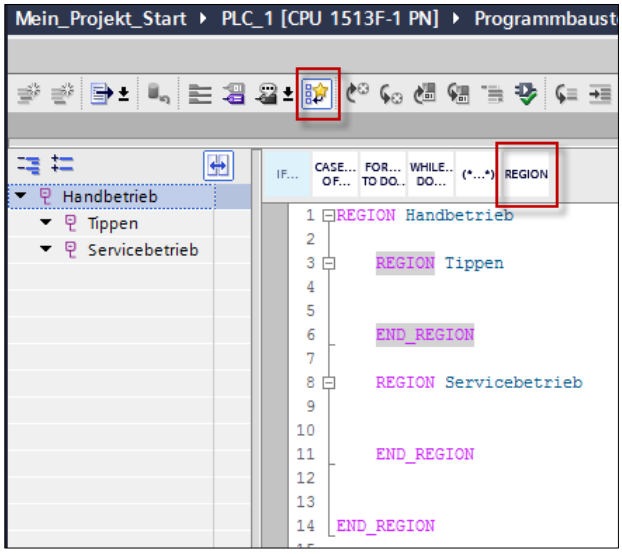
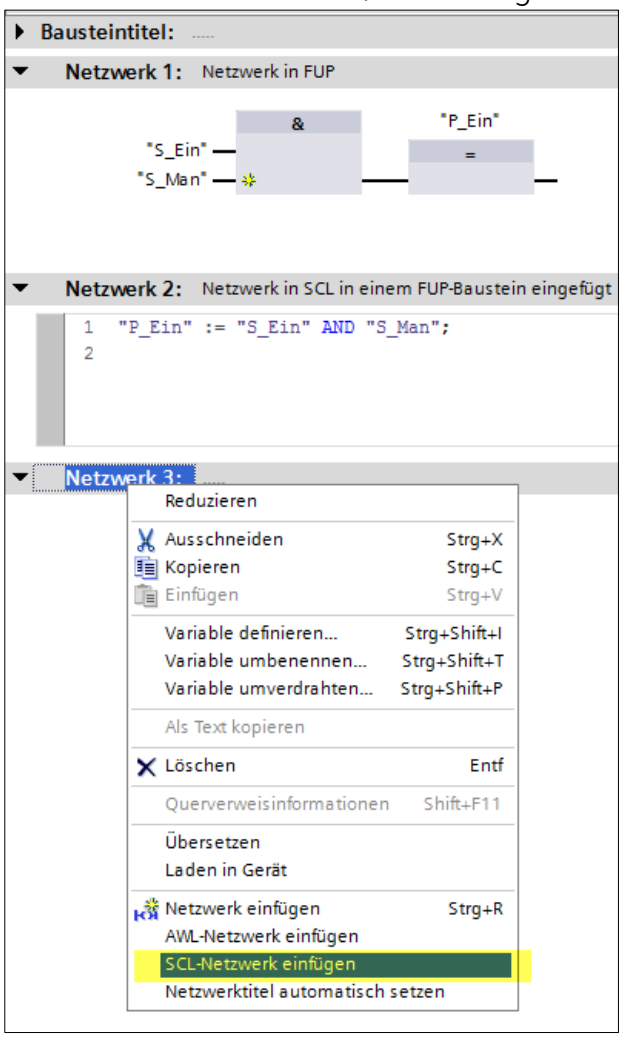
Neuerungen im SCL ab V14 SP1

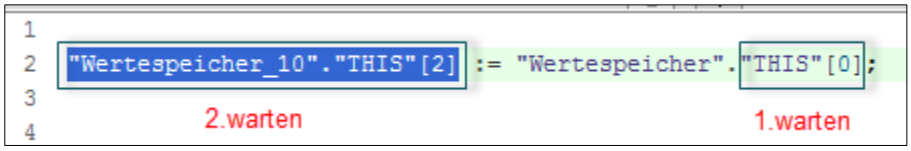
1. "Regions" zur besseren Strukturierung und Navigation



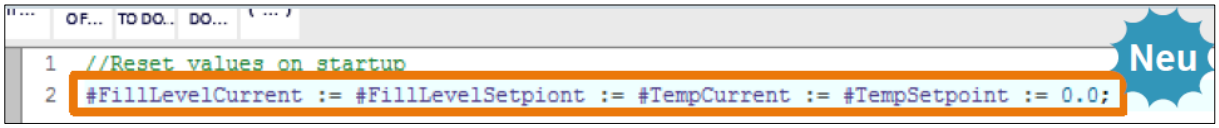
2. SCL-Netzwerke in KOP / FUP einfügen



3. strukturierte Variablen und ARRAY-Elemente per Drag&Drop ersetzen



4. Mehrfachzuweisungen



5. Umschalten zwischen Einfüge- und Überschreibemodus

Einfügemodus aktiv (normal)

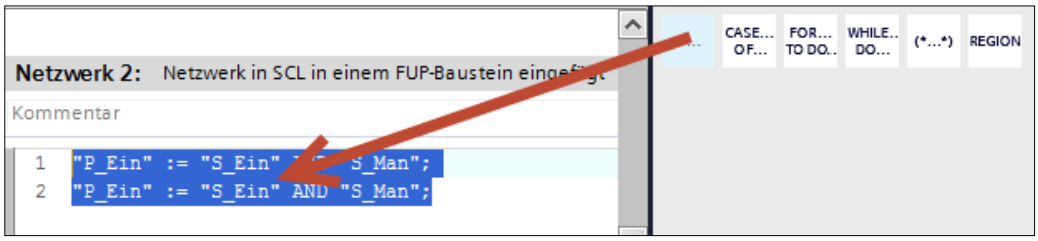
```
"P_Ein" := "S_Ein" AND "S_Man";
```

Überschreibmodus aktiv

```
"P_Ein" := "S_Ein" AND "S_Man";
```

Die Umschaltung erfolgt mit der Taste „Insert“

6. Selektierten Programmcode mit Strukturelementen umschliessen.




```
IF _condition_ THEN
    "P_Ein" := "S_Ein" AND "S_Man";
    "P_Ein" := "S_Ein" AND "S_Man";
END_IF;
```

7. STRG + Klick für Gehe zu Definition

```
END_IF;

#temp1 := "DeInkrement"(iValue := #temp1 , bDekrement := #toggle);
"dq_outb0" := #temp1;
"dq_outb0" := MIN(IN1 := "dq_outb0", IN2 := #temp1);
```

Neu

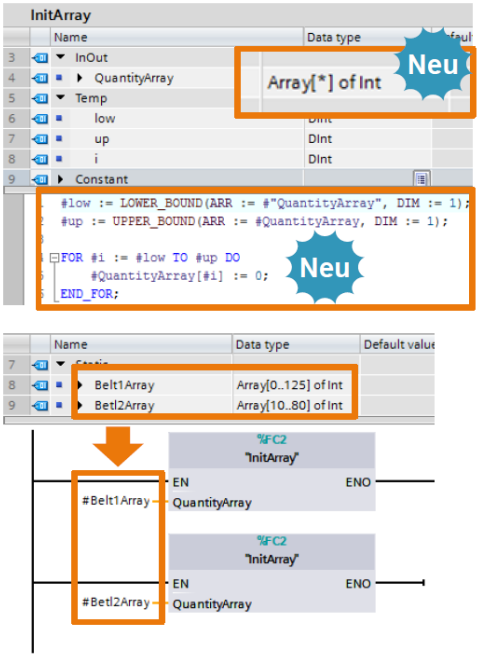
8. Array(*) zur Übergabe variabel langer Arrays

Funktionen

- Array(*) als neuer Parameter-Datentyp
- Neue Funktionen LOWER_BOUND, UPPER_BOUND zur Ermittlung der Array-Grenzen
- Für AWL, SCL, KOP, FUP

Nutzen

- Keine Pointer-Programmierung für Arrays unterschiedlicher Länge mehr notwendig
- Bessere Laufzeit Performance als Pointer (Any/Variant) und kopieren mittels MOVE_BLK
- Vollsymbolisch und bessere Lesbarkeit
- Ermöglicht generische Standardfunktionen für Arrays unterschiedlicher Länge.



9. Neue Arithmetik-Funktionen

+=, -=, *=, /=

```
#PieceCount := #PieceCount + 1;
#PieceCount += 1;

#Factor := #Factor / 2;
#Factor /= 2;
```

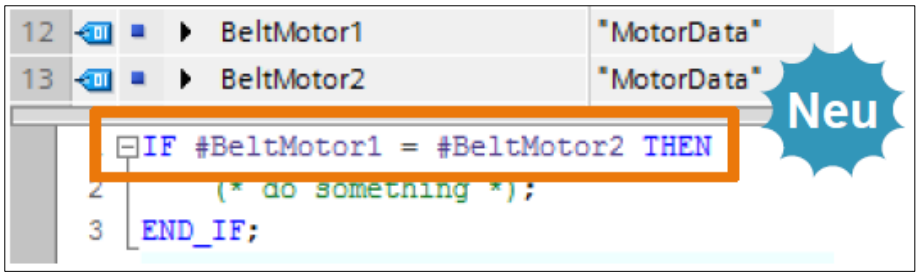
Neu

10. Konstanten für Strings

Bei der Deklaration eines Strings können jetzt auch Konstanten (lokal, global) für die Länge verwendet werden.

Neu

11. Vergleich von UDT-Variablen



12. Quellen exportieren