

NUMTOCHAR_CONV Conversion

When OpenInsight examines the contents of a string, if it evaluates to a number (such as **10.0**) and is compared with another string which evaluates to a number (such as **10.00**), then a comparison operator (such as **If**) does a numeric comparison. In this case the numbers would evaluate as equal.

There are situations (for example, where these strings represent product codes **10.0** and **10.00**, which are different products, where a string comparison is required. How to accomplish this? A simple user conversion, called **NUMTOCHAR_CONV**, does the trick, at the cost of adding an additional character to the stored string.

The principle is that on input, store the string with a character prefix (the conversion uses the letter **'Q'**, although any character will do). In the output conversion, return the string without the character prefix. The result is that the comparison of the two variables with this conversion will be based on their stored value. Since the stored value starts with a character, the comparison will be on a character-by-character basis, not as numeric values.

Below is the source:

```
SUBROUTINE NUMTOCHAR_CONV(Type, InVal, Branch, ReturnVal)
/*****

    PURPOSE : Force numbers to strings

    WARNINGS : The output conversion does not check the first character. If
                greater security is desired, check the first character before
                stripping it off.

    THEORY OF OPERATION :
    On ICONV, stuff a non-numeric character on the front of the string. On
    OCONV, strip the first character off.

    This conversion should be used whenever strings that look like numbers need
    to be treated as strings. For example, if you have codes like:

        1 1.0 01.0

    They are treated as being equal if they're seen as numbers. This conversion
    forces the system to treat them as strings, while allowing the user see
    them in their numeric form.

    *****/

if Type = 'ICONV' THEN
    ReturnVal = 'Q' : InVal
end else
    ReturnVal = InVal[2, LEN(InVal)]
end
return
```