



BOS voor (gebieds)besturing

Behorend bij Aquaview-X

Revisies

234700.000, 24-01-2024
Eerste uitgifte
234700.001, 07-02-2024
Revisielijst toegevoegd

Overzicht van de laatste revisies

Inhoudsopgave

Inhoudsopgave	1
1 BOS-module	3
1.1 Definities	3
1.2 Aquaview	3
1.3 Bedrijfsmodes	4
1.4 Aandachtspunten	5
2 Eenvoudige besturing	7
2.1 Aanmaken en/of verwijderen	7
2.2 Samenstellen van een eenvoudige besturing	8
2.3 Beschikbare blokken	8
2.4 Voorbeeld 1	9
2.5 Voorbeeld 2	10
2.6 Voorbeeld 3	10
3 Complexe besturing	13
3.1 Hulpvariabelen en getallen	13
3.2 Logische bewerkingen	13
4 Functies	15
4.1 @START_SCRIPTFORREGION / @END_SCRIPTFORREGION	15
4.2 @EVERY / @ENDEVERY	16
4.3 @SENDTOTAG	17
4.4 @TAGTOSEND	18
4.5 @BLOCKWEB	18
4.6 @BESYSLIGHT	19
4.7 @LOG	22
4.8 @FROMHIS	23
4.9 @TOHIS	24
4.10 @PROC / @ENDPROC	25
5 Samengestelde alarmen	27
5.1 @GET / @LOAD	28
5.2 @AND / @OR	28
5.3 @ANDGRT	28
5.4 ANDEQU, ANDLES, ORGRT, OREQU en ORLES	28
5.5 SET	28
5.6 STORE	29
5.7 Voorbeelden	29
Index	31

1 BOS-module

De BOS-module (Beslissings Ondersteunend Systeem) van Aquaview maakt het mogelijk om interactie tussen verschillende stations te automatiseren. Dit soort interactie is door (gebieds)afhankelijkheden altijd uniek en de ontwikkeling hiervan vereist een goed inzicht van de lokale omstandigheden en stuurparameters.

Op basis van deze kennis worden eisen en wensen van een BOS vastgelegd in een stuk programmacode wat op een ingesteld interval zal worden gestart. Hierbij worden indien nodig de benodigde gegevens bij stations opgevraagd of stuurparameters (setpoints) naar stations verstuurd waarbij het complete proces en de betrokken stations visueel inzichtelijk wordt gemaakt in afbeeldingen.

Voor complexe besturingen is de ontwikkeling meestal maatwerk. Programmeurs schrijven de benodigde programmacode en maken samen met de eindgebruikers bijpassende afbeeldingen. Voor eenvoudigere besturingen is er de mogelijkheid van 'Simpel Sturing'. Hierbij worden visueel de benodigde parameters, stations en relaties bij elkaar geklikt en verbonden. De bijbehorende programmacode en een afbeelding worden vervolgens automatisch gegenereerd.

Deze handleiding behandelt alle mogelijkheden van de BOS-module. Hierbij wordt eerst de mogelijkheid van de Simpele Sturing besproken en zullen vervolgens voor de complexe sturingen alle programmafuncties benoemd worden.

1.1 Definities

In deze documentatie wordt gebruik gemaakt van een aantal definities en termen:

Beslissing Ondersteunend Systeem (BOS)

Een schema of handleiding dat behulpzaam is bij het nemen van beslissingen, vaak geautomatiseerd.

Bron: Informatiehuis Water (AQUO).

Besys

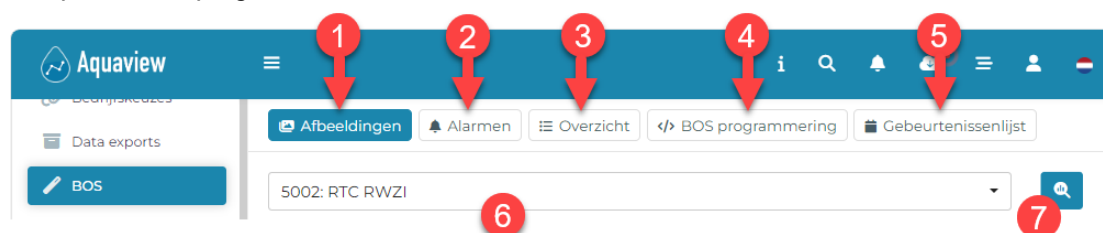
De naam van het Aquaview systeemonderdeel welke op de achtergrond de BOS-besturing uitvoert en acties in gang zet.

Real-Time Control (RTC)

De bedrijfsmode waarbij een station vanuit de BOS-besturing aangestuurd wordt.

1.2 Aquaview

De BOS-module heeft een eigen plaats in Aquaview. Hier worden visueel de aanwezige BOS-besturingen gepresenteerd, kunnen 'Simpel Besturingen' (p.7) worden bewerkt en kan de complexe BOS-programmacode worden bekeken.



Hierin is:

1. **Afbeeldingen**

Overzicht van de BOS-besturing(en), waarbij diverse parameters aangepast kunnen worden. Voor complexe BOS-programmering worden deze afbeeldingen 'op maat' gemaakt, voor simpele sturing worden deze automatisch gegenereerd. Om onderscheid te kunnen maken tussen verschillende BOS-programma's worden deze elk opgeslagen onder een (virtueel) station.

2. **Alarmen**

Overzicht van (samengestelde) alarmen vanuit de BOS-besturing.

3. **Overzicht**

Hier wordt alle programmacode weergegeven. De verschillende sturingen staan hierin onder elkaar uitgeschreven en met gekleurde bolletjes voor elke regel wordt statusinformatie van de regel aangegeven.

4. **BOS programmering**

Geeft een overzicht van de gedefinieerde simpele sturingen en is het startpunt om deze te bewerken, verwijderen of toe te voegen.

5. **Gebeurtenissenlijst**

Een overzicht van alle door BOS aangestuurde gebeurtenissen.

6. **Opslaglocatie (virtuele locatie)**

Als er meerdere onafhankelijke besturingen zijn dan worden deze weergegeven in verschillende afbeeldingen. Selecteer hier voor welke BOS-besturing de afbeelding moet worden weergegeven.

7. **Analyse**

Analyseer aan de hand van opgeslagen historische gegevens hoe de BOS-besturing in een opgegeven periode heeft gefunctioneerd.

1.3 Bedrijfsmodes

Voor de werking van BOS wordt gebruik gemaakt van een bedrijfsmode. Deze kan in de statusafbeelding van het desbetreffende station of vanuit het menu onder de kop 'Bedrijfskeuzes' worden aangepast. Het gebruik van het menu is vooral nuttig voor het aanpassen van de bedrijfsstatus van meerdere stations.

De ingestelde bedrijfskeuze is van belang voor de aansturing vanuit BOS:

- **RTC** De BOS-aansturing (**RTC**) is dominant en zal centraal blokkeringen en deblokkeringen versturen.
- **AUTO** De BOS-aansturing is niet langer dominant maar zal wel automatisch (blijven) deblokken. Verder functioneert het station **AUTO**noom.
- **HAND** De BOS-aansturing wordt gestopt. Alles blijft hierbij in de laatste stand staan. De gebruiker kan nu zelf de situatie **HAND**matig manipuleren.

LET OP:

- Om lokaal in te grijpen moet de installatie altijd eerst op "HAND" gezet worden.
 - Bij de gebruikersaccountgegevens kunnen gebruikers onder 'Instellingen' ervoor kiezen om stations die op de bedrijfskeuze 'AUTO' ingesteld staan te verbergen in het bedrijfskeuzescherf. Deze instelling heeft geen invloed op de beschikbaarheid van de bedrijfskeuze vanuit de statusafbeeldingen.
-

1.4 Aandachtspunten

Het schrijven van BOS-programmacode is niet voor iedereen weggelegd en vereist naast de kennis van de BOS-programmeerfunctionaliteit ook veel programmeursinzicht. Daarom is er de mogelijkheid 'Simpel Sturing' (p.7) waarmee grafisch een BOS sturing kan worden ontworpen. Kennis van de programmeertaal is in dat geval niet nodig, procesinzicht is wel vereist.



WAARSCHUWING:

Het is belangrijk om BOS-sturingen goed te testen op gewenste functionaliteit, potentiële fouten en risico's bij een verkeerde aansturing. Aquaview zal letterlijk uitvoeren wat er is ingesteld. Als er meerdere BOS-sturingen aanwezig zijn die een identiek object aansturen dan wordt alleen de laatste actie uitgevoerd. Als dus een eerst doorlopen BOS-sturing deze locatie blokkeert, maar een latere BOS-sturing deblokkeert hem, dan wordt de deblokking uiteindelijk naar het station verzonden.

LET OP:

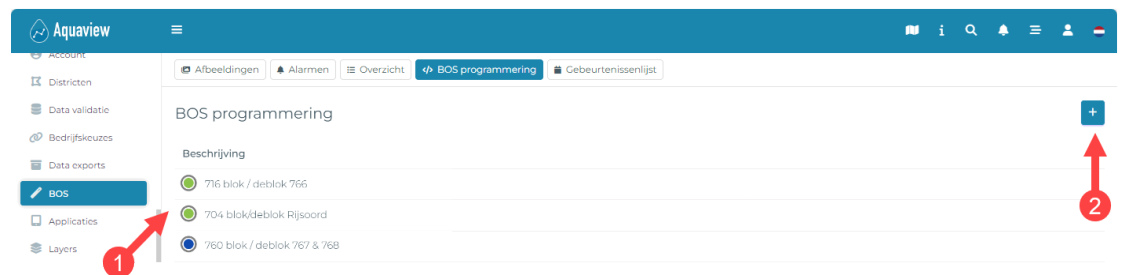
- Elke minuut worden de simpele sturingen doorgerekend.
 - Vanuit BOS wordt op intervalbasis (10 seconden) gestuurd. Voor een niet-Xylem gemaal wordt indien nodig eerst de actuele status van de locatie opgevraagd en zal pas na 10 intervallen (=100 seconden) de (de)blokkeringsactie volgen.
 - De snelheid waarmee gegevens van een station kunnen worden opgehaald of verstuurd hangt samen met het type onderstation en communicatie instellingen.
-

2 Eenvoudige besturing

De eenvoudige besturingsmogelijkheden zorgen ervoor dat ook zonder kennis van de programmeertaal op een visuele manier een besturing kan worden ontworpen. Kennis van het (gedrag van het) stelsel is hierbij wel van belang. Ga hiervoor naar het scherm 'BOS programmering'.

2.1 Aanmaken en/of verwijderen

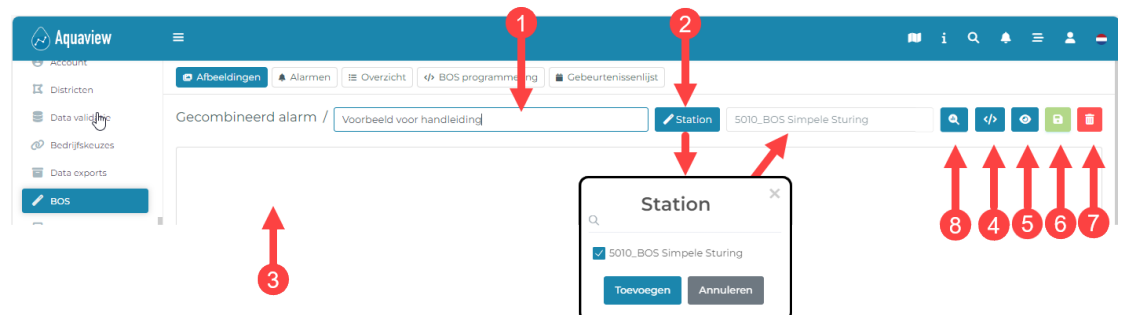
In het BOS programmering scherm staan alle opgeslagen eenvoudige besturingen. Klik op een regel om een eenvoudige besturing te openen, of op de '+'-knop om een nieuwe eenvoudige besturing aan te maken.



Hierin is:

1. Lijst met alle beschikbare eenvoudige besturingen.
De symboolkleur geeft weer of de betreffende eenvoudige besturing op dit moment zorgt voor een actieve blokkering. De symboolkleur wordt bepaald door de (persoonlijke) alarmstatuskleurinstellingen.
2. Knop voor het toevoegen van een nieuwe eenvoudige besturing.

Als een eenvoudige besturing wordt geopend dan verschijnt onderstaand scherm met daarin een aantal belangrijke elementen welke in onderstaande afbeelding worden benoemd.



Hierin is:

1. Een zelf op te geven naam voor de eenvoudige besturing. Deze moet uniek zijn.
2. Selecteer met deze knop onder welk (virtueel) station de eenvoudige besturing moet worden opgeslagen.
3. Het werkblad waarop de eenvoudige besturing wordt samengesteld.
4. Schakelknop om te switchen van de eenvoudige besturing naar complexe sturing. Dit is nog niet functioneel.
5. Met deze knop kan de gegenereerde programmacode worden bekeken.
6. Bewaar de eenvoudige besturing
7. Verwijder de eenvoudige besturing.
8. Analyseer aan de hand van opgeslagen historische gegevens hoe de eenvoudige sturing in een opgegeven periode heeft gefunctioneerd.

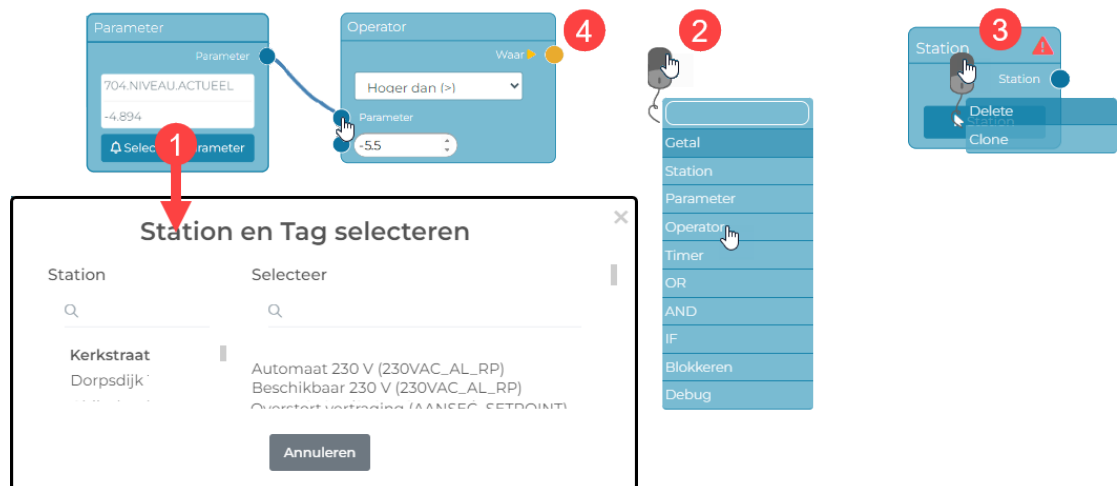
LET OP:

- Als de naam van de eenvoudige besturing (1) of het station (2) niet zijn ingevuld dan worden deze vakken rood omrand weergegeven om te wijzen op deze fout. Het is dan niet mogelijk om de besturing op te slaan.
- Vergeet niet om een gewijzigde eenvoudige besturing op te slaan.

2.2 Samenstellen van een eenvoudige besturing

Een eenvoudige besturing laat zich het beste vergelijken met een stroomdiagram, waarbij van links naar rechts blokken worden geplaatst die afhankelijk van de uitkomst van elk blok onderlinge relaties hebben. Klik met de rechtermuisknop op het werkblad om een blok toe te voegen en breng relaties aan door een in- of uitgang van een blok (weergegeven als bolletje) met de muis te selecteren en onderling te verbinden. Als blokken een koppeling hebben met een parameter dan zit er een knop op een dergelijk blok om een selectiedialoog te openen. Zoek daarin bij het gewenste station de betreffende parameter om deze te selecteren.

Al geplaatste blokken kunnen gedupliceerd worden (klonen) of verwijderd. Klik daarvoor met de rechtermuisknop op het betreffende blok en selecteer de gewenste actie.



Hierin is

1. Selecteren van het betreffende station en daarbij behorende tag.
2. Toevoegen van een nieuw blok.
3. Verwijderen of klonen van een bestaand blok.
4. De huidige situatie wordt weergegeven met oranje tekst en symbool. Dit geeft direct inzicht wat **op dit moment** de status is en welke gevolgen zoals blokkeringen er zijn.

LET OP:

- Als er bij een blok een rood waarschuwingssymbool met een '!' staat dan is er iets aan de hand met dat blok. Er mist dan bijvoorbeeld een instelling of een verbinding. Door deze fout(en) kan er geen geldige programmacode worden gegenereerd en kan daarom de eenvoudige besturing niet worden opgeslagen.

2.3 Beschikbare blokken

- **Getal**

Bij diverse blokken zoals een operator of timer kan een harde getalswaarde worden ingevoerd. Als er echter meerdere blokken zijn die allemaal aan dezelfde waarde moeten worden gekoppeld kan er ook een getalblok worden toegevoegd welke aan deze verschillende 'waarde' -instellingen wordt gekoppeld. Zo hoeft een wijziging van deze waarde nog maar op 1 plek te worden doorgevoerd.

- **Station**

Het stationsblok wordt gebruikt om een station aan een blokkeerblok te koppelen. Selecteer het betreffende station met de knop in het blok.

- **Parameter**

Het parameterblok specificeert een specifieke parameter welke gekoppeld kan worden aan een operatorblok om te toetsen of er aan een (de)blokkeringseis is voldaan.

- **Operator**

Dit blok kan worden gebruikt om een waarde of parameter te toetsen.

- **Timer**

Een timer kan worden gestart of gestopt als er aan bepaalde condities wordt voldaan. Daarmee laat de timer zich vergelijken met een kookwekker. Als de timer niet deze condities wordt gestopt dan loopt deze uiteindelijk af. Dit noemen we 'Timeout'. Aan de timeout kan een actie worden gekoppeld.

LET OP:

De timer wordt opgegeven in minuten.

- **OR**

Geeft als resultaat 'Waar' als er 1 van de beide testcondities is voldaan.

- **AND**

Geeft als resultaat 'Waar' als aan beide testcondities wordt voldaan.

- **IF**

Geeft het resultaat van de gekoppelde test.

- **Blokken**

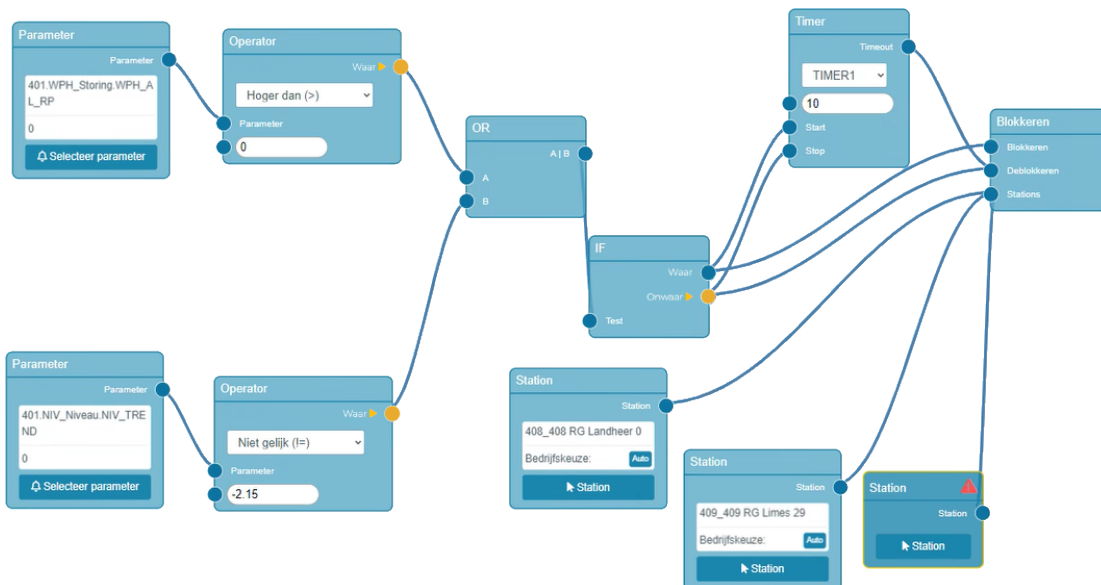
Koppel dit blok aan een conditietest die aangeeft of er locaties geblokkeerd of gedeblokkeerd moeten worden en voeg alle betrokken stations hieraan toe.

- **Debug**

Gebruik dit blok om de actuele waarde van bijvoorbeeld een parameter of operator uitkomst tijdens testen beter te kunnen volgen.

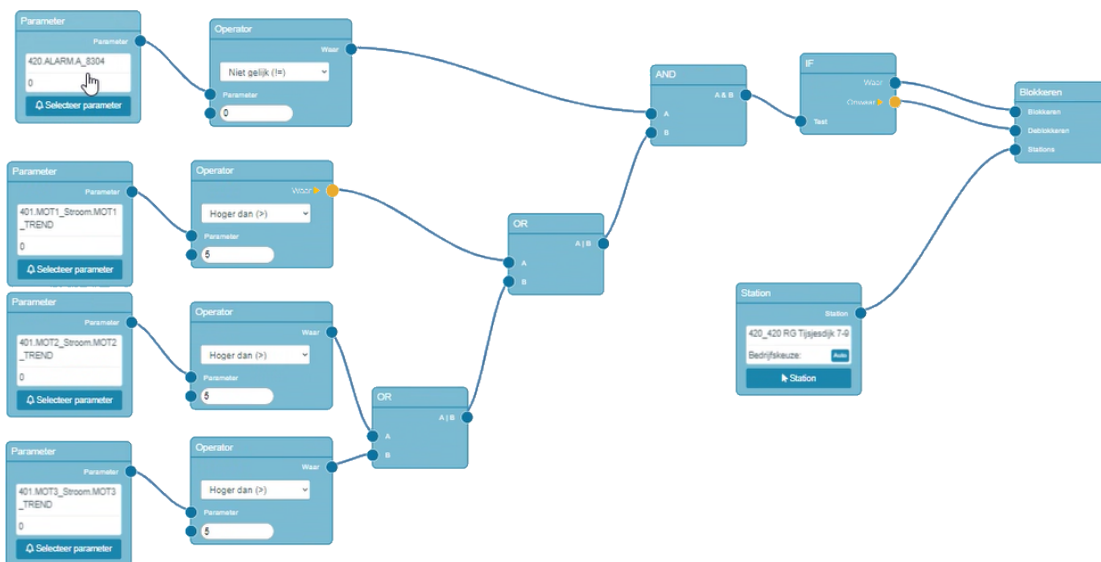
2.4 Voorbeeld 1

Als gemaal A in storing gaat, of er wordt een bepaald niveau bereikt dan worden de gemalen B, C en D een opgegeven tijd geblokkeerd. In onderstaand voorbeeld is station D nog niet ingesteld. Het waarschuwingssymbool geeft deze fout aan. Pas na een stationskeuze kan dit voorbeeld opgeslagen worden.



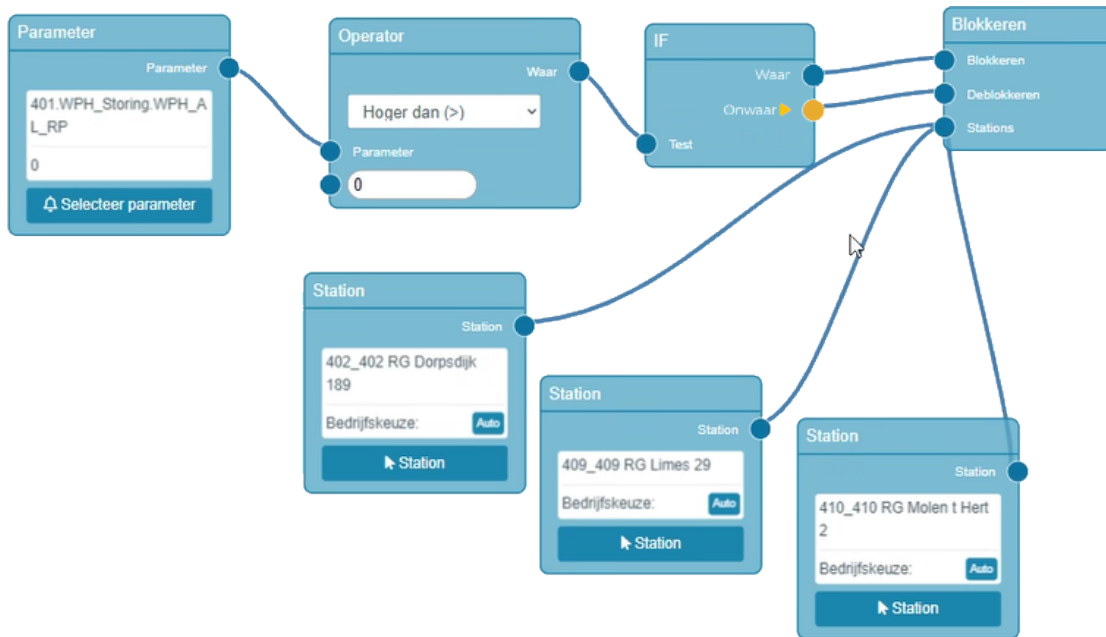
2.5 Voorbeeld 2

Als gemaal A draait, dan wordt gemaal B geblokkeerd, tenzij B een hoogwatermelding krijgt.



2.6 Voorbeeld 3

Als het niveau in gemaal / verzamelriool A te hoog wordt dan worden de gemalen B, C en D geblokkeerd



3 Complexe besturing

Als de complete BOS-besturing met de hand uitgewerkt wordt dan noemen we dit de 'complexe besturing'. Dit type besturing kan alleen bewerkt worden met de ConfigTool en vereist gedetailleerde kennis.

3.1 Hulpvariabelen en getallen

Er kan gebruik worden gemaakt van hulpvariabelen en getallen. Het formaat hiervan is als volgt:

Formaat	Toelichting
#xxxxx{.xxxxx}	Getallen worden voorafgegaan door een '#'. Getallen zijn hierbij floating points, decimalen zijn toegestaan.
<Hulpvar>	Hulpvariabelen worden voorafgegaan door een ''. Een hulpvariabele mag maximaal 20 karakters bevatten en is hoofdlettergevoelig. Bij het allereerste gebruik (bijvoorbeeld na het bewerken van de programmacode) zijn alle hulpvariabelen 0. Na elke RUN van BOS worden de waarden opgeslagen en vanaf dat moment gebruikt.

3.2 Logische bewerkingen

In de programmacode kan gebruik gemaakt worden van de volgende logische bewerkingen:

Bewerking	Betekenis
IF..ELSE..ENDIF	Test. Wanneer de bewerking na 'IF' de status 'TRUE' (= ongelijk aan 0) oplevert, wordt alles wat hierna komt uitgevoerd. Is de status 'FALSE' dan volgt het (optionele) deel achter 'ELSE'. Het blok wordt afgesloten met 'ENDIF'
ORIF	Logische of. Wordt gebruikt om extra testen toe te voegen na een IF. Staat altijd aan het begin van de regel
ANDIF	Logische en. Wordt gebruikt om extra testen toe te voegen na een IF. Staat altijd aan het begin van de regel
:=	Toekenning. De variabele links van de operator krijgt het resultaat toegekend van de bewerking rechts van de operator.
>	Groter dan
>=	Groter of gelijk aan
==	Gelijk aan
<=	Kleiner of gelijk aan
<	Kleiner dan
<>, !=	Ongelijk aan
//	Commentaar. Alles hierna op dezelfde regel wordt verwaarloosd.

Wanneer meerdere bewerkingen achter 'IF' of 'ELSE' nodig zijn, worden deze worden deze elk op een nieuwe regel geplaatst.

Voorbeeld

```
IF 401.WPH_Storing.WPH_AL_RP > #0
```

```
ORIF 401.NIV_Niveau.NIV_TREND > #-2.2
  //...
ELSE
  //...
ENDIF
```


4 Functies

Voor het programmeren van een BOS-besturing wordt er gebruik gemaakt van diverse functies. Onderstaande tabel geeft een globaal overzicht hiervan. De specifieke functies worden in detail in de volgende paragrafen besproken.

Functie	Toelichting	Pag.
@EVERY	Aanroep BeSys regels op tijdbasis	16
@LOG	Logging Event naar gebeurtenissenlijst	22
@TOHIS	Schrijven tagwaarde(n) naar trend (datum/tijd & waarde)	24
@FROMHIS	Bepalen actuele tagwaarde uit trend reeks	23
@SENDTOTAG	Waarden uit setpoint database kopiëren naar tagdatabase	17
@TAGTOSEND	Waarden uit tagdatabase kopiëren naar setpoint database	18
@BLOCKWEB	Webservice interface blokkeren locatie bijvoorbeeld waterschap → gemeente	18
@BESYSLIGHT	Besys light functionaliteit.	19
@PROC		25
@ENDPROC		
@START_SCRIPTFORREGION		15
@END_SCRIPTFORREGION		

LET OP:

Tenzij anders aangegeven moeten parameters bij een functie altijd ingevuld zijn anders wordt de functie niet uitgevoerd.

4.1 @START_SCRIPTFORREGION / @END_SCRIPTFORREGION functionaliteit

Functie om aan te geven dat de tussenliggende code bij een bepaald gebied/groep hoort. Besys negeert deze functionaliteit omdat alle code door Besys moet worden uitgevoerd. Maar in de BeysMonitor en BesysEditor is de tussenliggende code alleen zichtbaar als de ingelogde gebruiker toegang heeft tot de meegegeven groep.

parameters

```
@START_SCRIPTFORREGION (xx)
....
@END_SCRIPTFORREGION(xx)
```

	Parameter	Toelichting
1	xx	appgroep id van de groep

LET OP:

Het appgroup id dient te bestaan en moet zowel bij de start als bij het eind opgegeven worden.

**WAARSCHUWING:**

Deze functies mogen niet genest worden. Dit is dus NIET toegestaan:

```

1 | @START_SCRIPTFORREGION (xx)
2 | ...
3 |   @START_SCRIPTFORREGION (yy)
4 |   ...
5 |   @END_SCRIPTFORREGION(yy)
6 |   ...
7 | @END_SCRIPTFORREGION(xx)

```

Er mogen meerdere @START_-/@ENDSCRIPTFORREGION blokken in de Besysregels voorkomen, en eventueel ook meerdere keren met dezelfde appgroup id.

voorbeelden

In onderstaand voorbeeld staan algemene regels, en blokken specifiek voor Appgroup 12 en 13. Gebruikers met toegang tot beide appgroepen kunnen alle regels zien. Gebruikers die niet aan elke Appgroup zijn toegewezen krijgen alleen de algemene delen en de delen voor hun eigen Appgroup te zien (exclusief de @START_-/@ENDSCRIPTFORREGION regels).

Alle regels**Alleen 12****Alleen 13**

1
2	Regels A	Regels A	Regels A
3
4	@START_SCRIPTFORREGION (12)		
5
6	Regels B	Regels B	Regels C
7
8	@END_SCRIPTFORREGION(12)		
9
10	Regels C	Regels C	Regels D
11
12	@START_SCRIPTFORREGION (13)		
13
14	Regels D	Regels E	Regels E
15
16	@END_SCRIPTFORREGION(13)		
17		
18	Regels E		
19		

4.2 @EVERY / @ENDEVERY

functionaliteit

Functie om in te stellen om de hoeveel seconden een Besys (deel) programma moet worden uitgevoerd.

parameters

```
@EVERY (Interval)
```

	Parameter	Toelichting
1	Interval	Intervaltijd in seconden waarop het tussenliggende codeblok zal worden uitgevoerd. Het is ook mogelijk om het interval op te geven in 'uu:mm' en 'uu:mm:ss'. Dit wordt automatisch omgezet naar seconden.

voorbeelden

In onderstaand voorbeeld worden twee codeblokken opgegeven met elk een eigen intervaltijd.

```

1 | @EVERY (00:05:00)
2 | .....
3 | regels A
4 | .....
5 | @ENDEVERY
6 |
7 | @EVERY (10)
8 | .....
9 | regels B
10| .....
11| @ENDEVERY

```

De regels A worden om de 300 seconden uitgevoerd, de regels B elke 10 seconden.

4.3 @SENDTOTAG

functionaliteit

Functie om de actuele setpoint-waarde van één specifieke tag, of *alle* tags te kopiëren naar de actuele waarde van de tag(s).

parameters

```
@SENDTOTAG (ScadaId [, item.subitem])
```

	Parameter	Toelichting
1	ScadaId	Scada Id van locatie
2	item.subitem	item en subitem van de tag Deze parameter is optioneel: <ul style="list-style-type: none"> • Opgegeven alleen specifieke tag • Niet opgegeven alle tags

voorbeelden

Voorbeeld 1

Zoek de setpoint waarde op van de tag met ScadaId = 301 en item.subitem = LEVEL1.PARA en kopieer deze waarde naar de actuele waarde van de betreffende tag.

```
1 | @SENDTOTAG (301, LEVEL1.PARA)
```

Voorbeeld 2

Zoek de setpoint waarde op van alle tags met ScadaId = 301 en kopieer deze waarde naar de actuele waarde van de betreffende tag.

```
2 | @SENDTOTAG (301)
```

4.4 @TAGTOSEND

functionaliteit

Functie om de actuele waarde van een tag te kopiëren naar de actuele setpoint waarde van een andere tag. Wordt gebruikt bij FEWS down te kopiëren naar stop E2E bewaking.

parameters

```
@TAGTOSEND( ScadaId, item.subitem, ScadaId2, item.subitem)
```

	Parameter	Toelichting
1	ScadaId	Scada Id van de te zoeken tag waarde.
2	item.subitem	Item.subitem van de te zoeken tag waarde.
3	ScadaId2	Scada Id van de tag waar het setpoint heen gaat.
4	item.subitem	Item.subitem van de tag waar het setpoint heen gaat.

voorbeeld

Kopieer de waarde van de tag met ScadaId = 305 en item.subitem = WB04.A_DOWN als setpoint naar de tag met ScadaId = 44 en item.subitem = E2E.S_STOP

```
1 | @TAGTOSEND (305, WB04.A_DOWN, 44, E2E.S_STOP)
```

4.5 @BLOCKWEB

functionaliteit

Functie om de blokkeer opdrachten via bv SOAP van een Waterschap naar een gemeente in te stellen.

Globale werking:

1. Waterschap geeft commando BLOKKEREN
2. Gemaal wordt geblokkeerd als GEKOPPELD aan staat.
3. Bij HOOGWATER wordt de koppeling verbroken en pas hersteld bij peil X cm onder hoog water
Behalve als NEGEER HW geselecteerd staat, dan blijft het station geblokkeerd.

Optioneel:

Het station pas te deblokken als de opdracht deblokken gegeven is en het niveau van een ander station onder een ingestelde waarde komt.

LET OP:

Als de blokkering niet GEKOPPELD staat zal de blokkering altijd direct worden opgeheven.

parameters

```
@BLOCKWEB (ScadaId, ScadaIdBlock, ActLevel, HighLevel, ScadaIdLev, RefLev, rLevParam)
```

	Parameter	Toelichting
1	Scadald	Scadald van Webservice locatie waar de gegevens binnenkomen
2	ScadaldBlock	Scadald van locatie die geblokkeerd/gedeblokkeerd moet worden
3	ActLevel	actueel niveau van ScadaldBlock
4	HighLevel	hoog niveau instelling van ScadaldBlock
5	ScadaldLev	Scadald van locatie met niveau onder instelling om te deblokken.
6	RefLev	actueel niveau van ScadaldLev
7	rLevParam	instelling waar actueel niveau van ScadaldLev onder moet zitten om te deblokken (na opdracht deblokkeer)
Parameter 5, 6 en 7 zijn optioneel		

voorbeelden

```
1 | @BLOCKWEB (999, 49, PICTURE1.P_26, PICTURE1.P_22, 057, PICTURE1.P_2, STATIC.LEV4)
```

Afhandeling blokkering voor 1 locatie. Deblokkering pas als niveau (PICTURE1.P2) van locatie met Scadald = 57 lager is dan instelling in STATIC.LEV4

999 = Scada Id Webservice locatie. Hier staan voor elke locatie de afgesproken koppel tags.
49 = Scada Id te blokkeren locatie.

PICTURE1.P26 = actueel niveau te blokkeren locatie

PICTURE1.P22 = hoog niveau instelling te blokkeren locatie

```
2 | @BLOCKWEB (999, 49, PICTURE1.P_26, PICTURE1.P_22, 057)
```

Afhandeling blokkering voor 1 locatie . Deblokkering direct na opdracht

4.6 @BESYSLIGHT

functionaliteit

Deze functie kan een alarm hoog zetten of een sturing doen. (bijvoorbeeld blokkeren of deblokken). De sturing mag ook een andere toepassing zijn.

De gebruiker kan de voorwaarden voor het blokkeren of deblokken binnen BeSys zelf samenstellen.

Tevens kan er voor worden gekozen om de voorwaarden door een SMS bericht aan te sturen.

Het bericht wordt gekoppeld aan *BLOKxxx (SET xxx:YYY) en *DEBLOKxxx (CLR xxx:YYY)

Het is dus ook mogelijk een combinatie te maken van voorwaarden.

Bijvoorbeeld: Blokkeren op een SMS en weer vrijgeven op een voorwaarde vanuit BeSys

LET OP:

De SMS files worden centraal ingelezen. Komt de code: SET of CLR erin voor dan wordt de betreffende BesysTempVar gezet.

parameters

```
@BESYSLIGHT (*INTxxx, *BLOCKxxx, *DEBLOCKxxx, para4,
para5, para6, para7, para8, para9, para10, para11,
para12, para13)
```

Parameter	Toelichting
1	*INTzzz Unieke variabele om voortgang van het proces te bewaken
2	*BLOCKxxx Voorwaarde om te blokkeren vanuit BeSys of SMS. Bijvoorbeeld: SET 23:A1 of SET 23:BLOK → *BLOCK23 = 1 en *DEBLOCK23 = 0
3	*DEBLOCKxxx Voorwaarde om te deblokkeren vanuit BeSys of SMS. Bijvoorbeeld: CLR 23:A1 of CLR 23:BLOK → *BLOCK23 = 0 en *DEBLOCK23 = 1
Blokkeeropdracht te sturen tag bij *BLOCKxxx = 1 Vul deze parameters niet in als er alleen een ALARM wordt bestuurd.	
4	Scadald Scadald locatie
5	Item.Subitem te besturen tag
6	Te sturen waarde, bijvoorbeeld '1'.
Deblokkeeropdracht te sturen tag bij *DEBLOCKxxx = 1 Vul deze parameters niet in als er alleen een ALARM wordt bestuurd.	
7	Scadald locatie
8	Item.Subitem te besturen tag
9	Te sturen waarde, bijvoorbeeld '0'.
Te besturen alarm Parameters hoeven niet te worden ingevuld als er geen alarm wordt bestuurd.	
10	Scada ID locatie te besturen alarm.
11	Item.Subitem tag van te besturen alarm.
SMS (retour) bericht na uitgevoerde SMS opdracht Parameters hoeven alleen te worden ingevuld als een retourbericht gewenst is.	
12	SMS retour bericht na SET commando
13	SMS retour bericht na RESET commando
Interne parameter (deze mag niet worden gebruikt in de Besys Code)	
14	*TELxxx

**WAARSCHUWING:**

Als er gebruik gemaakt wordt van SMS dan moeten parameter 2 en 3 met respectievelijk *BLOCK en *DEBLOCK beginnen. De SMS wordt aan deze parameters gekoppeld.

**WAARSCHUWING:**

*DEBLOCKxxx gaat voor *BLOCKxxx. Met andere woorden RESET wint van SET.

voorbeelden

Voorbeeld 1

Niveau locatie met Scada ID 36 > 3.5 meter → blokkeer locaties met Scada ID 85 en 300

Niveau locatie met Scada ID 36 < 1.5 meter → deblokkeer locaties met Scada ID 85 en 300

```
IF 036.PICTURE1.P_9 > #3.50
  *BLOCK1 := #1
  *DEBLOCK1 := #0
ENDIF
IF 036.PICTURE1.P_9 < #1.50
  *BLOCK1 := #0
  *DEBLOCK1 := #1
ENDIF

@BESYSLIGHT (*INT1, *BLOCK1, *DEBLOCK1, 85, GEMAAL.BLOKKEREN, 1,
             85, GEMAAL.BLOKKEREN, 0);
@BESYSLIGHT (*INT2, *BLOCK1, *DEBLOCK1, 300, GEMAAL.BLOKKEREN, 1,
             300, GEMAAL.BLOKKEREN, 0);
```

Voorbeeld 2

Alarm-tag A_84 van locatie met ScadaID 345 zetten en gemaal met Scada ID 300 blokkeren bij:

- SMS in SET 1:BLOK of SET 1:A_xx
- SMS versturen [SET GELUKT]

Alarm tag A_84 van locatie met Scada ID 345 resetten en gemaal met Scada ID 300 deblokkeren bij:

- SMS in CLR 1:BLOK of CLR 1:A_xx
- SMS versturen [CLR GELUKT]

```
@BESYSLIGHT (*INT1, *BLOCK1, *DEBLOCK1, , , ,
             345, ALARM.A_84, SET GELUKT, CLR GELUKT);

@BESYSLIGHT (*INT2, *BLOCK1, *DEBLOCK1, 300, GEMAAL.BLOKKEREN, 1,
             300, GEMAAL.BLOKKEREN, 0);
```

Voorbeeld 3

Alarm tag A_84 van locatie met ScadaID 345 zetten en gemaal met ScadaID 300 blokkeren bij:

- SMS in SET 1:BLOK of SET 1:A_xx

Alarm tag A_84 van locatie met ScadaID 345 resetten en gemaal met ScadaID 300 deblokkeren bij:

- Niveau locatie met Scada ID 36 < 1.5 meter

```
IF 036.PICTURE1.P_9 < #1.50
  *BLOCK1 := #0
  *DEBLOCK1 := #1
ENDIF

@BESYSLIGHT (*INT1, *BLOCK1, *DEBLOCK1,
             ,
             ,
             ,
             345, ALARM.A_84);

@BESYSLIGHT (*INT2, *BLOCK1, *DEBLOCK1, 300, GEMAAL.BLOKKEREN, 1,
             300, GEMAAL.BLOKKEREN, 0);
```

4.7 @LOG

functionaliteit

Functie om rechtstreeks vanuit de BeSys-regels een gebeurtenis (event) te loggen in de gebeurtenissenlijst.

parameters

De functie heeft 2 aanroepmogelijkheden:

```
@LOG (station, item, subitem, type, waarde, tekst)
```

	Parameter	Toelichting
1	station	station naam of XXX.COM.NAAM waarbij XXX het Scadald is
2	item	tag item
3	subitem	tag subitem
4	type	gebeurtenistype (bijvoorbeeld 'Alarm')
5	waarde	actuele waarde van de gebeurtenis
6	tekst	Gebeurtenistekst

Parameter 2 t/m 4 mogen ook LEEG zijn ("").

```
@LOG (tekst)
```

	Parameter	Toelichting
1	tekst	Gebeurtenistekst

voorbeelden

Dit resulteert in 2 regels in de gebeurtenissenlijst. Beide met de locatiennaam van Scadald 123.

```
1 | @LOG (123.COM.NAAM, "Laatste", "Verbindingstijd", "Alarm" , "1",
   | "Tekst")
2 | @LOG (123.COM.NAAM, "Laatste", "Verbindingstijd", "Herstel", "0",
   | "Tekst")
```

Dit resulteert in 1 een regel in de gebeurtenissenlijst. Locatie naam = BeSys


```
3 | @LOG ("Tekst")
```

4.8 @FROMHIS functionaliteit

Functie om vanuit een (voorspellende) trend reeks de actuele waarde of setpoint te bepalen.
Voorbeeld: Wordt gebruikt om Voorspelde waterstanden, actuele windkracht, SOAP FEWS koppeling of eerstvolgende laag / hoog water uit een trendreeks te bepalen.

parameters

De functie heeft 3 aanroepmogelijkheden:

- Trendwaarde naar tag-waarde voor 1 tag.
- Trendwaarde naar tag-waarde voor alle tags.
- Trendwaarde naar setpoint-waarde andere (FEWS) tag.

	Parameter	Toelichting
1	Scadald	Scada Id van locatie
2	Tijd terug	Tijd terug vanaf huidig tijdstip waar vanaf gezocht moet worden
3	Tijd vooruit	Tijd vooruit vanaf huidig tijdstip tot waar gezocht moet worden
4	item.subitem	Item en subitem van de tag
5	FEWS Scada ID	Scada Id van de fews locatie die bestuurd wordt
6	FEWS item.subitem	item.subitem van de fews tag die bestuurd wordt

Para 2 en Para 3 hebben de volgende definitie

- Leeg is 1 dag
- < 600 dan hebben we het over dagen
- > 600 dan hebben we het over secondes

Para 4 leeg ("") dan alle tags van de locatie verwerken.

Para 5 en Para 6 zijn optioneel.

voorbeelden

Voorbeeld 1

Zoek in de trend reeks van de tag met scada id = 700 en tag item.subitem = DAMSTERDIEP.Q_RTC de laatste waarde voor de huidige tijd. De trendreeks omvat alle trendwaarden van 1 dag voor en 1 dag na nu. De gevonden waarde wordt als actuele waarde van de betreffende tag weggeschreven.

```
1 | @FROMHIS (700, 1, 1, DAMSTERDIEP.Q_RTC)
```

Voorbeeld 2

Zoek in de trend reeks van alle tags met scada id = 700 laatste waarde voor de huidige tijd. De

trendreeks omvat alle trendwaarden van 1 dag voor en 1 dag na nu. De gevonden waarde wordt als actuele waarde van de betreffende tag weggeschreven.

```
2 | @FROMHIS (700, 1, 1)
```

Voorbeeld 3

Zoek in de trend reeks van de tag met scada id = 700 en tag item.subitem = DAMSTERDIEP.Q_RTC de laatste waarde voor de huidige tijd. De trendreeks omvat alle trendwaarden van 1 dag voor en 1 dag na nu. De gevonden waarde wordt als setpoint weggeschreven naar een tag met Scada Id = 700 en item.subitem = CENTR.P_DEB

```
3 | @FROMHIS (700, 1, 1, DAMSTERDIEP.Q_RTC, 702, CENTR.P_DEB)
```

4.9 @TOHIS

functionaliteit

Functie om de actuele tag waarde op te slaan als een trend waarde. De actuele timestamp en de actuele tag waarde worden gebruikt.

Voorbeeld: Kan gebruikt worden om een extern signaal als trend op te slaan. Denk aan blokkering WS -> gemeente

parameters

```
@TOHIS (ScadaId [, item.subitem])
```

Parameter	Toelichting
1	ScadaId Scada ID van locatie
2	item.subitem item en subitem van de tag Deze parameter is optioneel: <ul style="list-style-type: none"> Opgegeven alleen specifieke tag Niet opgegeven alle tags

voorbeelden

Van locatie met Scada ID = 301 wordt de waarde van **de tag met item = LEVEL en subitem = PV** samen met de huidige datum/tijd weggeschreven naar de trend database.

Van locatie met Scada ID= 302 worden **alle tags** samen met de huidige datum/tijd weggeschreven naar de trend database.

```
1 | @TOHIS (301, LEVEL.PV)
2 | @TOHIS (302)
```

LET OP:

Bovenstaande geldt voor tags met een gedefinieerd trend adres.

4.10 @PROC / @ENDPROC

functionaliteit

In de BeSys regels kan je zelf procedures maken en aanroepen voor handelingen die vaak gedaan moeten worden met verschillende parameter. Als voorbeeld is een procedure gemaakt die voor de belangrijkste kunstwerken de laatste verbindingstijd controleert. Is de laatste verbindingstijd te oud dan wordt er een EVENT regel in de gebeurtenissenlijst geplaatst. Komt de verbinding terug, dan wordt er ook een EVENT regel geschreven in de gebeurtenissenlijst.

werking

Een procedureblok moet altijd worden afgesloten en ziet er globaal als volgt uit:

```
@PROC procedurenaam (para1, para2, etc)
    ....
@ENDPROC
```

Zowel de naam van de procedure als van de eventuele parameters zijn vrij te kiezen. In het procedureblok kan ook gebruik gemaakt worden van de meegegeven parameters.

De procedure aanroepen gebeurt door:

```
@procedurenaam (aap, noot, etc.)
```

Het aantal parameter is niet direct begrenst, maar met teveel parameters wordt het er niet makkelijk van.

LET OP:

Er moet een spatie staan tussen de procedurenaam en de parameterdefinitie.

voorbeeld

```
132 // Call procedure CheckLVB (locations with Scada ID: 138, 139,
133 // 155, 158, 160)
133 @CheckLVB(138)
134 @CheckLVB(139)
135 @CheckLVB(155)
136 @CheckLVB(158)
137 @CheckLVB(160)
138
139 // Procedure Check laatste verbindingstijd
140 // Langer dan 15 minuten geen verbinding -> Melding in
141 // gebeurtenissenlijst.
141 // -> Tag [COMFOUT] bijwerken
142 @PROC CheckLVB (XXX)
143 IF XXX.COM.LVT > #15
144 ANDIF *XXXStatusLVT == #0
145     +XXXStatusLVT := #1
```

```
146     $XXX.WABIS.A_COMFOUT := #1
147     @LOG (XXX.COM.NAAM, "Laatste", "Verbindingstijd", "Alarm",
148     "1",
149     "Meetgegevens ouder dan 15 minuten")
149 ENDIF
150 IF XXX.COM.LVT < #16
151 ANDIF *XXXStatusLVT == #1
152     +XXXStatusLVT := #0
153     $XXX.WABIS.A_COMFOUT := #0
154     @LOG (XXX.COM.NAAM, "Laatste", "Verbindingstijd", "Herstel",
155     "0",
156     "Meetgegevens ouder dan 15 minuten")
156 ENDIF
157 @ENDPROC
```

LET OP:

In plaats van het Scada ID mag ook de locatienaam worden gebruikt.

5 Samengestelde alarmen

Het is mogelijk om in een BOS-besturing alarmen samen te stellen uit verschillende tags van (verschillende) stations. De volgende functies zijn hiervoor beschikbaar:

Funcie	Toelichting
@LOAD	Start met samengesteld alarm
@GET	
@OR	'or' functie
@AND	'and' functie
@ANDGRT	'andgrt' functie: en tagwaarde groter dan andere tagwaarde/vaste waarde
@ANDEQU	'andequ' functie: en tagwaarde gelijk aan andere tagwaarde/vaste waarde
@ANDLES	'andles' functie: en tagwaarde kleiner dan andere tagwaarde/vaste waarde
@ORGRT	'orgrt' functie: of tagwaarde groter dan andere tagwaarde/vaste waarde
@OREQU	'orequ' functie: of tagwaarde gelijk aan andere tagwaarde/vaste waarde
@ORLES	'orles' functie: of tagwaarde kleiner dan andere tagwaarde/vaste waarde
@STORE	Slaat het samengestelde alarm op.
@SET	

Omdat alles binnen Aquaview tag-gerelateerd is, hebben we ook voor de samengestelde alarmen tags nodig. Deze tags kunnen in een locatie geïmporteerd worden. Dit mag een unieke locatie zijn, maar ook een bestaande locatie voldoet.

In onderstaand voorbeeld zijn de volgende tags geïmporteerd in een nieuwe locatie.

RECORD#	RTUNAME	ITEM	SUBITEM	ALARM	PRIOR	1	ALARM	P	LOCAL	AL	CENTRAL	ALARM	PRINT	RTI	ALARM	P	ALARM	P	RTU	LABEL	WRITE	DESCRIPTION	RTU	ADD#
1	ALARM	SA1		0	0	0	0	1				1	0	0								0 By Besys filled in		
2	ALARM	SA2		0	0	0	0	1				1	0	0								0 By Besys filled in		
3	ALARM	SA3		0	0	0	0	1				1	0	0								0 By Besys filled in		
4	ALARM	SA4		0	0	0	0	1				1	0	0								0 By Besys filled in		
5	ALARM	SA5		0	0	0	0	1				1	0	0								0 By Besys filled in		
6	ALARM	SA6		0	0	0	0	1				1	0	0								0 By Besys filled in		
7	ALARM	SA7		0	0	0	0	1				1	0	0								0 By Besys filled in		
8	ALARM	SA8		0	0	0	0	1				1	0	0								0 By Besys filled in		
9	ALARM	SA9		0	0	0	0	1				1	0	0								0 By Besys filled in		
10	ALARM	SA10		0	0	0	0	1				1	0	0								0 By Besys filled in		

Deze nieuwe locatie bestaat niet fysiek en heeft dan ook geen communicatieprotocol of andere hardware gerelateerde zaken nodig.

LET OP:

Als Scada ID mag elk uniek ID gebruikt worden. Vanwege de herkenbaarheid wordt geadviseerd om een hoog nummer (5000 range) te gebruiken.

5.1 @GET / @LOAD

```
1 | @GET (555.ALARM.A_81)
```

De GET functie heeft 1 parameter. Met de GET functie starten we met een nieuw samengesteld alarm. De waarde van de parameter wordt als start waarde meegegeven. Parameter 1 mag een tag, metatag of waarde zijn.

Tag: scadald.item.subitem (555.ALARM.A_81)

MetagTag: scadald.Metatag (555.ALARMXXX)

Waarde:(1)

De LOAD functie is gelijk aan de GET functie. De gebruiker gebruikt de benaming die hem/haar het meest aanspreekt.

5.2 @AND / @OR

De AND en OR functies hebben dezelfde parameter opbouw als de GET functie.

```
1 | @AND (555.ALARM.A_83)
2 | @OR (555.ALARM.A_82)
```

De AND functie doet een logische AND met de waarde van de parameter op het resultaat tot nu toe.

De OR functie doet een logische OR met de waarde van de parameter op het resultaat tot nu toe.

5.3 @ANDGRT

De ANDGRT functie heeft 3 parameters.

```
1 | @ANDGRT (120, 555.PICTURE.P_9, 2.55)
```

Het resultaat doet een logische AND op het resultaat tot nu toe.

Parameter	Toelichting	Voorbeeld
Waarde	waarde in minuten hoe oud de waarde mag zijn. Is de waarde ouder dan wordt het alarm gereset.	120
Tag / waarde	waarden waarover de GRT functie wordt uitgevoerd.	555.PICTURE1.P_9
MetagTag / waarde		2.55

5.4 ANDEQU, ANDLES, ORGRT, OREQU en ORLES

De bewerkingen zijn anders maar verder hebben de functies ANDEQU, ANDLES, ORGRT, OREQU en ORLES dezelfde parameters.

```
1 | @SET (5000.ALARM.SA1, DIT IS VIA BESYS INGEVULD)
```

5.5 SET

De SET functie heeft 2 parameters. Met de SET functie wordt het resultaat van alle

bewerkingen opgeslagen in de Tag of Metatag die in parameter 1 wordt omschreven.

Parameter	Toelichting	Voorbeeld
Tag	scadald.item.subitem	5000.ALARM.SA1
MetagTag	scadald.Metatag	5000.ALARMSA1

In de 2^e parameter wordt de tekst van het nieuwe samengestelde alarm opgegeven.

5.6 STORE

De STORE functie is gelijk aan de SET functie. De gebruiker gebruikt de benaming die hem/haar het meest aanspreekt.

```
1 | @STORE (5000.ALARM.SA2, Hoogwater locatie 555 - niveau > 2.55 meter)
```

5.7 Voorbeelden

Hier onder enkele mogelijk toepassingen van de omschreven samengestelde alarmen.

```
1 | @GET (555.ALARM.A_81)
2 | @OR (555.ALARM.A_82)
3 | @OR (555.ALARM.A_83)
4 | @SET (5000.ALARM.SA1, DIT IS VIA BOS INGEVULD)
5 |
6 | @LOAD (555.ALARM.A_81)
7 | @ORLES (120, 555.PICTURE1.P_9, 2.55)
8 | @STORE (5000.ALARM.SA2, Hoogwater locatie 555 - niveau > 2.55 meter)
```

Index

A

Analysemodule

BOS 4

B

BeSys

Functies 15

@BESYSLIGHT 19

@BLOCKWEB 18

@END_SCRIPTFORREGION 15

@ENDEVERY 16

@ENDPROC 25

@EVERY 16

@FROMHIS 23

@LOG 22

@PROC 25

@SENDTOTAG 17

@START_SCRIPTFORREGION 15

@TAGTOSEND 18

@TOHIS 24

Getallen 13

Hulpvariabelen 13

Samengestelde alarmen 27

@AND 27

@ANDEQU 27

@ANDGRT 27

@ANDLES 27

@GET 27

@LOAD 27

@OR 27

@OREQU 27

@ORGRT 27

@ORLES 27

@SET 27

@STORE 27

R

Regelingen

Logische bewerkingen 13

Xylem |'zīləm|

- 1) het weefsel in planten dat water van de wortels naar boven brengt;
- 2) een toonaangevend watertechnologiebedrijf.

Wij zijn een wereldwijd team dat verenigd is in een gemeenschappelijk doel: geavanceerde technologische oplossingen creëren voor de wateruitdagingen van de wereld. De ontwikkeling van nieuwe technologieën die het gebruik, de conservering en het hergebruik van water in de toekomst zullen verbeteren, staat centraal in ons werk. Onze producten en diensten verplaatsen, behandelen, analyseren, bewaken en voeren water terug naar de omgeving in toepassingen van openbare nutsvoorzieningen, industriële, residentiële en commerciële gebouwdiensten. Xylem biedt ook een toonaangevend portfolio van slimme meters, netwerktechnologieën en geavanceerde analyseoplossingen voor water-, elektriciteits- en gasvoorzieningen. In meer dan 150 landen onderhouden we sterke, langdurige relaties met klanten, die vertrouwen op onze krachtige combinatie van toonaangevende productmerken en toepassingsexpertise met een sterke focus op het ontwikkelen van omvattende, duurzame oplossingen.

Ga voor meer informatie over hoe Xylem u kan helpen naar www.xylem.com/nl.



Digital transformation start met Xylem Vue

Digitale implementatie is niet langer een optie, het is noodzaak om de toenemende druk waarmee u te maken heeft het hoofd te bieden. Xylem Vue brengt ons volledige portfolio van digitale oplossingen samen om uw meest urgente wateruitdagingen op te lossen tijdens uw digitale reis. Laten we kijken wat voor u mogelijk is.

Voor meer informatie over Xylem Vue, ga naar xylem.com/XylemVue



Xylem Water Solutions Nederland B.V.
Pieter Zeemanweg 240
3316 GZ Dordrecht
Tel +31 78 654 84 00
www.xylem.com/nl

Visit our Web site for the latest version of this document

The original instruction is in Dutch. All non-Dutch instructions are translations of the original instruction

© 2024 Xylem Inc.