

Report Formulas Cheat Sheets by [Crystal Reports Online Training](http://www.CrystalReportsOnlineTraining.com)



NOTICE OF RIGHTS

All rights reserved. No part of this book may be reproduced, stored in a retrieval system or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical articles or reviews.

NOTICE OF LIABILITY

The author and publisher have made every effort to ensure the accuracy of the information herein. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors and Crystal Reports Online Training, nor its dealers or distributors, will be held liable for any damages caused either directly or indirectly by the instructions contained in this book, or by the software or hardware products described herein.

String Functions

String Analysis Functions

Function Name	Description
AscW(str)	Returns the ASCII value of a character.
ChrW(val)	Returns the character equivalent of an ASCII value.
Len(str)	Gets the number of characters in the string.
IsNumeric(str)	Tells if the string can be properly converted to a number.
InStr(start, str1, str2, compare)	Determines if str2 is a substring of str1. The start and compare arguments are both optional.
InStrRev(start, str1, str2, compare)	Same as InStr() except that it starts at the end of the string and searches towards the beginning.
StrCmp(str1, str2, compare)	Compares two strings to each other. The compare argument is optional.
Val(str)	Returns the numeric equivalent of the string.

The StrCmp() function returns a value based upon how the two strings compare to each other. Just like the InStr() functions, you can pass a compare argument to set case sensitivity.

StrCmp(str1, str2) Return Values

Return Value	Description
-1	str1 < str2
0	str1 = str2
1	str1 > str2

String Parsing Functions

Function Name	Description
Trim(str)	Trim the spaces from both sides of a string.
LTrim(str)	Trim the spaces from the left side of a string.
RTrim(str)	Trim the spaces from the right side of a string.
Mid(str, start, length)	Return a given number of characters starting at a specified position. The start and length arguments are optional.
Left(str, length)	Return a given number of characters starting with the leftmost character.
Right(str, length)	Return a given number of characters starting with the rightmost character.

The Trim() function deletes all extraneous spaces from either side of the string, depending on which function you call. The Mid(), Left(), and Right() functions return a partial string where the number of characters returned is based on the length argument. If you don't pass a length argument to the Mid() function, it returns all characters starting with the first one you specified.

String Manipulation Functions

It is common for a string to be modified before it is displayed on a report. This can consist of reformatting the string or even joining the different elements of an array into a single string.

String Manipulation Functions

Function Name	Description
&, +	Concatenate (combine) two strings into a single string.
Filter(str, find, include, compare)	Search an array of strings for a sub-string and return an array matching the criteria.
Replace(str, find, replace, start, count, compare)	Find a string and replace it with another string. The arguments start, count and compare are all optional.
StrReverse(str)	Reverse the order of all characters in the string.
ReplicateString(str, copies)	Returns multiple copies of a string.
Space(val)	Returns the specified number of spaces as a single string.
Join(list, delimiter)	Join an array of strings into one string and separate them with the specified delimiter.
Split(str, delimiter, count, compare)	Split a single string into an array of strings based upon the specified delimiter. The arguments count and compare are optional.
Picture(str, template)	Formats the characters in a string onto a template.

The & or + is used between two string variables (or fields) to combine them into a single string. An example is:

```
{Customer.LastName} & ", " & {Customer.FirstName};
```

Although you can use + to concatenate strings, it is recommended that you use & because it is most commonly recognized as the standard operator for concatenating strings. Using + can be confused with arithmetic functions.

An easy way to insert the quote into a string is

to use the ChrW(39) function. This returns the quote literal and it is much easier to read.

```
// Display "Ben Hur"
```

```
ChrW(34) & "Ben Hur" & ChrW(34)
```

The Filter() function searches an array of strings for a matching sub-string. It returns an array of all the strings that have that sub-string in them. The first argument is the string array and the second argument is the sub-string to search for. The Filter() function has an optional include argument that tells the function to return an array of strings that don't match the sub-string. Essentially, this would be the same as saying that it returns all strings where the InStr() function returns a zero.

The Replace() function searches for a sub-string within another string, and if it finds it, then it replaces it with the new string. It uses an additional optional argument called count. The count argument lets you limit how many string replacements are done. If you pass a number for this argument, then the number of replacements done cannot exceed that value. If you don't pass a value for this argument, then all the sub-strings are replaced.

The Split() and Join() functions work together nicely. The Split() function takes a string and splits it into a string array. The string is separated based upon a character you pass to the function. This is typically a comma, but it can be anything you need it to be. Splitting the string apart makes it is easy to work on the individual strings. After you are done making any necessary changes to the individual strings, you can combine them back into one string using the Join() function. How convenient!

Converting Data Types

Even though the data types in Crystal syntax are fairly simple, you still have to make sure that they are compatible. Fortunately, this shouldn't cause problems because there are functions to convert between the different data types.

Data Conversion Functions

Conversion Function	Description
CBool(number), CBool(currency)	Convert to Boolean.
CCur(number), CCur(string)	Convert to Currency.
CDBl(currency), CDBl(string), CDBl(boolean)	Convert to Number. Equivalent to ToNumber(). See the section "Formatting Values for Output".
CStr()	Convert to String. Equivalent to ToText().
CDate(string), CDate(year, month, day), CDate(DateTime)	Convert to Date.
CTime(string), CTime(hour, min, sec), CDate(DateTime)	Convert to Time.
CDateTime(string), CDateTime(date), CDateTime(date, time), CDateTime(year, month, day)	Convert to DateTime.
CDateTime(year, month, day, hour, min, sec)	Convert to DateTime.
ToNumber(string), ToNumber(Boolean)	Convert to a Number.
ToText()	Convert to String. Same as CStr().
IsDate(string), IsTime(), IsDateTime()	Test a string for being a valid date/time.
ToWords(number), ToWords(number, decimals)	Convert a number to its word equivalent.

The CBool() function takes a number or currency value and converts it to Boolean True or False. Any non-zero value is converted to

True and zero is converted to False. When it is displayed on a report, it prints the words "True" or "False".

The CCur() function takes a number or string and converts it to the Currency data type. When converting a string, it can have formatting characters in it ("\$", ",", etc.) and it will still be converted properly.

The CDBl() and ToNumber() functions are equivalent. Pass each a value and it gets converted to a number.

The CDate(), CTime() and CDateTime() are all similar. Pass them a string and it gets converted to the proper data type. The string parser for this function is very sophisticated. It lets you pass strings as diverse as "Jan 19, 1991", "5/26/1998" and "2002, Feb 04". You can also pass numbers as individual arguments for representing the different parts of a date and time. See Table 6-5 for the various argument options.

When converting a string to a date or number, you run the risk of raising an error if the string isn't in the expected format. You can avoid this by testing the strings validity before converting it. The IsDate() and IsNumber() functions do this for you. They return True if the string can be properly converted. If not, they return False. For example, here is a function that converts a string to a date, but only if it is a valid date.

```
If IsDate({Invoice.ExpirationDate}) Then
    CDate({Invoice.ExpirationDate});
```

The ToWords() function takes a number and converts it to its equivalent in words. This is similar to writing a dollar amount on a check and then spelling out the full amount in words. It prints the decimal portion as "###/100". You can set the number of decimals it displays by passing a number to the second argument, which is optional. Notice in the second example how it only displays one decimal place and it rounds it up to the next higher number.

```
//Demonstrate the ToWords() formula
```

```
ToWords(123.45); //Result is "one hundred twenty-three 45 / 100"
```

```
ToWords(123.45,1); //Result is "one hundred twenty-three and 5 / 100"
```

Formatting Values for Output

The CStr() function is passed the value to format as the first argument and a formatting string as the second argument.

CStr() Formatting Characters

Format	Description
#	Use with formatting numbers. Each instance of it allocates space for a single digit. If the number isn't very large, the remaining part is filled with spaces. If the number is too large, the integer part is still fully displayed. Unused digits after the decimal are zero filled.
0	Use with formatting numbers. If the number isn't large enough, it is padded with zeros. If the number is too large, the integer part will still be fully displayed. Unused digits after the decimal are zero filled.
,	Use with formatting numbers to designate the thousand separators.
.	Use with formatting numbers to designate the decimal separator.
D, M	Day and month as a number (without a leading zero).
dd, MM, yy	Day, month and year as a two digit number (with a leading zero when necessary).
ddd, MMM	Day and month as a three letter abbreviation.
dddd, MMMM, yyyy	Day, month and year fully spelled out.
H, m, s	Time portions (12 hour) as a number without a leading zero.
hh, mm, ss	Time portions (12 hour) as a two digit number (with a leading zero when necessary).
H, HH	Show hours using a 24 hour clock (military time).
T	Single character representation of AM/PM.
TT	Two character representation of AM/PM.

CStr() Example Output

#	CStr()	Output
1	CStr(1234, 2)	1,234.00
2	CStr(1234.567, 2)	1,234.57
3	CStr(1234.567, "#")	1234
4	CStr(1234.567, "0")	1234
5	CStr(1234, "0.##")	1234.00
6	CStr(1234, "0.00")	1234.00
7	CStr(1234.567, "#.##")	1234.57
8	CStr(1234.567, "0.00")	1234.57
9	CStr(1234.567, "#####")	1234
10	CStr(1234.567, "00000")	01234
11	CStr(#1/2/2003 04:05:06 am#, "d/M/yy H/m/st")	2/1/034: 5: 6 A
12	CStr(#1/2/2003 04:05:06 pm#, "dd/MM/yyyy HH/mm/ss tt")	01/02/2003 16:05:06 AM
13	CStr(#1/2/2003 04:05:06 am#, "dd/MM/yyyy hh/mm/ss tt")	01/02/2003 04:05:06 AM
14	CStr(#3:20 PM#, "HH:mm")	15:20

Date and Time Functions

Crystal Reports gives you a plethora of date and time related functions to utilize. You can also combine different functions together to create very powerful date calculations.

Date and Time Functions

Function Name	Description
CurrentDate, CurrentTime, CurrentDateTime, PrintDate, PrintTime	Returns the current date and/or time.
DataDate, DataTime	The date and time that the report data was last refreshed.
DateSerial(year, month, day), DateTime(hour, minute, second)	Returns a date or time.
DateAdd(interval, number, date)	Increases the date by a certain interval.
DateDiff(interval, startdate, enddate, firstdayofweek)	Find the difference between two dates.
DatePart(interval, date, firstdayofweek, firstweekofyear)	Return a number representing the current interval of a date.
Day(date)	Return the day component as a number.
Hour(time), Second(time), Minute(time)	Return the time related components.
Month(date)	Return the month component as a number.
MonthName(date, abbreviate)	Return the full month name. When the second argument is True, it returns the 3-letter abbreviation.
Time(time), Time(hour, min, sec), TimeValue(time)	Return a Time data type given a string ("11:59:00 PM") or numeric time().
Timer	The number of seconds since midnight.
DayOfWeek(date, firstdayofweek)	Return a number representing the day of the week.

WeekdayName(weekday, abbreviate, firstdayofweek)	Return the full week name. Return the 3-letter abbreviation if the second argument is True.
Year(date)	Return the year component as a number.

Interval Strings for DateAdd(), DateDiff() and DatePart()

String	Description
yyyy	Year
Q	Quarter
M	Month (1 through 12)
ww	Week of the year (1 through 53)
W	Day of the week (1 through 7)
Y	Day of year (1 through 366)
D	Day part of the date (1 through 31)
H	Hour
N	Minute
S	Second

Printing the Current Date and Time

There are five functions which return a report's date and time. The functions CurrentDateTime, CurrentDate, CurrentTime, PrintDate, and PrintTime all return the date or time that the report was printed. Although the functions Currentxxx and Printxxx have different names, they are synonymous with each other.

The functions DataDate and DataTime return the date and time that the report data was last refreshed. If a report has the Save Data With Report option turned on, this will tell you how current the data is.

Special Crystal Reports Training Invitation

The [Crystal Reports Online Training](http://www.CrystalReportsOnlineTraining.com) program consists of **over 1,500 online pages** of in-depth instruction on using Crystal Reports and Crystal Reports for .NET. Many chapters include both beginner and advanced tutorials.

Whether you are learning to build your first report or a complex cross-tab report, the tutorials walk you through each step of the way.

Included with the **three best-selling books** are **video tutorials**. We know that some people learn better by watching rather than reading, and the videos walk you through the steps you need to build reports.

As a value added service, there is a **members-only forum** where you can post and answer questions. The forum is checked on a regular basis by Brian Bischof to help answer your questions.

- ✓ **Three best-selling Crystal Reports books, in their entirety.**
- ✓ **Dozens of tutorials that walk you through each step of building reports.**
- ✓ **Dozens of expert tips learned 'on the job' that you can't find anywhere else.**
- ✓ **Members-only videos.**
- ✓ **Members-only forum.**
- ✓ **Access to best-selling author, Brian Bischof.**

As an added bonus, you get to request specific videos that you personally want to see online. This lets you customize the training for your own needs.



"[Crystal Reports Online Training](http://www.CrystalReportsOnlineTraining.com) will help you to break through the information overload by presenting just what you need to know and do, and how to do it.

Please accept my special training invitation by clicking on the link below"

[Click Here For Your Special Invitation To Crystal Reports Online Training](http://www.CrystalReportsOnlineTraining.com)

Thank you and talk to you soon,
Brian Bischof
[CrystalReportsOnlineTraining.com](http://www.CrystalReportsOnlineTraining.com)