Rocket Uniface Library 10.4

Logical Operators

Logical operators take two conditions as operands, and return a Boolean value that expresses whether one, both, or none of the conditions is true.

Table: Logical Operators

Operator	Description	Priority
`!`	Logical NOT	1
`&`	Logical AND	2
`|`	Logical OR	3

Logical operators are used to connect smaller expressions to create a larger expression with multiple conditions. For example, the following code contains three simple expressions:

```
trigger loseFocus
if ( DOCCODE!="REM" & DOCCODE!="QUO" & DOCCODE!="" )
 message "Code not valid!"
 return (-1) ;negative value prevents leaving field
endif
end ; loseFocus
```

The simple expressions `doccode!="REM"`, `DOCCODE!="QUO"`, and `DOCCODE!=""` are combined to make one complex expression. If the data in the DOCCODE field does not match any of the constant strings, Uniface sends the message and prevents the user from leaving the field.

Logical operators have a lower priority than all other operators. When two or more logical operators are used in the same expression, `!` has the highest priority, followed by `&` and then `|`.

Note: This order of precedence conforms to the American National Standards Institute (ANSI) SQL, which gives priority to the AND operator when it is used in combination with the OR operator. ANSI SQL also attributes a higher priority to the NOT operator when used with AND or OR.

Consider the following statement:

```
if ( a & b | c & d )
```

Uniface interprets this as:

```
if ( (a&b) | (c&d) )
```