

Using the TRANSFORM Function to Create Numeric Formats

Product: R&R Report Writer® for DOS **Oper Sys:** DOS®
Version: All

Summary

The TRANSFORM function enables you to format character or numeric data by using one of several predefined formats or by supplying a picture or template to represent the desired format.

The /Field Format command in R&R Report Writer provides you with six numeric formats. Two of these formats, Currency and Comma, place negative numbers in parentheses instead of using the negative sign. For example, -2500 is displayed as (\$2,500.00) and (2,500) in those formats. If you prefer not to put negative numbers in parentheses, you can create user-defined functions that use the TRANSFORM function to create non-parenthesized numeric formats.

The TRANSFORM Function

The TRANSFORM function enables you to format character or numeric data by using one of several predefined formats or by supplying a picture or template to represent the desired format. The function's first argument is the value to be formatted. The function's second argument provides the formatting instructions, which can be a code specifying the predefined format and/or a template.

For example, to create a numeric template to format any nine-digit number with dashes after every three numbers apply the template 999-999-999 to a numeric field called NUMBER. You create a calculated field with the expression

```
TRANSFORM (NUMBER, '999-999-999')
```

where the 9s represent the digits of the input value and the dashes show where dashes are to be inserted in the numeric value.

NOTE: More information is available in the TRANSFORM section of the Calculated Field Expressions chapter of the R&R User's Guide.

Comma Format

You can create a user-defined format (UDF) that transforms a number into a character string to produce a comma format with a specified number of decimal places and negative numbers identified with a negative sign rather than enclosed in parentheses.

For negative numbers, the character string will consist of a minus sign followed by the number. For positive numbers, the string will consist of just the number. The declaration and formula for this UDF are:

```
Declaration:  COMMA(N_NUMBER)
Formula:      IIF (NUMBER<0, "-", "") + LTRIM (TRANSFORM (ABS
              (NUMBER), '999,999'))
```

According to its declaration, this UDF named COMMA takes one numeric argument. According to the formula, the UDF performs these operations on the value that is supplied as the numeric argument:

- First, the NUMBER value is tested to see if it is negative, that is, less than zero. If the number is negative, the IIF function returns a string consisting of a single minus sign. If the number is positive or zero, the IIF function returns an empty string.
- The ABS function returns the absolute value of the NUMBER value, which in effect turns any negative number into a positive number.
- The TRANSFORM function formats the value returned by the ABS function according to the template you provide. In this case the template is '999,999', which indicates that the number will have a thousands comma separator and no decimal places.

NOTE: The template must have enough characters to accommodate the maximum width of the NUMBER value. You can edit the template to provide the number of decimal places needed. For example, use a '999,999,999.99' template to format number values in the millions, a comma separator, and two decimal places.

- The LTRIM function trims any leading spaces from the value returned by the TRANSFORM function.
- Finally, the value returned by LTRIM is concatenated with the value already returned by the IIF function (a minus sign for negative NUMBER values and an empty string for positive values).

You can use this UDF in a calculated field expression. If you want to format the AMOUNT field in a report to use a negative sign rather than enclose negative numbers in parentheses,

apply this format to the field. To apply the format, create a calculated field that returns a formatted string corresponding to the value in the AMOUNT field.

The expression for this field (named FAMOUNT, for example) is `COMMA(AMOUNT)`. When you provide the AMOUNT field as the numeric argument for the COMMA UDF, R&R Report Writer performs the operations defined in the UDF's formula on the value in the AMOUNT field.

AMOUNT	FAMOUNT
123	123
123.4	123
-123	-123
-123.5	-124
-1,234	-1,234
-123,456.45	-123,456

Figure 1. Various AMOUNT Values and Corresponding FAMOUNT Values

Currency Format

You can create another UDF that transforms a number into a character string to produce a currency format with a specified currency sign and number of decimal places, and has negative numbers identified with a minus sign.

For negative numbers, the character string will consist of a negative sign followed by a specified currency sign (in this case the dollar sign) followed by the number. For positive numbers, the string will consist of the currency sign followed by the number. The declaration and formula for this UDF are:

Declaration: **CURRENCY (N_NUMBER)**

Formula: **IIF (NUMBER < 0, "-", "") + "\$" + LTRIM (TRANSFORM (ABS
(NUMBER), '999,999.99'))**

According to its declaration, this UDF named **CURRENCY** takes one numeric argument. According to the formula, the UDF performs these operations on the value that is supplied as the numeric argument:

- First, the **NUMBER** value is tested to see if it is negative; that is, less than zero. If the number is negative, the **IIF** function returns a string consisting of a single minus sign. If the number is positive or zero, the **IIF** function returns an empty string.

- The value returned by the IIF function is concatenated with a string consisting of a dollar sign. Note that you can change this string to consist of any currency sign and that you could change its location in the expression so that it appears at the beginning or end of the returned value.
- The ABS function returns the absolute value of the NUMBER value, which in effect turns any negative number into a positive number.
- The TRANSFORM function formats the value returned by the ABS function according to the template you provide. In this case the template is '999,999.99', which indicates that the number will have a thousands comma separator and two decimal places.

NOTE: The template must have enough characters to accommodate the maximum width of the NUMBER value. You can edit the template to provide the number of decimal places needed. For example, use a '999,999,999.99' template to format number values in the millions, a comma separator, and two decimal places.

- The LTRIM function trims any leading spaces from the value returned by the TRANSFORM function.
- Finally, the value returned by LTRIM is added to the previously constructed string, which consists of the IIF value (a minus sign for negative NUMBER values and an empty string for positive values) plus the dollar sign.

You can use this UDF in a calculated field expression. If you want to format the BALANCE field in a report to a currency format and use a negative sign rather than enclose negative numbers in parentheses, apply this format to the field. To apply the format, create a calculated field that returns a formatted string corresponding to the value in the BALANCE field.

The expression for this field, named FBALANCE for example, is CURRENCY(BALANCE). When you provide the BALANCE field as the numeric argument for the CURRENCY UDF, R&R Report Writer performs the operations defined in the UDF's formula on the value in the BALANCE field.

BALANCE	FBALANCE
123	\$123.00
123.4	\$123.40
-123	-\$123.00
-123.5	-\$124.50
-1,234	-\$1,234.00
-123,456.45	-\$123,456.45

-123,456.123

-\$123,456.12

Figure 2. Various BALANCE Values and Corresponding FBALANCE Values.

Other Numeric Formats

You can use the general techniques explained in this article to produce UDFs that will format numeric and character values in many ways. For example, TRANSFORM is particularly useful for formatting phone numbers, postal codes, and social security numbers. In fact, you can use a TRANSFORM UDF to reproduce the format of almost any structured number or character string. For more information about user-defined functions, see the Calculated Field Expressions chapter of the R&R Report Writer User's Guide.

All trademarks are the property of their respective owners. The information contained in this technical bulletin is subject to change without notice. Liveware Publishing Inc. provides this information "as is" without warranty of any kind, either expressed or implied, but not limited to the implied warranty of merchantability and fitness for a particular purpose. Liveware Publishing may improve or change the product at any time without further notice; this document does not represent a commitment on the part of Liveware Publishing. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the licensing agreement.