Table of contents

Retain variable - RETAIN

Retain variable - RETAIN

Retain variables are declared by the keyword RETAIN is added in programming objects in the scope VAR, VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT, VAR_STAT, or VAR_GLOBAL.

Syntax bei der Deklaration

```
<scope> RETAIN
   <identifier>: <data type> ( := <initialization> )? // ( ... )? : Optional
END_VAR
<scope> : VAR | VAR_INPUT | VAR_OUTPUT | VAR_IN_OUT | VAR_STAT | VAR_GLOBAL
```

Assignment of input, output or memory addresses with the keyword AT is not allowed.

Example

In einer POU:

```
VAR RETAIN
   iVarRetain: INT;
END_VAR
```

In einer GVL:

```
VAR_GLOBAL RETAIN
   g_iVarRetain: INT;
END_VAR
```

Possible declaration locations

Locally in a program	Only the variable lies within the retain memory area. Info: When using redundancy, the entire program with all its data resides in the retain memory area.

Global in a global variable list	Only the variable lies within the retain memory area. Info: When using redundancy, the entire global variable list with all its data can be found in the retain memory area.
Local in a function block	The entire instance of the function block with all of its data lies within the retain memory area. Only the declared retain variable is protected.
Locally in a function	The variable is not within the retain memory area. This declaration does not have any effect.
Local and persistent in a function	The variable is not within the retain memory area. This declaration does not have any effect.

Whenever possible, avoid marking variables of a function block with RETAIN.

Also refer to

- ↘ “Declaring VAR PERSISTENT Variables ”
- ↘ “Command 'Add all instance paths'”
- ↘ “Preserving Data with Retain Variables”