http://www.sqlservercentral.com/articles/IsNumeric/71512/
Printed 2012/09/17 12:30PM

Why doesn't ISNUMERIC work correctly? (SQL Spackle)

By Jeff Moden, 2010/10/28

"SQL Spackle" is a collection of short articles written based on multiple requests for similar code. These short articles are NOT meant to be complete solutions. Rather, they are meant to "fill in the cracks".

--Phil McCracken

Introduction

This is a very old subject but, according to a lot of posts, a lot of folks still don't understand that ISNUMERIC is NOT an "IsAllDigits" function. There are many cases where you need to ensure that the string data you are working with includes only numeric digits. Most Developers will use the built in ISNUMERIC function to make such a check. Here's why that's a bad idea and what to do about it.

What is ISNUMERIC?

"Books OnLine" summarizes the description of the ISNUMERIC function as...

"Determines whether an expression is a valid numeric type."

...and that's a 100% accurate description that leaves much to be desired. Just what is a "valid numeric type"? Reading further in BOL (short for "Books OnLine), we find additional information:

"ISNUMERIC returns 1 when the input expression evaluates to a valid integer, floating point number, money or decimal type; otherwise it returns 0. A return value of 1 guarantees that expression can be converted to one of these numeric types."

Again, read the wording... "when the input expression evaluates to a valid integer", etc, etc. And, that's the catch. There are many different things that you may not expect that will evaluate to one of the data types listed in the description of ISNUMERIC and a lot of them are NOT the digits 0 thru 9. ISNUMERIC will return a "1" for all of them.

Let's consider the most obvious... what will ISNUMERIC('-10') return? What will ISNUMERIC('1,000') return? And how about the not-so-obvious... what will ISNUMERIC('0d1234') or ISNUMERIC('13e20') return? How about ISNUMERIC('1,2,3,4,5')? There are many different combinations of letters, numbers, and symbols that can actually be converted to numeric data types and ISNUMERIC will return a "1" for all of them. It's not a flaw... that's the way it's supposed to work!

What IS Actually Considered "Numeric" by ISNUMERIC?

This code will show all of the single characters that ISNUMERIC thinks of as "Numeric"...

```
------ Return all characters that ISNUMERIC thinks is numeric
     -- (uses values 0-255 from the undocumented spt_Values table
     -- instead of a loop from 0-255)
 SELECT [Ascii Code]        = STR(Number),
        [Ascii Character]   = CHAR(Number),
        [ISNUMERIC Returns] = ISNUMERIC(CHAR(Number))
   FROM Master.dbo.spt_Values
  WHERE Type = 'P'
    AND Number BETWEEN 0 AND 255
    AND ISNUMERIC(CHAR(Number)) = 1
```

That code produces the following list of characters...

```
Ascii Code Ascii Character ISNUMERIC Returns
---------- --------------- -----------------
         9                 1
        10                 1
        11                 1
        12                 1
        13                 1
        36 $               1
        43 +               1
        44 ,               1
        45 -               1
        46 .               1
        48 0               1
        49 1               1
        50 2               1
        51 3               1
        52 4               1
        53 5               1
        54 6               1
        55 7               1
        56 8               1
        57 9               1
       128 €               1
       160                 1
       163 £               1
       164 ¤               1
       165 ¥               1
```

What are those characters?

Ascii Code 9 is a TAB character and is included because a column of numbers is frequently delimited by a TAB.

Ascii Code 10 is a Line Feed character and is included because the last column of numbers is frequently terminated by a Line Feed character.

Ascii Code 11 is a Vertical Tab character and is included because the last column of numbers is frequently terminated by a Vertical Tab character.

Ascii Code 12 is a Form Feed character and is included because the last column numbers of the last row is sometimes terminated by a Form Feed character.

Ascii Code 13 is a Carriage Return character and is included because the last column of numbers is

frequently terminated by a Carriage Return character.

Ascii Codes 36 (Dollar sign), 128 (Euro sign), 163 (British Pound sign), and 164 (Yen sign) are included because they are frequently used as enumerators to identify the type of number or, in this case, the currency type the number is meant to represent.

Ascii Codes 43 (Plus sign), 44 (Comma), 45 (Minus sign), and 46 (Decimal place) are included because they are frequently included in numeric columns to mark where on the number line the number appears and for simple formatting.

Ascii Code 160 is a special "hard space" and is included because it is frequently used to left pad numeric columns so the column of numbers appears to be right justified.

Ascii Code 32 is a "soft space" and is not included because a single space does not usually represent a column of numbers. Ascii Code 32 is, however, a valid numeric character when used to create right justified numbers as is Ascii Code 160 but a single Ascii Code 32 character is NOT numeric. In fact, a string of Ascii Code 32 spaces is not considered to be numeric but a string of spaces with even a single digit in it is considered to be numeric.

Ascii Code 164 is a special character and is included because it is frequently used by accountants and some software to indicate a total or subtotal of some type. It is also used by some to indicate they don't know what the enumerator is.

Ascii Codes 48 thru 59 are included because they represent the digits 0 through 9

Sets of Characters Treated as "Numeric" by ISNUMERIC

Do notice that "e" and "d" (everybody forgets about this) are not included as numeric in the results because a single "e" or "d is NOT considered to be numeric. HOWEVER, these letters represent two different forms of numeric notation (the one with the "e" is Scientific Notation). So, if you have anything that looks like the following, ISNUMERIC will identify them as "Numeric"...

SELECT ISNUMERIC('0d2345')
SELECT ISNUMERIC('12e34')

The "Regular" Solution

Hopefully, I've proven that ISNUMERIC is NOT the way to determine if a value or a column of values IS ALL DIGITS. So, what to do? We could write something really complex that loops through each character to see if it's a digit... or ... we can use a very simple (almost) regular-expression to do the dirty work for us. The formula is...

NOT LIKE '%[^0-9]%'

... and it can be used directly (preferred method for performance reasons)...

```
SELECT *
  FROM sometable
 WHERE somecolumn NOT LIKE '%[^0-9]%'
```

... or you can create your own "IsAllDigits" function (this one is an "Inline Table Valued Function" or "iTVF").

```
 CREATE FUNCTION dbo.IsAllDigits
/**********************************************************************
 Purpose:
 This function will return a 1 if the string parameter contains only
 numeric digits and will return a 0 in all other cases.  Use it in
 a FROM clause along with CROSS APPLY when used against a table.

 --Jeff Moden
**********************************************************************/
------ Declare the I/O parameters
        (@MyString VARCHAR(8000))
RETURNS TABLE AS
 RETURN (
         SELECT CASE
                WHEN @MyString NOT LIKE '%[^0-9]%'
                THEN 1
                ELSE 0
                END AS IsAllDigits
        )
```

The reason why the formula works is that the "^" means "NOT". So the formula is stating "find everything that's not like something that has something not like a numeric digit." Doing the Boolean math, it means "If everything in the string is a digit from 0 to 9, return a 1".

Crack filled! ;-)

Thanks for listening, folks.

--Jeff Moden
