





Standaardwerk naar TIA Portal Converter - Project Documentatie

Project Overzicht

Doel

Een converter ontwikkelen die Standaardwerk-stappenprogramma's kan omzetten naar Siemens TIA Portal functieblokken. Het project wordt stapsgewijs ontwikkeld, beginnend met een parser/validator om de syntax te verfijnen.

Huidige Status

-  **Fase 1:** Basis parser/validator webapp ontwikkeld
-  **Fase 2:** Syntax verfijning en multi-programma ondersteuning (in uitvoering)
-  **Fase 3:** TIA Portal conversie engine
-  **Fase 4:** Volledige converter webapp

Standaardwerk Syntax Analyse

Cruciale Ontdekking: Voorwaarden Logica

BELANGRIJK: Aanvankelijk werd gedacht dat voorwaarden onder een stap bij die stap hoorden, maar dit is **ONJUIST**.

Correcte logica:

- RUST heeft GEEN voorwaarden (het is de startpositie)
- Voorwaarden onder RUST zijn voorwaarden om van RUST naar STAP 1 te gaan
- Voorwaarden onder STAP 1 zijn voorwaarden om van STAP 1 naar STAP 2 te gaan
- Etc.

TIA Portal implementatie bevestigt dit:

Network 2: RUHE stap (SR latch gereset door STAP 1)

Network 3: STAP 1 (wordt geset door RUHE EN voorwaarden die onder RUHE stonden)

Syntax Elementen (Geïdentificeerd)

Programma Structuur

[Programma Naam] [FB Nummer]
Symbool IDB: [Instance_DB_Naam]

RUST: [Beschrijving]
[Voorwaarden voor STAP 1]

STAP 1: [Beschrijving]
[Voorwaarden voor STAP 2]

STAP X: Fertig/Klaar

Voorwaarden Types

1. **Basis voorwaarden:** Eenvoudige statements
2. **OR logica:** Regels beginnend met
3. **Negatie:** prefix
4. **Externe referenties:**
5. **Timer condities:**
6. **Vergelijkingen:** , , , , , ,
7. **Sensoren:**
8. **Buttons:**

Variabele Definities

[Variabele Naam] =
[Voorwaarden voor deze variabele]

TIA Portal Mapping (Uit Analyse)

- **Stappen:** array
- **Hulpmerkers:** array voor variabelen
- **Timers:** array
- **Networks:** Elke stap wordt een aparte network met SR latch logica

Huidige Parser Implementatie

Architectuur

javascript

```
parseStandaardwerk(code) {  
  // Parses multiple programs  
  // Returns structured data with steps, variables, validation  
}
```

Data Structuur

javascript

```
{  
  programName: string,  
  functionBlock: string,  
  symbolikIDB: string,  
  steps: [{  
    type: 'RUST' | 'STAP',  
    number: number,  
    description: string,  
    transitionConditions: [ConditionGroup],  
    lineNumber: number  
  }],  
  variables: [{  
    name: string,  
    conditions: [Condition],  
    lineNumber: number  
  }],  
  errors: [ValidationError],  
  warnings: [ValidationWarning]  
}
```

Validatie Regels

1. RUST stap verplicht
2. RUST mag geen transition conditions hebben
3. Fertig/Klaar stap aanbevolen
4. Stap nummering controle
5. Externe referentie validatie (toekomstig)

Volgende Ontwikkelingsstappen

Prioriteit 1: Syntax Verfijning

- ☐ Aparte configureerbare syntax rules sectie
- ☐ Meer gedetailleerde parsing van condition types

- ☐ Edge cases identificeren en oplossen

Prioriteit 2: Multi-Programma Ondersteuning

- ☐ Parser uitbreiden voor meerdere programma's in één bestand
- ☐ Cross-referentie validatie tussen programma's
- ☐ Programma dependency analyse

Prioriteit 3: TIA Portal Conversie Voorbereiding

- ☐ Template systeem voor TIA Portal networks
- ☐ Mapping van Standaardwerk elementen naar TIA componenten
- ☐ Code generatie engine

Technische Implementatie Details

Huidige Parser Features

javascript

// Robuuste error handling

try-catch blokken overall

safeArray() en safeGet() helper functies

// Syntax parsing

- Nederlandse/Duitse terminologie ondersteuning
- Flexible condition parsing
- External reference detection
- Timer condition parsing

Code Structuur

StandaardwerkParser Component





- ├─ parseStandaardwerk() - Main parser
- ├─ parseCondition() - Condition parsing
- ├─ parseConditionDetails() - Detailed analysis
- ├─ validateStructure() - Validation rules
- └─ renderCondition() - UI rendering

Bekende Issues & Oplossingen

Issues Opgelost

1. ☒ Undefined array access - Opgelost met safe array functions
2. ☒ Incorrect condition assignment - Opgelost met pendingConditions logic
3. ☒ Missing error handling - Comprehensive try-catch added

Nog Te Oplossen

1.  Complete syntax specification neddig
2.  Multi-program parsing
3.  Cross-reference validation
4.  TIA Portal template mapping

Standaardwerk Voorbeeld (Real World)

L01: Hoofdprogramma Kraan FB300

Symbool IDB: L01_Hoofdprogramma_kraan

RUST: L01: Hoofdprogramma kraan

Kraan beweging gewenst

Geen kraan route actief of gestart

STAP 1: Kraan naar losstation?

Ja, kooi is bijna leeg:

Aanvraag lege kraan naar losstation

NIET Aanvraag kooi van losstation

STAP 2: Start kraan zonder kooi naar losstation

Geen aanvraag of aanvraag bevestigd (*L01: Kooi status losstation L30* NIET STAP 9-10)

Geen aanvraag of aanvraag bevestigd (*L01: Kooi status losstation L31* NIET STAP 9-10)

// ... meer stappen

TIA Portal Target Structure

FB300: L01 Hoofdprogramma Kraan

- ├── Static Variables
 - | ├── Stap[0..51] of Bool
 - | ├── Hulp[1..32] of Bool
 - | └── Tijd[1..10] of IEC_TIMER
- ├── Networks
 - | ├── Network 1: RUST logic (SR latch)
 - | ├── Network 2: STAP 1 logic (SR latch + conditions)
 - | └── ... (one per step)
- └── Variable assignments (Hulp[] calculations)

Development Roadmap

Korte Termijn (1-2 weken)

1. Syntax rules configuratie systeem
2. Multi-programma parsing
3. Cross-reference validatie
4. Edge case identificatie

Middellange Termijn (1 maand)

1. TIA Portal template engine
2. Code generatie proof-of-concept
3. Export functionaliteit
4. Batch processing

Lange Termijn (2-3 maanden)

1. Volledige converter webapp
2. Import/export van TIA Portal bestanden
3. Reverse engineering (TIA → Standaardwerk)
4. Integration met TIA Portal API

Belangrijke Aandachtspunten

Voor Syntax Definitie

- **Consistent gebruik** van Nederlandse vs Duitse termen
- **Whitespace regels** - tabs vs spaties
- **Case sensitivity** - hoofdletters vs kleine letters
- **Special characters** in namen en referenties
- **Comment syntax** - als die bestaat

Voor TIA Portal Mapping

- **Data types** - Bool, Int, Time, etc.
- **Memory layout** - hoe arrays worden georganiseerd
- **Network optimization** - redundante logica vermijden
- **Performance** - grote programma's efficiënt maken

Voor Validation

- **Circular references** detectie
- **Dead code** identificatie
- **Unreachable steps** detectie

- **Missing dependencies** waarschuwingen

Contact & Handoff Info

- **Parser Locatie:** Artifact `standaardwerk-parser`
- **Laatste Status:** Basis parser werkend, ready voor uitbreiding
- **Test Data:** Nederlandse Standaardwerk syntax in parser
- **Kritieke Discovery:** Voorwaarden horen bij VOLGENDE stap, niet huidige stap

Documentatie laatst bijgewerkt: [Vandaag] Voor nieuwe gesprekken: Begin met deze documentatie en verwijst naar artifact `standaardwerk-parser`