Tips on Using Crystal Reports

and

Tips on Converting Custom Reports to the Tigerpaw Business Suite 9, 32-bit Database Structure

and

Useful Formulas

Revised: 1/2/2003

Please route this document to the person in your organization responsible for using Crystal Reports.

If you have tips / suggestions concerning this document that that you think would be helpful to someone else, please email them to Ed Spence at

eds@tigerpawsoftware.com

Table of Contents

Introduction

This document is divided into three sections. The first section consists of tips on using Crystal Reports – most specifically, version 8.5. These tips are intended to help you write new custom reports or modify system-supplied reports that are written against the 32-bit Access database used in our Tigerpaw Business Suite 9 software.

The second section of this document is titled "Tips on Converting Custom Reports to the Tigerpaw Business Suite 9 Database Structure". This section is intended to help people who wrote or modified reports in Crystal version 6 for our Tigerpaw Business Suite 8 database structure, and need to convert those reports to Crystal 8 (or 8.5) for our version 9 database structure.

The third section consists of useful formulas I have developed. Feel free to use them if you have use for them.

Using the Correct Version of Crystal

Note: the version number of our software is on the "splash screen" displayed when you launch the software, and it's on the Help menu's "About Tigerpaw Business Suite" option. Version 9.0+ is our latest version. Note that you must use Crystal version 8.0+, 32-bit, when writing reports against that database; previous or later versions of Crystal will not work properly.

If the version number of our software is 8.0+, that's our previous product and you must use Crystal version 6.0+, 32-bit to write reports against that database; previous or later versions of Crystal will not work properly. Note that Seagate Software no longer officially supports Crystal's version 6.

You can check the version of Crystal by clicking Help | About Seagate Crystal Reports. Check the middle of the second line – it will read "CRW32" and the very next number is the version number, which is then followed by other numbers, separated by commas. The format is:

CRW32 V, n, n, nnn" where the "V" is the version number, followed by sub-version numbers.

Note: If you do not have the correct version of Crystal Reports, you can contact Seagate Software at 604-681-3435 or by email at sales@seagatesoftware.com.

This document was written primarily for Crystal version 8.0+ and uses screen shots from version 8.5.0.217.

Other Documentation You Should Have

If you intend to write / modify reports against our database, you should have a "data dictionary", a list of all the tables and fields and the relationships between the tables. You can print the data dictionary from (Microsoft) Access by following the steps listed below:

If you are using Access 2000, you may have to "enable" the database before you can print the data dictionary. For instructions on enabling the Access database, watch the video titled "Enabling a Version 9 Database" in the category titled "Upgrading from V8 to V9 Topics", on our web page at

www.tigerpawsoftware.com/support/supportvideos_ov.asp

If you are using Access 97 or a database that has been enabled for Access 2000:

1. Launch Access
2. Select the database
3. Click Tools | Analyze | Documenter
4. From the Documenter form, click Options to display the Print Table Definition form. From that form, select the fields you want to see. I suggest that under "Include for Tables", you select “Relationships”, under "Include for Fields", you select "Names, Data Types, and Sizes", and under "Include for Indexes", you select "Names, and Fields".
5. Click OK to close the Print Table Definition Form and redisplay the Documenter form.
6. On the Tables tab, select the tables you want printed - if you want all the tables, click the Select All button.
7. Click the OK button - the system displays a print preview of the output, on the Object Definition form.
8. Click the Print icon on the form to print the report.

Note that if you elect to print data from all the tables (using the options suggested above), the report will be approximately 350 pages long (that’s how long it is in Access 97).

You should also have a copy of the Reports Guide – from the main page of our web page

www.tigerpawsoftware.com

click “Downloads” to open a drop-down menu and select “Suite 9” to display the Suite 9 Downloads page. From that page, you can download the Reports Guide, which includes a sample of every report in the system. If you need a report we don’t currently have, using the Reports Guide, you can (hopefully) find a report similar to the report you’re dreaming about, and use that as the starting point for a custom report. Don’t reinvent the wheel; we encourage you, if possible, to take advantage of the work we’ve already done.

Expediting the Report Development Process

You can save yourself considerable report development time if you do these three things:

1. Run both Crystal and the Tigerpaw Business Suite application at the same time.

2. Make sure that when you Open a report in Crystal, you Open it from the directory that is used by our application when it calls for a report.

3. In Crystal, work with a small set of known data, specifically data that you can quickly edit in our application. See “Working With Specific Data Records” below.

The idea behind doing these things is that you enter / change the data in our Tigerpaw Business Suite application and then immediately Print Preview the report in Crystal to verify that the data is being properly processed.

After you are sure that a report is working as expected in Crystal, you should run the report from within our application. That’s the acid test - only when a report runs properly from our application can it truly be said to be working properly.

Running Reports from within the Tigerpaw Business Suite

Running a report from within the Tigerpaw Business Suite software may involve creating a Report Definition record in the Report Manger in our software. For additional information, refer to “Using the Report Manager” in the Reports Guide.

Section 1. Tips on Using Crystal Reports

Version 9 of our software includes over 320 reports, all of which have been converted to work with Crystal version 8.5, 32-bit. As of the writing of this document, 8.5 is Crystal's latest version.

When you develop a report in Crystal, the end result is an ".rpt" (report) output file. We supply the .rpt files for all the reports in our Tigerpaw Business Suite 9 software. You can use Crystal (version 8.0+, including version 8.5) to modify any of those reports, or you can write your own reports from scratch, and you can print or print preview those reports from within our application – for an explanation of that process, refer to "Using the Reports Manager" in the Reports Guide.

This document is designed to assist people who are not experienced Crystal users. It is intended to help those people get started and quickly learn some of the most important things about Crystal Reports. Emphasis is on the "big picture" things in Crystal that can cause you much grief if you don't know how they work.

Obviously, we can't teach you everything there is to know about Crystal in this document (they have a 720 page manual designed to do that), but, **if you are not already an experienced Crystal user, knowing the things in this document can save you many hours of work**.

Note: in addition to Crystal, you can also use Microsoft Access to write reports against our Tigerpaw Business Suite 9 database. Reports written in Access, however, can't be launched from within our software, they must be launched from within Access. This document does not discuss report writing in Access – refer to the Access User's Manual or Help Text for information on that subject.

If you previously used our version 8 software and wrote / modified any of the reports, and are now upgrading to our version 9 system, refer to Section 2 in this document, "Tips on Converting Custom Reports to Version 9 of the Tigerpaw Business Suite".

Crystal's Help Text

Learn to use Crystal's extensive Help system. Click the word "Help" on the right side of the menu bar to display the Help menu. Next, click Crystal Reports Help. On the Contents tab, double-click and read any topics of interest, including:

- Welcome to Crystal Reports | About the Online help
- Quick Start | Learning how to use Crystal Reports | Quick Start for new users (you really should go through this material if you're a new user)
- Report Design Concepts
- Introduction to Reporting

Do the same on the Index and Search tabs. On the Index tab, browse through:

- Application tutorials (this has a good summary list of useful-to-know topics)
- Fields (many subtopics)
- Formula Editor
- Formulas (many subtopics)
- Formulas In Action Index
- Functions Index (Alphabetical and many subtopics)
- Glossary
- Group Command (Insert Menu)
- Index (multiple subtopics)
- Linking
- Record Selection
- Relational Databases
- Reports – basic design
- Running Totals
- Section, Section Characteristics
- Subreport (multiple subtopics)
- Troubleshooting

The Undo and Redo Buttons (Very Useful)

The single most important function in Crystal is the "Undo" function, which undoes the last (undoable) thing you di d on a report. Multiple actions can be undone, beginning with the most recent action, by clicking the Undo button multiple times. **The button for this feature has an icon of a blue arrow that is curved to the left. There is also a Redo button, just to the right of the Undo button;** it has an icon of a blue arrow curved to the right; it redoes whatever you just undid. **You will use the Undo button often.**

Opening a Report

To open a report, first launch Crystal's Report Designer and then click File | Open. Next , browse to find and select a report. As indicated previously, in order to expedite the development process, you should make sure that when you Open a report in Crystal, you Open it from the directory that is used by our application when it calls for a report.

Modifying Existing Reports

If you need a report that's not included in the system, assuming it's not an obvious "custom" report, one that includes your logo, for instance, contact us to see if such a report is in our future plans. If we don't plan to add a report like that, you can write the report yourself, or ask about us writing it for you.

If you're going to write custom reports, we encourage you to take advantage of the work we have already done – find a report that includes most of what you want and then modify that report. Ideally, you'll be able to find a similar report that already has all the tables and table relationships included, the proper totals defined, the proper group breaks defined, etc. The point is, don't reinvent the wheel, you can probably save a lot of time if you first check the Reports Guide mentioned previously to see if we have a similar report.

Custom Reports Naming Conventions

Important Note: If you want to modify a report that was supplied with the software, the first thing you should do after you open the appropriate file, is to click File | Save As and save it with a different name (typically just add a "Z" to the front of the name, see below).

If you don't change the name of the report, but instead you use the same name as was originally used for the report, your report will be overwritten the next time you update our software. To prevent the problem of duplicate report names, use a report name that starts with the character "Z", except where noted below. We will never use a Z as the first character in any of the reports we write.

The Tigerpaw Business Suite includes several reports that we call "documents" – these are reports that are to be sent to an outside company, like a quote or an invoice. Whenever you print or print preview one of these documents, the system displays a form that includes a "Custom" print option.

The "documents" are listed below, along with the Crystal Report File Name you **must** use if you want to customize one of the documents. When you print or print preview one of these documents, the system looks for the "Custom Crystal Report File Name" and sets the "Custom" print option as the default if it finds the appropriate file name. Note that the names that must be used for these custom document reports are exceptions to the suggestion about starting customized reports with a file name that has a "Z" as the first character. We will never include reports in our updates that use the Custom Crystal Report File Names.

Document	Our Crystal Report File Name	Custom Crystal Report File Name
Quote	Quote.rpt	Cquote.rpt
Service Order	Newsodtl.rpt	Newsocus.rpt
Invoice	Invoice.rpt	Cinvoice.rpt
Contract Invoice	Coninv.rpt	Cconinv.rpt
Purchase Order	Porder.rpt	Cporder.rpt

To make it easy to determine the Crystal report file name used to produce a report, the file names are, with few exceptions, printed in the page footer at the bottom of every page of the report. In addition, if a report is printed from the Report Manager, you can determine the name of a report by clicking the Title in the Report Manager, and then click Edit to display the Report Definition form, with the report's Crystal Report File Name displayed at the top of the form.

Options File Settings (Very Important)

When you first begin using Crystal, you should check the settings of some variables in the Options File – click File | Options to display the Options form. This form contains several tabs with variables that control various functions. The setting of mo st of those variables is a matter of personal preference but there are a few that must be set to specific values. (Note that setting these variables is basically a one-time job although you may have to change some of the settings if you also write Crystal reports against databases in some other vendor's software).

1. On the Layout tab, under View Options, check "Show Short Section Names in Design" and Show Section Names in Preview". This causes Crystal to abbreviate the section names, "Page Header" becomes "PH", etc., leaving more room for the report itself. Also, under Field Options, check "Show Field Names" (which replaces all the "Xs" with field names on the Design form). On the Grid options, I prefer a grid setting of "0.32".

2. On the Database tab**, make sure that the Auto-SmartLinking checkbox is not checked.** Crystal has the ability to automatically create links between the tables used in a report but you **do not** want it to perform this function - it makes many links that are not correct. **In all cases, you want to create your own links between the tables in a report.** Note: even with Auto-SmartLinking turned off, Crystal sometimes performs the Auto-SmartLink function, but I have not been able to define when / why.

3. On the Reporting tab of the Options form, under When Reading Data, set the Convert Date-Time Field to "To String". This setting causes Crystal to read all date and/or time fields in the database as 22 character "string" (alphanumeric) fields (as required by our software). Note: because dates are read as 22 character dates, displaying a date in a report requires you to use a formula (see below). If you do not set this to the To String setting, the report will work OK in Crystal but it may fail when you run it from our application.

4. Also, on the Reporting tab, check the "More Report Engine Error Messages" and the "Display Alerts on Refresh" checkboxes. Leave the other checkboxes unchecked, although you might consider using the Autosave Reports function. Make sure that the Formula Language dropdown is set to Crystal Syntax" (unless you know and prefer Basic Syntax).

Make sure you save changes you have made to the Options form by clicking the OK button to save / close the form.

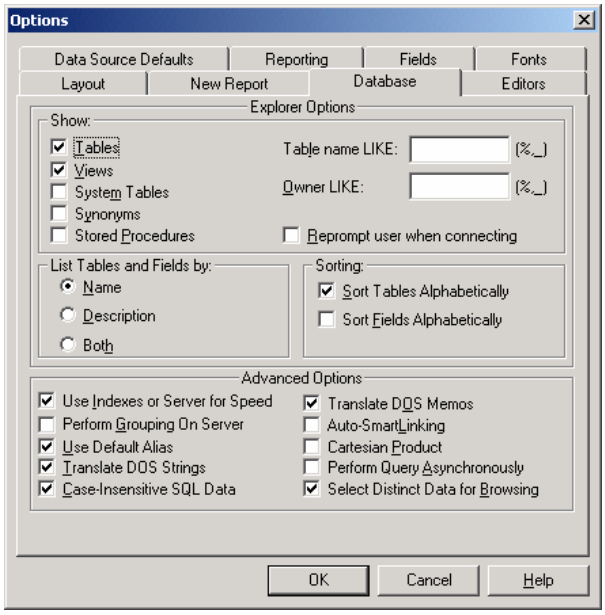

Figure 1. Make sure Auto-SmartLinking is turned off

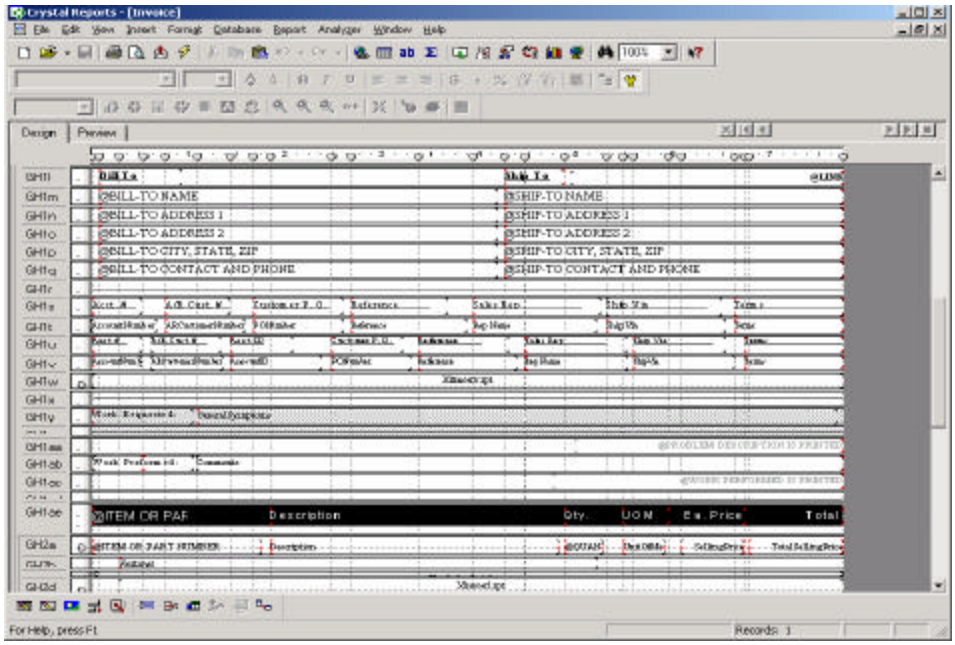

Figure 2. The Design form will look like this if you set the proper File | Options settings

Report Options (Very Important)

The Report Options form contains variables that are specific to a particular report (the one currently displayed on Crystal's Design form). Several of the variables were copied from the Options file (see "Options File Settings" above) when the report was first saved. Nonetheless, you should check this form when you first begin working with a report. To display the Report Options form, click File | Report Options. Make sure that:

1) the Convert Date-Time Field is set to the value “To String”
2) the Convert NULL Field Value to Default check box is not checked

If you change the setting of the Convert Date-Time Field, when you click OK to leave the Report Options form, Crystal automatically “verifys the database”, (see “The ‘Verify Database’ Operation” below).

See the sections below titled “Date Fields” and “Time Fields” for a description of the formulas used to print string-format date and time fields.

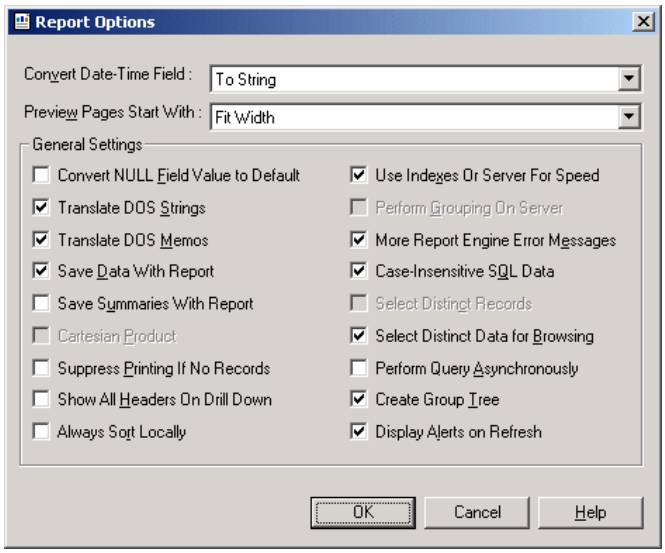

Figure 3. The Report Options form

“Setting the Location” of the Database (Very Important)

In order to be able to print or print preview a report from within Crystal, using data from your database, you must first provide Crystal with the Path and database name, via the Set Location function.

You only have to do this function one time for each report, unless you change the location or name of your database, and then you’ll have to do it again if you want to work with any given report in Crystal. Note that you must run this function before you can successfully run the “Verify the Database” function explained below.

To Set the Location (to point to your database), from the Design form:

1. Click Database | Set Location to display the Set Location form shown below. Note: As shipped by us, the path \ database used in all the reports is I:\32bit Reports\sample.mdb, as is indicated in the small Location window near the bottom of the form in the Figure below.

If the table shown in the Location window is already pointing at your database (or a copy thereof), then scroll to the bottom of the list in the large Databases window at the top of the form and check the last table in the list. If both tables are pointing at your database, then you can skip the rest of this process – click the Done button to close the Set Location form, and then proceed to the Verify the Database function (see below).

If you find that any table does not point at your database, note the name of the table (tblAccounts, in this case). You will need to know that table name later in this process.

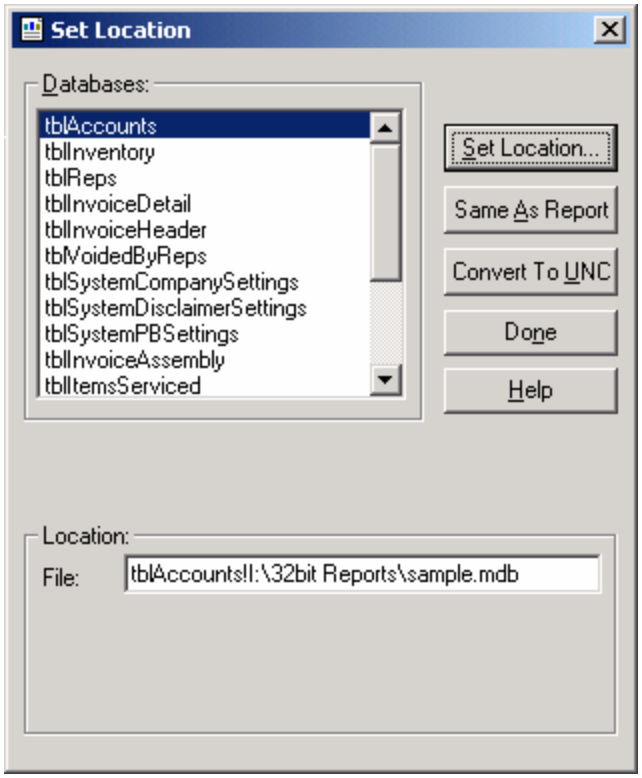

Figure 4. The Set Location form

2. On the Set Location form, click the Set Location button to display the Data Explorer shown below. If necessary, click “Database Files” to view the contents of that folder, as shown below. If you have already done the Set Location operation on some other report, you will see the database you’re looking for under the heading “Find Database File”, otherwise you must double-click on “Find Database File” and then browse and find the correct database file.

Note: if you don’t know the location of the database, you can determine where the database is that our application is using, by clicking Help | About Tigerpaw Business Suite | Current Database (in our application).

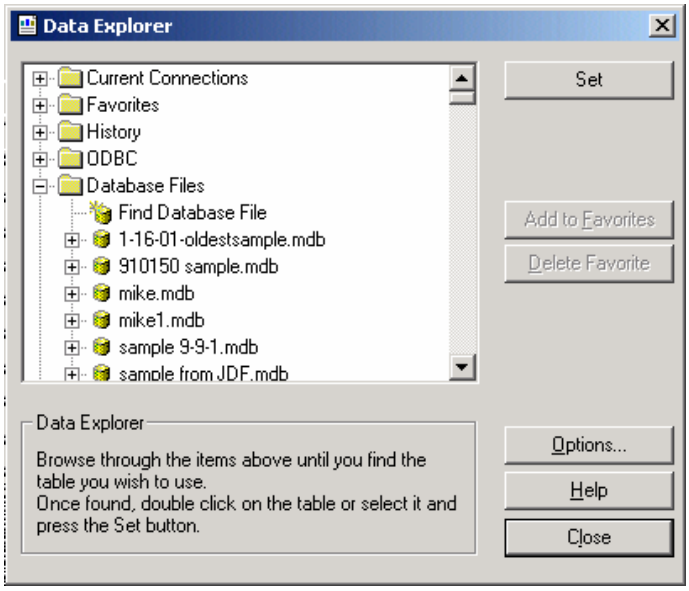

Figure 5. The Data Explorer form

3. Once you have found the correct database from which you want to add tables to the report, double-click the database name to display a list of all the tables in the database.

4. Next, click to highlight the table name that is the same as the table name that was highlighted on the Set Location form (tblAccounts, in this example), and then click the Set button in the top right corner of the data Explorer form.

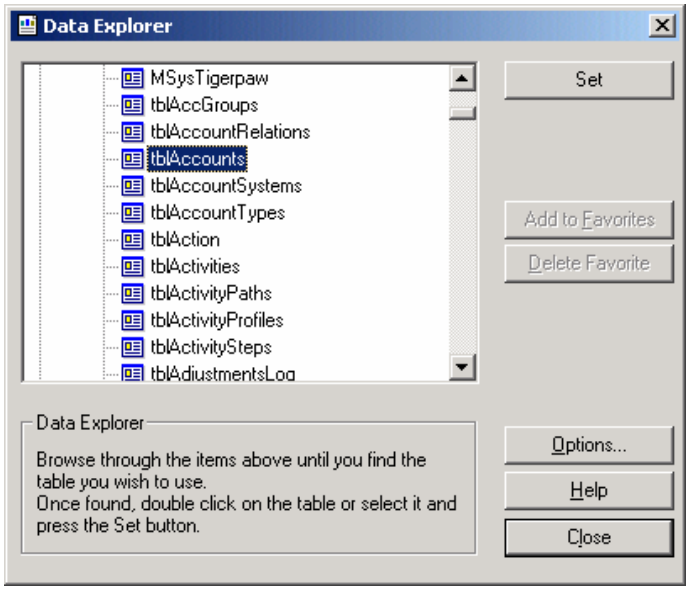

Figure 6. Highlight the table noted on the Set Location form

After you click Set, Crystal redisplays the Set Location form, overlaid by a message asking if you want to "Propagate server and database changes across tables with the same information?" .**Always click Yes in response to that message; never point a table to a database that is different than the database that is pointed to by some other table in the report.**

Next, click the Done button on the Set Location form. At that time, Crystal will do an automatic Verify Database function – nonetheless, you will still need to manually do the Verify Database function, as explained below.

Note: if the report you're working on includes subreports, Crystal automatically performs the Set Location function in each of the subreports, to point to the same database that you specified in the main report. At least that's what happens in most cases. In some cases, however, Crystal does not change the database that is being pointed to in all the subreports (for reasons that I do not understand), and this causes a minor problem in the Verify the Database function, see below.

The "Verify Database" Operation (Very Important)

When the Tigerpaw Business Suite application gets a request to print or print preview a report, it calls the Crystal Report Engine. The print engine first compares the table / field structure of the database you're using to an internal database structure map it stored when the report was last "verified against the database". If the internal map does not match the structure it finds in your database, the print engine displays an error (most, but not all of the time), and is unable to print the report properly.

Normally, all this is of no consequence to you, the user, because we always "verify" the reports that are included with our software. The problem arises if you have written or modified any reports. If we supply you with an update to our software that includes database changes, then reports that you have written or modified that use any of the tables / fields that we have modified, those reports may not work properly because the database structure is different.

Therefore, if you have written / modified any reports using Crystal Reports, you should verify those reports against your (updated) database (or a copy of that database) <u>after</u> you open the database with the most recent version of our software. Opening a database with the most current version of our software, updates the database to the most current database structure, if necessary.

In most cases, you don't really have to perform the "verify" operation since the database changes we made probably didn't effect any of the tables you're using, at least not in a way that would require you to verify the database. But you don't have any good way of knowing whether the verify operation is or isn't required. And since it won't hurt anything if you do the verify operation even when it isn't necessary, and since the verify process should only take a couple of minutes anyway, the best bet is to just go ahead and verify the database.

Note that the Verify Database function reads the structure of the database that is identified on the Set Location form. That's why the Set Location form must be pointing at your real live database (or a copy of that database).

In some circumstances, Crystal performs an automatic verification of the database. If, for instance, you "Set the Location" as explained above, Crystal automatically performs the verify function when you click the Done button on the Set Location form.

Nonetheless, because of some idiosyncrasies in Crystal, I strongly encourage you, in all cases, to manually verify the database, as explained below, any time you suspect that the database structure may have changed since you last successfully completed the verify the database function.

To manually verify the database, from Crystal's Design form:

a) Click the Database | Verify Database option.

b) If it finds any differences in the database structure, Crystal displays a message to the effect that "The database file 'tblxxxx' has changed. Proceeding to fix up the report!". You will see this message once for each table that does not match Crystal's internal map of that table. Click the OK button (your only option) in response to that message.

If a report includes subreports, then, when you do a verify the database operation (from the main report), Crystal checks the main report and all the subreports to see if any of the tables have changed. If it finds any tables that have changed, it displays the same "……Proceeding to fix up the report!" message and you must click OK on all those messages. Next, you must rerun the verify operation again, until you get the "The database is up to date" message, once for the main report and once for each subreport. (For example, if a report includes 3 subreports, you will get the "The database is up to date" message four times).

It is of utmost importance that you run the Verify Database function, as many times as necessary, until you get the "The database is up to date" message, once for the main report and once for each subreport, without getting any intervening "……Proceeding to fix up the report!" messages. In order for this to happen, it may be necessary to run the verify operation multiple times.

During the Verify Database function, if the report includes one or more subreport, you may get a message(s) to the effect that a table can't be found, and asking you to supply the location of the table. This is the condition mentioned above in the Set Location discussion. Note that you normally won't get this message, but I have not been able to figure out when and why it shows up, but when it does, **just point to the same database that is specified on the Set Location form, and continue the Verify Database function. Never point to a database in a subreport that is different from the database that is pointed to in the main report.**

As soon as you complete the Verify Database operation successfully, click the Save icon so you don't have to repeat this process, even if Crystal crashes later.

Working With Specific Data Records

When working on a report in Crystal, you should work with a small set of data, maybe only one record, at least initially. This is a real time-saver, so understand how this process works. The idea is to make Crystal process a specific record(s) that you have entered or modified in our application, so that you know what to expect when you run the report in Crystal. An additional benefit is that it typically takes less time to read / process a small set of data than a large set of data.

You can enter a Selection Formula in Crystal by clicking Report | Edit Selection Formula | Record. For example, the following Selection Formula causes Crystal to process only Service Order Number 47:

{tblServiceOrderMaster.SONumber} = 47

The following Selection Formula causes Crystal to process only Accounts that have a Type of "Customer":

{tblAccounts.AccountType} = "Customer"

The Selection Formula in Figure 4. below causes Crystal to process only Invoice Number 299:

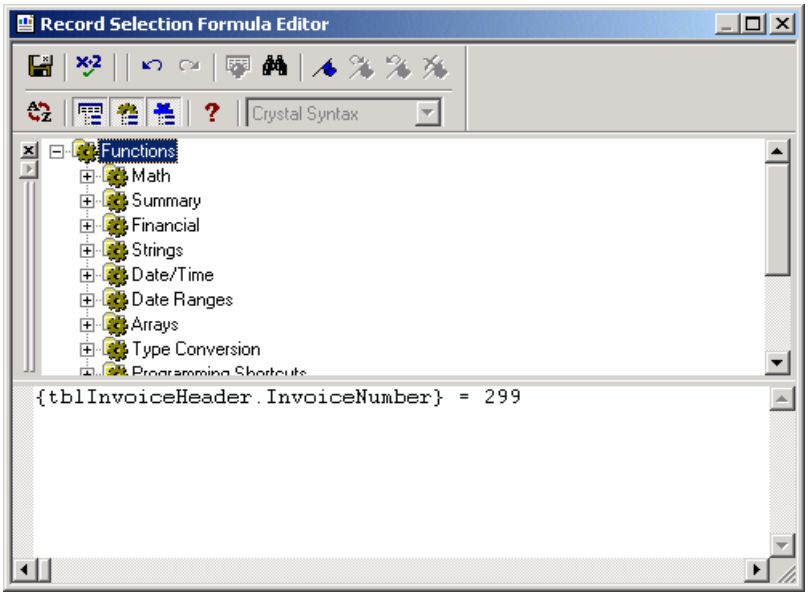

Figure 7. A Selection Formula to process only Invoice # 299

It is important to note that the Selection Formula you use in Crystal is moot when you run the report from our Tigerpaw Business Suite application. Our application creates its own Selection Formula or asks you to enter the appropriate information when you run a report.

Working With Subreports

A subreport is a report within a (primary) report. Subreports allow you to do things in Crystal that can't be done any other way. Per Crystal's Help text, there are four instances in which a subreport would typically be used:

1) To combine unrelated reports into a single report.
2) To coordinate data that cannot otherwise be linked.
3) To present different views of the same data within a single report.
4) To perform one-to-many lookups from a field that is not indexed on the lookup field.

The process of creating a subreport is similar to the process of creating a primary report. A subreport has most of the characteristics of a report. The main differences between a subreport and a primary report are that a subreport:

1) is inserted as an object into a primary report (via Insert | Subreport)
2) cannot stand on its own (although a subreport can be saved as a primary report)
3) can be placed in any report section and the entire subreport will print in that section
4) cannot contain another subreport.

If a report includes subreports, you can right-click the subreport and then Edit Subreport to display the subreport on the Design form. Working with a subreport, at that point, is the same process as working with the main report, and you can do almost everything in the subreport that you can do in the main report, the big exception being that subreports only go one level deep – subreports can't contain other subreports.

A frequent source of problems with subreports is getting the link between the main report and the subreport made correctly. If you are going to use subreports, make sure you understand how to link them back to the main report, if that's required. Refer to "Subreports" on the Contents tab and on the Index tab of Crystal's Help text for assistance.

It's typically not difficult to get the link made correctly between a primary report and a single subreport. The real linking problems that you'll encounter, at least in my experience, are often the result of adding the link for the second (or third) subreport, and, in the process, screwing up the link to the first subreport. At times I've had to delete all the links (not the subreports themselves, just the links) and create them all over again from scratch Linking a subreport to a primary report might not be much of a problem in Crystal's version 8+ but it was definitely a problem in Crystal version 6+.

Helpful hint: when you link a subreport to the primary report, Crystal creates a Selection Formula in the subreport, and I have often found it valuable to inspect the Selection Formula to ma ke sure that it says what I expect it to say. See "Working With Specific Data Records", above, and see Figures 8 – 11 below.

I have found it a very useful technique to include the same table in a subreport that is used as the main table in the primary report. For instance, assume that you have a primary report that processes service orders and that you have a subreport that processes the Items Serviced records that are attached to a service order. In this example, the primary report would contain the Service Order Master table, and the subreport would contain the Items Serviced table, and you could link the Service Order Number in the Service Order Master table in the primary report to the Service Order Number in the Items Serviced table in the subreport.

A better way to link the reports together, however, one that is less subject to problems, is to include the Service Order Master table in the subreport, and to link the Service Order Number in the Service Order Master table in the primary report, to the Service Order Number in the Service Order Master table in the subreport. And then in the subreport, link the Service Order Number in the Service Order Master table to the Service Order Number in the Items Serviced table. I promise you that this is a better way to link a subreport to the primary report, when that is required. See Figures 8 – 11 below for an example of the least error-prone way to link a subreport to a primary report.

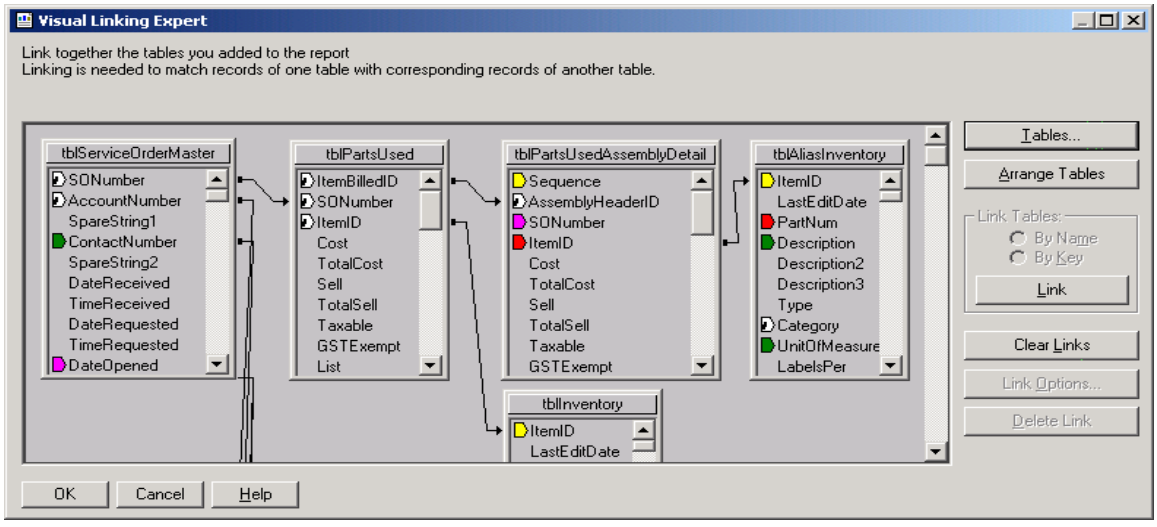

Figure 8. The primary report contains the Service Order Master table

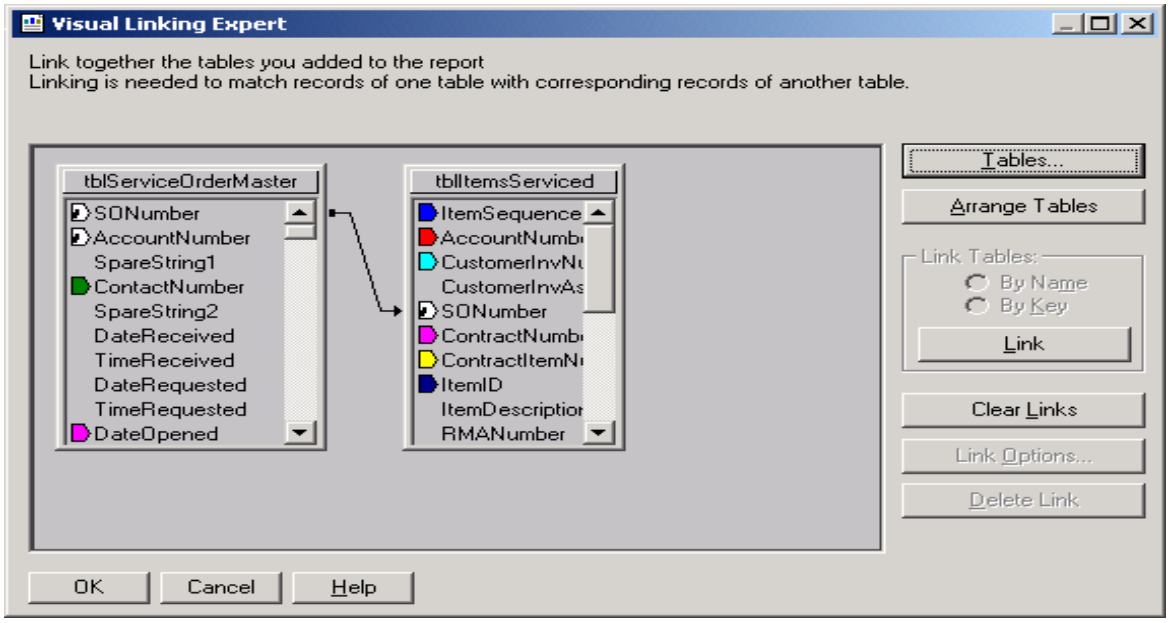

Figure 9. The subreport also contains the Service Order Master table

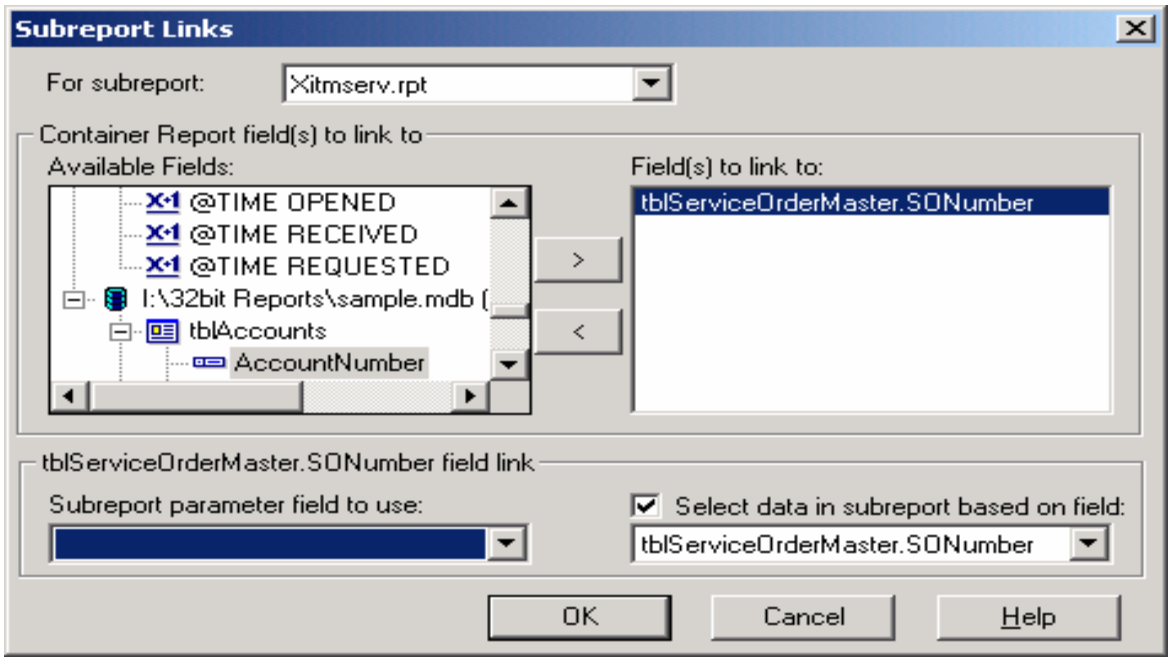

Figure 10. The subreport is linked to the primary report via the Service Order Number in the Service Order Master table (which is in both reports)

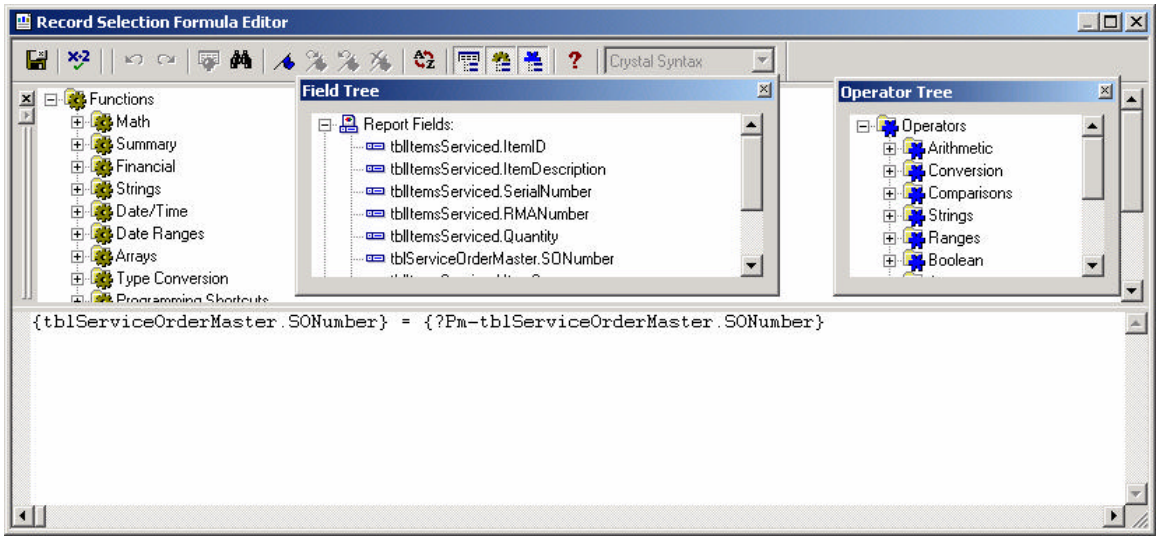

Figure 11. The Selection Formula in the subreport shows the proper link to the primary report

Store and Fetch Commands

The Store and Fetch commands can be used to pass data between a main report and a subreport, but (with version 8+) there is an easier way to do that (see "Shared Variables" below) so I don't recommend using Store and Fetch commands anymore.

If you do use those commands, be aware of the following problem – if you use the Store / Fetch commands and are testing the report in Crystal, make sure that the data you specify in the Report Selection Formula (if any) actually exists in the database. For instance, if the record selection formula calls for service order 47, make sure that there is a service order with that number in the database. If a service order with that number does not exist, Crystal may generate an error (probably a "Variable not found" message) and may even bomb out with an execution error; in some cases, you won't have any indication as to why Crystal bombed out.

See "Shared Variables…." Below.

Shared Variables – a replacement for the Store and Fetch Commands

I'm not sure when Crystal introduced Shared Variables (version 8?), but I'm glad they did. Shared Variables effectively replace the Store and Fetch commends, and the Shared Variable are much easier to use. With the Shared command all you have to do is declare that the variable is shared – of course you do that in every formula where the variable is used, both in the main report and in the subreport. For example:

Shared StringVar HasInvoiceBeenPrinted := "Yes":

or

Shared CurrencyVar InvoiceTotal;

Inserting Objects on a Report

In a Crystal Report, there are basically six kinds of fields you can print:

1) text fields
2) database fields
3) formulas
4) parameter fields - seldom used, but useful to input a parameter used in the report
5) special fields (such as the current date or a page number)
6) group names

All these fields can be inserted on a report via the Insert menu, and then clicking either Text Object or Field Object.

The fastest way to insert a text object on a report, however, is via the Insert Text Object tool button, with an icon with the letters "ab". After clicking the tool button, position the mouse wherever you want the text object to appear and click the mouse – the text object will appear, surrounded by a blue (highlight) border, with the cursor positioned to accept text. You may have to make the text box larger if you enter a lot of text in the text box.

When using a formula, you often don't want anything to display (print) as a result of the formula – to suppress the printing of a formula (or any other object), right-click the object and then click Format Field to display the Format Editor and then check the Suppress checkbox on the Common tab.

Eliminating Tables That Are Not Used In A Report

In general, the reports included with our software, as written by us, don't contain tables that are not used by the report – this makes the reports print slightly faster and makes them easier to develop.

One of the first things you should do when you start working on a report is to eliminate all the tables that you don't need for the report. This, of course, means you need some knowledge of the database tables - refer to http://www.tigerpawsoftware.com/download/dtps.htm for documentation on the table and field names, and table relationships, etc.

The report you're writing probably requires very few tables, maybe only 4 or 5 tables (seldom does a report contain 10+ tables. So before you do anything else, remove tables that are not required. It's easy to add a table back into a report, so it's no big deal if you happen to delete one that is required.

You eliminate tables via Database | Remove From Report, which displays the Remove from Report form. On that form, click the entry for the table you want to remove, and then click the Remove button. If you get the message "There are fields in the report from this file. Continue?" you probably want to click Cancel so you don't delete the field, at least until you determine for sure you don't need any of the fields from that table. In general, you should determine where in the report a field from that table is being used, then delete those references to the table, and then you can safely delete the table from the report.

Adding Tables to a Report

If the report you're working on does not include all the tables you need for the report, you can add the missing tables via Database | Add Database to Report, which displays the Data Explorer from which you specify the database that contains the tables that you want to add to the report. Before you do that, however, make sure that the report is already pointing at the correct database, as specified on the Set Location form – refer to "Setting The Location Of Your Database" above. In order to avoid a real mess, make sure that you

specify the exact same database when adding tables to the database as is specified on the Set Location form.

When you specify the database on the Choose Database File form, Crystal displays a list of the tables in that database. From that list, highlight the desired table and click the Select button.

Creating Links Between Tables

When writing Crystal reports for our version 9 software, you are responsible for creating your own links between the tables in the database – for help in understanding the relationships between tables in the database, refer to the table / field names and relationships documentation discussed earlier in this document.

Linking tables correctly is one of the major keys to getting a report to print properly. If the links are not correct, the report won't run correctly.

Do not let Crystal create the links between tables – specifically, make sure that the Auto-Smart Linking option is unchecked (not set) in the File Options table (as explained in "Options File Settings"), and never click the Smart Linking button on the Visual Linking Expert form (see below).

Links between tables can be added or removed from the Visual Linking Expert form (via Database | Visual Linking Expert). To create a link on the Visual Linking Expert form, position the mouse over the field in the table that you want to link from, then click, hold, and drag the mouse pointer and position it near the bottom of the appropriate field in the table that you want to link to, so that the mouse pointer turns into a "lightening-bolt arrow" and then release the mouse. Crystal will draw a link (line) pointing from the first field to the second field.

Although this next step is probably no longer necessary, previous versions of Crystal had a quirk that got me in the habit of manually setting the Index In Use field, as described below. After you create a link between two tables, the link will be highlighted (bolded), or you can bold it by clicking on the link. With the link bolded, click the Options button at the bottom of the Visual Linking Expert form to display the Link Options form. On that form, note the value in the field labeled "Index In Use" on the left side of the form – it probably has a value of "<no specific index>". This field can have several different values – some of which could cause the report to do things you don't want it to do.

Click the down-arrowhead button on the right side of the Index in Use field and note the options available. Although normally only two options are shown, there are four possible options:

a) Primary Key

b) The name of a specific field, such as SONumber or AccountNumber

c) <no specific index>

d) Something unintelligible like EMHX001-XMECC4.

If you have a choice, don't use "c" or "d" in the list above; make sure that the Index In Use field has a value of "Primary Key" or the name of a specific field (a or b. above).

To remove a link between fields, click on the arrow that devotes the link on the Visual Linking Expert form, to highlight it, and then press the Delete button.

The Format Section Expert

Right-click in the gray area on the far left side of the Design form to display a menu and then select Format Section to display the Section Expert form. This is a frequently used form; it is used primarily to force page breaks, reset the page number to zero, and to suppress sections. The real power of this is that the functions available from this form can be made to happen conditionally, based on the result of a formula that you write, and each section of a report has its own set of these conditional functions.

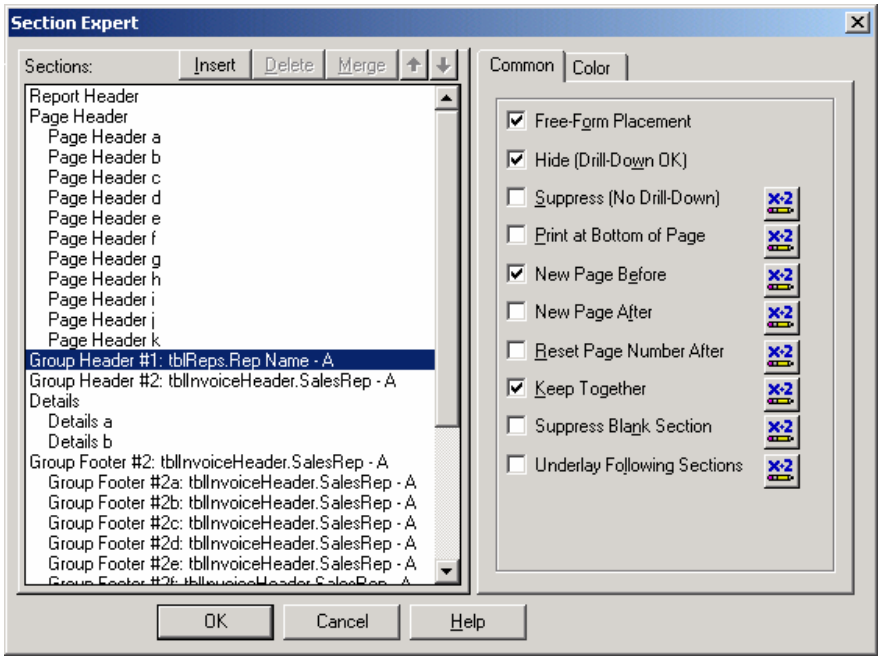

Figure 12. The Section Expert is a frequently used form in Crystal Reports

As you click on a section in the large white window on the left side of the form, Crystal displays the check box settings applicable to that section, on the right side of the form. In the example shown below, “Group Header #1”, which is triggered when the Rep Name changes, causes a “New Page Before” the Group Header is printed.

If the X-2 formula button to the right of a check box is blue, there is no formula behind the function. If the X-2 button to the right of a check box is red, then there is a formula behind the function. The formula can be entered or accessed by clicking the appropriate X-2 button. When the formula behind a check box evaluates to true, then the function is performed. When the formula behind a check box evaluates to false, then the function is not performed.

For example: assume that you have a report that prints invoices, and in a particular section of the report, say in Group Header 1a, you have some text that reads “This Invoice has been voided”. Obviously, you only want to print that section if the invoice has been voided (Status = Void). In all other cases, you want that section to be suppressed.

To accomplish this, in the Section Expert for Group Header 1a, you should have a formula behind the Suppress (No Drill Down) check box to test for a status “not equal to void”. If the formula evaluates to true, which means that the Invoice is not void, then the section will be suppressed. If the formula evaluates to false, which means that the invoice has been voided, then that section, with the “This Invoice has been voided” literal, will be printed. The applicable formula would look like this:

{tblInvoiceHeader.Status} <> "Void"

Note, if you want any of these conditions to always occur, simply check the appropriate check box. If, for example, you always want to skip to a new page before printing a section, then just check the "New Page Before" check box.

If you do not check the check box for a function, but instead use a formula, then the function will occur only if the formula behind the X-2 button for that function evaluates to a true condition.

Printer Considerations

Crystal allows you to designate that a given report is to be printed on a specific printer or on the "default printer" (as defined in the operating system). You should be aware that **the printed output of a report is somewhat printer specific – a given report that is printed on two different printers looks different. The difference is most notable on a laser printer versus a dot matrix printer.**

In addition, if you designate a specific printer for the report and then later change the setting to a different printer, the vertical and horizontal alignment often automatically changes (without warning) in Crystal's Design form, and, as a result, it also changes on the printed output. This can cause problems, especially if the report is to be printed on a preprinted form, one where vertical and horizontal alignments are "tight" and don't allow much room for misalignment.

The general rule is to use the "default printer" and "default options" whenever possible. However, if you are creating a report that is to be printed on a custom form with critical vertical and horizontal alignments, you should plan on printing the report on a specific make and model of printer, and designate that printer in that report (see below). If you are ever forced to print the report on a different printer, however, be aware that the output probably won't look the same.

To designate the printer that is to be used to print a report:

a) Click File | Printer Setup to display a Print Setup form.

b) On the Print Setup form, check either the Default Printer radio button or the Specific Printer radio button.

If you choose the default printer, you should also check the Default Options checkbox. If you use the default printer but don't have the Default Options checked, Crystal won't use the options that are set on the printer at print time - it won't, for instance, allow you to print to a different tray even though that is the setting on the printer.

If you choose to use a specific printer, you will have to specify the printer (from a list of printers that have been registered in the Windows operating system).

In most cases, when you change something (anything) in a report and then close the report, Crystal displays a message asking if you want to save you changes. Note that if the only things you changed were all on the Print Setup form, and you then close the report, you will not get the "save changes" message and your changes will not be saved.

Also be aware that Crystal is not truly "WYSIWYG". In print preview, adjacent fields sometimes overlap, but they print with proper spacing. You have to go by what the printed output looks like, not what the screen display looks like.

Reports for Dot Matrix Printers

If the report you're creating is going to be printed on a dot matrix printer, make sure you don't use True Type fonts in the report, since dot matrix printers can't print them and have to convert to a printable font – the results often leave much to be desired. Per Seagate Software, the makers of Crystal Reports, you'll get the best results on most dot matrix printers if you use a Courier 10 point (non-True Type) font.

Snap To Grid, Showing the Grid, Guidelines, and Rulers, etc.

Right-click in a white blank area of a report to display a menu with several useful options, including snap to grid, show the grid, show guidelines, and show rulers. These options are available on both the Design and the Preview mode.

Objects (text fields, database fields, formulas, etc.) can be "tied" to a guideline – this is very useful when trying to align objects in different sections so they are all aligned on the left or right side.

Experiment with the options on the right-click menu (from a white blank area of a report) and you will quickly see the usefulness of the various options.

Using Zoom

When working on a report that has tight vertical or horizontal alignment requirements, use Crystal's Zoom feature - click View | Zoom to display the Magnification form. A value of 400 is the highest possible magnification and is very useful for precision positioning of fields, lines, etc. For normal purposes, I typically use a magnification of 100.

Tips on Drawing Evenly Spaced Lines

Drawing lines on a report so that they are evenly spaced is relatively easy if you:

1. Use the Zoom feature from the View menu with a magnification factor of 400 (the maximum).
2. Remove all the Guideline Markers on the left side of the form for the section that you are working on – remove them from the bottom up in a section; this allows you to position fields anywhere you want in the section.
3. Display the Gridlines (right-click in a blank area of the design view to display a menu and select "Show Grid in Design"), and use the grid lines to help exactly position the lines you want drawn on the report. If you position a line exactly over the dots that mark a grid line, the dots turn white – then you can do the same thing to draw lines on other grid line (dots). You can change the distance between grid lines from File | Options to display the File Options form – the grid setting is on the Layout tab.

The grid lines can also be used to position fields so that they print evenly across the page – if you move fields up / down, you can position the tops of the fields on the dots of a grid line very exactly, by watching how the grid dots change from black to white as you move the field.

Aligning Objects

Crystal has some formatting idiosyncrasies that you should be aware of:

1. If you drag a database field and let it touch a Text Object, Crystal sometimes makes the text in the Text Object smaller (but does not indicate the change in the Text Point Size window on the form). Most, but not

all of the time, if you click on the Text Object, it will change back to the proper size. Also, if you position a database field so that it touches a Text field, Crystal combines the two fields – use the Undo key to fix this problem. The Undo key is a very useful key for this and many other reasons.

2. If you have objects or text boxes with different point sizes in a section, you can have difficulty getting the fields/text boxes to line up properly on the line. To align all the objects in a section at the top of the section, first align one of the objects properly, at the top of the section. Then select all the other objects in the section "simultaneously", by depressing the Shift key and then clicking on each object in turn. Each object is displayed with an enhanced border when the object is selected. Select the object that you previously positioned at the top of the section last – it's the last object selected that determines the alignment of all the objects when, after you have selected all the objects you want aligned, you right-click to display a menu and then select Align and then select Top. Note that you have other possible alignments, such as Bottom, Left, etc.

To get different objects lined up in a section, in some cases, I've had to remove the horizontal tab guides from the white ruler area, on the far left margin of a section. This allows you to position objects anywhere you want in a section, both vertically and horizontally, but you may have to move each object individually.

If you want to move the objects to the top of the section but they won't move up, the problem may be that one of the fields is "locked" (for some unknown reason) in that (vertical) position. First, right-click in the far left, dark gray margin of the Design tab to display a menu and then insert a new section. Drag the bottom of the new section down to make it obviously larger than other sections, so you won't forget to delete it later, after you're done with the section. Next, temporarily drag the left-most object in the section in which the objects are not lined up, to the new section and then drag the object back to the proper section. Hopefully, the tops of all the objects in the section are now properly aligned.

At times, to get objects in a section lined up properly, I've had to create a new section below the one in question, move all the objects to that section and then delete the (now empty) old section.

In extreme cases, in order to get objects to line up properly, I have had to delete and recreate text objects.

Making Objects the Same Size

Crystal now has a real easy way to make multiple objects the same size - first make one of the objects the size you want all the objects to be and then, with the Shift key depressed, click on each object, one at a time, selecting the object that you want the other objects to match size-wise last. Then right-click on one of the objects to display a menu and select Size. You can then select to make all the objects the same width, height, or both width and height.

Memo Fields

You can't use Access "Memo" (64kb) fields in formulas with an exception noted below. Memo fields, in fact, are not even displayed in the Formula Editor, but they are displayed on the Database tab of the Insert Fields form and on the Visual Linking Expert form. The bottom line is that you can't use Memo fields in a formula - among other things, this means that you can't test a Memo field to see if it is blank.

You can however, use a memo field in a formula to control printing an entire section or Group. For example, right-click the section/group to display a menu and then select "Format Section" to display the Selection Expert form (Common tab). On that form, click the blue "X-2" formula button to display the Format Formula Editor form that controls suppressing a section / group. On that form, you can manually enter an appropriate formula using a memo field in the formula such as

```
IsNull ({tblServiceOrderMaster.MemoField)
```

In this case, the section / group will be suppressed (not printed) if the Service Order Master Memo field is null (blank). Note that you can't pick the Memo field name from the list of field names on the Format Formula Editor form because the names of Memo fields do not display in that list – but you can manually type the field name into the formula.

Note also that if you use Memo fields in a formula, you do so at your own risk. It works as intended in the example explained above, but it is apparently an unsupported function and may not work in later versions.

Date Fields

As previously explained in the section above titled Report Options, you must setup Crystal to convert date and time fields to string fields (via File | Report Options). If the Convert Date-Time Field variable is not set to "To String", you will get errors when you run the report from within our application, usually an "Error 515, error in formula".

If you find that the Convert Date-Time Field variable is set to anything other than "To String", be aware that after you change it and then click OK to leave the Report Options form, Crystal automatically "verifies the database", (see "The 'Verify Database' Operation" above).

Because the Convert To String option is (must be) set for Date-Time fields, you must use a formula whenever you want to display a date or time field.

To display a date field, use Crystal's DTSToDate function, for example:

```
DTSToDate ({tblServiceOrderMaster.DateReceived})
```

To include a date in a formula that includes text characters, you must also use the ToText command:

```
"Closed Date: " + ToText (DTSToDate ({tblServiceOrderMaster.DateReceived}))
```

The DTSToDate command allows the system to display and print dates in the format used by the Windows operating system, as defined in the Regional Settings file in Window's Control Panel, effectively allowing a user to dictate the date format (mm/dd/yy versus yy/mm/dd, for example). To set the "Use Windows Default" for a date formula, right-click on the formula to display a menu and then select Format Field to display the Format Editor; from that form, click "Use Windows Default Format".

Crystal has many powerful commands to manipulate dates – enter "date" in the Index tab of the Help function for a list of different date functions.

Time Fields

As previously explained, all Date and Time fields must be interpreted (in Crystal) as 22 character string fields. Time fields, like date fields, have to be printed by means of a formula. The following is a formula I use to display time fields in AM/PM format. Per the comments in bold below, substitute the Time field you want displayed, for the field on the right side of the StringVar ConvertTime equation, replacing the {tblServiceOrderMaster.TimeReceived} field.

```
// Set the StringVar "ConvertTime" to the date/time field from which
//  you want to display time in AM/PM format
```

```
StringVar ConvertTime := {tblServiceOrderMaster.TimeReceived};

If (ConvertTime [12 to 13]) = "00"
  Then "12:" + ConvertTime [15 to 16] + " AM"
  Else (If (ToNumber (ConvertTime [12 to 13])) < 12
        Then ConvertTime [12 to 13] + ":" + ConvertTime [15 to 16] + " AM"
        Else (If (ConvertTime [12 to 13]) = "12"
              Then "12:" + ConvertTime [15 to 16] + " PM"
              Else ToText(((ToNumber(ConvertTime [12 to 13]) - 12)),0) + ":" + ConvertTime [15 to 16] + " PM"))
```

Note that Crystal has many powerful commands to manipulate time fields – enter "time" in the Index tab of the Help function for a list of different time functions.

Print/Print Preview Speed Tips

The following speed tips can greatly influence how long a report takes to print (or print preview), both when you are printing a report from Crystal or from our Tigerpaw Business Suite software.

Speed Tip #1

An example is the best way to explain this factor. Assume that you want to create a simple report that prints data from our Service Order database. The report is to include data fields from both the Service Order Master table and the Items Serviced tables. When you print the report in a live production environment, you intend to input the number of the service order you want printed.

Both the Service Order Master table and the Items Serviced table contain a service order number field and that's the field you use to link the two tables together (in the Visual Linking Expert, on the Database menu drop-down). Crystal allows you to make a link either from the Service Order Master table to the Items Serviced table, or from the Items Serviced table to the Service Order Master table. Either way works and I don't think that the way the link points (in this example) has any effect on print speed. For this discussion, assume that the link points from the Service Order Master table (the "from" table) to the Items Serviced table (the "to" table), as shown in the screen-shot of the Visual Linking Expert below .

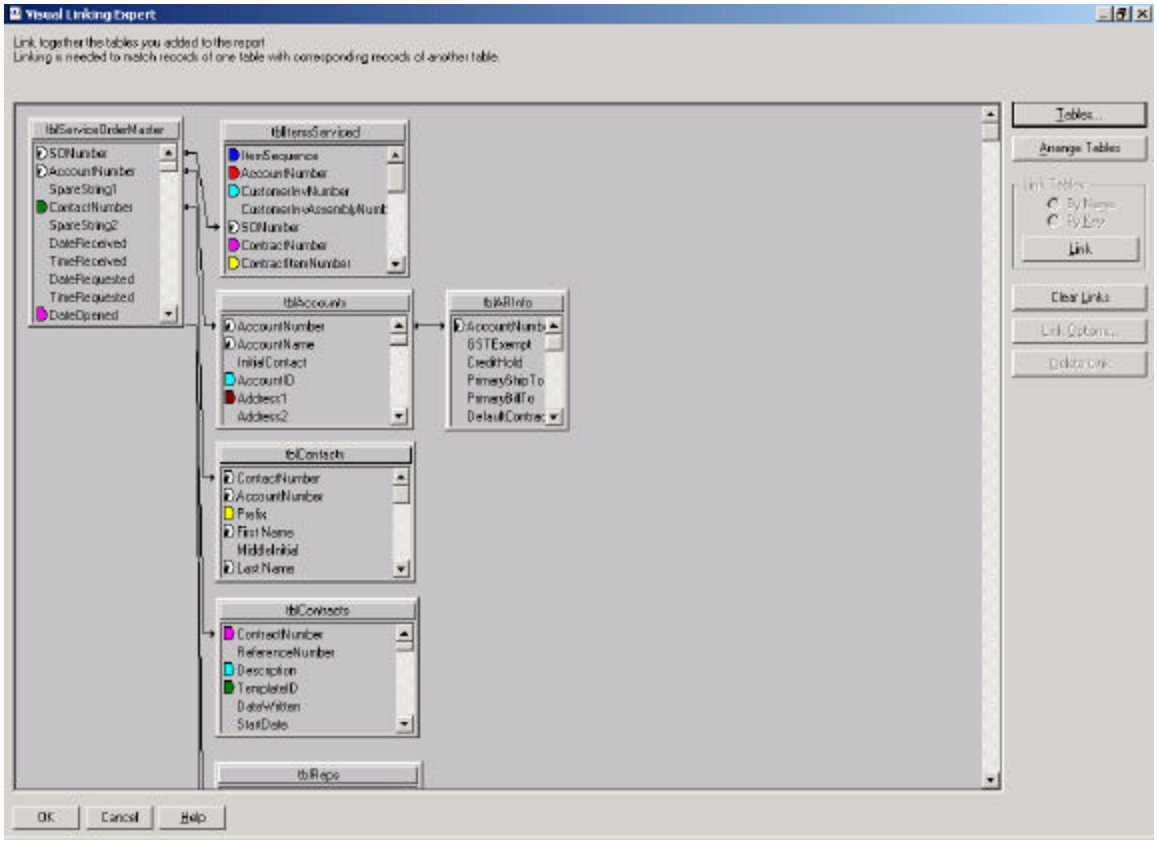

Figure 13. The Visual Linking Expert Form

Here's the important part: in formulas or input parameters when running the report from Crystal, or in Selection Formulas used in our Report Manager when running the report from our software, use the name of the "from" table, not the name of the "to" table.

For example, if you wanted to process just service order number 175 when you are running the report from Crystal, you should use this selection formula:

{tblServiceOrderMaster.SONumber} = 175

Do not use this selection formula:

(tblItemsServiced.SONumber} = 175

Using the "to" table , the Items Serviced table in this example, causes the report to print (or print preview) **much** slower (in some tests, it took as much as 800 times longer to print).

The same principle applies if you print the report from our software – make sure that the Selection Formula you use in TIGERPAW BUSINESS SUITE 9's Report Manager tests against the "from" table not the "to" table. In the example stated above, the Selection Formula you should use in the Report Manager is:

{tblServiceOrderMaster} = "<%S. O. Number>"

Do not use a selection formula like this:

{tblItemsServiced.SONumber} = "<%S. O. Number>"

Note: for a discussion of Selection Formulas in the Report Manager, refer to the Reports Digest mentioned previously - see "Other Documentation You Should have".

Speed Tip #2

If you are running a report from Crystal or from our TIGERPAW BUSINESS SUITE 9 program, and the database the report is using is open in Microsoft Access, the report will take at least twice as long to print / print preview.

Adding a Company Logo or other graphic to a report

To add a logo or other graphic image to a report:

1. Open the report
2. From the Insert menu, click the Picture option, to display the Open form
3. Next, from the Open form, browse to find and select the graphic image and then click Open, which causes Crystal to create a shadowed graphic image
4. Next, position the shadowed graphic on the report and left-click to place the image on the report. You will then be able to move or resize the image.

Section 2. Tips on Converting Custom Reports to the Tigerpaw Business Suite 9, 32-bit Database Structure

This section of the document is intended to help you convert reports from Crystal version 6.0+ (used with version 8 of our Tigerpaw Business Suite software) to Crystal version 8.0+ (used with our most current version, version 9).

Note that you must use Crystal version 8.0+, 32-bit (including version 8.5) to modify a report so that it will work with version 9 of the Tigerpaw Business Suite. Previous versions of Crystal will not work.

You can check the version from Crystal's Help menu – select About Seagate Crystal Reports to display an About… form. Check the middle of the second line and it should read "CRW32 **8**, n, n, nnn", which means "Crystal Reports, 32-bit, version 8".

In general, modifying a report to work with version 9 of our software is a simple process that should only take a couple of minutes per report. In essence, all you have to do is:

1. Open the report.
2. Using the Set Location function, point the report to a database that has been opened, at least once, with the most current version of Release 9 of the Tigerpaw Business Suite software (which means that the database has the most current database structure). The report may already be pointing at the proper database, but you need to make sure that's the case.
3. Perform the "Verify Database" operation so that Crystal reads and stores a map of the new database structure. This is really the main part of the conversion process.
4. After insuring that the report is using a Selection Formula to point to a known set of data, as explained in "Expediting the Report Development Process", test the report – refresh the data and see if any problems come up. Most reports will not have problems that need to be resolved.
5. Resolve problems with the modified report, if there are any.
6. After resolving all the problems, click Save and then click File | Close.
7. Test the report from our application – remember it's not until a report runs successfully from our application that it can be considered to be working properly. If it works from our application (and it should), then you're done. If it doesn't work from our application, open the report in Crystal and start debugging.

The reason you have to do all this is because Release 9 of the Tigerpaw Business Suite includes many changes to the database structure used in the Release 8. Most of those changes will not cause problems during the conversion process, but three conditions need to be considered:

1. Many new tables and / or fields were added to the database. Since those tables / fields were not previously available, they do not have an effect on the conversion of any of the reports.
2. A few tables and / or fields were eliminated from the database. With one exception, the tables / fields were not previously populated with data. You may have used one of these fields in a report, but there was never any data in that field to be printed - so eliminating those tables / fields from the database, worst case, is only a minor problem.
3. The names of a few tables and / or fields have changed. In some cases, only the field name has changed. In other cases we have moved the field to a different table, with or without a new field name. In all these cases, during the "verify the database" operation Crystal will not be able to find the (old) table.field and it will give you an opportunity to "map" that field to some other field.

As a result of the database changes, when you perform the verify the database function:

a) tables may disappear from the Visual Linking Expert form
b) links between tables may disappear from the Visual Linking Expert form
c) fields may disappear from the report
d) formulas may no longer work properly
e) group sort levels may become undefined

Basically, that's all that can go wrong, but remember, most of the time, nothing goes wrong - the report works as expected after you complete the Verify Database function.

The Text Point Size and Alignment Problems

Actually, there are two other related minor problems that often crop up. When converting from Crystal version 6.0+ to Crystal version 8.0+, Crystal often reduces the point size of most of the text fields, especially if the text field is bolded. It usually decreases the size by one point but sometimes even assigns a half point size (8.5, for instance) to the text field.

Even after you reset the desired point size set on all the objects in a section, the tops of the objects often are not properly aligned, which makes for a very sloppy-looking report. The makers of Crystal suggested that I try converting a report from Crystal 6.0+ to 7.0+ and then convert the report from 7.0+ to 8.0+; they also suggested that I leave unchecked the "7.0 Text Compatibility" checkbox on the Options | Reporting tab. I don't know if either of these suggestions would alleviate the problem.

Getting the objects in a section to line up can be a real chore – see "Aligning Objects", in Section 1 above.

First Things First – Before You Convert a Report

I'll admit that I no longer do the things mentioned in this section, because I've converted so many reports that I know what to look for without taking screen shots, etc., but I do recommend that you do the things mentioned below if you only have a few reports to convert.

First, make sure that you have a backup copy of the report; it's unlikely that you'll need it, but it's cheap insurance. The fastest way to do this is to just take a copy of all the reports before you start the conversion process.

Next, print a sample of the report from our version 8 software so that you can later compare the format of the report with a sample of the report that you will produce from our version 9 software.

Next, making sure that you are using the correct version of Crystal (version 8.0+, 32-bit, including version 8.5), open the report and get printed screen shots (see below for instructions) of the following three forms, before you do anything else.

1. The Design form (via the Design tab, if necessary). Make sure the display is set to show the field names, not just "XXXX"s (see "Options File Settings").
2. The Visual Linking form (from Database | Visual Linking).
3. The Record Sort Order form (from Reports | Sort Records).

The screen shots and the printed sample report make the conversion process as painless as possible – they make it relatively easy to spot problems. You compare them against their counterparts after you have completed the verify database function, looking for any differences. Once you have the printed screen shots, then you can begin the actual process of converting the report – see below.

Printing a Screen Shot

To get a printout of the image of a screen, first copy the image to the Windows Clipboard by pressing the "Alt" and the "Print Screen" (or "Print Scrn") keys simultaneously. You can then paste the screen-shot from the clipboard into a Word Processing document, an email message, or an image-processing program such as "Paint", and then, if desired, you can produce a printout of the screen-shot. Since I have Microsoft Outlook running all the time, I create an email message, which I never send, but instead I just print to produce the printed output:

1. Display the screen that you want to get a printout of.
2. Press the Alt and Print Screen keys simultaneously.
3. Click to give control to Microsoft Outlook.
4. Click New to start an email message.
5. Press the "Ctrl" and the "v" keys simultaneously to copy the contents of the clipboard into the email message.
6. Click Outlook's Print icon to print the email message.
7. Click the X in the top right corner to Close (without saving) the email.

Converting a Report to Our 32-bit Database Structure

After printing the "before conversion" materials mentioned above, you need to perform the "Set Location" and "Verify the Database" functions, as explained in the first section of this document. The Set Location function tells Crystal where your database is located. Make sure that you point it at a database that has been opened, at least once, with version 9 of our software. Note that the report may already be pointing at the proper database – but check to make sure or you won't be able to count on the verify database function working properly. If necessary, refer to the first section of this document for instructions.

During the verify-database process, if Crystal finds any differences, and it will find differences in almost every report, it displays a message(s) to that effect

> "Table xxxxxx has changed. Proceed to fix up the report?"

You must respond "Yes" to all those messages to allow Crystal to generate a new internal map of the 32-bit database structure.

Mapping table.field Names that Have Been Changed or Deleted, to New table.field Names

During the Verify Database function, if Crystal detects that a table.field name is no longer in the database (because it has been changed or deleted), it displays a Map Fields form and allows you to "map" the (no longer used) table.field names to some other field in the database.

There is a list of table.field name changes at the end of this section – that list will help you map no-longer-used table.field names to their new table.field name equivalents in Release 9. Referring to that list and Figure 14 below, to map an old table.field name to its Release 9 equivalent, first of all, ignore the two "Mapped fields" windows at the bottom of the form – the mapping is done using only the top two "Unmapped fields" windows. Crystal highlights an old, no longer used table.field name in the window on the left and it's your job to find and click the equivalent new table.field name in the window on the right (and as listed in the list at the end of this section) and then click the Map button, which causes Crystal to replace the old field name with the new field name, everywhere applicable in the report. Click OK at the

bottom of the form and Crystal continues with the Verify Database function. Note that if you do not map a replacement table.field name, when you click OK, Crystal deletes the old table.field name everywhere its used in the report, and that causes problems that will have to be resolved manually.

One field deserves special attention – it's {tblServiceOrderMaster.Location} which is where we store the field labeled "Directions to Service Location" on the Account Info tab of a service order. The field is populated from the Instructions tab of the Account view when you create a service order. We changed this "Location" field from a 255 character text field in our Release 8 software and made it a 64K Memo field in Release 9. Note that we did not change the name of the field but we did change both the field size and type.

During the Verify Database operation, if you used {tblServiceOrderMaster.Location} in a report, Crystal displays the Map Fields form shown below, even though that field name was not changed. Unfortunately, even if you uncheck the Match Type checkbox on the Map Fields form, Crystal often does not display the {tblServiceOrderMaster.Location} field in the window on the right side, which would allow you to map to that field. Sometimes Crystal displays the field, other times it doesn't, and I have not been able to determine the rules it uses to display or not display the field name. At any rate, if you can't map that field, when you click OK to close the Map Fields form, Crystal deletes all references to that field in the report, and you have to manually re-enter that field on the report, everywhere it's supposed to be, just as if you were just now adding that field to the report for the first time.

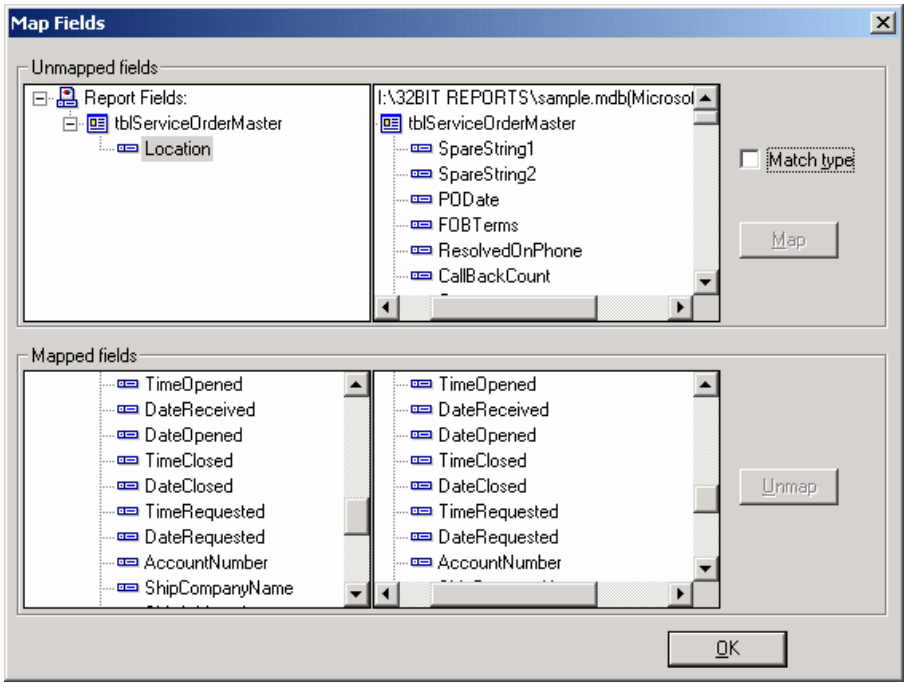

Figure 14. The Map Fields form is used to map new table.field names to table.field names that have changed or been deleted

Per instructions in the first section of this document, do the Verify Database function as many times as necessary, until you get the "Database is up to date", once for the main report and once for each subreport, without any intervening messages that a "Table XXX has changed. Proceed to fix up the report?", and without any intervening field Mapping issues . Then save the report so you don't have to repeat the process, even if you have other problems later.

After completing the Verify Database function, make sure you working with a known set of data, as was discussed in "Expediting the Report Development Process", and then refresh the data (via the yellow

lightening bolt icon), and see what happens. In most cases, everything will work as expected (with the exceptions noted in "The Text Point Size and Alignment Problems", above).

Creating Links that Disappeared from the Converted Report

When you refresh the data, if any of the links between tables disappeared (because table.field names changed), the system will display a message to the effect that "XXX table could not be linked". To resolve that problem, you will have to:

1) refer to the "before conversion" screen shot of the Visual Linking form to determine the field that was previously used to link the tables
2) determine the replacement field name from the table below
3) use that field to create a new link between the tables

If you don't know the mechanics of creating links between tables, refer to the Crystal Tips section of this document (see Creating Links Between Tables in the Database).

Restoring Fields That Disappeared From The Converted Report

As previously explained, the Verify Database will prompt you to map a table.field name that no longer exists, to some valid table.field name. If you don't map the field to a valid table.field name, then Crystal deletes the field from the report.

If you notice that the converted report has some "holes" (typically a text object without a corresponding database field), you can use the "before conversion" Design form screen-shot to determine the name of the field that got deleted, and then find the replacement field in the list below, and insert the proper table / field into the report.

Restoring Group Sort Fields that Disappeared from the Converted Report

When you refresh the data, if any of the Group Sort fields disappeared (due to table.field name changes), you will get an appropriate error message. Or, before you refresh the data, you can click Report | Sort Records to display the Record Sort Order form. Check the box on the right side to make sure that all the Group control breaks were converted properly. Any Group control breaks that have a question mark ("?") instead of a field name did not convert, because you didn't map a no-longer used table.field name to a new field.

To change the field referred to by a Group Sort record, first determine the correct field from the table below. Next, right-click in the gray area on the left side of the appropriate Group section and then select the Change Group option. On the Change Group Options form, browse to find and select the correct field and then click OK.

Fixing Formulas That Don't Work In The Converted Report

If a formula in a report uses a table.field name that has been changed or deleted, the formula will not work properly. The easiest way to detect formulas that don't work is to simply refresh the report (via the yellow lightening bolt). If a formula doesn't work, it will be displayed on the Formula Editor form with an appropriate error message. To fix the problem, you must replace the no-longer valid table.field name with the valid table.field name – refer to the list below.

After fixing a formula that didn't previously work, refresh the report again and Crystal will display the next formula that doesn't work, if there are any left, or it will run the report.

Tables and Fields that Have Been Deleted From the Database

Several fields have been deleted from the database. They were never populated with data in previous versions and it's unlikely that you used any of those fields in any of your reports. Even if you did use one of those fields in a report, worst case, it's a minor problem.

In the list of Old / New Field Names below, if the New Field Name is "THIS FIELD HAS BEEN DELETED" then the field listed in the Old Field Name column has been deleted from the database.

During the "verify the database" operation, Crystal will detect that the table.field name (of a deleted field) is no longer in the database, and the system will give you an opportunity to "map" that field to some other field. Since these fields have been deleted and there are no fields to map to, simply click the OK button on the Map form and Crystal will close the form and eliminate the field(s) from the report. Note that you will have to manually remove the text label used on the report, if there is one, that identified the field(s) that have been deleted.

When Crystal removes a (deleted) field from the report, formulas that used that field will no longer work, and group section breaks that used those fields will no longer work.

The table tblClipped is no longer used and has been deleted. Previously, that table was used to store the Account Numbers of Accounts that a Rep had put in a special list called the Clip List. The table was deleted because Account Groups can be used to produce lists like the Clip List.

The Security subsystem has been completely rewritten in Release 9 and the Security table has been completely redesigned. Prior to Release 9, the Security table was named {tblSecurity}. In Release 9, the table is named {tblSecurityGroupAssignedPrivileges}. See the Crystal report Security.rpt for details.

The Transfer sub-system has been substantially rewritten and the table structure has been changed to allow multiple items to be transferred on a Transfer ticket. Prior to Release 9, there was one Transfer table, {tblTransfers}. In Release 9, there are two Transfer tables: {tblTransfers} and {tblTransferDetail} and {tblTransferSerialNumbers}. See the Crystal report Transfers.rpt for details.

Table.Field Names That Have Changed / Been Deleted

The following is a list of table.field names that have been changed or deleted. Note that the fields marked with an asterisk have been moved to a different table:

Old Field Name	New Field Name
{tblActivities.RepNumber}	{tblActivities.TaskRepNumber}
{tblActivities.MailFromRepType}	{tblActivities.EmailFromRepType}
{tblActivities.MailFromRepNumber}	{tblActivities.EMailFromRepNumber}
{tblARHeader.TaxCode}	{tblARHeader.SalesTaxCode}
{tblARHeader.RetainageDate}	THIS FIELD HAS BEEN DELETED
*{tblARInfo.DirectionsToLocation}	{tblAccounts.DirectionsToLocation}
{tblARInfo.LaborCode}	THIS FIELD HAS BEEN DELETED
*{tblARInfo.ServiceZone}	{tblAccounts.ServiceZone}

{tblAltBillTo.AccountNum}	{tblAltBillTo.AccountNumber}
{tblAltShipTo.AccountNum}	{tblAltShipTo.AccountNumber}
{tblGLHeader.FiscalPerios}	{tblGLHeader.FiscalPeriod}
{tblQuotes.GLIncomeCode}	THIS FIELD HAS BEEN DELETED
{tblQuotes.GLCostCode}	THIS FIELD HAS BEEN DELETED
{tblServiceOrderMaster.Location}	Refer to the special note on this field in the section titled “Mapping table.field Names that Have Been Changed or Deleted, to New table.field Names”

Section 3. Useful Formulas

1. Date Field formulas:

Because the Convert to String option must be set for Date-Time fields, you must use a formula whenever you want to display a date or time field. The easiest way to display a date field is by using Crystal's DTSToDate function, for example:

DTSToDate ({tblServiceOrderMaster.DateReceived})

To include a date in a formula that includes text characters, you must use the ToText command. For example:

"Closed Date: " + ToText (DTSToDate ({tblServiceOrderMaster.DateReceived}))

2. A generalized formula to display a date from a 22 character date/time string; the date is displayed in the format:

Sunday January 6, 2000

```
StringVar ConvertDate;

// Replace the date field in the next line with the name of the date field you want converted.

ConvertDate := {tblInvoiceHeader.InvoiceDate};

// Replace the wording in the following line or delete the line.

"Invoices for:  " +

["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"] [DayOfWeek
(DTSToDate (ConvertDate))]
+ "  -  "
+ (If ConvertDate [6 to 7] = "01" then "January"
     else If ConvertDate [6 to 7] = "02" then "February"
     else If ConvertDate [6 to 7] = "03" then "March"
     else If ConvertDate [6 to 7] = "04" then "April"
     else If ConvertDate [6 to 7] = "05" then "May"
     else If ConvertDate [6 to 7] = "06" then "June"
     else If ConvertDate [6 to 7] = "07" then "July"
     else If ConvertDate [6 to 7] = "08" then "August"
     else If ConvertDate [6 to 7] = "09" then "September"
     else If ConvertDate [6 to 7] = "10" then "October"
     else If ConvertDate [6 to 7] = "11" then "November"
     else If ConvertDate [6 to 7] = "12" then "December")
+ "  "
```

```
+ (If ConvertDate [9] = "0" then ConvertDate [10] else ConvertDate [9 to 10])
+ ",  " + ConvertDate [1 to 4]
```

3. The following is a formula that displays time fields in AM/PM format, by converting the appropriate part of the standard 22 character Date-Time field:

```
// Set the StringVar "ConvertTime" to the date/time field from which
//  you want to display time in AM/PM format


StringVar ConvertTime := {tblServiceOrderMaster.TimeReceived};

If (ConvertTime [12 to 13]) = "00"
  Then "12:" + ConvertTime [15 to 16] + " AM"
  Else (If (ToNumber (ConvertTime [12 to 13])) < 12
          Then ConvertTime [12 to 13] + ":" + ConvertTime [15 to 16] + " AM"
          Else (If (ConvertTime [12 to 13]) = "12"
                  Then "12:" + ConvertTime [15 to 16] + " PM"
                  Else ToText(((ToNumber(ConvertTime [12 to 13]) - 12)),0) + ":" + ConvertTime [15 to
16] + " PM"))
```

4. A formula to determine if a Contract has expired – it prints “Remaining” data, based on the Contract Basis field:

```
// Has the Contract Expired? Check the Expiration Date and then check the Basis field
//       (T = Time, H = Hourly, M = Metered, P = Prepaid, U = Units, I = Incidents)
//   Modified 8/15/01 to indicate that the Contract has / has not been renewed.

(If ((DTSToDate ({tblContracts.ExpirationDate}) - Today) <= 0) then "Note: This Contract has expired -
the Expiration date has passed"
    else If {tblContracts.Basis} = "T"
            then (If ((DTSToDate ({tblContracts.ExpirationDate}) - Today) <= 0)
                    then "Note: This Contract has expired - there is no Time Remaining"
                    else "")
            else If {tblContracts.Basis} = "H"
                    then (If IsNull ({tblContracts.OverUnderIndicator}) or
({tblContracts.OverUnderIndicator} = "+")
                            then (If (({tblContracts.HoursRemaining} = 0) and
({tblContracts.MinutesRemaining} = 0))
                                    then "Note: This Contract has expired - there are no Hours Remaining"
                                    else "")
                            else (If (({tblContracts.HoursRemaining} = 0) and
({tblContracts.MinutesRemaining} = 0))
                                    then ""
                                    else "Note: This Contract has expired - Hours Remaining is
negative"))
                    else If {tblContracts.Basis} = "M"
                            then (If ({tblContracts.MeterRemaining} <= 0)
                                    then "Note: This Contract has expired - there are no Meter Clicks
Remaining"
```

```
                                    else "")
                            else If {tblContracts.Basis} = "P"
                                    then (If ({tblContracts.DollarsRemaining} <= 0)
                                            then "Note: This Contract has expired - there is no Credit Amount
Remaining"
                                            else "")
                                    else If {tblContracts.Basis} = "U"
                                            then (If ({tblContracts.UnitsRemaining} <= 0)
                                                    then "Note: This Contract has expired - there are no
Units Remaining"
                                                    else "")
                                            else If {tblContracts.Basis} = "I"
                                                    then (If ({tblContracts.UnitsRemaining} <= 0)
                                                            then "Note: This Contract has expired - there are
no Incidents Remaining"
                                                            else "")
                                                    else "Error - call Tigerpaw Software")
+ "  " + (If IsNull ({tblContracts.RenewContractNumber})
  then "The Contract has not yet been renewed."
  else "The Contract has been renewed by Contract #  " + ToText
(({tblContracts.RenewContractNumber}), 0, "") + ".")
```

5. A formula to add Billable / Not Billable time (Hours and Minutes) and the Amount, and Contract Chargeable / Not Contract Chargeable Time (Hours and Minutes) and the Amount. For an example of printing these fields, see formulas 6 – 10 below:

```
// Add Billable or Not Billable Hours & Minutes, and Amount.

WhilePrintingRecords;

NumberVar Minutes;
CurrencyVar Amount;

CurrencyVar L1BillAmount;
CurrencyVar L1NoBillAmount;
CurrencyVar L1ContractChargeAmount;
CurrencyVar L1NoContractChargeAmount;
NumberVar L1BillMinutes;
NumberVar L1NoBillMinutes;
NumberVar L1ContractChargeMinutes;
NumberVar L1NoContractChargeMinutes;

// Calculate Minutes

Minutes := ({tblSOLog.LogHours} * 60) + {tblSOLog.LogMinutes};

// Add to Billable or Not Billable based on the setting in the Time Log

If {tblSOLog.Billable} then (L1BillAmount := L1BillAmount + Amount;
                                L1BillMinutes := L1BillMinutes + Minutes)
                        else (L1NoBillAmount := L1NoBillAmount + Amount;
                                L1NoBillMinutes := L1NoBillMinutes + Minutes);
```

```
// Add to Contract Chargeable or Not Contract Chargeable based on the setting in the Time Log

If {tblSOLog.Chargable} then (L1ContractChargeAmount := L1ContractChargeAmount + Amount;
                                L1ContractChargeMinutes := L1ContractChargeMinutes + Minutes)
                        else (L1NoContractChargeAmount := L1NoContractChargeAmount + Amount;
                                L1NoContractChargeMinutes := L1NoContractChargeMinutes + Minutes)
```

6. A formula to print the L1BillMinutes calculated in Formula # 5 above:

```
// Have totaled L1BillMinutes in the formula ADD TO LEVEL 1. Convert the Minutes
//      to Hours and Minutes and print.

WhilePrintingRecords;

NumberVar L1BillMinutes;
NumberVar PrintHours;
NumberVar PrintMinutes;

PrintHours := Truncate ((L1BillMinutes) / 60);
PrintMinutes := Remainder ((L1BillMinutes), 60);

// Test for zero Hours and Minutes. Put in 2 digits if 9 Minutes or less (add a zero).

If ((PrintHours = 0) and (PrintMinutes = 0)) then "0:00"
            else ToText ((PrintHours), 0) + ":"
                + (If ((PrintMinutes) > 9) then ToText ((PrintMinutes), 0)
                    else "0" + ToText ((PrintMinutes), 0))
```

The End…..