

SRP_Num

Determines if a value is a number upon which arithmetic can be done.

Syntax

```
Result = SRP_Num(Expression, Default)
```

Parameters

Title Field	Description
Expression	The expression to evaluate.
Default	The default value to return if the expression is not a number. (<i>Optional</i>)

Remarks

SRP_Num has two modes: to determine if an expression is a number or to default an expression to a number. Mode 1 occurs when *Default* is omitted, and mode 2 occurs when *Default* is set.

Mode 1

In Mode 1, SRP_Num returns 1 if the expression is a number or 0 if not. Unlike the built-in Num() function, SRP_Num only returns 1 if *Expression* is safe for arithmetic.

```
// A is not assigned, so IsNum will be 0. Using Num() would have caused a VNAV error.
IsNum = SRP_Num(A)

// B is "", which might be different than 0 in this application
// Num() would have returned 1 and calculating C would have assumed "" is 0
B = ""
If SRP_Num(B) then
    C = B + 1
end
```

Mode 2

In Mode 2, SRP_Num returns *Expression* if *Expression* is a number or *Default* if it is not. This is particularly useful for quickly defaulting numeric function parameters.

```

Compile Function Divide(Dividend, Divisor)

// The usual way of defaulting parameters, which will break this routine if Divisor is ""
If Unassigned(Dividend) then Dividend = 0
If Unassigned(Divisor) then Divisor = 1

// To be even safer, we need to check for both "" and unassigned... but this will break too
// because BASIC+ evaluates all expressions in an condition, even if the first expression
// makes the condition true. Thus, these statements will break to the debugger if Dividend
// or Divisor are unassigned.
If Unassigned(Dividend) OR Dividend EQ "" then Dividend = 0
If Unassigned(Divisor) OR Divisor EQ "" then Divisor = 1

    // This is the safest way to ensure this routine never breaks to the bugger.
    If Unassigned(Dividend) then
        Dividend = 0
    end else
        If Dividend EQ "" OR Not(Num(Dividend)) then Dividend = 0
    end
    If Unassigned(Divisor) then
        Divisor = 1
    end else
        If Divisor EQ "" OR Not(Num(Divisor)) then Divisor = 1
    end

    // SRP_Num makes it so much simpler
    Dividend = SRP_Num(Dividend, 0)
    Divisor = SRP_Num(Divisor, 1)

Return Dividend / Divisor

```