2	Image 3	Image 4

InfoText1	Carts _More, LLC
InfoText2	446 9th Avenue South, Plymouth, MN 55441
InfoText3	Phone: 612-788-2221; Toll-Free: 800-788-2221
InfoText4	Visit us online at www.cartsandmore.com
InfoText5	
InfoText6	
InfoText7	
InfoText8	
InfoText9	
InfoText10	

Creating Custom Scripts

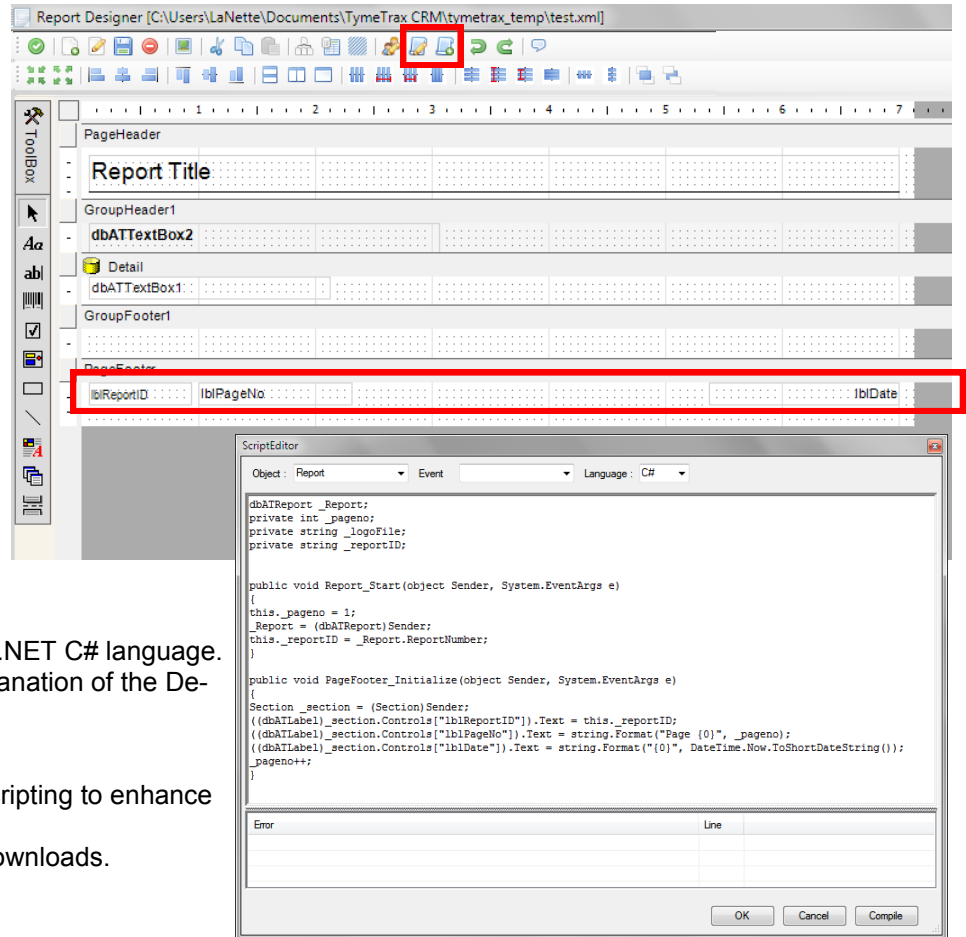
The combined features of a custom DataQuery and the Layout Designer are typically more than enough to generate the reports you want. However, the Layout Designer also supports custom scripting to really tweak your reports to a very professional output.

The default reports that are copied from the Sample Database use a script to populate the Page Number on each page, to add the Report ID field value (which helps to identify which report was printed), and to put the current date on each page (right).

The “Edit Script” button displays the current scripts, if any, and the ‘Add Default Scripts’ button on the right side of it adds the basic scripts needed to automatically put the current date and page numbers on the page, and any Company logos (images) and business information you’ve defined in the Company Information section.

All scripts are written in the Microsoft.NET C# language. See the next page for a detailed explanation of the Default Scripts and how to edit them.

There are many other ways to use scripting to enhance report designs; check our website at www.TymeTrax.com for free script downloads.



Creating Custom Scripts (Continued)

```
dbATReport _Report;
private int _pageno;
private string _Image1FilePath;
private string _Image2FilePath;
private string _Image3FilePath;
private string _Image4FilePath;
private string _reportID;
private string _InfoText1;
private string _InfoText2;
private string _InfoText3;
private string _InfoText4;
private string _InfoText5;
private string _InfoText6;
private string _InfoText7;
private string _InfoText8;
private string _InfoText9;
private string _InfoText10;
private string _QueryPromptText;
private string _QueryResultsText;
```

```
public void Report_Start(object Sender, System.EventArgs e)
{
```

```
    _Report = (dbATReport) Sender;
    this._pageno = 1;
    this._Image1FilePath = _Report.Image1FilePath;
    this._Image2FilePath = _Report.Image2FilePath;
    this._Image3FilePath = _Report.Image3FilePath;
    this._Image4FilePath = _Report.Image4FilePath;
    this._reportID = _Report.ReportNumber;
    this._InfoText1 = _Report.InfoText1;
    this._InfoText2 = _Report.InfoText2;
    this._InfoText3 = _Report.InfoText3;
    this._InfoText4 = _Report.InfoText4;
    this._InfoText5 = _Report.InfoText5;
    this._InfoText5 = _Report.InfoText6;
    this._InfoText6 = _Report.InfoText7;
    this._InfoText7 = _Report.InfoText8;
    this._InfoText8 = _Report.InfoText9;
    this._InfoText10 = _Report.InfoText10;
    this._QueryPromptText = _Report.QueryPromptText;
    this._QueryResultsText = _Report.QueryResultsText;
}
```

```
public void PageHeader_Initialize(object Sender, System.EventArgs e)
{
```

```
    Section _section = (Section) Sender;
    if (System.IO.File.Exists(_Image1FilePath))
    {
        ((dbATPicture)_section.Controls["picImage1"]).Image = System.Drawing.Image.FromFile(_Image1FilePath);
    }
    ((dbATLabel)_section.Controls["lblInfoText1"]).Text = this._InfoText1;
    ((dbATLabel)_section.Controls["lblInfoText2"]).Text = this._InfoText2;
    ((dbATLabel)_section.Controls["lblInfoText3"]).Text = this._InfoText3;
    ((dbATLabel)_section.Controls["lblInfoText4"]).Text = this._InfoText4;
    ((dbATLabel)_section.Controls["lblInfoText5"]).Text = this._InfoText5;
}
```

```
public void PageFooter_Initialize(object Sender, System.EventArgs e)
{
```

```
    Section _section = (Section) Sender;
    ((dbATLabel)_section.Controls["lblReportID"]).Text = this._reportID;
    ((dbATLabel)_section.Controls["lblDate"]).Text = string.Format("{0}", DateTime.Now.ToShortDateString());
    ((dbATLabel)_section.Controls["lblInfoText6"]).Text = this._InfoText6;
    ((dbATLabel)_section.Controls["lblInfoText7"]).Text = this._InfoText7;
    ((dbATLabel)_section.Controls["lblInfoText8"]).Text = this._InfoText8;
    ((dbATLabel)_section.Controls["lblInfoText9"]).Text = this._InfoText9;
    ((dbATLabel)_section.Controls["lblInfoText10"]).Text = this._InfoText10;
    ((dbATLabel)_section.Controls["lblPageNo"]).Text = string.Format("Page {0}", _pageno);
    _pageno++;
}
```

When you click the 'Add Default Scripts' button on the Layout Designer toolbar, the following scripts are copied to the report design. Click the "Edit Scripts" button to edit them.

You can remove any of the sections you don't need by deleting the entire line of script by color.

Legend:

Bold Black	Basic Function structure; required if any code within the curly braces { } is used.
Blue	Page Numbering
Red	Images (Image1) - use similar code for any image in any section
Purple	Report_ID field
Orange	Company Info (header)
BlueGreen	Company Info (footer)
Brown	Today's Date
Green	Query Prompt Parameters*

*The **QueryPromptText** property is the text explanation of what the user selected when the DataQuery prompts the user for filter values. For example, if the DataQuery asks the user for date range on a field called "Transaction_Date", this text will say: **Transaction Date >= 01/01/2011 AND Transaction Date <= 03/31/2011**. This text can be added to a Page or Report Header (add a Label and use the same script as the PageHeader example below to put the text into the label) so users know what range of data is being displayed in the report. The **QueryResultsText** property on the report object is the results of a query defined in the Report Design.Results_Query field. This value can be displayed on the report, or used to calculate other values for C# scripts.

Tutorial - Creating A Report Design

Creating A New Report

Every sample database has default report designs that are automatically copied to every new database you create. These reports provide a 'starting point' for each table in the system and you may find it's much easier simply to copy a report that's "close" using the 'Copy' button on the Toolbar, and then editing it to make the necessary changes.

Starting with a usable report and copying it is a recommended best practice once you have at least one report that's formatted the way you want. However this tutorial will use the 'start from scratch' method to show you every aspect of using the Report Designer.

Before You Start

You can go through this tutorial either using the SampleDatabase.mdb, which is included when installing TymeTrax, or if you have already created a Company database, you will need to have entered the following information in it:

- 1) Company logo and information Lines on the Custom Information/Custom Text tab in the Report Designer.
- 2) At least one Company record and one Contacts record.

This tutorial will walk through each step in logical order, and it's best to simply complete each step along in the order it's being explained. If you need to stop, be sure to save your work by clicking "Save" on the Report Designer toolbar. **Please note that Layout Designer changes are NOT saved to the database until you save the entire record by clicking 'Save' on the Report Designer (NOT the Layout Designer) toolbar.**

This tutorial has numbered steps, and within each step, tasks noted by a, b, c... Within each step is an explanation of what you're doing and why to help you make sense of how everything works.

What Should The Report Show?

Before you begin a new report, make sure you've defined what information the report should deliver. Make a few notes, or print out an existing report and mark it up with needed changes.

For this exercise, we'll build a form (report) to send to all Company entries where the Contact Type is a "Customer". Our policy is to have each customer confirm their contact information each year, and also ask for any feedback they may be willing to give us.

We'll be showing their current Contact information and then providing a place for corrections. We'll also add a place for customers to write in any other feedback that they want to tell us. We'll also need to break the page after each customer record so we can send individual letters.



Tutorial - Creating A Report Design

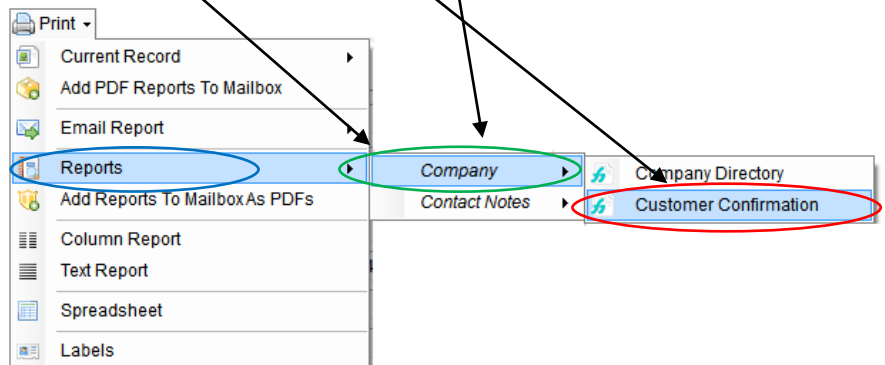
1. Create A New Report Design

In the Report Designer, click 'New' on the form toolbar (at the top of the page) to create a new Report Design record.

a. Enter these values:

Report Title	Customer Confirmation
Description	We send this to all Customers to confirm their contact information.
Page Menu*	Company
Menu Order	3

The **Report Title** will be displayed in the **Company** Print menu in the **2nd** position under **Reports** (because this report will print multiple Company records, which is determined by Unchecking the "Print Current Record Only" option). The "Print Current Record Only" reports are displayed under "Current Record" on the print menu and prints only the data associated to the currently selected record.



The **Menu Group** value is not required, but lets you "group" similar types of reports together on the Print menu simply by assigning each report the same group value.

Checking the "Check here to begin a new section on the Print Menu" option would put a separator line above "Customer Confirmation" on the print menu, but since there are only a few reports, we don't need to separate them to make any of them easier to find. (But this option is available as you build more reports and it helps to groups them logically to make them easier to find.)

The **Report No.** is generated automatically and is always a unique value. The last three characters are the initials of the logged-in user; in this case we do NOT have security enabled, so the default "Manager" initials are used (MGR). This value is available to script onto any section in a report from the Report's "ReportNumber" property (see page 20 for this reference: `this._reportID = _Report.ReportNumber;)` ... and can be used to easily identify which report you're using especially when changes need to be made.

*Adding Reports To Other Departments

The Page Menu field determines in which print menu each report will be displayed. You can add a report from one table/department to another by copying the Report (click the "Copy" button on the menu, choose "1", and click "OK"), then change the Page Menu field value to another table / page. For example, if a user should NOT have Security access to open the Product Catalog, but still needs to print a Catalog or Product Sheet report, copy those reports, change the Page Menu from Product Catalog to Sales (or Invoice) and save it. The Product Catalogs will appear on the Sales/Invoice print menu, giving the user the data they need without seeing the full Product Catalog.

Tutorial - Creating A Report Design (Continued)

2. Creating the Report Query

The next step is to tell the report what data to read when displaying the results. The Query Builder dialog lets you simply point and click to create a database query. See Page 7 for more information about the Query Builder.

- Click the “Report Query” button to open the **Query Builder** dialog.
- Choose the Company table as the Primary (parent) table from the first list.
- Choose the Contacts table as the Linked (or child) table.

Choose Table(s) and Fields

Choose Primary Table: Company

Choose Linked Table: Contacts

Choose How to Link Tables:

- Include ALL Primary table records even if there are no matches in the linked table(s)
- Include ONLY records where there are matching values in the linked tables
- Include linked table records even if there are no matching Primary table records

LEFT JOIN

Custom: LEFT JOIN [Contacts] ON [Company].GuidID=[Contacts].Company_GuidID

Note that Query Builder chooses a default for “How To Link Tables”, which is USUALLY the first option. The JOIN text is also created automatically when you click on a Linking option.

The Custom Join Button opens a dialog where you can choose other key fields (if they exist) to link two tables together with. Please see the System Admin guide for all possible linking combinations among tables.

More Info: Every data table in the system has a unique key field, called “GuidID”, and child tables are linked to the parent/primary tables by a field named using the Primary table’s name plus “_GuidID”. For example: The linking field for both Contacts (and Contact_Notes) is “Company_GuidID”. The JOIN text uses this information to link every related Contact to its correct Company record.

3. Defining What Data Is Used

The “Search for data WHERE” section lets you define which records will be included in the report. If you didn’t add anything here, the report would display ALL of the records in the database.

When you selected “Company” as the Primary table, the first line in the WHERE list was entered automatically:

“**[Company].IsArchived = 0**”. This entry excludes all Archived records (see the user guide on Database Utilities/Archiving for details). Sometimes you may want to include Archived data in a report such as when reporting on sales trends, etc., in which case you would click on it, then click “Remove” to delete it. But since we are sending this report out to only current customers we’ll leave the line in.

Search for data WHERE

Field Name	Operator	Value
[Company].IsArchived	=	0
AND [Company].Contact_Types	LIKE	"%Customer%"

Buttons: Add, Remove, Clear All

The second line tells the report to only print data for Company records where the “Customer” option is checked in Contact Types.

- Select the Field Name: **Contact_Types**; choose the Operator: **Contains**; and enter **Customer** in the value field. Click “Add” to add it to the list, which should then appear as above.

More Info: We use the CONTAINS option because the Contact_Types field can have multiple values (Customer, Lead, Personal, etc.) and we want to include all records that “contain” the Customer value. The “LIKE” text is how a database interprets “contains” so its automatically changed for you in the query.

The order of WHERE clauses in this list does not matter; the query will consider all pieces equally when all conditions must be met, which is determined by using the “AND” concatenator between conditions. See Page 10 for help on looking for data based on “OR” conditions, which needs to be ordered correctly to be understood correctly.



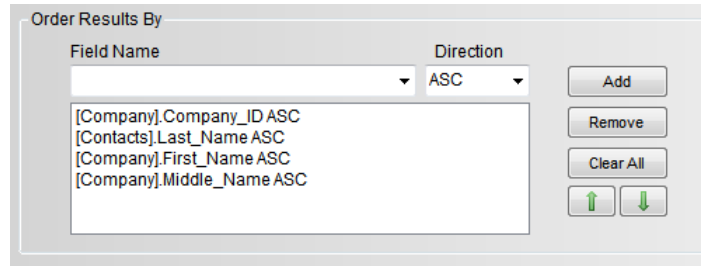
Tutorial - Creating A Report Design (Continued)

4. Setting the Data Order

Our plan is to print each Company record with any associated Contacts in alphabetical order. Since we'll also be printing labels for mailing envelopes, it's easier to stuff them if both are printed in the same order.

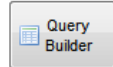
The "Order Results By" section tells the query how to order the data as its read from the database.

a. Choose each of these items from the Field Name dropdown list in order, keep the default sort direction (ASC = Ascending) and click "Add" for each field name below:



[Company].Company_ID
[Contacts].Last_Name
[Contacts].First_Name
[Contacts].Middle_Name

Report Query



```
SELECT [Company].GuidID, [Company].EditDate, [Company].Editor,  
[Company].IsActive, [Company].IsDeleted, [Company].IsEdited,  
[Company].IsArchived, [Company].Company_Name, [Company].First_Name,  
[Company].Middle_Name, [Company].Last_Name, [Company].Address,  
[Company].City, [Company].State, [Company].Zip_Code, [Company].Company_ID,
```

Your results should look like the image above. Our Report Query is complete!

SPECIAL NOTE: To order reports by the Company_ID field in tables that do not have the Company_ID field, use the Company_GuidID field in the ORDER BY section. TymeTrax will recognize it and actually use the Company_ID value to sort the data.

b. Click "OK" on the toolbar and the query is displayed in the field.

PLEASE NOTE: While the Report Query is editable in the field (double-click to open a larger edit window), it is not recommended that you edit it manually unless you are very skilled in editing SQL queries. Even so, editing the query manually may make it unreadable by Query Builder and also by the Report code when it tries to run the report.

5. Adding an Additional Query

The Additional Query field can be used to manipulate data from the database. See the info on Page 6, "Additional Query Scripts" for examples. For this report, we need to get a value using this logic: If the Company_Name is NOT empty, use Company_Name, else use Last_Name, First_Name, Middle_Name.

Database queries support "IF" statements to give us that information.

a. Enter this text (or copy and paste it) into the Additional Query field without any hard carriage returns:

```
IIF([Company].Company_Name <> '', [Company].Company_Name, [Company].Last_Name + ', ' +  
[Company].First_Name + ' ' + [Company].Middle_Name) AS FullName, IIF([Contacts].Last_Name <>  
'', [Contacts].Last_Name + ', ' + [Contacts].First_Name + ' ' + [Contacts].Middle_Name,  
[Contacts].First_Name + ' ' + [Contacts].Middle_Name) AS ContactName
```

More Info: These scripts must be exact; note we're using a "plus (+)" sign for concatenating text values, and single quotes to denote hard-coded text, like the comma and spaces. Also the table name, Company, MUST be enclosed in the brackets, followed by a dot (.), and then the field name. This is standard SQL query language format and any changes will result in an error message.

This script will decide which field values to use for each record as the data is read and store the results in a field called "FullName" and another called "ContactName". When we open the Layout Designer, at the end of the list of Data Fields you'll see these new fields that are only created when the query runs.

b. Click **Save** on the Toolbar to save your report so far. It's best to save at this point BEFORE starting the Layout Designer and to save often as you make changes to it.



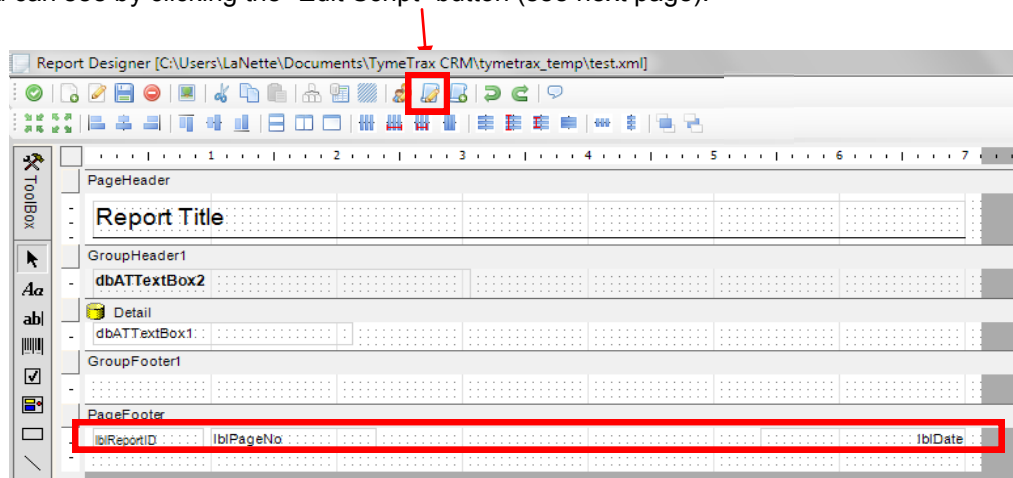
Tutorial - Creating A Report Design (Continued)

6. Creating the Report Layout

Click the “Open Designer” button to open the Layout Designer.

If you start a report from scratch as we’ve done in this tutorial, a system-based default report will be displayed in the Layout Designer, which includes a PageHeader with a label (“Report Title”) on it, one Group Header with a field on it, and one field in the Detail section. The GroupFooter is empty, but the PageFooter has three labels on it that are populated by scripts, which you can see by clicking the “Edit Script” button (see next page).

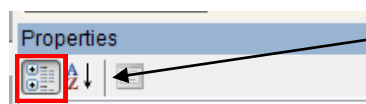
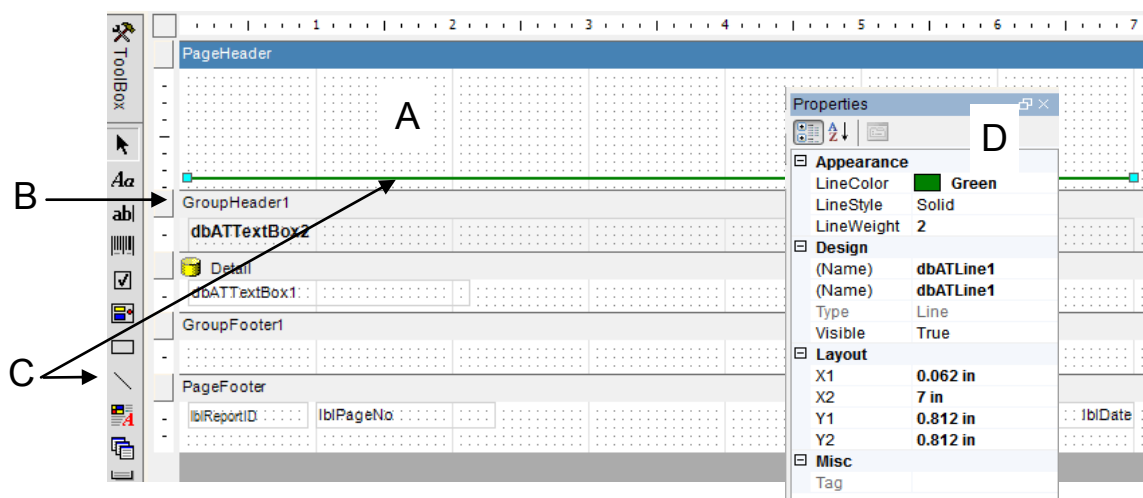
These labels could be removed simply by right-clicking on each and choosing “Delete”, however since we would like to use this information on every page, we’ll leave them. We will, however change the scripts so we can also add our logo and some company information to this report.



7. Defining the Page Header

The PageHeader contains information that’s displayed on the top of every page in the report.

- Since we want to put our Company logo and other information here, we’ll start by removing the label with “Report Title” in it by right-clicking on it and choosing “Delete”.
- Then hold your mouse on the top of the GroupHeader gray bar and pull it down to make the PageHeader bigger. The example below stretches the PageHeader to about 7/8 of an inch. (use the left ruler to measure).
- Using the line tool on the side toolbar, draw a line at about the 3/4” mark. You can drag either end up, down or sideways to center it and line up with the outside edges of the “IbIReportID” and “IbIDate” controls below.
- Make sure it’s selected (it will have a little square mark on each end of it as below) then in the Properties window, change the LineColor to “Green” (click in the field and a little button will give you a drop-down list of colors) and the LineWeight to 2 by changing the default value of 1 to 2.

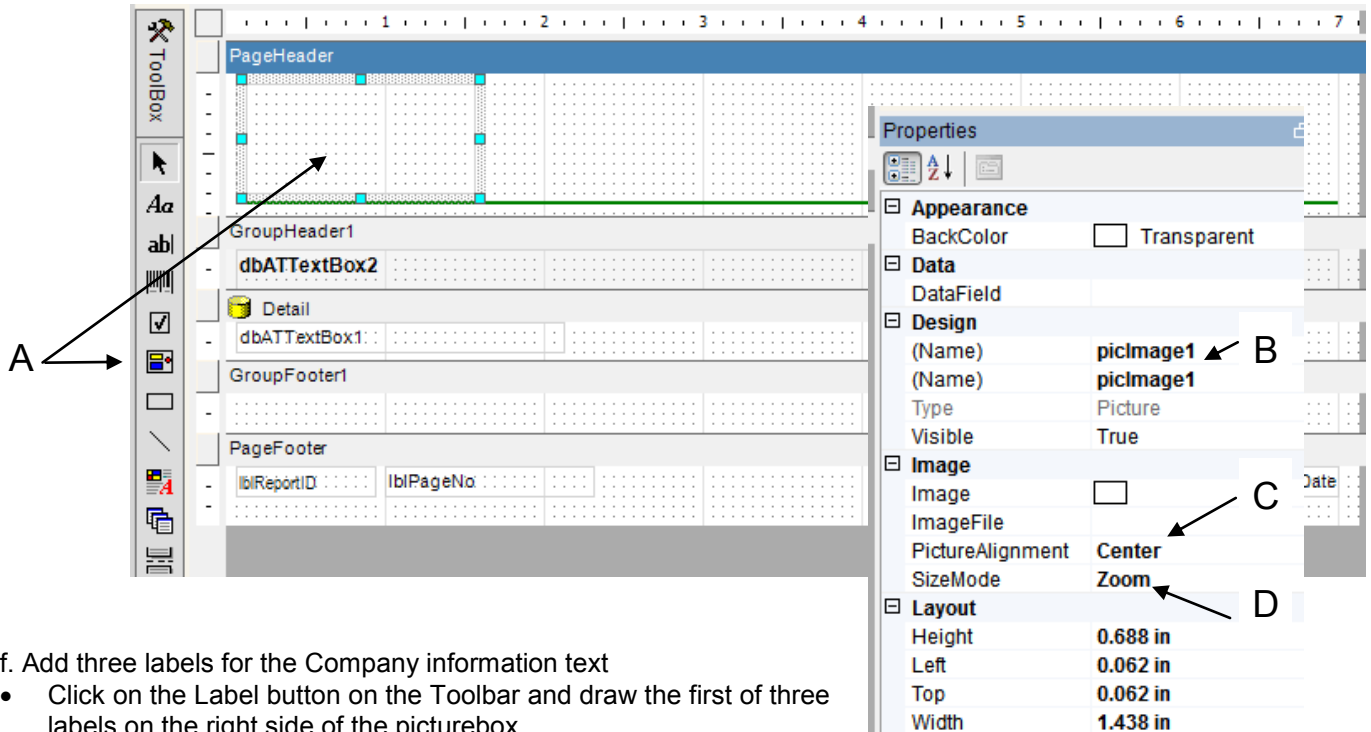


NOTE: In this exercise the ‘Categorized’ button has been clicked on the Properties box to organize all properties by the type of Property each one is.

Tutorial - Building A Report Design (Continued)

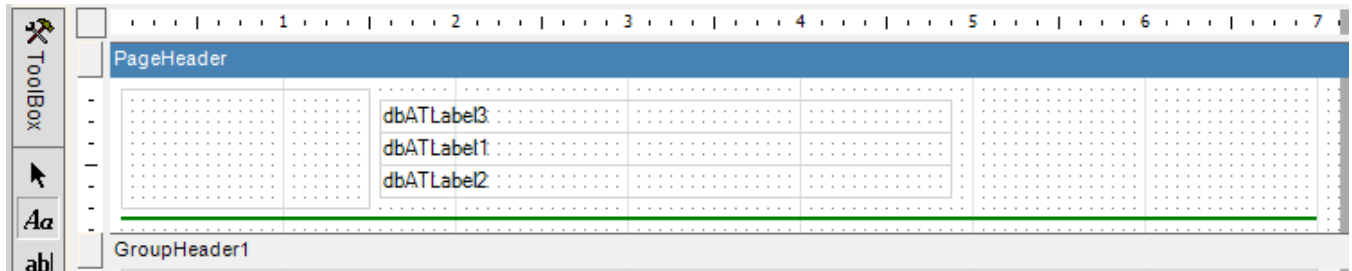
e. Adding a PictureBox Control for the Logo

- A. Click on the PictureBox button on the Toolbar to activate it, then draw a square in the PageHeader for the Company logo. It's left edge should line up with the green line and you can make it about 1.5 inches wide. If you are using your Company database for this exercise and your company logo is different proportions, change the height / width of the picture box to mimic the shape of your logo.
- B. Make sure the picture box is highlighted (will have little squares on each side and corner), then click on the Properties window and change the Design.Name to **picImage1** which will match the default scripts we'll edit later. **Note:** changing the first one will automatically change the second 'Name' property.
- C. Change the Image.Picture Alignment property to **Center**.
- D. Change the Image.SizeMode property to **Zoom**.



f. Add three labels for the Company information text

- Click on the Label button on the Toolbar and draw the first of three labels on the right side of the picturebox.
- Right-click on the first label and choose, "Copy", then right-click and choose "Paste" to create a second copy; please note that the copy will be laying directly over the original label. Drag it to a position directly below the first label, aligned on the left.
- Repeat Step B to create the third label. The PageHeader should now look like this:



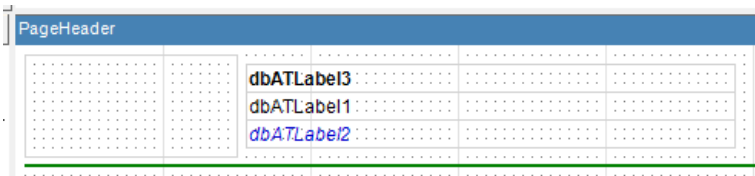
Tutorial - Building A Report Design (Continued)

g. Format the Labels

To format the labels in the same font as the rest of the report (the sample reports are all in **Arial** font), hold down your Shift Key and click on each one of the labels to select them all.

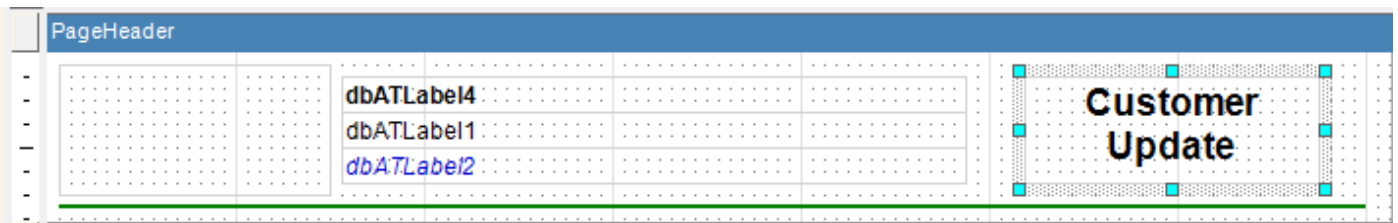
- In the Properties dialog, click on the FormatText.TextFont property, click the little gray button on the left and choose **Arial**, style = **Regular**, size **9** and click OK.
- Select JUST the first label, click on the FormatText.TextFont property and change the style to **Bold**.
- IF YOU ARE WORKING IN THE SAMPLE DATABASE, or your company info, line 4 has your email/website info: Select JUST the last label, click on FormatText.TextFont property and change the style to **Italic**. Then click in the Appearance.Forecolor property and change the color to **Medium Blue**.

The PageHeader labels should look like this:



Now we'll add a title to this form also on the PageHeader.

- ### h. Right-click on the top label (with bold font) and choose "Copy", then choose "Paste". Move the copy over to the right side of the PageHeader and resize it to center in the space, like this:



- ### i. In the Properties box change the Design.Text property to **Customer Update**; change the FormatText.TextAlign property to **Center**; and the FormatText.TextFont to size 14 (14.25).

The image shows three instances of the Properties dialog box for different labels, with arrows pointing from the text instructions to specific properties:

- Format Text** (Top): RightToLeft: False, TextAlign: TopLeft, **TextFont: Arial, 9pt**, WordWrap: True.
- Format Text** (Middle): RightToLeft: False, TextAlign: TopLeft, **TextFont: Arial, 9pt, style=Bold**, WordWrap: True.
- Appearance** (Bottom): BackColor: Transparent, ForeColor: **MediumBlue**.
- Design** (Bottom): (Name): dbATLabel2, (Name): dbATLabel2, Text: (empty), Type: Label, Visible: True.
- Format Text** (Bottom): RightToLeft: False, TextAlign: TopLeft, **TextFont: Arial, 9pt, style=Italic**, WordWrap: True.

The image shows the Properties dialog box for the 'Customer Update' label:

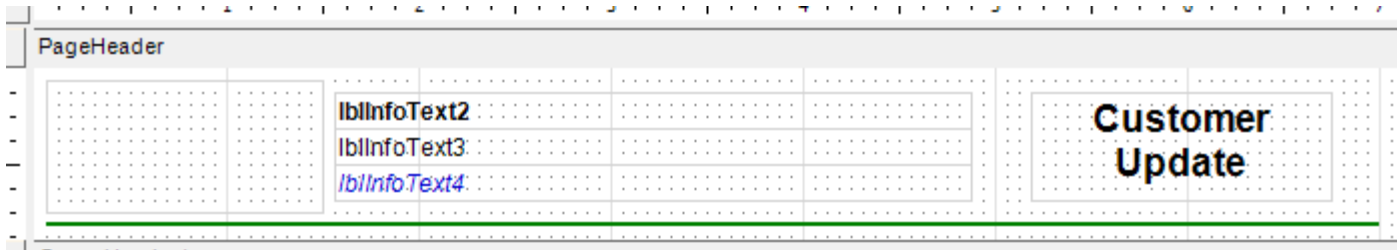
- Appearance**: BackColor: Transparent, ForeColor: **Black**.
- Design**: (Name): dbATLabel3, (Name): dbATLabel3, **Text: Customer Update**, Type: Label, Visible: True.
- Format Text**: RightToLeft: False, TextAlign: **TopCenter**, **TextFont: Arial, 14.25pt, style=Bold**, WordWrap: True.

Tutorial - Building A Report Design (Continued)

Rename the first three labels to match the default scripts for adding company information:

- j. Click on the top label, change the property Design.Text to **lblInfoText2**
- k. Click on the middle label, change the property Design.Text to **lblInfoText3**
- l. Click on the bottom label, change the property Design.Text to **lblInfoText4**

The PageHeader should now look like this:

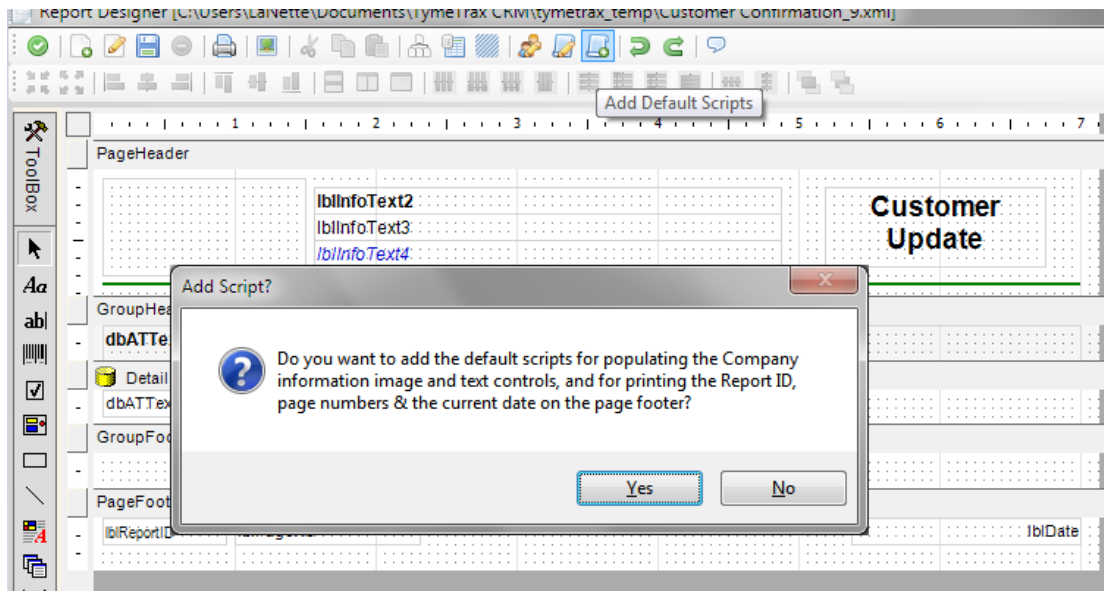


Click on the Green Check mark “Close” button on the Toolbar, then click “Save” on the toolbar to save your work. Click “Edit” again to edit it, and click “Open Designer” to return to the Layout Designer.

8. Updating the Scripts

So far we’ve added a logo picture box, named “picImage1” and three labels on the PageHeader (lblInfoText2, lblInfoText3 and lblInfoText4) which will display the Company information. In this step we’ll update the scripts to populate those controls at runtime.

Click on the “Add Default Scripts” button on the Layout Designer toolbar, then click ‘Yes’ to the prompt, and click “OK” on the reminder to refer to this guide for more information.



Click on the “Edit Scripts” button to open the Script Editor with all of the default scripts in it. Refer to the next page to remove all lines in red, which are not needed in this report.

Tutorial - Building A Report Design (Continued)

```
dbATReport _Report;
private int _pageno;
private string _Image1FilePath;
private string _Image2FilePath;
private string _Image3FilePath;
private string _Image4FilePath;
private string _reportID;
private string _InfoText1;
private string _InfoText2;
private string _InfoText3;
private string _InfoText4;
private string _InfoText5;
private string _InfoText6;
private string _InfoText7;
private string _InfoText8;
private string _InfoText9;
private string _InfoText10;
private string _QueryPromptText;

public void Report_Start(object Sender, System.EventArgs e)
{
    _Report = (dbATReport)Sender;
    this._pageno = 1;
    this._Image1FilePath = _Report.Image1FilePath;
    this._Image2FilePath = _Report.Image2FilePath;
    this._Image3FilePath = _Report.Image3FilePath;
    this._Image4FilePath = _Report.Image4FilePath;
    this._reportID = _Report.ReportNumber;
    this._InfoText1 = _Report.InfoText1;
    this._InfoText2 = _Report.InfoText2;
    this._InfoText3 = _Report.InfoText3;
    this._InfoText4 = _Report.InfoText4;
    this._InfoText5 = _Report.InfoText5;
    this._InfoText5 = _Report.InfoText6;
    this._InfoText6 = _Report.InfoText7;
    this._InfoText7 = _Report.InfoText8;
    this._InfoText8 = _Report.InfoText9;
    this._InfoText10 = _Report.InfoText10;
    this._QueryPromptText = _Report.QueryPromptText;
}

public void PageHeader_Initialize(object Sender, System.EventArgs e)
{
    Section _section = (Section)Sender;
    if(System.IO.File.Exists(_Image1FilePath))
    {
        ((dbATPicture)_section.Controls["picImage1"]).Image = System.Drawing.Image.FromFile(_Image1FilePath);
    }
    ((dbATLabel)_section.Controls["lblInfoText1"]).Text = this._InfoText1;
    ((dbATLabel)_section.Controls["lblInfoText2"]).Text = this._InfoText2;
    ((dbATLabel)_section.Controls["lblInfoText3"]).Text = this._InfoText3;
    ((dbATLabel)_section.Controls["lblInfoText4"]).Text = this._InfoText4;
    ((dbATLabel)_section.Controls["lblInfoText5"]).Text = this._InfoText5;
}

public void PageFooter_Initialize(object Sender, System.EventArgs e)
{
    Section _section = (Section)Sender;
    ((dbATLabel)_section.Controls["lblReportID"]).Text = this._reportID;
    ((dbATLabel)_section.Controls["lblDate"]).Text = string.Format("{0}", DateTime.Now.ToShortDateString());
    ((dbATLabel)_section.Controls["lblInfoText6"]).Text = this._InfoText6;
    ((dbATLabel)_section.Controls["lblInfoText7"]).Text = this._InfoText7;
    ((dbATLabel)_section.Controls["lblInfoText8"]).Text = this._InfoText8;
    ((dbATLabel)_section.Controls["lblInfoText9"]).Text = this._InfoText9;
    ((dbATLabel)_section.Controls["lblInfoText10"]).Text = this._InfoText10;
    ((dbATLabel)_section.Controls["lblPageNo"]).Text = string.Format("Page {0}", _pageno);
    _pageno++;
}
```

c. Carefully remove each line in Red from the scripts in the Script Editor

or

Copy and Paste the code on the next page into the Script Editor window, completely replacing the current code.



Tutorial - Building A Report Design (Continued)

```
dbATReport _Report;
private string _Image1FilePath;
private string _reportID;
private string _InfoText2;
private string _InfoText3;
private string _InfoText4;
```

d. Copy and Paste this code on nto the Script Editor window, completely replacing the existing code.

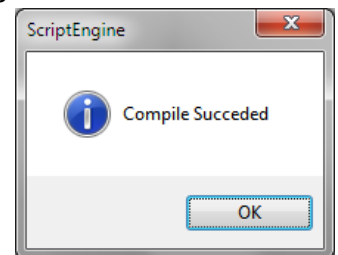
```
public void Report_Start(object Sender, System.EventArgs e)
{
    _Report = (dbATReport)Sender;
    this._Image1FilePath = _Report.Image1FilePath;
    this._reportID = _Report.ReportNumber;
    this._InfoText2 = _Report.InfoText2;
    this._InfoText3 = _Report.InfoText3;
    this._InfoText4 = _Report.InfoText4;
}
```

```
public void PageHeader_Initialize(object Sender, System.EventArgs e)
{
    Section _section = (Section)Sender;
    if(System.IO.File.Exists(_Image1FilePath))
    {
        ((dbATPicture)_section.Controls["picImage1"]).Image = System.Drawing.Image.FromFile(_Image1FilePath);
    }
    ((dbATLabel)_section.Controls["lblInfoText2"]).Text = this._InfoText2;
    ((dbATLabel)_section.Controls["lblInfoText3"]).Text = this._InfoText3;
    ((dbATLabel)_section.Controls["lblInfoText4"]).Text = this._InfoText4;
}
```

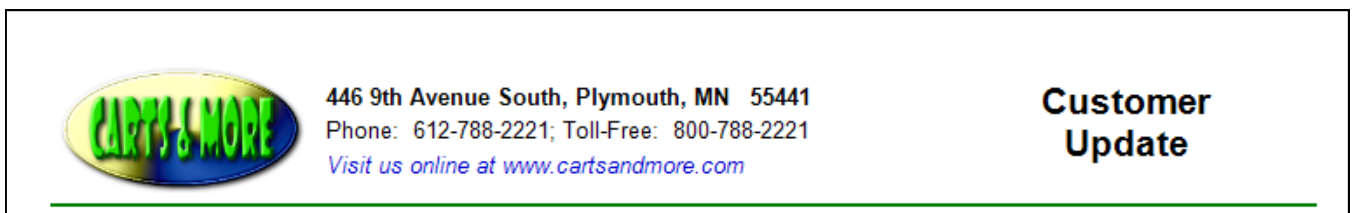
```
public void PageFooter_Initialize(object Sender, System.EventArgs e)
{
    Section _section = (Section)Sender;
    ((dbATLabel)_section.Controls["lblReportID"]).Text = this._reportID;
    ((dbATLabel)_section.Controls["lblDate"]).Text = string.Format("{0}", DateTime.Now.ToShortDateString());
}
```

e. On the Page Footer, right-click on the “lblPageNo” label and choose “Delete” to remove it. Usually you’ll want to print a page number on a report, especially if it’s a multi-page report, however since we’re only sending one page to each Customer, we’ll remove it as it wouldn’t make sense for customer #23 to see “Page 23” on their copy.

f. To test your changes, click on the “Compile” button on the Script Editor; you should get a message box like the one at right if everything is correct. If not, you can close the Script Editor, click the “Add Default Scripts” button again to reload the defaults.



g. Once the scripts are correct, close the Layout Designer, save the report and then click the “Preview Report” button to see the changes on the PageHeader. It should look something like this in the Print Preview window:



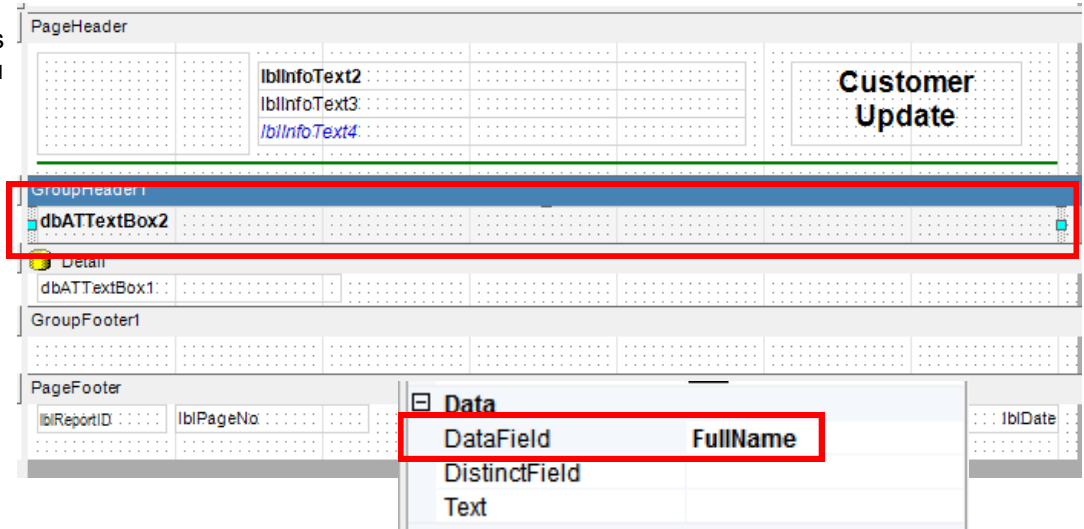
h. Close the Print Preview window and reopen the Layout Designer to continue.

Tutorial - Building A Report Design (Continued)

9. Customizing the GroupHeader1

a. Click on the field in the GroupHeader1 section to select it, then in the Properties window, click on Data.DataField and choose "FullName" at the bottom of the list which is one of the results of the Additional Query. This will be the Company_Name, if there is one, else the Last Name, First Name, Middle Name values for each Company record.

This field is formatted as **Arial, 9pt, style=Bold**; as with any field or label, you can change the **Font** properties to whatever you like.

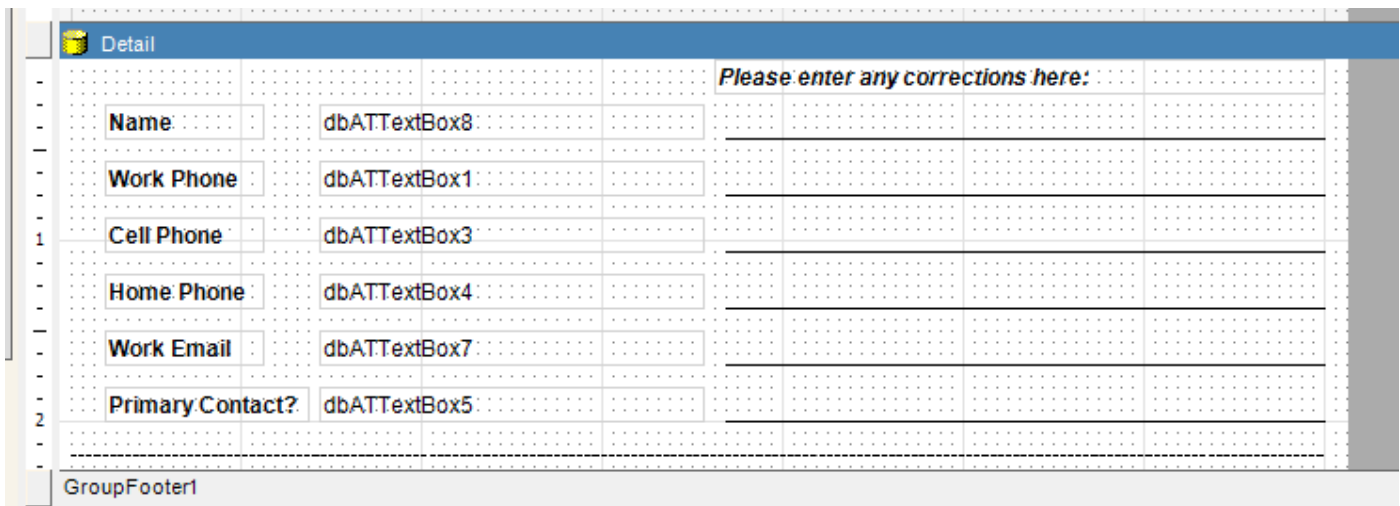


10. Adding the Detail Fields

All Group Header sections are used to categorize ("group") data in logical sections and typically only use one or just a few fields to do so. The "Detail" area of the report is where the actual details of the report are displayed. Most reports will display this data in a column format with a number of fields lined up in the Detail section in a readable order. However, you can also "stack" fields in any section (see the "Company Directory" report in the Company reports as an example) if there is not enough room to place the fields side-by-side.

The purpose of this report is to send each customer their phone & email information and have it verified and returned with any changes. To do that, we'll need to show their current Contacts information, then provide a place for them to write in any corrections.

a. Hold your mouse over the top border of the Group Footer 1 section and stretch the Details section to about 1-1/2 inches high. When we're done adding fields to the Detail Section, it will look like the image below.

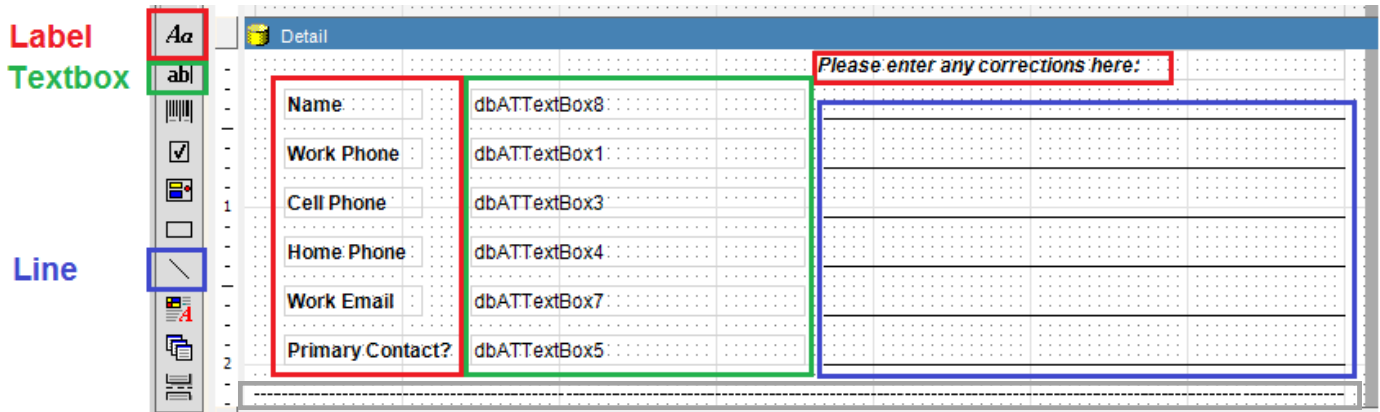


Tutorial - Building A Report Design (Continued)

Detail Section (Continued)

b. Refer to the diagram below to add Labels, Textbox (fields) and Lines to the Details section. Don't worry about formatting (fonts, etc.) until you have all of the items on the page. We'll format the text and labels in groups in the next step.

Note the spacing of the labels at the left; we've moved them in a couple of gridlines to indent the group on the page. Set the "Text" property of each Label with "Name", "Work Phone", "Cell Phone", "Home Phone", "Work Email" and "Primary Contact?" in the order shown below. Set the Label text above the lines to "Please enter any corrections here:".



c. Click on each of the Textboxes and then set the "DataField" property of each to these fields (in order): "ContactName", "Work_Phone", "Cell_Phone", "Home_Phone", "Work_Email" and "Is_Primary_Contact".


d. Behind each Textbox using the Line tool, draw a line where customers will write in corrections for each field. These should be the same length, and should line up with the bottom of each Textbox.

e. Click on the background to clear all selections, then click on the first ("Name") label. Hold down a Shift key, then click on each of the other labels in the column. All labels should show the selection tags (light blue squares on the borders). Set the TextFont property to Arial, 9 pt, Bold by clicking the button on the right side of the property to open the Font selection dialog.

f. Select all of the Textbox controls, then set the TextFont property to Arial, 9 Pt Regular. Select the "Please enter..." Label above the lines and set the font to Arial, 9 Pt, Bold Italic.

g. At the bottom of all of the controls, add another line (boxed in purple above), which will separate multiple Contacts. Click on the line to select it, then change the Line Style property to "Dash".

h. Click on the green checkmark on the top toolbar to Close the Designer. Click "Save" to save your work so far, then click Edit to resume editing. Click "Preview Report" to view the results... it should look something like this:

		446 9th Avenue South, Plymouth, MN 55441 Phone: 612-788-2221; Toll-Free: 800-788-2221 Visit us online at www.cartsandmore.com	Customer Update
<hr/> ABC Company, Inc.			
			<i>Please enter any corrections here:</i>
Name	Johnson, Brad	_____	
Work Phone	212-555-2323	_____	
Cell Phone	444-555-6666	_____	
Home Phone		_____	
Work Email	sales@abcompanyincnet	_____	
Primary Contact?	Yes	_____	
<hr/>			

Tutorial - Building A Report Design (Continued)

11. Customing Group Footer 1

For each Group section, the Header is the “beginning” of the group and the Footer is the “ending” of the Group. Since we’re Grouping records by the Company FullName field, and our intent is to break to a new page for each Company, we need to add the feedback section to the Group Footer 1 section so it’s only displayed once for each Company.

a. Click on the border on the bottom of the Group Footer 1 section and stretch the height of it to about 2 inches. You can adjust the finished height after you’ve added all controls.

In this section we’ll add two Labels and four Lines, using the Toolbar at the left. Click on the top Label and set the TextFont to Arial, 9 Pt. Bold Italic and the Text to “Please include any feedback about our services below:”. Click on the bottom Label, set the TextFont to Arial 9 Pt. Italic Regular and the Text property to “If you have made any corrections or have feedback, please return this to our address above within 10 days.”

Copying Controls

b. Draw the first line the length you want, then right-click on it and choose “Copy”. Move the first Line to the next position, then right-click on the section and choose “Paste”. The new copied line will be pasted into the original line’s position. This feature is standard with all controls... the copied control is pasted into the same position as the original control.

c. Space the items as you see below; as with all controls, you can always make changes after you’ve Previewed the report. The results should look something like this:

The screenshot shows a report designer interface for a section titled "GroupFooter1". The section is a rectangular area with a dotted grid background. At the top, there is a label with the text "Please include any feedback about our services below:" in a bold, italicized font. Below this label are four horizontal lines, each representing a text input field. At the bottom of the section, there is another label with the text "If you have made any corrections or have feedback, please return this to our address above within 10 days." in a regular, italicized font.

d. Close the Designer and Save the Report Design. Click “Preview Report” to view the results. You can edit again as necessary to get the layout you want.

12. Setting Page Breaks

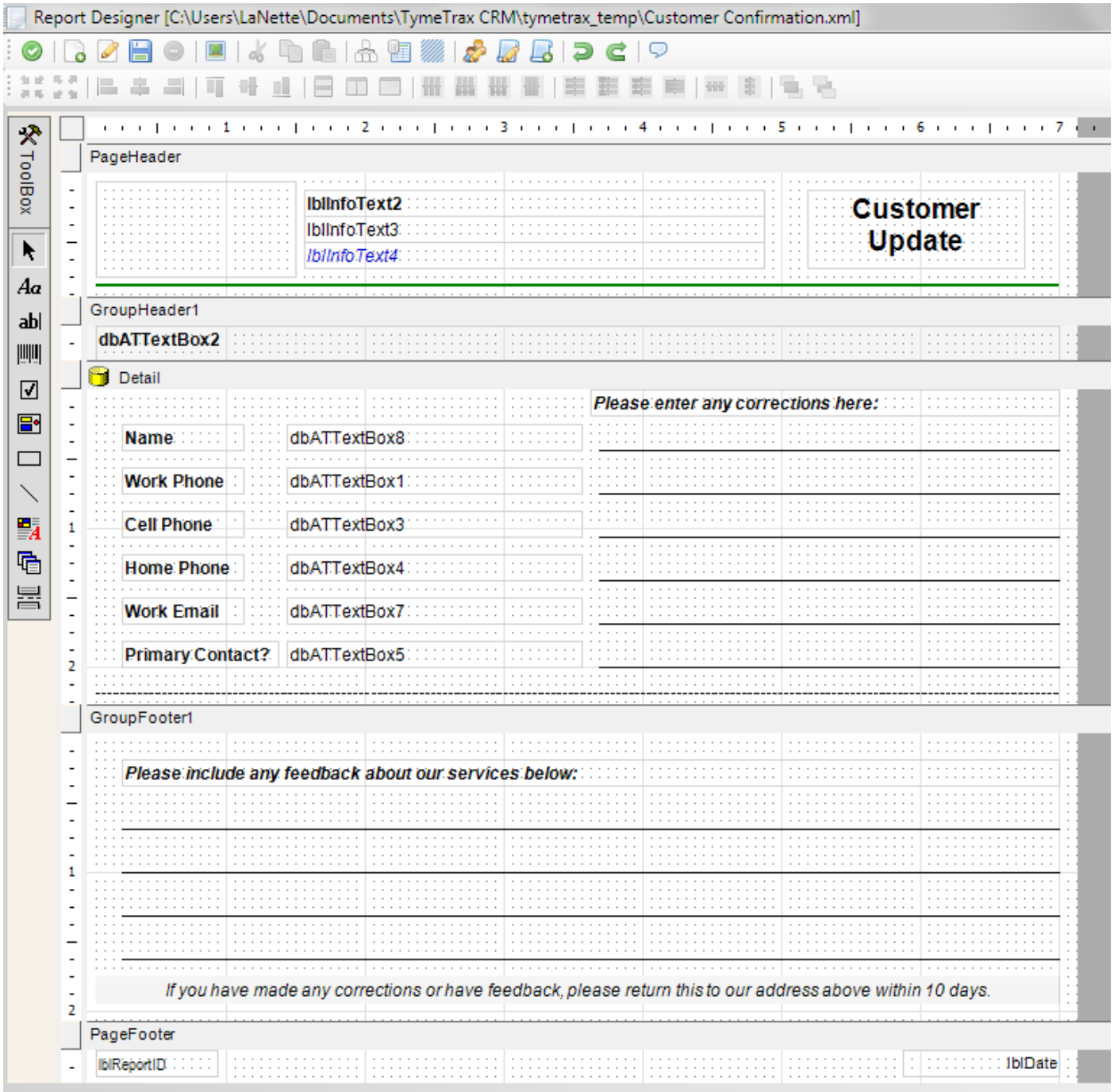
The last step is determining how to keep the data from breaking across pages at the wrong point. For sure, we want to break after each Company so we can simply mail out the forms without sharing information across Customers.

- For each of the Group Header 1, Detail and Group Footer1 sections, click on the section, then set the “Keep Together” Property to “True”.
- On the Group Footer 1 Section, set the “New Page” Property to “After”, which will start a new page for each customer regardless of how many pages it takes to print all Contacts for each customer.

Tutorial - Building A Report Design (Continued)

13. Viewing The Results

Your design should look something like this:




A screen print of the finished Report is on the next page.



Tutorial - Building A Report Design (Continued)

14. The Finished Report

Your Report should Print Preview like this, and each customer name should start on a new page:



446 9th Avenue South, Plymouth, MN 55441
Phone: 612-788-2221; Toll-Free: 800-788-2221
Visit us online at www.cartsandmore.com

Customer Update

ABC Company, Inc.

Name Johnson, Brad

Work Phone 212-555-2323

Cell Phone 444-555-6666

Home Phone

Work Email sales@abcompanyinc.net

Primary Contact? Yes

Please enter any corrections here:

Name Johnson, Marcy

Work Phone 212-555-8390

Cell Phone

Home Phone

Work Email

Primary Contact? No

Please enter any corrections here:

Please include any feedback about our services below:

If you have made any corrections or have feedback, please return this to our address above within 10 days.

100031-MGR09/19/2011



TymeTrax System & User Requirements

TymeTrax Software products are designed to be straightforward and easy to use, with minimal administration by the customer, both on a hardware and software level. We strive to quickly resolve any and all issues regarding installation, performance and daily usage; however the following General and System Requirements must be met before we can effectively do so.

General Requirements

- ◇ Contact Person - For multiple user customers, Crow River Tech requires that **one person** from within your company be designated as the Contact Person for exchanging information about updates, support issues, and any other communication with Crow River Tech regarding the training, installation and general daily operations of the software product. We do not support each user on a one-to-one basis in multi-user situations; the Contact Person within the company is responsible for mitigating on-site issues and communicating those personally to the support staff at Crow River Tech and then reporting any resolutions, update notices, etc., to the licensed users within the company.
- ◇ Users must have an overall general understanding of how to use a computer, including but not limited to*:
 - *Basic Windows skills, such as copy/paste functions and file & folder navigation in Windows Explorer*
 - *General Outlook email familiarity and ability to adjust personal settings such as Outlook Rules & Alerts and edit Mail Account settings*

System Requirements

- ◇ Supported Operating Systems (please make sure all updates and service packs are applied regularly)
 - *Windows XP(2)*
 - *Windows XP(3)*
 - *Vista (any version)*
 - *Windows 7*
- ◇ Supported Database Platforms
 - *Microsoft Access - does NOT require the Office Access application installed on the user's computer; database support is integrated into our software and does not require additional purchase by customer*
 - *Microsoft SQL Server 2005 or 2008 - requires SQL Server installed and operating on a company network and is NOT included in our software packaging, installations or pricing.**
 - *Microsoft SQL Server EXPRESS 2008 - requires SQL Server Express 2008, which is a free product available from Microsoft's website (<http://www.microsoft.com/sqlserver/2008/en/us/editions.aspx>). We recommend downloading and installing the SQL Server Management Studio 2008 (also free) to facilitate setting up the application login and password required by our software. Please refer to Microsoft documentation for system requirements for their products.***
- ◇ User Hardware Requirements
 - *MINIMUM 2GB RAM on XP & Windows 7 Systems; 3GB RAM on Vista. More RAM results in better performance.*
 - *10MB Hard Drive Space for setup and installation; additional storage required for database (up to 2GB for a local Access database.)*
 - *Video Resolution Minimum of 1280 x 800, 96 dpi (or 100% on Windows 7 Systems); settings that vary from the default proportions for your system may cause some pages to appear distorted*
 - *Other computer hardware less than 3 years old.*
- ◇ Additional Requirements (to use the listed specific features only)
 - *Microsoft Outlook 2000, 2003, 2007 or 2010 (32-bit only) (for email and remote replication)*
 - *Microsoft Streets & Trips 2008, 2009 or 2010 (for applications that support mapping from CRM/Company data)*
 - *Internet Connection (for updates and remote replication)*
 - *For Replicator Server Utilities - Windows Server NOT required (Replicator can run on a workstation); however the workstation should be an up-to-date computer with a minimum of 4GB of RAM to efficiently process replication packages for client systems.*

*The Internet offers a wealth of information for learning computer basics, such as these FREE tutorials:

- ◆ <http://www.basics4beginners.com/>
- ◆ <http://www.homeandlearn.co.uk/bc/bcs1p1.html>
- ◆ ..or simply search for "free computer tutorials" using your search engine to find topics to fit your needs

**While we provide instructions for setting up an application role login and password on SQL servers, we do not support the overall operations of any SQL Server products as that is the responsibility of the vendor from which it was purchased.





TymeTrax Software Report Designer Guide

Version 3.01

*All user guides are available for download
on our support site at
www.TymeTrax.com*

This publication is the property of

Crow River Tech, LLC

64806 318th Street
Litchfield, MN 55355 USA
www.CrowRiverTech.com

*and may not be copied or reproduced
without express written permission.*

*Microsoft Windows XP, Vista, Windows 7, SQL Server
and Microsoft Outlook, Word, PowerPoint, Excel & Access
are trademarks of Microsoft Corporation.
Any other product names are trademarks
of their respective companies.*

*TymeTrax Software / Crow River Tech, LLC. Copyright 2011.
All Rights Reserved.*

