

SAP Crystal Solutions
SAP Crystal Reports and SAP Crystal Server

Step by Step guide: Formulas With SAP Crystal Reports 2016





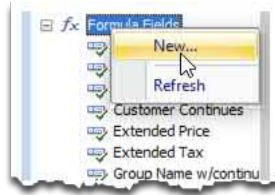
It won't be too long before you encounter a situation where the fields supplied by your report's data source don't fully meet all your custom reporting requirements. For example, you may need to create a numeric value that adds sales tax to an order amount, even if a sales tax rate isn't in the data source.

While the number of days it took to ship an order isn't in the data source, you may need to determine it by examining order and ship dates. Or, you may wish to navigate a complex set of sales rules to determine your sales team's end-of-year bonus based on such factors as a sales person's years of service, total sales for the year, and overall discount that the sales person gave customers.

These requirements, and many more like them, can be solved with use of SAP Crystal Reports formulas. A formula is a custom calculation, created with Crystal Reports comprehensive formula language. Formulas appear in their own category in the Field Explorer, ready to be placed on the report, and formatted, just like a standard data source field.

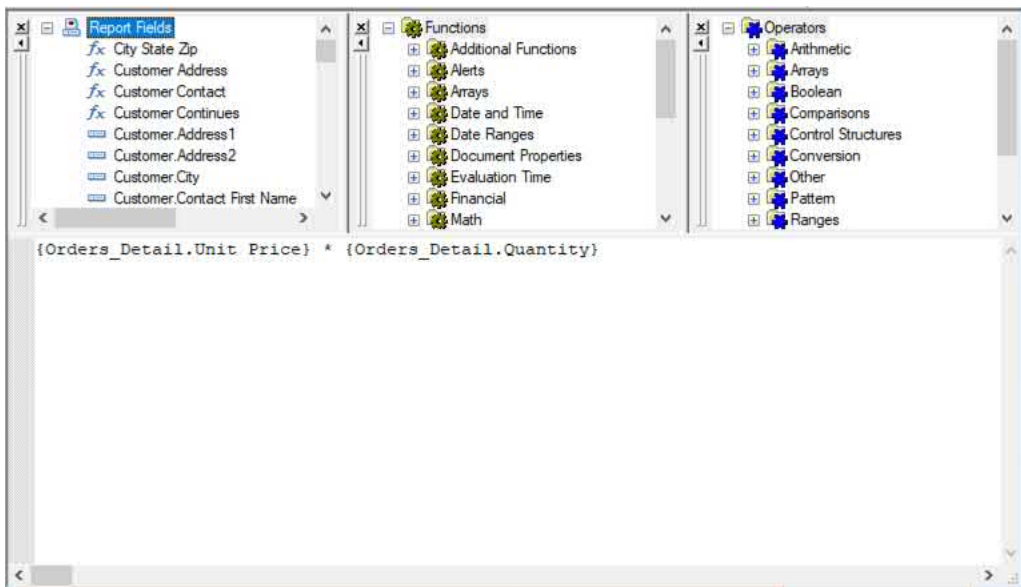
CREATING A FORMULA

While there are several ways of creating formulas in SAP Crystal Reports, the quickest and most straightforward is to RIGHT-CLICK on the Formula Fields category in the Field Explorer. Select New from the context menu.



You are prompted for the formula name. Choose a descriptive name for the formula that doesn't duplicate an existing formula (and, while you can create a formula that duplicates the name of a data source field, this isn't a best practice – there may be confusion between data source and formula fields with identical names). The Formula Editor, a text editor designed for Crystal Reports formulas, will appear inside the encompassing Formula Workshop dialog box (more information on the Formula Workshop appears later in this white paper).

The Formula Editor consists of four general areas:



- **Fields Box:** A list of existing fields already in use on the report appear within the Report Fields category. All available data source fields appear within another node of the Fields Box denoted by the name of the data source.
- **Functions Box:** Built-in formula language functions appear in this box, denoted by function category. Expand a category to see functions available within it. Functions appear with the values (sometimes referred to as arguments) that you may supply to the functions within parentheses. Functions provide various types of helpful features, such as converting from one data type to another or performing sophisticated date math.

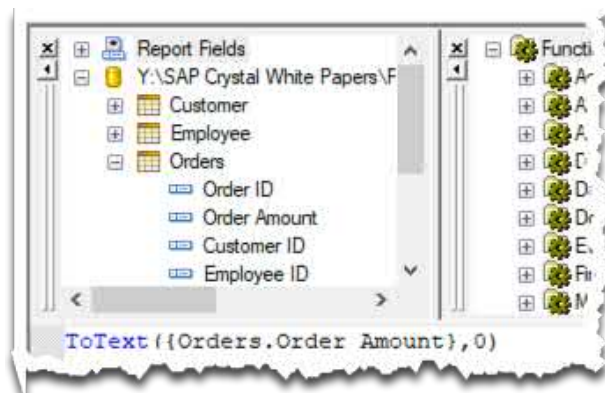


- **Operators Box:** Built-in formula language operators appear in this box, denoted by operator category. Expand a category to see operators available within it. While it's not always clear how an operator is different from a function, operators typically involve capabilities to perform addition, subtraction, less than/greater than comparison, and so forth.

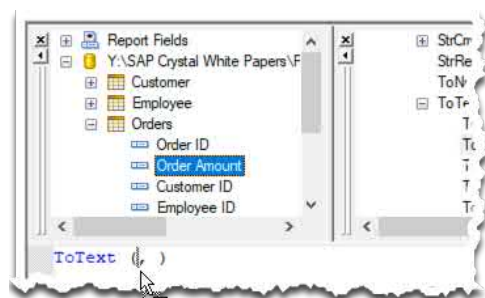
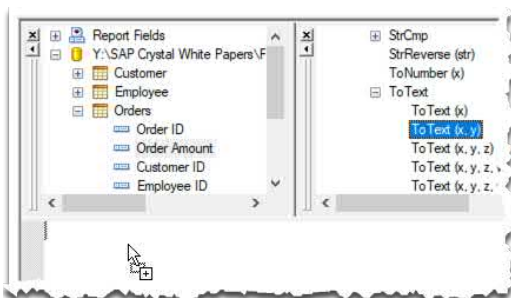
- **Formula Text Area:** This larger white area encompasses the entire bottom of the Formula Editor. Here, you may simply type in formula language fields, functions and operators directly. You may also drag and drop from the other three boxes into this area, or double-click items from the other three boxes to add them to the formula text area.

There are several ways of creating the actual formula text in the formula text area. For example, if you are creating a simple formula to convert the Order Amount field (from the Orders table) to a string data type with no decimals shown:

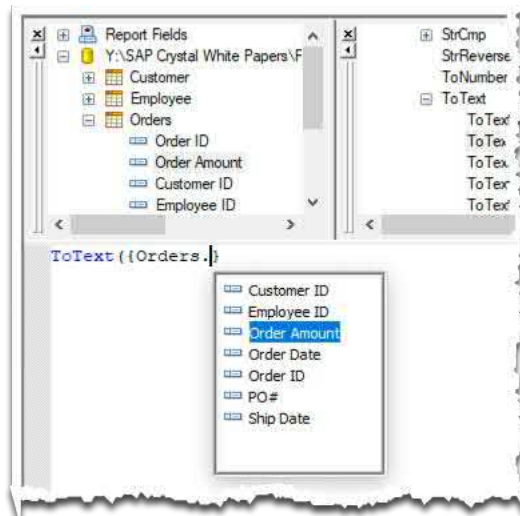
- **Type the formula directly:** When adding a function, enclose arguments in parentheses and separate each with a comma. When referencing a data source field, make sure to surround the field name with curly braces, and separate the table name from the field name with a period. This approach is the fastest – once you become familiar with the Crystal Reports formula language, you'll probably just type most of the formula in manually (perhaps making some use of DOUBLE-CLICK or drag-and-drop options discussed next).




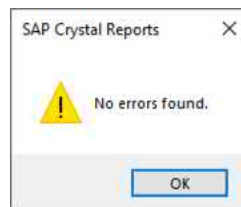
- **Double-click or drag and drop:** Expand the Strings category in the Functions box. Within the Strings category, expand ToText. DOUBLE-CLICK the two-argument version `ToText (x, y)`. You may also drag the function from the functions box to the formula text area. With the cursor appearing after the opening parenthesis, DOUBLE-CLICK on Order Amount within the Orders table from the Fields Box. You may also drag the field from the fields box and drop it after the opening parenthesis. The database field will appear in the formula text area with proper syntax (curly braces surrounding the field, and the table name separated from the field name by a period). Using the right-arrow or mouse, position the cursor after the comma. Type the number zero as the second argument to the ToText function (deleting any trailing spaces is optional).



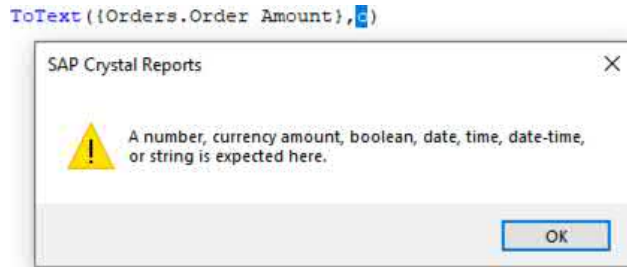
- Use the auto-complete feature: When you are first learning Crystal Reports formulas, you may prefer to use auto-complete. It may not be the fastest approach, but will help familiarize you with language options. Press CTRL-SPACE. This will enable auto-complete mode for functions. A small dialog box will appear below the cursor showing all available functions. Type the first few characters of ToText – notice the dialog box narrows available choices. Using the mouse, or a combination of arrow and TAB or ENTER keys, select ToText. Then, type an open parenthesis. With the cursor positioned after the opening parenthesis, type an opening curly brace { character (the shift portion of a left square bracket). This will enable auto-complete mode for fields. A small dialog box will appear below the cursor showing all available fields from the Fields Box. You may either DOWN-ARROW in the dialog box until you find the Orders table or type the first few characters of “Orders.” Use the DOWN-ARROW key or your mouse to select the table and then press TAB, ENTER or DOUBLE-CLICK to expand it. Use a similar procedure to select Order Amount from within the Orders table. You may also type freely as necessary (for example, continue typing the remainder of the formula by typing a comma, followed by the number zero and a closing parenthesis).




Once you have finished creating the formula, you will probably want to check the formula's syntax (the combination of properly-spelled field names, curly brace characters, proper operators and so forth). Click the Check Syntax button  in the toolbar above the Formula Editor, or press the ALT-C keyboard combination. If Crystal Reports encounters no syntax errors, you'll be so notified.



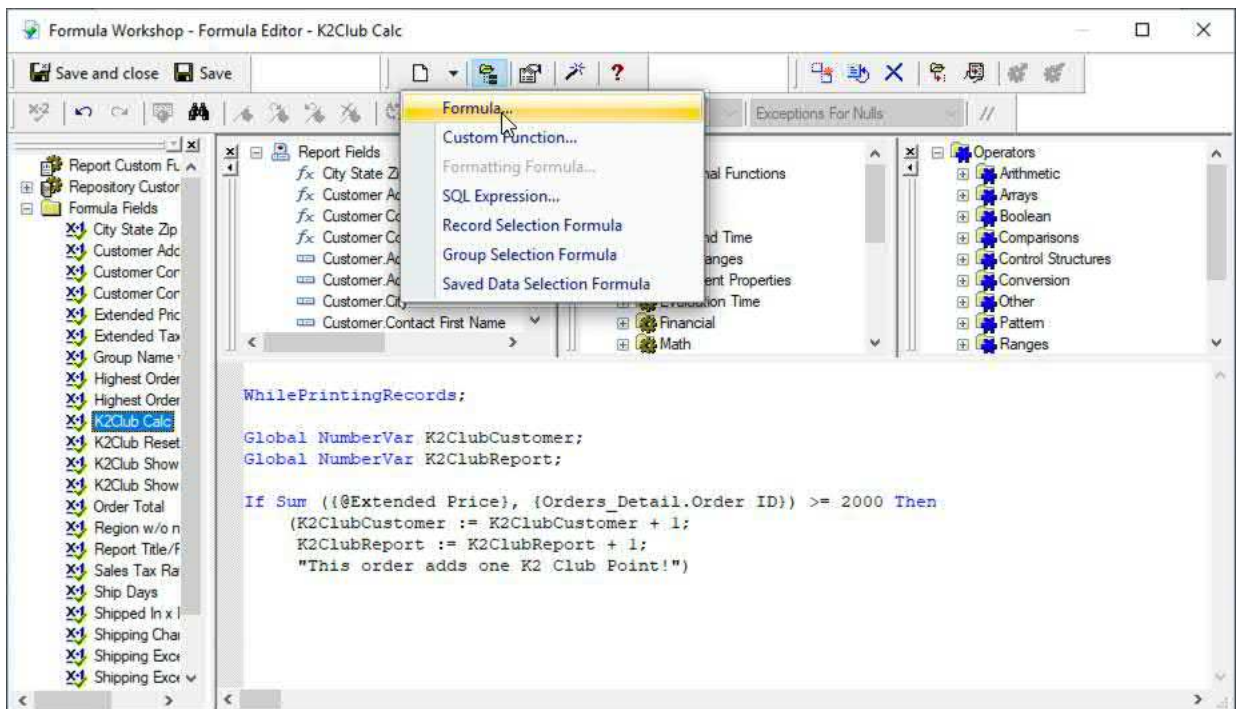
However, if the formula is not syntactically correct, an error message will appear and the formula will be highlighted where Crystal Reports no longer understands the formula syntax (for example, if you supplied the letter O instead of the number zero as the second ToText argument). Error messages may be very descriptive, or vague, depending on the type of error encountered. Click OK, correct the formula at the point of the blue highlighting, and check syntax again.



Once the formula is error-free, click  Save and close in the toolbar to save the formula and close the Formula Workshop. The new formula will appear in the Formulas category of the Field Explorer. You may now drag the formula to the report, as you would any other database field.

THE FORMULA WORKSHOP

The Formula Editor, discussed previously, is actually encompassed by the larger Formula Workshop dialog box. The Formula Workshop surrounds the Formula Editor with two toolbars at the top and a navigation tree on the left.



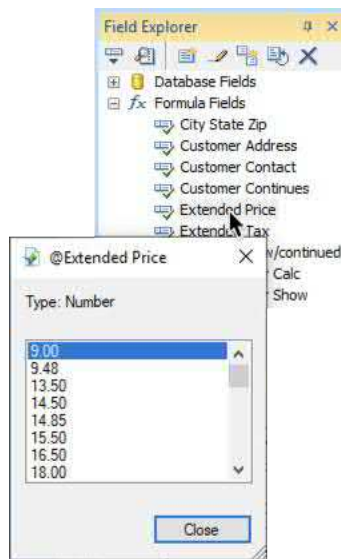
Use toolbar buttons to create new formulas in the Formula Workshop, check syntax on the currently-open formula, undo or redo typing, and so forth. The navigation tree can be used to select existing formulas to modify, or to examine custom functions you've created in the current report or that are contained in a connected SAP Crystal Server repository. Crystal Reports online help provides complete details on the Formula Workshop.



Use toolbar buttons to create new formulas in the Formula Workshop, check syntax on the currently-open formula, undo or redo typing, and so forth. The navigation tree can be used to select existing formulas to modify, or to examine custom functions you've created in the current report or that are contained in a connected SAP Crystal Server repository. Crystal Reports online help provides complete details on the Formula Workshop.

FORMULA TYPES

Like data source fields, each formula will be assigned a data type (string, number, date, and so forth). Unlike database fields, which are assigned a data type by the database designer, you as a report designer will determine the data type of a formula by the fields, functions, and operators you use in the formula. To confirm the data type of formula you have created, RIGHT-CLICK on the formula in the Field Explorer and choose Browse Data from the context menu. Sample formula values, as well as the formula's data type, will appear.



NUMBER

The formula returns a numeric value, based on containing other numeric fields, numeric operators (such as multiplication), or numeric functions (such as the Val function that converts other data types to numbers).

```
{Orders_Detail.Unit Price} * {Orders_Detail.Quantity}
```

STRING

The formula returns a text or string value. The formula may concatenate (tie together) other strings using the + or & operator, use literal text surrounded by quotations marks, or make use of string functions, such as ToText.

```
{Customer.City} + ", " + {Customer.Region} + " " + {Customer.Postal Code}
```

DATE OR DATE/TIME

The formula returns a date or date/time value, depending on fields of functions used in the formula. For example, you might calculate the due date of an invoice by use of the DateAdd function.

```
DateAdd("d", 30, {Invoices.Invoice Date})
```



IF-THEN-ELSE

If-Then-Else formulas make use of traditional logic used in standard computer programming languages, as well as many other analytical software programs (such as spreadsheet programs). One or more tests (using IF) can be evaluated with one or more THEN results being returned by the formula based on the results of the IF test. Optionally, an ELSE clause can return a value if all the previous IF tests fail. The actual data type returned by the formula is based on the fields or values that are referenced by THEN and ELSE portions (which must all return the same data type).

```
If {Invoices.DaysPastDue} > 90 Then
    "Send to Collections"
Else If {Invoices.DaysPastDue} > 30 Then
    "Overdue"
Else
    "Current"
```

BOOLEAN

Similar to just the IF portion of an IF-THEN-ELSE formula, this formula simply returns a TRUE or FALSE value. For example, if a shipping exception occurs when an order takes longer than 3 days to ship, a formula testing another formula that calculates the days to ship will return either TRUE or FALSE.

```
{@Ship Days} > 3
```

ADVANCED VARIABLE USAGE

While SAP Crystal Reports is not positioned as an application development tool, its formula language actually provides many features found in high level programming languages. For advanced reporting requirements, you may even declare and assign values to formula variables. As report logic progresses through detail and group sections, formula logic can test for specific conditions and increment or assign values to variables. These variables may then be placed in other formulas to return values that have been accumulated as report processing progresses.

```
WhilePrintingRecords;

Global NumberVar K2ClubCustomer;
Global NumberVar K2ClubReport;

If Sum ({@Extended Price}, {Orders_Detail.Order ID}) >= 2000 Then
    (K2ClubCustomer := K2ClubCustomer + 1;
    K2ClubReport := K2ClubReport + 1;
    "This order adds one K2 Club Point!")
```



© 2019 SAP AG or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG (or an SAP affiliate company) in Germany and other countries. Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices. Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP AG or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP AG or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP AG or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP AG or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP AG's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP AG or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

