Calculation Commands

The third line of the sample screen C_HOME_PHONE has no formatting in the fieldcalculation. This instructs the printer to left justify the home phone of the patron starting in row 15 and column 1040. To justify the phone number place the field in brackets and prefix it with the letters 'jst'. Follow the field (inside the bracket) with a comma and a single quote to start the formatting instructions, enter the formatting instructions, and close with a single quote. For example, jst(C_HOME_PHONE,'-8') formats the field calculation to allocate 8 spaces for the home phone number and to right justify when printing.

String Justification

String	Description	Example
jst ()	jst(string1,number1,[string2,number2]...) Returns a string containing the specified string left or tight justified with sufficient spaces added to make a total length specified by number. The jst() function also includes concatenation. If number is negative the resulting string is right justified, if the number is positive the string is left justified; number must be in the range of 1 to 999.	jst('This is left justified',30) gives |This is left justified | jst('This is right justified',-30) gives | This is right justified|
^n	(caret) Causes the data to be centered in the field n characters wide.	jst('This is centered',^30) gives | This is centered |
-n	(minus) Causes the data to be right justified in a field n characters wide.	jst('This is Right',-20) gives |*******This is Right|where * represents spaces
$	Places a $ sign in front of the data.	jst(PS_TOTAL_PAID,'$' gives $12.00 if the ticket price paid was 12.00
Pc	Causes the part of the field not filled by the data to be filled by character c.	jst(PS_TOTAL_PAID,'-7P*' gives '**12.00' if the ticket price paid was 12.00
X	Causes the data to be truncated if its length exceeds the field length. The default is not to truncate.	jst('abcdef','4') gives 'abcdef jst('abcdef','4X') gives 'abcd'
U	Causes the data to be converted to upper case.	jst('this IS upper Case','U' gives 'THIS IS UPPER CASE'
L	Causes the data to be converted to lower case.	jst('this IS lower Case','L' gives 'this is lower case'
C	Causes the data to be capitalized.	jst('this IS Capitalized','C' gives 'This Is Capitalized'
Nnn	Causes the data to be treated as a fixed decimal number with nn decimal places.	jst(0.235,'N') gives '0.235' jst(0.235,'N2') gives '0.24'
E	Applies to numbers only and displays a blank when the number is zero.	jst(0,'N2') gives '0.00' jst(0,'N2E') gives ' '
,	Applies to numbers only and adds commas as number separators.	jst(1234,'N2') gives '1234.00' jst(1234,'N2,') gives '1,234.00'

© 2011, Arts Management Systems

String Functions

String	Description	Example
cap()	Returns the capitalized representation of a string, that is, the first letter of each and every word in the string is capitalized.	cap('tHeATre MANAGER') returns 'Theatre Manager'
con(string1,string2[,string3]...)	Returns a string concatenating or combining two or more string values. Theatre Manager will build concatenations. Simply select more than one field from the available fields list when editing the field calculation. To format the concatenation simply treat the concatenation as one field calculation.	con(C_FIRST_NAME,'/',C_ADDRESS1,'/',C_POSTAL_CODE). First name, address, and postal code information together on the same line and places a slash between each item. Formatting is limited to the group of fields linked together.
		jst(con(MS_SECTION,', Row ',MS_ROW_NUMBER,', Seat ',MS_SEAT_NUMBER),'^38x'). Centers section, row, and seat number in a space 38 characters wide. A comma and row precede the row number and separate it from the actual number. A comma and seat precede the seat number and separate it from the actual number.
		jst(con(MS_SECTION,', Row ',MS_ROW_NUMBER,', Seat ',MS_SEAT_NUMBER),'^38x'). Centers section, row, and seat number in a space 38 characters wide. A comma and row precede the row number and separate it from the actual number. A comma and seat precede the seat number and separate it from the actual number.
len()	Returns the length of the string, that is, number of characters.	len('abcd') returns '4' len('abcd') + 15 returns '19'
low()	Returns the lower case representation of a string. Any non-alphabetic characters in the strings are unaffected by low().	low('tHeATre MANAGER') returns 'theatre manager' low('1234') returns '1234'
mid(string,position,length)	Returns a substring of a specified length, starting at a specified position, from a larger string. If position is less than 1 it is taken as 1, that is the first character; if it is greater than the length of the string, an empty string is returned. If length is greater than the maximum length of any substring of string starting at position, then the returned substring will be the remainder of string starting at position..	mid('Theatre Manager',9,3) returns 'Man' mid('Theatre Manager',9,24) returns 'Manager'
pos(substring,string)	Returns the position of a substring within a larger string. The substring must be contained within string in its entirety for the returned value to be non-zero.	pos(' ',J. Smith') returns 2 (the position of the space character) pos('Mouse','Mickey Mouse') returns 8 pos('mouse','Mickey Mouse') returns 0 - note the case of the substring
upp()	Returns the upper case representation of a string. Any non-alphabetic characters in the strings are unaffected by low().	upp('tHeATre MANAGER') returns 'THEATRE MANAGER' upp('1234') returns '1234' upp(mid(dtw(PB_PERFORM_DATE),1,3)) returns 'MON'

© 2011, Arts Management Systems

Date/Time Functions

The date/time functions operate on a datestring to return date values or strings representing some part of a date/time. A datestring is a string recognizable as a date using a date format.

String	Description	Example
dat(datestring|number[,dateformat])	Converts a datestring or number to a date value using a second optional argument, a dateformat string. The date formatting string can be composed of the following symbols: Y Year 98 (no century) y Year 1998 (with century) C Century 19 m Month short form (JUN) M Month number (06) n Month long form (June) D Day of month (12) d Day of month (12th) W Day of week (5) w Day of week long form (Friday) V Day of week short form (FRI) E Day of year (between 1 and 366) G Week of Year (between 1 and 52) F Week of Month (between 1 and 6)	dat('Dec 21 11','MDY') returns '122111' dat('Dec 21 11','D m Y') returns '21 DEC 11' dat('Dec 21 11','w, d n, y') returns 'Wednesday, 21st December, 1911' dat('Dec 21 98','V m D Y') returns 'WED DEC 21 98'
dim(datestring,number)	Increments a datestring by a number of months. Months containing different numbers of days are accounted for. Jan 31 96 increased by one month gives Feb 29 96, but beware, Feb 29 96 decreased by one month (using a negative number) returns Jan 29 96, not Jan 31 96.	dim(dat('Dec 21 98',3)) returns 'Mar 21 99'
dtcy(datestring)	Returns the year and century of a datestring as a string.	dtcy(dat('Dec 21 98')) returns '1998'
dtd(datestring)	Returns the day part of a datestring unless it is part of a calculation when it is returns as a number.	dtcy(dat('Dec 21 98')) returns '21st' dtcy(dat('Dec 21 98')) + 0 returns '21' dtcy(dat('Dec 21 98')) + 1 returns '22'
dtm(datestring)	Returns the month part of a datestring.	dtm(dat('Dec 21 98')) returns 'December'
dtw(datestring)	Returns the week part of a datestring.	dtw(dat('Dec 21 98')) returns 'Monday' upp(mid(dtw(PB_PERFORM_DATE),1,3)) returns 'MON'
dty(datestring)	Returns the year part of a datestring.	dtcy(dat('Dec 21 98')) returns '98'
tim(timestring[,timeformat])	Converts a timestring to a time value using a second optional argument, a timeformat string. The time formattinG string can be composed of the following symbols: H Hour (between 0 and 23) h Hour (between 1 and 12) N Minutes S Seconds s Hundredths A AM/PM	tim('13:25:57','h:N A')) returns '1:15 PM' tim('13:25:57','h:N.S A')) returns '1:15.57 PM'

© 2011, Arts Management Systems

Lookup Functions

The lookup functions provide a method for selecting items from a list of values.

String	Description	Example
max(value1[,value2]...)	Returns the maximum value from the list of values. The values should all be numbers when numeric comparison is used or all strings when string comparison is used.	max(3,6,2,7) returns '7' max('dagger','dog','digger') returns 'dog'
min(value1[,value2]...)	Returns the minimum value from the list of values. The values should all be numbers when numeric comparison is used or all strings when string comparison is used.	min(3,6,2,7) returns '2' min('dagger','dog','digger') returns 'dagger'
pick(number,value0,value1,[value2]...)	Selects an item from a list of string or numeric values. The number argument is rounded to an integer and used to select the item. value0 is returned if the result is 0, value1 if the result is 1, value2 if the result is 2, and so on. If the number is less than zero or greater than the number of values in the list, then an empty value is returned. The list of values can be a mixture of string and numeric values.	pick(2,'A','B','C','D') returns 'C' pick(2>4,'A','B','C','D') returns 'A', as 2 is not greater than 4, thus it is False, where False equals 0. pick(2<4,'A','B','C','D') returns 'B', as 2 is less than 4, thus it is True, where True equals 1. pick(pos('A','DEF')>0,'Price Not Found','Price Found') returns 'Price Not Found' pick(pos('A','ABC'),'Price Not Found','Price A','Price B','Price C') returns 'Price A'

Number Functions

String	Description	Example
abs(number)	Returns the magnitude of a real number ignoring its positive or negative sign.	abs(1002) returns '1002' abs('-203.45') returns '203.45' abs('12ABC') returns '0'
int(number)	Returns the integer part of a number; it does not round to the nearest integer.	int(23.1056) returns '23' int('-2.66') returns '-2' abs(int('-2.66')) returns '2'
mod(number1,number2)	Returns the remainder of a number division, that is, when number1 is divided by number1 to produce a remainder; it is a true modulus function.	mod(6,4) returns '2' mod(6,4) returns '2' mod(9,1.5) returns '0'
rnd(number,dp)	Rounds a number to a specified number of decimal places by dp .	rnd(2.105693,2) returns '2.11' rnd(0.5,0) returns '1'

Labels

Entering a label, or string of characters that will be the same on every ticket, is accomplished by placing the label in single quotes. Example: 'PRESENTS' places the word PRESENTS on the ticket.

© 2011, Arts Management Systems