R&R Report Writer *Technical Bulletin*

Focus on the TRANSFORM Function

Product: ARPEGGIO™
R&R Report Writer® for Windows®
Version: All
Host: N.A.

NIC: N.A.
Interface: N.A.
Oper Sys: Microsoft® Windows® NT®

Summary

The TRANSFORM function provides a powerful tool for formatting character and numeric data. It works by converting a character string or numeric value into a format you specify. Common applications include phone numbers, zip codes, and social security numbers. For example, you can use TRANSFORM to convert a phone number stored as 8143403 into (425) 814-3303.

TRANSFORM takes two arguments. The first argument is the value you want to format, which must be character or numeric. The second argument provides the formatting instructions, which can be either a code specifying the format or a template. For example, you might use a numeric template such as 999-999-999 to print dashes after every third number in a 9-digit numeric value. To apply this template to a numeric field called NUMBER, you would create a calculated field whose expression is TRANSFORM(NUMBER,"999-999-999"), where the 9s represent the digits of the input value and the dashes represent the literal dashes you want to insert.

For detailed instructions on using the TRANSFORM function, refer to Chapter 11, “Using R&R Functions,” in *Using R&R*.

Common Transforms

The following sections give examples of some of the most common uses of the TRANSFORM function. You may have to make minor changes to the expressions before using them in your applications, depending on how your data is stored.

Telephone Numbers

In the following expressions for telephone numbers, the database field PHONE stores telephone numbers as 10-digit numbers, without punctuation, which represent the area

code, exchange, and number. You can use TRANSFORM to convert the value 4258143403 into (425) 814-3403 using one of the following expressions:

If PHONE is a character field:

```
TRANSFORM(PHONE,"@R (XXX) XXX-XXXX")
```

If PHONE is a numeric field:

```
TRANSFORM(PHONE,"(999) 999-9999")
```

In the first expression, the @R function indicates that characters such as the parentheses, dash, and spaces are literals and should be included in the output string. The data from the PHONE field replaces the "X" characters. In the second expression, the data from the PHONE field replaces the "9" characters.

If the PHONE field stores 7-digit numbers without area codes, you can use an expression like the one below to output the area code with the phone numbers, so 8143403 will be converted into (425) 814-3403.

If PHONE is a character field:

```
TRAN(PHONE,IIF(LEN(TRIM(PHONE))=7,"@R (425) XXX-XXXX","@R (XXX) XXX-XXXX"))
```

If PHONE is a numeric field:

```
TRAN(PHONE,IIF(PHONE<=9999999,"(425) 999-9999","(999) 999-9999"))
```

You might also want to right-align the phone numbers in a column when some values include the area code and others don't, as shown below:

You want this	**Not this**
(312) 825-4416	(312) 825-4416
814-3403	814-3403

To right align–the phone numbers, you can use one of the following expressions:

Character:

```
TRAN(PHONE,IIF(LEN(TRIM(PHONE))=7,"@R       XXX-XXXX","@R (XXX) XXX-XXXX"))
```

Numeric:

```
TRAN(PHONE,IIF(PHONE<=9999999,"       999-9999","(999) 999-9999"))
```

In both of these examples, you type spaces in the expression to replace the zip code portion of the template, so that the 7-digit numbers are right-aligned.

Zip Codes

If you have a field ZIP that contains both 5- and 9-digit zip codes, use an expression like one of the following to convert 9-digit zip codes such as 123456789 into 12345-6789:

```
TRANSFORM(ZIP,IIF(LEN(TRIM(ZIP))=9,"@R XXXXX-XXXX","@R XXXXX"))
TRANSFORM(ZIP,IIF(ZIP>99999,"99999-9999","99999"))
```

If your database includes a COUNTRY field and some of your customers are Canadian, use the following expression to apply the proper format for Canadian postal codes, which do not have the same format as U.S. zip codes. A Winnipeg postal code, for example, is R3G 0M8.

```
TRANSFORM(ZIP,IIF(COUNTRY="Canada","@R XXX XXX",IIF(LEN(TRIM(ZIP))=9,
"@R XXXXX-XXXX","@R XXXXX")))
```

Social Security Numbers

Social security numbers are easy to format using TRANSFORM, since the data is all in the same form. For example, if you have a field SS_NO that contains 9-digit numbers, use an expression like one of the following to convert a social security number such as 123456789 into 123-45-6789:

```
TRANSFORM(SS_NO,"@R XXX-XX-XXXX")
TRANSFORM(SS_NO,"999-99-9999")
```

Creating Numeric Formats

The Windows International settings in the Windows Control Panel control formatting characteristics of numeric values on R&R Report Writer reports. For example, for currency values, you can specify the character you want to use (such as $ or £), how you want to format negative numbers, how many decimal characters you want to show, and other characteristics. When you apply the R&R Report Writer Currency format to a numeric field on a report, the International settings you selected control how the field prints on the report.

However, the International settings for non–currency numeric values do not offer as many choices as the currency settings. If you want to apply numeric formats that are not available through Windows, you can create user–defined functions that use the TRANSFORM function. For example, non–currency negative numbers are shown with a minus sign (as in -2,500) by default, but you may prefer to parenthesize negative numbers. To produce a parenthesized format with a specified number of digits, you can create a User-Defined Function (UDF) that identifies negative numbers and converts them into parenthesized formats. You can use the same techniques to develop UDFs that apply other numeric formatting characteristics.

The declaration and formula for this UDF, which we will call PARENS, are:

Declaration:

```
PARENS(N_NUMBER)
```

Formula:

```
IIF(NUMBER<0,LTRIM(TRANSFORM(ABS(NUMBER),"(999,999)")),"")
```

According to this declaration, the name of this UDF is PARENS and it takes one numeric argument. According to the formula, the UDF performs the following operations on the value that is supplied as the numeric argument:

1. First, the NUMBER value is tested to identify whether it is less than zero. If the number is negative, the IIF function returns the formatted string. If the number is positive or zero, the IIF function returns an empty string.
2. The ABS function returns the absolute value of NUMBER. Effectively, it turns the negative number into a positive number. Its purpose here is to remove the minus sign from the NUMBER value.
3. The TRANSFORM function formats the value returned by the ABS function according to the template. In this case the template is "(999,999)", which indicates that the number will have a thousands comma separator, no decimal places, and enclosing parentheses. You can change this template to provide any number of decimal places you like. For example, you could use the template "999,999,999.99" to format number values in the millions with a comma thousands separator and two decimal places. The template must allow for the maximum number of digits that the field being transformed can contain. If the template is longer than the maximum field with, the LTRIM function will trim the extra spaces.
4. The LTRIM function trims any leading spaces from the value returned by TRANSFORM.

You can now use this UDF in a calculated field expression. For example, if you want to apply the parenthesized format to the AMOUNT field in a report, create a calculated field FAMOUNT that supplies the AMOUNT field as the numeric argument for the PARENS user-defined function. The FAMOUNT expression would be:

```
PARENS(AMOUNT)
```

Insert the FAMOUNT calculated field on your report layout instead of the AMOUNT field. When you run the report, R&R Report Writer will format value in the AMOUNT field according to the formula in the UDF.

For example, the figure below shows some of the possible values of AMOUNT and FAMOUNT.

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