Rocket Uniface Library 10.4

Operators

Operators perform arithmetic, relational, logical, string manipulation, struct member de-reference, operation de-reference, and field indirection operations in ProcScript instructions.

The following operators can be used in ProcScript expressions.

Table: ProcScript Operators

Type	Operator	Description	Precedence
Indirection	@	Field Indirection	1
De-reference	-> *Identifier* { }	Struct Dereference and Operation Activation Struct Index	2
Extraction	[ ]	Extraction	3
Indirect de-reference	-> "*SubstitutionString*"	Dereference with string substiution	4
Arithmetic Operators	* / %	Multiplication Division Modulus	5
	+ -	Addition Subtraction	6
Relational Operators	< <= != = == >= >	Less than Less than or equal to Not equal to Equal to Equal to Greater than or equal to Greater than	7
Logical Operators	!	Logical NOT	8
	&	Logical AND	9

Type	Operator	Description	Precedence
	|	Logical OR	10

Note: Do not confuse operators with profile characters that can appear as data in fields. Profile characters can be placed in fields during query-by-form operations, either by ProcScript statements or by entering directly.

Precedence

Operators are evaluated in order of precedence, with precedence 1 being the highest. The operator precedence defines which operator takes precedence when Uniface evaluates expressions with several operators. In an expression:

- All operators with a higher precedence are evaluated before those of a lower precedence.
- Operators with the same precedence are evaluated from left to right.
- An expression in (parentheses) is evaluated before the outer part.

For example, the expression:

```
3 * 6 - 4 / 2
```

evaluates to 16, while the following expression evaluates to 3:

```
3 * ( 6 - 4 ) / 2
```

Compound Operators

Compound operators combine the assignment operator (=) with an arithmetic operator. They provide a shorthand notation when performing an arithmetic calculation on the value of a field or variable and assigning the result to the same field or variable. For example:

- `A += 1` is equal to `A = A + 1`
- `A -= 1` is equal to `A = A - 1`
- `A *= 2` is equal to `A = A * 2`
- `A /= 3` is equal to `A = A / 3`
- `A %= 10` is equal to `A = A % 10`

Related concepts

Expressions
Access Operators
Extracting Values From String Data
Substitution in String Values
Fields