

# SIVAK

## SYSTEEMDOCUMENTATIE

**Simulatiepakket voor de VerkeersAfwikkeling bij Kunstwerken**

van

Rijkswaterstaat

SIVAK	versie v3.00
Documentatie	versie v1.00
December 2000	

BIBLIOTHEEK  
Bouwdienst Rijkswaterstaat  
Postbus 20.000  
3502 LA Utrecht



## 1. Inleiding

### *doelstelling*

Dit document bevat de systeemdokumentatie voor het model SUBMOD2 dat onderdeel is van de SIVAK programmatuur.

### *globale opbouw van de documentatie*

*Hoofdstuk 2* van de documentatie geeft een overzicht van de structuur van het SIVAK systeem.

*Hoofdstuk 3* beschrijft de globale aspecten van SUBMOD2.

*Hoofdstuk 4* beschrijft de layout van de data bestanden.

*Hoofdstuk 5* bevat de procesbeschrijvingen van de componenten en klassen.

*Hoofdstuk 6* bevat de geannoteerde bronteksten van de onderdelen van SUBMOD2.

*Hoofdstuk 7* bevat een korte handleiding voor het maken en installeren van duplicates.

### *typografie*

In deze handleiding staan de typische PROSIM woorden en namen gedrukt in KLEINE HOOFDLETTERS. Namen van bestanden (files) staan in HOOFDLETTERS. De bronteksten zijn in een niet-proportioneel lettertype uitgevoerd. Benadrukte woorden en passages zijn *cursief* gedrukt.

### *verantwoording*

Het SIVAK systeem is in opdracht van- en in nauwe samenwerking met de toenmalige Hoofdafdeling Scheepvaart i.c. P.H. Hiddinga van de Dienst Verkeerskunde ontwikkeld door bureau Sierenberg en de Gans te Waddinxveen.

Deze documentatie is vervaardigd door de rechtsopvolger van dat bureau, Prosim bv te Zoetermeer.

## 2. Overzicht van het SIVAK systeem

Het hele systeem bestaat uit:

- zes modellen,
- een onbepaald aantal data files.

De zes modellen zijn:

- model SUBMOD2:  
het eigenlijke simulatie model van een netwerk van vaarwegen waarin tenminste één sluis , één brug of één vernauwing voorkomt,
- model SUBMOD1:  
dit model zorgt voor het genereren van schepen per traject van het te simuleren netwerk. Een *traject* is een opeenvolging van waterwegen tussen twee uiteinden van het netwerk,
- model QUERY:  
met dit model is de gebruiker in staat de uitvoer van SUBMOD2 op een flexibele manier te bekijken,
- model SLUISANI:  
dit model maakt een FIGURE FILE waarmee een animatie kan worden aangestuurd van een sluis in het met SUBMOD2 gesimuleerde netwerk,
- model BRUGANI:  
dit model maakt een FIGURE FILE waarmee een animatie kan worden aangestuurd van een brug in het met SUBMOD2 gesimuleerde netwerk,
- model SHP2FILE:  
dit model geeft een overzicht van de inhoud van met model SUBMOD1 verkregen files van te simuleren schepen.

In technische zin kent PROSIM vier soorten data files:

- FIGURE FILES:  
zo'n file wordt gemaakt door sluisani of brugani.  
Aan het einde van een run met één van deze modellen, krijgt de gebruiker de gelegenheid de FIGURE FILE van een naam te voorzien waaronder dan die file in een bibliotheek wordt opgenomen. Iedere file in deze bibliotheek kan in de ANIMATION SUPPLEMENT gebruikt worden om een animatie aan te sturen.
- STORE FILES:  
tijdens de run van ieder van de modellen wordt een STORE FILE gemaakt, waarin een aantal tijdreeksen met betrekking tot de functie van dat model is opgeslagen. Deze tijdreeksen kunnen tijdens het runnen van het model (met behulp van DATA HANDLING) worden geïnspecteerd ter beoordeling van het functioneren van het model. Evenals FIGURE FILES kunnen ook STORE FILES aan het einde van een run worden bewaard voor later gebruik; bijvoorbeeld in de GRAPHICS en STATISTICS SUPPLEMENTS.
- MATRIX FILES:  
dit zijn direct access files, die door het SIVAK-systeem automatisch worden gemaakt en aangesloten. Deze files dienen voor een zeer efficiënte data communicatie tussen de verschillende modellen. De gebruiker wordt niet geacht iets aan deze bestanden te wijzigen.
- USER DATA FILES:  
dit zijn alle overige data files, die de voornaamste input en output van de modellen uitmaken.

De USER DATA FILES zullen we functioneel in drie groepen verdelen:

- **BIBFILES:**  
een BIBFILE (bibliotheek file) bevat algemene gegevens betreffende wegverkeer en scheepvaart, die door de AVV in stand worden gehouden. De diverse modellen zullen, waar nodig, gegevens putten uit deze bestanden. Deze files hebben een voorgeschreven structuur en de "onderdelen" daarin zijn doorgaans op een éénduidige manier geïdentificeerd. De gebruiker zal dus niet zomaar zo'n bestand wijzigen. Als dus de gebruiker bijvoorbeeld wil weten wat er gebeurt als de kosten van een scheepvaartklasse verdubbelen, dan zal hij eerst een kopie maken van de BIBFILE BIBKOSTEN om daarin te wijzigen en vervolgens die kopie aan te sluiten aan het betreffende model in plaats van de oorspronkelijke BIBFILE.
- **RAPPORTEN:**  
een rapport is een uitvoer file van een model, die niet functioneert als invoer voor een ander model. Een rapport is dus een eindproduct van het systeem. De gebruiker kan dus zonder bezwaar (althans in technische zin) een rapport wijzigen, door bijvoorbeeld commentaar toe te voegen of iets te verwijderen. In de meeste gevallen echter wordt een rapport rechtstreeks naar de printer gestuurd.
- **MODELFILES:**  
alle overige files zijn MODELFILES; zo'n file is dus een invoer file voor tenminste één model. Dat betekent dat de structuur vast ligt en dus door de gebruiker moet worden aangehouden. In het algemeen wordt zo'n file of gemaakt door een model (als uitvoer file dus) of gemaakt door de gebruiker (met behulp van de editor). In de praktijk zal de gebruiker nimmer zo'n file vanaf "blanco" aanmaken, omdat zoiets vrijwel onmogelijk is zonder te zondigen tegen de vastgestelde structuur. In voorkomende gevallen zal de gebruiker steeds een bestaande file kopiëren om die kopie vervolgens te veranderen onder handhaving van de structuur.

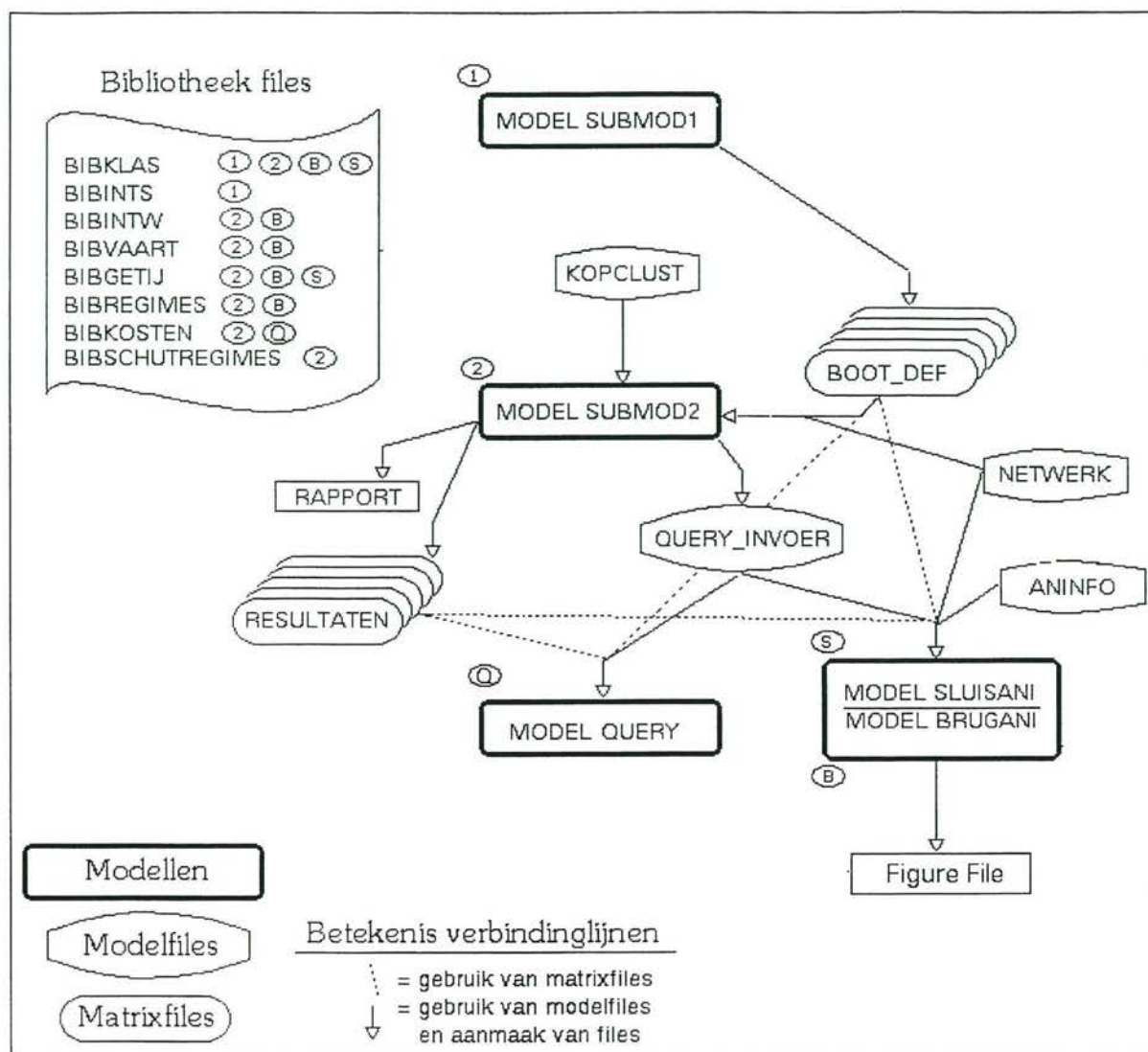
### ***environments***

Om een model te kunnen runnen zullen er BIBFILES en MODELFILES aangesloten moeten worden.

Iedere complete verzameling files nodig om een model te kunnen runnen heet in PROSIM een ENVIRONMENT. Alle in gebruik zijnde ENVIRONMENTS worden door de MODEL MAINTENANCE bewaard. Om meer ENVIRONMENTS per model mogelijk te doen zijn, wordt intern in een model gebruik gemaakt van formele namen van data files (INPUTSTREAMS en OUTPUTSTREAMS). In een ENVIRONMENT wordt dan vastgelegd welke actuele data file verbonden zal worden met welke formele afspiegeling hiervan in het model.

In het volgende schema wordt aangegeven op welke manier de modellen en de data files aan elkaar gekoppeld zijn. Er is gebruik gemaakt van de formele namen die binnen de modellen deze files representeren.





### legende:

#### BOOT\_DEF

een MATRIX FILE die het resultaat is van het gebruik van model SUBMOD1. Dergelijke files bevatten gegevens op grond waarvan schepen worden gegenereerd en de gegevens van de zodanig gegenereerde schepen. Met SUBMOD1 kunnen dergelijke files niet alleen worden aangemaakt maar ook worden onderhouden.

Elke file van dit type heeft een, door de gebruiker opgegeven, naam. Deze namen worden in de onder beschreven file NETWERK gebruikt om schepen te koppelen aan de daarin te specificeren trajecten.

#### NETWERK:

een MODELFILE waarin de gebruiker de configuratie van het te simuleren netwerk aangeeft.

#### KOPCLUST:

een MODELFILE met informatie voor model SUBMOD2 met betrekking tot de lay-out van een rapport (aanhef en clustervorming van scheepsklassen).

#### ANINFO:

een MODELFILE die stuurinformatie bevat voor model SLUISANI of BRUGANI op grond waarvan het te animeren object uit het netwerk wordt geselecteerd en een aantal opties worden ingesteld.

**BIBKLAS:**

een BIBFILE met statistische informatie over de karakteristieken van de verschillende scheepsklassen en de indeling van deze klassen naar categorieën.

**BIBINTS:**

een BIBFILE met aanbodspatronen voor het scheepvaartverkeer.

**BIBINTW:**

een BIBFILE met aanbodspatronen voor het wegverkeer.

**BIBVAART:**

een BIBFILE met informatie op grond waarvan de in- en uitvaartijden bij sluizen en de (onder)doorvaartijden bij bruggen bepaald kunnen worden.

**BIBGETIJ:**

een BIBFILE met doodtij- en springtijtabellen.

**BIBREGIMES:**

een BIBFILE met tabellen met bloktijden en openingsduurbeperingen zoals deze voor kunnen komen bij bruggen.

**BIBSCHUTREGIMES:**

een BIBFILE met schutregimes zoals deze voor kunnen komen als schutwaterbesparende maatregelen bij sluizen.

**BIBKOSTEN:**

een BIBFILE met de verschillende soorten kosten voor het scheepvaartverkeer.

### 3. Proces beschrijvingen

#### 3.1 Overzicht

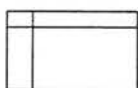
Het te simuleren netwerk bestaat uit vaarwegen, bruggen, sluisen en vernauwingen. Deze onderdelen, ook wel *waterwegen* genoemd, zijn met elkaar verbonden door NODES (knooppunten, knopen). Ook aan ieder uiteinde van het netwerk bevindt zich een knoop. Een brug, sluis of vernauwing is in dit verband een waterweg waarin zich een kunstwerk bevindt. In deze zin is dus het hele netwerk bevaarbaar.

Een schip beweegt zich door het netwerk vanaf zijn beginknoop over een daarmee verbonden waterweg naar de knoop aan de andere kant van die waterweg, dan over een met die knoop verbonden waterweg naar de knoop aan de andere kant van die waterweg, enzovoort, totdat zijn eindknoop is bereikt. De zo bevaren verzameling waterwegen heet zijn *traject*, de verzameling daarbij gepasserde knopen zijn *route*.

Een schip beweegt zich in principe zelfstandig door het netwerk. Zijn gedrag is daarbij afhankelijk van het type waterweg dat bevaren gaat worden:

- *Vaarweg*  
Het schip bevaart de vaarweg zelfstandig. Maar is de volgende waterweg van het schip een brug, sluis of vernauwing dan zal het zich aanmelden (met als uitzondering een garantieschip dat een sluis nadert: dat meldt zich een waterweg eerder aan).
- *Brug*  
Een schip dat onder de brug door kan varen en de brug gesloten aantreft zal gewoon doorvaren als ware de waterweg een vaarweg. Anders komt het in ieder geval in de wachtrij, maar bovendien in het brugplan als de brug voor passage geopend moet zijn. De feitelijke passage geschiedt onder regie van het *brugcomplex*, zijnde een samenstel van een BRUG, een BRUGWACHTER, twee REGELAARS, een KLOK en een STROOM.
- *Sluis*  
Komt het schip aan bij een sluis dan vindt interactie plaats met het sluiscomplex om vast te stellen of het schip eerst naar de wachtruimte moet of direct naar de opstelruimte kan. Daarna vindt de feitelijke passage plaats onder regie van het *sluiscomplex*, zijnde een samenstel van een SLUIS, een aantal KOLKEN, een WEKKER en de MACRO SCHUTPLANNING als representatie van de 'virtuele' component *sluismeester*.
- *Vernauwing*  
Komt het schip aan bij een vernauwing dan vindt interactie plaats om vast te stellen of het schip direct kan passeren. Zo niet dan gaat het naar de wachtruimte totdat passage wel mogelijk is.

Met de volgende toelichting



betekent 'interactie met'



betekent 'de regie overdragen'



betekent autonome beïnvloeding



betekent trigger

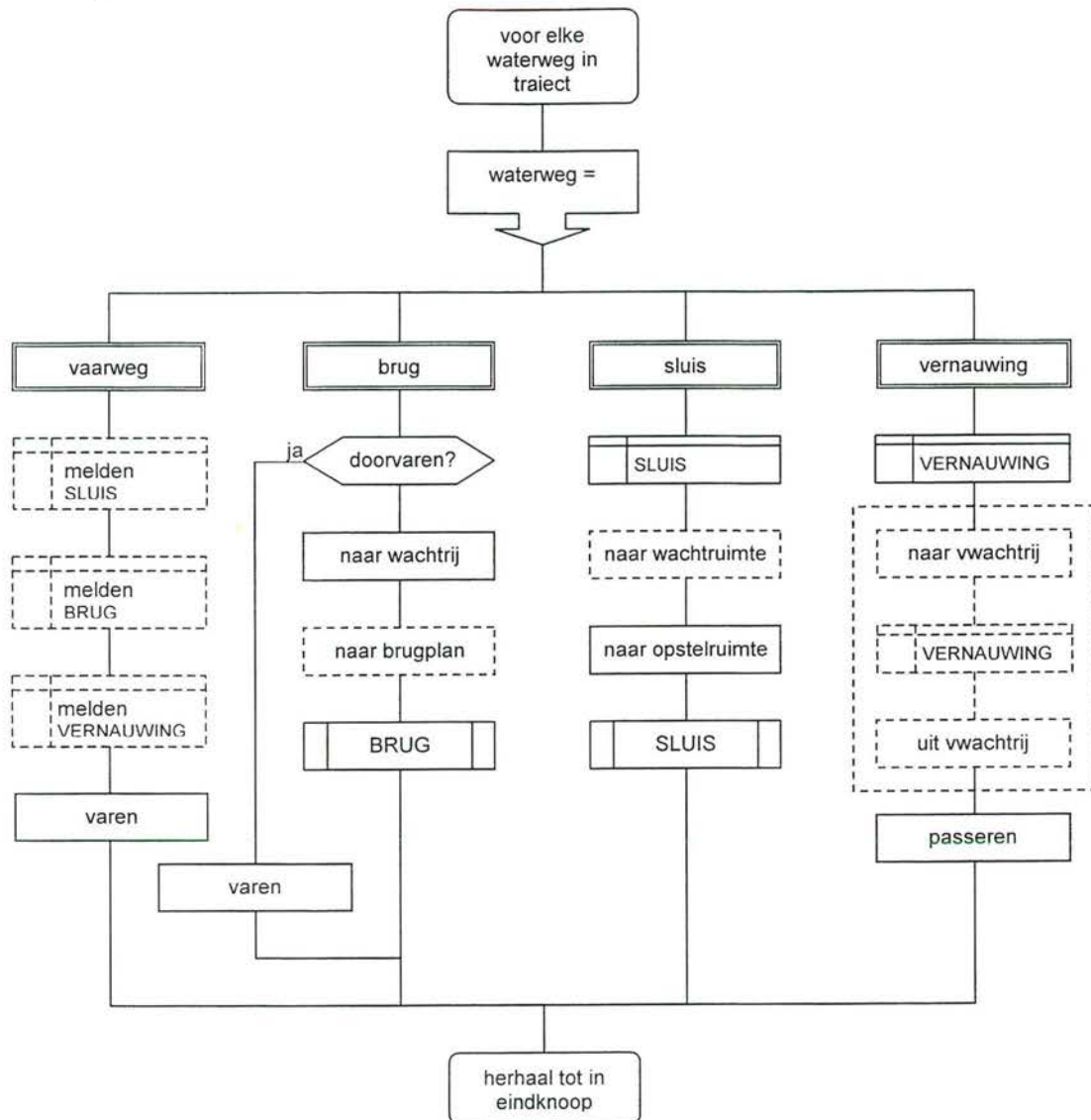


betekent consultatie

woorden in kapitaal verwijzen naar VERNAUWING dan wel brug- of sluiscomplex, symbolen in stippellijnen betekenen 'eventueel' cq 'indien van toepassing',

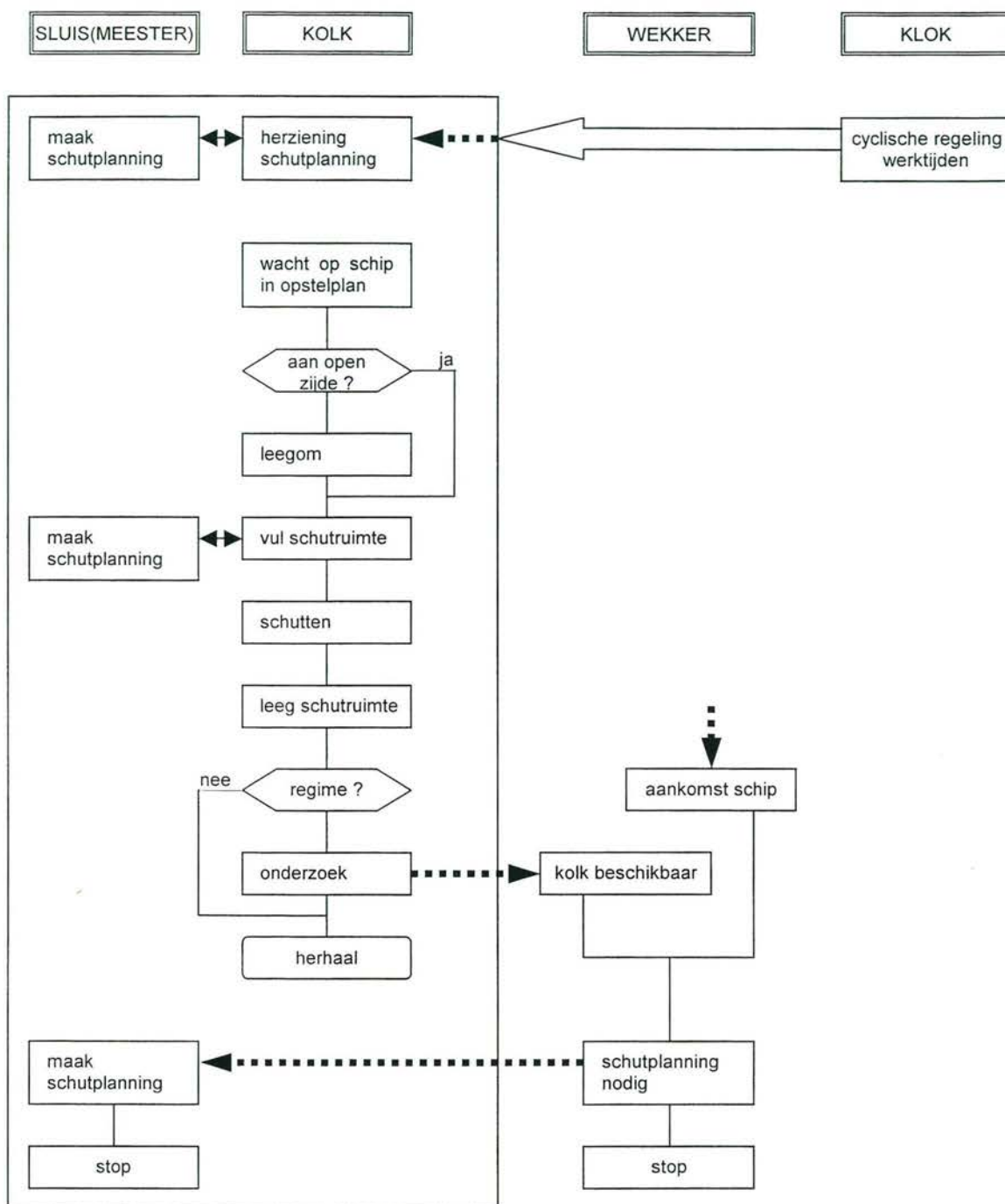
kan één en ander worden samengevat in het volgende schema's:

Voor een schip:

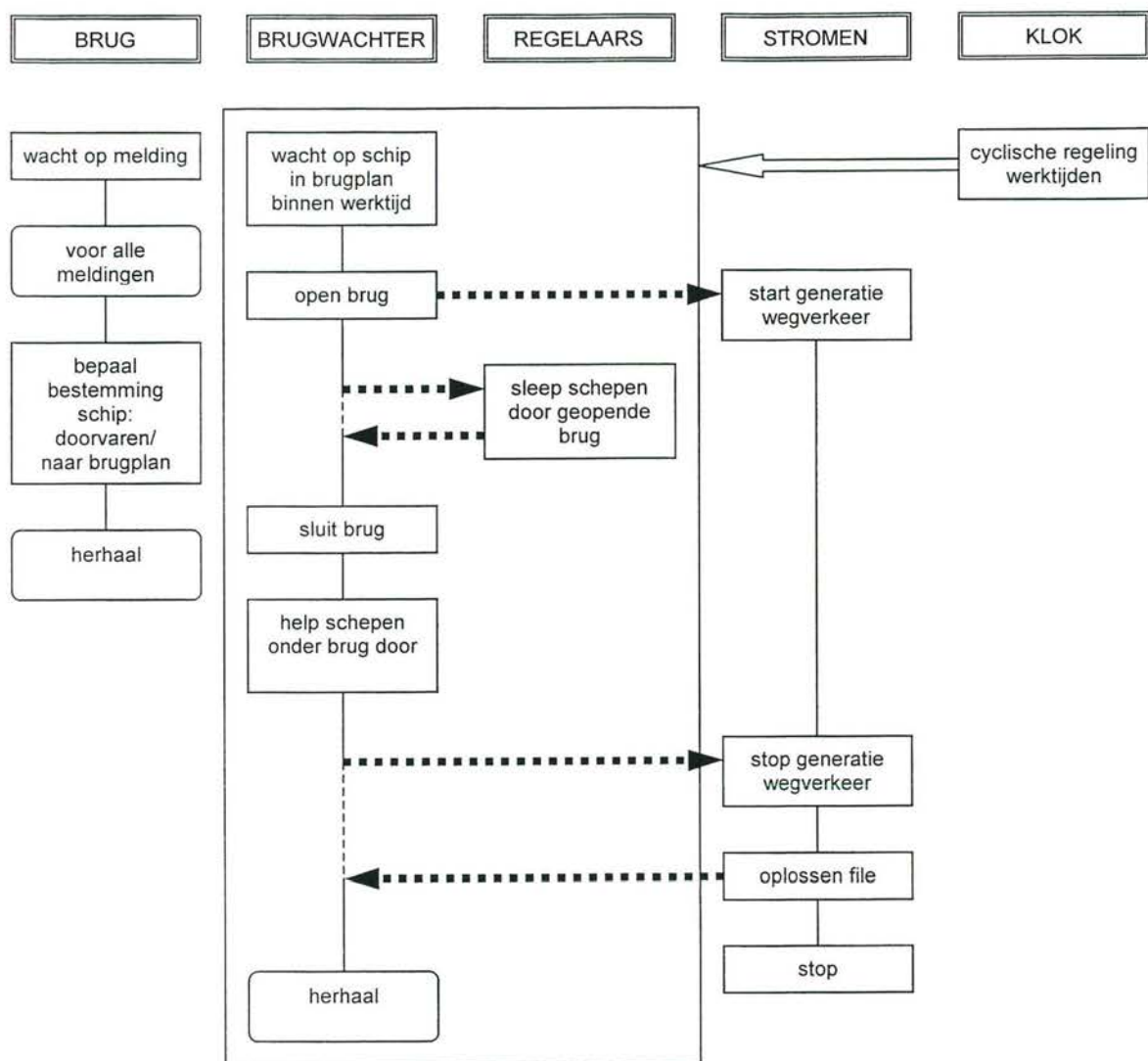




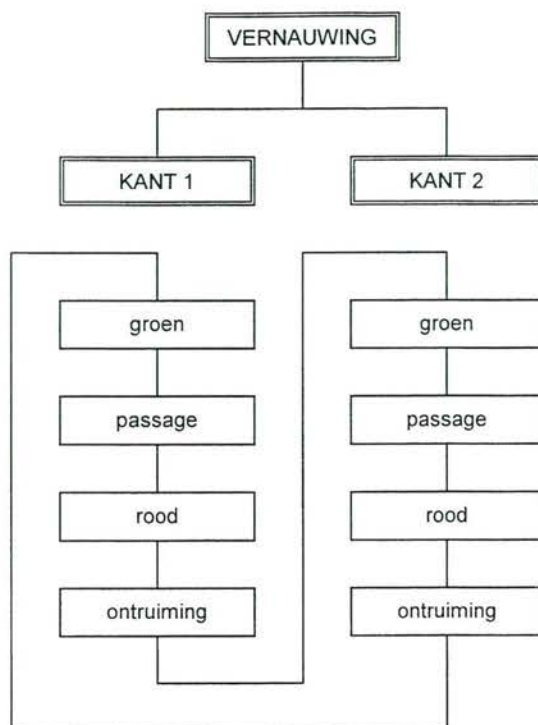
Voor het sluiscomplex:



Voor het brugcomplex:



Voor een vernauwing:



### 3.2 Brug

Een brug doet het voorbereidende werk voor de passage van een schip en schuift het daarna door naar zijn brugwachter.

1. Wacht zolang er geen aangemelde schepen zijn.
2. Kies een kijkrichting van waaruit zich ten minste 1 schip heeft gemeld.
3. Behandel het eerst gemelde schip uit die richting.
4. Bepaal voor dat schip welke doorgang het moet passeren (DOORGANG) en onderzoek of de brug daartoe geopend moet worden (ONDERDOOR = FALSE).
5. Draag het schip over aan de brugwachter.
6. Ga naar stap 1.

### 3.3 Brugwachter

Een brugwachter heeft de supervisie over de schepen die er onderdoor varen en het wegverkeer dat er overheen rijdt. De besturing van de wegverkeersstromen is in handen van de brugwachter zelf, voor de passage van schepen beschikt de brugwachter aan elke zijde van de brug over een REGELAAR.

Voor de bepaling van de opening- en sluittijdstippen heeft de brugwachter per kant van de brug de beschikking over een planningsset BRUGPLAN, die de schepen bevat die om een brugopening vragen.

1. Wacht zolang beide brugplannen leeg zijn en de bedrijfstijd nog niet verstreken is.
2. Indien de bedrijfstijd is verstreken wacht dan tot de nieuwe bedrijfstijd ingaat en herhaal vanaf stap 1.
3. Selecteer uit de beide brugplannen het schip dat het eerst is aangekomen of zal aankomen.
4. Bepaal het uiterste tijdstip waarop de brug geopend moet worden om het schip ongehinderd te laten passeren en wacht tot dit tijdstip is aangebroken.
5. Ga na of een brugopening nu in conflict is met een bloktijdenregeling. Zo ja, wacht tot de bloktijd voorbij is en herhaal vanaf stap 1.
6. Bepaal DEADLINE, zijnde het uiterste tijdstip waarop de brug gesloten moet zijn in verband met de bloktijdenregeling of de brugopeningsduurbepaling.
7. Zet de verkeerslichten op ROOD en open de brug.
8. Als de brug een draaibrug is, activeer dan beide regelaars; wacht tot die hun taak voltooid hebben en ga door met stap 11.
9. Activeer de regelaar aan de kant van het in stap 3 geselecteerde schip en wacht tot die klaar is.
10. Activeer de andere regelaar en wacht tot die klaar is.
11. Sluit de brug en zet de verkeerslichten op GROEN.
12. Laat alle schepen die inmiddels bij de brug zijn aangekomen (in WACHTRIJ zitten) en onder de brug door kunnen varen hun weg vervolgen.
13. Wacht tot de wegverkeerfiles zijn opgelost en herhaal vanaf stap 1.

### 3.4 Generator

Het werk van een generator bestaat uit wachten en het creëren en activeren van schepen.

De informatie betreffende de te creëren schepen is aangemaakt door model SUBMOD1 en bevat in een *schipregel*. Schipregels zijn gegroepeerd in *bladzijden* en R\_SCHEPEN is de verzameling bladzijden voor de betreffende generator.

1. Herhaal voor elke BLADZIJ in R\_SCHEPEN:
  1. Herhaal voor elke regel in BLADZIJ:
    1. Roep MACRO GEMAAKT aan zodat een schip wordt gecreëerd volgens de data in de datafile van de generator.
    2. Wacht tot het aankomstmoment van het schip, activeert dat en herhaal vanaf stap 1.1.1.



### 3.5 Klok

Deze klasse wordt gebruikt voor de besturing van de beschikbaarheid van kunstwerken

#### KLOK VAN DE BRUGWACHTER

Voor elke brugwachter is er een klok die er voor zorgt dat de brugwachter buiten bedrijfstijden niet werkt.

1. Zet het attribuut `WERKT` op `FALSE`.
2. Beschouw de volgende bedrijfstijd. Stop als die er niet is.
3. Wacht tot het aanvangsmoment en zet `WERKT` op `TRUE`.
4. Wacht tot het eindmoment.
5. Verleg de tijdstippen van de bedrijfstijd 1 week en herhaal vanaf stap 1.

#### KLOK VAN DE KOLK

Voor elke kolk is er een klok die er voor zorgt dat de kolk buiten bedrijfstijden niet werkt.

1. Beschouw de volgende bedrijfstijd en maak de kolk onzichtbaar.
2. Stel de kolk buiten bedrijf als hij niet bezig is met schutten.
3. Actualiseer te toestand van wachtende schepen die uitsluitend van deze kolk gebruik kunnen maken.
4. Stop als er geen bedrijfstijd is.
5. Wacht tot het begin van de bedrijfstijd.
6. Maak de kolk weer zichtbaar
7. Maak nieuw(e) opstelplan(nen).
8. Wacht tot het eind van de bedrijfstijd
9. Verleg de tijdstippen van de bedrijfstijd 1 week en herhaal vanaf stap 1

### 3.6 Kolk

Het werk van een kolk bestaat uit het schutten van schepen die door de sluismeester in zijn `OPSTELPLAN` zijn gezet. Schepen die daadwerkelijk bij de sluis zijn aangekomen worden opgenomen in `OPSTELRUIMTE`. De kolk is verantwoordelijk voor de schepen die zich in `OPSTELPLAN` bevinden vanaf `OPSTELRUIMTE` totdat ze de kolk verlaten hebben.

Bij iedere toestandsovergang zal de kolk daartoe op toestemming vragen door aanroep van de `MACRO` die aan het attribuut `VOORWAARDE` is gekoppeld. Indien nodig zal de kolk eerst wachten totdat de voorwaarden toestaan om door te gaan.

1. Wacht tot de sluismeester bepaalt dat er weer geschut moet worden.
2. Wacht, indien van toepassing op het begin van een operationele periode.
3. Duw elk schip in `OPSTELPLAN` de schutruimte in; als zo'n schip nog niet in de `OPSTELRUIMTE` zit moet daarop gewacht worden.
4. Laat een nieuw opstelplan maken.
5. Sluit de deuren.
6. Nivelleer het water in de kolk.
7. Open de deuren
8. Duw de schepen één voor één de kolk uit en activeer ze.
9. Herhaal vanaf stap 1

### 3.7 Main

MAIN is de standaard PROSIM component die op het hoogste niveau de besturing van de simulatie regelt.

1. Specificeer de gebruikte continue attributen.
2. Initialiseer het model:
  1. Initialiseer de nodige grootheden
  2. Inlezen gegevens
    1. scheepsklassen met vooraanmelding
    2. categorieën
    3. vaarwegen
    4. sluizen
    5. bruggen
    6. vernauwingen
    7. getijden
    8. wegintensiteiten
    9. regimes
    10. generatoren
    11. vaarkrommen
  3. Initialisatie debiet storestreams
3. Stop indien initialisatie mislukt.
4. Wacht totdat de simulatieduur is verstreken
5. Stel de rapporten samen
6. Sla de uitvoer ten behoeve van model QUERY op.

### 3.8 Observer

De observator zorgt ervoor dat de regimes wekelijks herhaald worden en dat regelmatig resultaten worden opgeslagen.

1. Indien het globale attribuut EINDIGEN de waarde TRUE heeft: zorg ervoor dat de simulatie wordt afgesloten en stop.
2. Wacht 1 uur
3. Verschuif niet meer actuele regimes een week.
4. Noteer een aantal grootheden met betrekking tot statistiek van bruggen.
5. Als er een dag is verstreken verleng dan de storestreams met betrekking tot debietregistratie.
6. Herhaal vanaf stap 1

### 3.9 Regelaar

Een brugwachter heeft voor elke kant van de brug een regelaar. Een regelaar laat de schepen aan zijn kant de geopende doorgang passeren.

1. Beschouw het schip in BRUGPLAN dat het eerst is aangekomen of aan zal komen.
2. Is er niet zo'n schip ga dan naar stap 7.
3. Bepaal of het schip nog mee kan in verband met regimes. Zo niet, ga dan naar stap 7.
4. Wacht eventueel tot het schip is aangekomen (zich in WACHTRIJ bevindt).
5. Wacht totdat het schip de brug is gepasseerd en activeer het.
6. Herhaal vanaf stap 1
7. Wordt een enkele of een dubbele brug geregeld, activeer dan de brugwachter en stop.
8. Wacht totdat er weer een schip in BRUGPLAN komt óf de andere regelaar ook in stap 8 van zijn proces is.
9. Is er weer een schip in BRUGPLAN herhaal dan vanaf stap 1.
10. Activeer de brugwachter (indien dat nog kan) en stop.

### 3.10 Schip

Het proces van een schip bestaat uit varen op een vaarweg, zich melden voor kunstwerken, wachten voor een brug, sluis of vernauwing, of het passeren van een gesloten doorgang van een brug. De procesbeschrijving valt daarom uiteen in een aantal delen.

Een schip in vrije vaart:

1. Indien het een speciaal schip is dat zich op een geëigende afstand van een sluis bevindt doet het een vooraanmelding
2. Indien het schip de beginknoop van een kunstwerk nadert meldt het zich aan naar gelang de aard van het naderende kunstwerk. Indien van toepassing wordt zijn ETA aangepast om inhalen te voorkomen.
3. Wacht totdat het traject bevaren is.
4. Bepaal het volgende traject.
5. Stop als er geen vervolgetraject meer is.
6. Ga naar stap 1 van het bij dat traject behorende proces.

Schip komt bij sluis:

1. Wacht op instructie van de sluismeester.
2. Neem plaats in de WACHTRUIMTE of de OPSTELRUIMTE al naar gelang de instructie.
3. Geef de controle over aan het sluispersoneel.

Schip komt bij brug:

1. Wacht op instructie van de brugwachter.
2. Kan en mag het schip onder een vaste doorgang door, passeer dan de brug en ga verder met stap 4 van het proces van een schip in vrije vaart.
3. Neem plaats in WACHTRIJ en indien van toepassing in BRUGPLAN.
4. Geef de controle over aan het brugpersoneel.

Schip komt bij een vernauwing:

1. Wacht totdat het schip de vernauwing mag invaren.
2. Passeer de vernauwing
3. Ga verder met stap 4 van het proces van een schip in vrije vaart.

### 3.11 Sluis

1. Geef het te behandelen schip INSTRUCTIE met betrekking tot de voortgang in zijn proces.
2. Stop.

### 3.12 Stroom

Het wegverkeer bij een brug is gemodelleerd als een continu proces.

Per categorie en per richting wordt een component STROOM gemaakt. Zo'n stroom komt alleen in actie als de verkeerslichten van zijn brug op "ROOD" gezet zijn. Hij zal dan al integrerende drie continue attributen bijhouden:

INHOUD	het aantal verkeersequivalenten in de file voor de brug
TOTAAL	het totale aantal door de brugopening beïnvloede verkeersequivalenten
CUM_WACHTTIJD	de totale wachttijd van alle door de brugopening beïnvloede verkeersequivalenten.

De STROOM zal blijven integreren tot de file opgelost is en de verkeerslichten weer op "GROEN" staan.

De waarden van de continue attributen worden bepaald door de differentiaal vergelijkingen in de module MAINMOD. Hierin wordt:

de verandering van INHOUD bepaald door het verschil in grootte tussen de aankomende en vertrekkende verkeersstroom,

de verandering in TOTAAL bepaald door de grootte van de aankomende verkeersstroom,

de verandering in CUM\_WACHTTIJD bepaald door de waarde van INHOUD (= grootte van de file).



### 3.13 Vernauwing

De vernauwing zelf is gedurende de simulatie slechts actief als bediende van zijn verkeerslichten en speelt dan ook geen actieve rol als de ontmoetingsregeling geldt. Het te volgen proces is afhankelijk van het type verkeerslichtregeling.

Starre verkeerslichtregeling

1. Begin aan kant  $k$ .
2. Zet het licht aan kant  $k$  op groen.
3. Laat, indien die er zijn, garantieschepen passeren en ga dan door met stap 5.
4. Wacht de maximale groentijd.
5. Zet het licht aan kant  $k$  op rood.
6. Wacht totdat de ontruimingstijd is verstreken.
7. Wissel van kant ( $k$  wordt  $3 - k$ ).
8. Herhaal vanaf stap 2.

Verkeersafhankelijke regeling

1. Begin aan kant  $k$ , noem de overkant  $l$ .
2. Zet het licht aan kant  $k$  op groen.
3. Wacht totdat er aan kant  $l$  een schip arriveert. Nu gaat de maximale groentijd in.
4. Laat garantieschepen aan kant  $k$  passeren.
5. Laat nog wachtende schepen passeren zolang de maximale groentijd niet wordt overschreden.
6. Zet het licht aan kant  $k$  op rood.
7. Wacht totdat het laatste schip uit de vernauwing is gevaren.
8. Wissel van kant ( $k$  wordt  $3 - k$ ).
9. Herhaal vanaf stap 2.

### 3.14 Wekker

Deze component zet of herziet het moment van activeren van de sluis.



## 4. De globale aspecten van SUBMOD2

In de volgende paragrafen wordt, in alfabetische volgorde, voor alle categorieën een opsomming gegeven van de gedefinieerde entiteiten, voor zover van toepassing met hun attributen, en een korte omschrijving van de betekenis. Indien die omschrijving ontbreekt dan is er sprake van een hulpgroothed.

### 4.1 Component

#### 4.1.1 MAIN

MAIN is een standaard component die alle niet tot een specifieke klasse of component behorende grootheden als attribuut heeft. Deze attributen zijn:

A_INTENS	REFERENCE TO SET	
AB	INTEGER	constante
AF	INTEGER	constante
AFH_KLAS[30]	INTEGER	
BEPAAL_DEADLINE	INTEGER	constante
BRUGGEN	REFERENCE TO SET	verzameling van alle bruggen
BUF[1000]	INTEGER	buffer t.b.v. werken met MATRIX files
CAP[2,2]	REAL	
CATEGORIE[30]	INTEGER	categorie van scheepsklasse <i>i</i>
CATWIS	INTEGER	
CC[6]	INTEGER	
CHAR	CHARACTER(1)	
CH10	CHARACTER(10)	
CH30	CHARACTER(30)	
CH4	CHARACTER(4)	
CH8	CHARACTER(8)	
CLUSTERS	REFERENCE TO SET	verzameling van alle clusters
COLORS[4,2]	INTEGER	constante
D_DICHT	INTEGER	constante
D_OPEN	INTEGER	constante
DAG[7]	REFERENCE	constante
DATA	STREAMREFERENCE	
DATE[4]	INTEGER	
DOSPAD	CHARACTER(76)	
DOSPADSHP	CHARACTER(76)	
DSTREEP	CHARACTER(127)	
EINDIGEN	LOGICAL	FALSE, het wijzigen naar TRUE beëindigt de simulatie en initieert de generatie van uitvoer-gegevens.
ESTREEP	CHARACTER(127)	
FILENAAM	CHARACTER(8)	
G[10]	REAL	
GENERATOREN	REFERENCE TO SET	verzameling van alle generatoren
GROEN	INTEGER	constante
HELP[2]	REFERENCE TO SET	
HULP	REFERENCE TO SET	
HULP2	REFERENCE TO SET	
HULP3[10]	REFERENCE TO SET	
I[2]	INTEGER	
IDLE	INTEGER	constante
II	INTEGER	
IKB	INTEGER	constante
IKF	INTEGER	constante
INFO1[7]	DOUBLE	
INFO2[7]	DOUBLE	
INVAREN	INTEGER	constante
J	INTEGER	
JJ	INTEGER	
K	INTEGER	

KK	INTEGER	
KLAS_BIJ_BRUG[30]	LOGICAL	
KLAS_BIJ_SLUIS[30]	LOGICAL	
KOSTEN[3,30]	REAL	passeerkosten voor scheepsklasse <i>j</i> bij een sluis ( <i>i</i> =1), een brug ( <i>i</i> =2) of een vernauwing ( <i>i</i> =3)
L	INTEGER	
LAATSTEBOOT	REFERENCE	
LEGBLOK	REFERENCE TO REGIMEBLOK	regimeblok zonder regimes
LINENR	INTEGER	
LL	INTEGER	
M	INTEGER	
MB	INTEGER	constante
MF	INTEGER	constante
N	INTEGER	
NAAR_OPSTELRUIMTE	INTEGER	constante
NEXTSTAT	REFERENCE	
NIEUW	LOGICAL	
NIEUW_OPSTELPLAN	INTEGER	constante
NIVELLEREN	INTEGER	constante
NODES	REFERENCE TO SET	verzameling van alle knooppunten
ORIGINEEL[6]	REFERENCE TO SET	
ORANJE	INTEGER	constante
PAGNR	INTEGER	
PASSEER_KNOOP	INTEGER	constante
PLAATS[3]	REFERENCE	
PLEK	REFERENCE	
QFILENAAM	CHARACTER(12)	naam van query file
REF[2]	REFERENCE	
REFBRUG	REFERENCE	
REFKOLK	REFERENCE	
REFNODE	REFERENCE	
REFREGIME	REFERENCE	
REFSHIP	REFERENCE	
REFSLUIS	REFERENCE	
REFVAARWEG	REFERENCE	
REGEL	CHARACTER(80)	
REGELAANTAL	INTEGER	
REGELOPH	INTEGER	
REGIMEBLOKKEN	REFERENCE TO SET	verzameling van alle regimeblokken
ROOD	INTEGER	constante
S_RECL	INTEGER	recordlengte van de scheepsbestanden
SB	INTEGER	constante
SCHIP_BIJ_MELDRAAI	INTEGER	constante
SCHIP_BIJ_SLUIS	INTEGER	constante
SCHIPNAAM	CHARACTER(8)	
SCHIPSTATS	REFERENCE TO SET	
SCHUTREGIMES	REFERENCE TO SET	verzameling van alle schutregimes
SCHUTREGISTRATOR	REFERENCE TO GENERATOR	generator die bestand met schuttingen registreert
SF	INTEGER	constante
SGARKLAS[30]	INTEGER	schepen van klasse <i>i</i> worden SPECIAAL m.b.t. sluisen
SIMULATIEDUUR	REAL	simulatieduur in uren
SLUIS_ZELF	INTEGER	constante
SLUIZEN	REFERENCE TO SET	verzameling van alle sluisen
STACK	REFERENCE TO SET	
START_DOORVAREN	INTEGER	constante
START_OVERLIGGEN	INTEGER	constante
START_SCHUTTEN	INTEGER	constante
T	REAL	

TEKST	CHARACTER(75)	
TEXT	CHARACTER(52)	
TEXT10	CHARACTER(20)	
TIME[4]	INTEGER	
TOON_VERNAUWING	INTEGER	0: geen tekst-animatie van een vernauwing 1: tekst-animatie van een vernauwing 2: tekst-animatie van een vernauwing met pauze na elke verandering
TOT	REAL	
UIT_OPSTELRUIMTE	INTEGER	constante
UITVAREN	INTEGER	constante
VAARWEGEN	REFERENCE TO SET	verzameling van alle vaarwegen
VERNAUWINGEN	REFERENCE TO SET	verzameling van alle vernauwingen
VGARKLAS[30]	INTEGER	schepen van klasse <i>i</i> worden VSPECIAAL m.b.t. vernauwingen
VIA_DOORGANG	INTEGER	constante
VIA_KOLK	INTEGER	constante
VOLGLENTE[3,3]	REAL	vrije volglenkte van schepen van categorie <i>i</i> op schepen van categorie <i>j</i> bij een vernauwing
VOOR_AANMELDING	INTEGER	constante
VOORKENNISBOOM	REFERENCE TO SET	
VOORW_INIT	MACRO	referentie naar MACRO ter initialisatie van de kolkvoorwaarden
VRIJE_KOLK	INTEGER	constante
W[2]	REAL	
WACHTEN	INTEGER	constante
WIT	INTEGER	constante
X	REAL	
XXX[6]	REAL	
X1	REAL	
X2	REAL	
Y	REAL	
Z	REAL	

#### 4.1.2 OBSERVATOR

Deze component is verantwoordelijk voor de wekelijkse herhaling van de regimes alsmede de registratie van brugopeningen per uur en kolkdebieten per dag, en heeft geen eigen attributen.



## 4.2 Class

### 4.2.1 BLADZIJ

Een BLADZIJ is een representant voor een blok informatie in een MATFILE en heeft geen eigen proces.

BLADINHOUD[10000]	REAL	inhoud van de bladzij
REGELS	INTEGER	aantal regels op de bladzij
ST_REC	INTEGER	eerste corresponderende record in matrixbestand
VOLLE_REGELS	INTEGER	aantal volledig gevulde regels op de bladzij

### 4.2.2 BRUG

Een BRUG representeert een brug in het netwerk.

B_NRTIJ	INTEGER	0: constant peil >0: nummer van te gebruiken getijdetabel
B_PEILPAR	REAL	B_NRTIJ[I]=0: waterstand t.o.v. gekozen referentievlak B_NRTIJ[I]>0: tijdtranslatie
BAFW[10]	REAL	standaardafwijking van de meetgrootheden
BCATij	REAL	extra doorvaartijd voor een schip van categorie <i>j</i> als dit achter een schip van categorie <i>i</i> de brug passeert
BGEM[10]	REAL	gemiddelde van de meetgrootheden
BMAX[10]	REAL	maximum van de meetgrootheden
BMIN[10]	REAL	minimum van de meetgrootheden
BOPEN	REAL	
BRFUIKLENGTE[2]	REAL	fuiklengte aan kant <i>i</i> van de brug
BSCHIPSTATS	REFERENCE TO SET	verzameling SCHIPSTATS
HOOGTE[2]	REAL	hoogte van doorgang <i>i</i> t.o.v. het gekozen referentievlak
LICHT	CHARACTER(5)	verkeerslichten voor het wegverkeer ("ROOD" of "GROEN")
N_UREN	INTEGER	aantal geregistreerde uren
NBOPEN	INTEGER	aantal brugopeningen sinds laatste registratie
NRBRUG	INTEGER	identificatienummer
NRDOORGANGEN	INTEGER	aantal doorgangen
OPEN	LOGICAL	toestand van het beweegbare gedeelte van de brug
OPLOS	REAL	oplostijd van de file
REGIMES	REFERENCE TO SET	verzameling regimes
TBOPEN	REAL	benodigde tijd om de brug te openen
TBSLUIT	REAL	benodigde tijd om de brug te sluiten
TOTOPEN	INTEGER	aantal brugopeningen
TYPE	CHARACTER(6)	soort brug: ENKEL, DUBBEL of DRAAI
VRIJVERKEER	INTEGER	aantal opgeloste files bij de brug
WACHTRIJ[2]	REFERENCE TO SET	verzameling wachtende schepen aan kant <i>i</i> van de brug
WATERPAD[2]	REFERENCE TO VAARWEG	vaarweg "in zicht" aan kant <i>i</i> van de brug
WEGVERKEER	REFERENCE TO SET	verzameling wegverkeersstromen
WIJDTE[2]	REAL	breedte van doorgang <i>i</i> van de brug
WKOSTEN[2]	REAL	kosten voor het wegverkeer per categorie
ZIJDE[2]	REFERENCE TO NODE	NODE aan kant <i>i</i> van de brug
ZIJNWACHTER	REFERENCE TO BRUGWACHTER	de brugwachter



### 4.2.3 BRUGWACHTER

Deze klasse representeert de brugwachters.

BRUGHULP[2]	REFERENCE TO REGELAAR	de regelaar aan kant <i>i</i> van de brug
BRUGKANT	INTEGER	de kijkrichting van de brugwachter
BRUGPLAN[2]	REFERENCE TO SET	verzameling schepen aan kant <i>i</i> van de brug die een brugopening vragen
DEADLINE	REAL	tijdstip waarop de brug gesloten moet worden in verband met regimes
WERKT	LOGICAL	TRUE als de brugwachter in bedrijf is
ZIJNBRUG	REFERENCE TO BRUG	de brug

### 4.2.4 CLUSTER

Een CLUSTER definieert een verzameling scheepsklassen en heeft geen eigen proces.

C_AANTAL	INTEGER	
C_KOSTEN	REAL	
C_NOVER	INTEGER	
C_PASTIJD	REAL	
KLASSEN[30]	LOGICAL	TRUE/FALSE als scheepsklasse <i>i</i> wel/niet tot cluster behoort
NRCLUSTER	INTEGER	identificatienummer

### 4.2.5 DAGTYPE

Een hulpklasse ter benoeming van de dagen van de week. Daarvoor wordt het systeem attribuut NAME gebruikt, zodat verder geen attributen nodig zijn. Deze klas kent geen eigen proces.

### 4.2.6 GENERATOR

Voor iedere te beschouwen route wordt een generator gebruikt die de sequentie van schepen, zoals die bepaald is met behulp van model SUBMOD1, reconstrueert.

AANTAL_SCHEPEN	INTEGER	aantal schepen voor route
BLADEN	REFERENCE TO SET	verzameling bladzijden met tijdgegevens
HIST_FULLNAME	CHARACTER(76)	de volledige DOS naam van het bestand met historische gegevens
HISTORY_FILENAAM	CHARACTER(12)	de naam van het bestand met historische gegevens
HSCHIP[8]	INTEGER	MATFILE handle
HTIJD[8]	INTEGER	MATFILE handle
PALEN_HEEN	REFERENCE TO SET	verzameling door zijn schepen te passeren mijlpalen
PALEN_TERUG	REFERENCE TO SET	verzameling door zijn schepen te passeren mijlpalen
R_SCHEPEN	REFERENCE TO SET	verzameling bladzijden met scheepsgegevens
REGELLENGTE	INTEGER	regellengte op een bladzij
REGELS_PER_BLADZIJ	INTEGER	aantal regels per bladzij
ROUTE_FILENAAM	CHARACTER(12)	de naam van het bestand met route gegevens
ROUTE_HEEN	REFERENCE TO SET	verzameling met door zijn schepen te volgen NODES
ROUTE_TERUG	REFERENCE TO SET	verzameling met door zijn schepen te volgen NODES
SCHIPREGEL	INTEGER	
TIJDRECEORDS	INTEGER	
TSCHIP[6]	INTEGER	
WEG_HEEN	REFERENCE TO SET	verzameling met door zijn schepen te volgen trajecten
WEG_TERUG	REFERENCE TO SET	verzameling met door zijn schepen te volgen trajecten
NR_GEN	INTEGER	identificatienummer

GETA	REAL	ETA van het eerstvolgende te genereren schip
------	------	--

#### 4.2.7 IPUNT

Een hulpklassse bij het bepalen van intensiteitspatronen zonder eigen proces.

NN1	INTEGER
NN2	INTEGER

#### 4.2.8 KLOK

Een KLOK verzorgt de wekelijkse herhaling van de operationele periodes van brugwachters en kolken.

INHIBITOR	REFERENCE	de component die door de klok wordt gedomineerd
SHIFT	REFERENCE TO SET	verzameling werktijden
WERK	REFERENCE	de actuele werktijd

#### 4.2.9 KOLK

Een KOLK modelleert een kolk in een sluis.

AANTAL	REAL	
BEROEPS_RUIMTE[2]	REAL	door beroepsvaart bezette ruimte in de kolk vanaf kant <i>i</i> inclusief veiligheidszones
BEZIG	LOGICAL	
BRDHOOFD	REAL	breedte van het kolkhoofd
BRDTKOLK	REAL	breedte van de kolk
BUITEN_BEDRIJF	REAL	
BUURKOLK	REFERENCE TO KOLK	de kolk waarmee een afhankelijkrelatie van toepassing is
CAT_IN_KOLK	INTEGER	t.b.v. animatie
CAT <i>i</i> <sub>j</sub>	REAL	breedte van de veiligheidszone bij gemengd schutten tussen schepen van categorie <i>i</i> en categorie <i>j</i>
CUR_DEBIET	REAL	
DEB_REDUCTIE	REAL	factor waarmee de totale nivelleertijd wordt vermenigvuldigd in geval van hevelen
DEBIET[2]	REAL	
DREMPEL[2]	REAL	hoogte van de drempel aan kant <i>i</i> t.o.v. gekozen referentievlak
DUUR	REAL	
FACNIVEL	REAL	factor voor tijdsduur bij nivelleren
FUIKLENGTE[2]	REAL	fuiklengte aan kant <i>i</i>
GARKOLK[30]	INTEGER	geeft aan of een schip van klasse <i>i</i> met een garantieschip meegeschat mag worden
GEEN_UITSTEL[2]	LOGICAL	
GESCHIKT[30]	LOGICAL	TRUE als schepen van klasse <i>i</i> van de kolk gebruik mogen maken
HEVEL_FACTOR	REAL	
ICAT <i>i</i> <sub>j</sub>	REAL	extra vaartijd voor een schip van categorie <i>j</i> als dit achter een schip van categorie <i>i</i> de kolk invaart
IDLE_TIME	REAL	
INDELING[2]	REFERENCE TO SET	verzameling hoekpunten in de vrije ruimte van een planning voor kant <i>i</i>
KMAX	LOGICAL	TRUE in geval van volschutting, anders FALSE
KOLK_STAT[2]	REFERENCE	de KOLKSTAT per richting <i>i</i>
KOLK_STATUS	INTEGER	de toestand van de kolk
KOLK_WENS	INTEGER	de door VOORWAARDE geconditioneerde toestand

LENGKOLK	REAL	lengte van de kolk
NRKOLK	INTEGER	identificatie; indien oneven/even dan begint de kolk open aan kant 1/2
NULNIVEL	REAL	standaard tijdsduur voor nivelleren
OPP_KOLK	REAL	t.b.v. animatie
OPPREND	REAL	kolkoppervlakte rendement
OPSTELPLAN[2]	REFERENCE TO SET	planningset voor schepen bij kant <i>i</i> van de kolk
OPSTELRUIMTE[2]	REFERENCE TO SET	verzameling aan kant <i>i</i> opgestelde schepen
POSTSCHIP	REFERENCE TO SCHIP	
PRESCHIP	REFERENCE TO SCHIP	
SCHUTRUIMTE	REFERENCE TO SET	verzameling schepen in de schutruimte
SCHUTTENDE	LOGICAL	
SIDE	INTEGER	kant waar de kolk naar gericht staat
START_IDLE	REAL	
TOPEN[2]	REAL	benodigde tijd voor het openen van de kolkdeuren aan kant <i>i</i>
TOT_DEBIET[8,2]	REAL	totaal debiet op dagtype <i>i</i> naar richting <i>j</i>
TOTLAADVERM	REAL	
TSLUIT[2]	REAL	benodigde tijd voor het sluiten van de kolkdeuren aan kant <i>i</i>
UCAT <sub>ij</sub>	REAL	extra vaartijd voor een schip van categorie <i>j</i> als dit achter een schip van categorie <i>i</i> de kolk uitvaart
VOORWAARDE	MACRO	verwijzing naar de MACRO waarin de kolkvoorwaarden zijn verwoord
VULGRAAD	REAL	t.b.v. animatie
VW_START	REAL	
WACHT_OP_DEADLINE	LOGICAL	
WACHT_OP_PERIODE	LOGICAL	
WACHT_OP_VOORW	REAL	
WATERLENGTE	REAL	lengte ter bepaling van het schutwaterverlies
ZIJNSLUIS	REFERENCE TO SLUIS	de sluis

#### 4.2.10 KOLKSTAT

Een klasse van componenten ten behoeve van het verzamelen van statistische informatie betreffende kolken. Deze klasse heeft geen eigen proces.

FACT[2]	REAL	richtingscoëfficiënt in de formule van de cyclustijd van volschuttingen voor richting <i>i</i>
INTC[2]	REAL	constante term in de formule van de cyclustijd van volschuttingen voor richting <i>i</i>
KGEM[3]	REAL	gemiddelde per schutting voor resp. het aantal geschutte schepen, de schutduur en het geschut laadvermogen
LEEGOM	INTEGER	aantal leegom schuttingen
R[5]	REAL	
S[5]	REAL	
TOTOVER	INTEGER	totaal aantal geschutte schepen
TOTSCHUT	INTEGER	aantal schuttingen
VOLSCHUT	INTEGER	aantal volschuttingen

#### 4.2.11 MIJLPAAL

Een klasse van componenten ten behoeve van het verzamelen van statistische informatie betreffende mijlpalen, zijnde een karakteristiek punt in een vaarroute. Deze klasse heeft geen eigen proces.

M_INDEX	INTEGER	index in matrixbestand
M_KOP	INTEGER	identificatie aan de kop van het bestand



M LENGTE	INTEGER	aantal gegevens
M_TYPE	INTEGER	soort gegevens
		0 = passeren knoop
		1 = passeren sluis
		2 = passeren brug
		3 = passeren vernauwing

#### 4.2.12 NODE

Klasse die de knopen in het netwerk representeert. Deze klasse heeft geen eigen proces.

MELDINGEN	REFERENCE TO SET	verzameling bij het kunstwerk gemelde schepen
NRNODE	INTEGER	identificatienummer

#### 4.2.13 PUNT

Een hulpklasse die gebruikt wordt bij het vullen van een kolk. Deze klasse heeft geen eigen proces.

XX	REAL
YY	REAL

#### 4.2.14 REGELAAR

Een klasse van componenten die de verkeersafhandeling bij een brug afhandelt.

BRIDGE	REFERENCE TO BRUG	de brug
BSIDE	INTEGER	de kant van de brug
CASE	INTEGER	de wijze waarop zijn schip de brug passeert (eerste schip, volgschip, lusschip, lopende vaart)
LASTBOOT	REFERENCE TO SCHIP	
NUMBER	INTEGER	
NXTBOOT	REFERENCE TO SCHIP	
WATCHER	REFERENCE TO BRUGWACHTER	de "werkgever" van de regelaar

#### 4.2.15 REGIME

Een klasse van datacomponenten die regimes met betrekking tot bruggen vastleggen. Deze klasse heeft geen eigen proces.

GELDTOT	REAL	tijdstip waarop het regime stopt
GELDVAN	REAL	tijdstip waarop het regime ingaat
PARAM	REAL	de maximale brugopeningsduur in geval van een brugopeningsduurbeperring
SOORT	INTEGER	1 = bloktijd 2 = duurbeperring

#### 4.2.16 REGIMEBLOK

Een blok van REGIMES. Deze klasse heeft geen eigen proces.

BLOKNR	INTEGER	identificatienummer
R_INHOUD	REFERENCE TO SET	verzameling regimes

#### 4.2.17 SCHIP

De representaties van de schepen waar het allemaal om draait.

BLAD	REFERENCE TO BLADZIJ	te beschrijven bladzij
BLADSTART	INTEGER	plaats op de bladzijde waar de gegevens van dit schip beginnen
BREEDTE	REAL	breedte van het schip
CREATIETIJD	REAL	tijdstip van aanvang van de route
CUMAFST	REAL	
CUR_PLEK	REFERENCE	het traject waarin het schip zich bevindt
DIEPGANG	REAL	diepgang van het schip
DOEL	REFERENCE TO NODE	de eerstvolgende NODE



DOORGANG	INTEGER	nummer van de te passeren doorgang van een brug
ETA	REAL	berekende aankomsttijd bij het eind van de vaarweg
GARANTIE	LOGICAL	geeft aan of het schip gebruik maakt van de garantieregeling bij het passeren van een sluis
GEL	INTEGER	ladingstoestand 1 = ongeladen 2 = geladen
GEN	REFERENCE TO GENERATOR	de generator
INSTRUCTIE	REFERENCE	verwijzing naar te passeren kunstwerk (BRUG, SLUIS of KOLK)
KIELSPELING	REAL	kielspeling van het schip
KLAS	INTEGER	nummer van de scheepsklasse
KLEINSTE_KOLK	REFERENCE TO KOLK	kolk met het minste schutwaterverlies waar het schip in past
LAADVERM	REAL	laadvermogen van het schip
LENGTE	REAL	lengte van het schip
MPALEN	REFERENCE TO SET	verzameling te passeren mijlpalen
ONDERDOOR	LOGICAL	geeft aan of het schip de aangewezen doorgang in gesloten toestand mag passeren
PLANBREEDTE	REAL	extra gereserveerde breedte m.b.t. kolkplanningen
PLANLENGTE	REAL	extra gereserveerde lengte m.b.t. kolkplanningen
RICHTING	INTEGER	
ROUTE	REFERENCE TO SET	een set met de passeren NODEs
SNELHEID	REAL	snelheid van het schip op de vaarwegen
SPECIAAL	LOGICAL	TRUE als het schip zich eerder aanmeldt bij een sluis in verband met de garantieregeling
START	REFERENCE TO NODE	de laatst gepasseerde NODE
SCHRIKH	REAL	schrikhoogte van het schip
STRIJKH	REAL	strijkhoogte van het schip
TBIJSLUIS	DOUBLE	aankomsttijd bij een sluis
TL	REAL	
TS	REAL	
TW	REAL	
VAARPLAN	REFERENCE TO SET	een set met te bevaren trajecten
VORIGE_MIJLPAAL	REFERENCE TO MIJLPAAL	te passeren of net gepasseerde mijlpaal
VSPECIAAL	LOGICAL	TRUE als het schip zich eerder aanmeldt bij een vernauwing in verband met de garantieregeling
WACHTEND	LOGICAL	beschrijft toestand in geval van oponthoud TRUE = wachtend FALSE = overliggend

#### 4.2.18 SCHIPSTAT

Een klasse van componenten ten behoeve van het verzamelen van statistische informatie betreffende schepen. Deze klasse heeft geen eigen proces.

S_AANTAL[2, 30]	INTEGER	aantal gepasseerde schepen uit richting <i>i</i> van scheepsklasse <i>j</i>
S_CUMKOSTEN[2, 30]	REAL	totale passeerkosten van schepen uit richting <i>i</i> van scheepsklasse <i>j</i>
S_CUMPASTIJD[2, 30]	REAL	totale passeertijd van schepen uit richting <i>i</i> van scheepsklasse <i>j</i>

S_NOVER[2,30]	INTEGER	aantal overliggers onder schepen uit richting <i>i</i> van scheepsklasse <i>j</i>
S_VOLGNR	INTEGER	gebruikte kolk/doorgang van het betreffende kunstwerk of richting bij vernauwing

#### 4.2.19 SCHUTREGIME

Een klasse van datacomponenten die regimes met betrekking tot schutten vastleggen. Deze klasse heeft geen eigen proces.

BENUTTINGSPERC[2]	REAL	minimale benuttingsgraad voor <i>goede</i> en <i>verkeerde</i> kant
COMBITYPE	CHARACTER(7)	type vergelijking
SCHUTPERIODES	REFERENCE TO SET	periodes m.b.t. periodiek schutten
SREGIMENR	INTEGER	identificatie nummer
VOORKENNISTIJD[2]	INTEGER	voorkennistijd per kant <i>i</i>
WACHT_GEWICHT_CAT3	REAL	gewichtsfactor voor de wachttijd van de recreatievaart
XTRA_WACHTTIJD	INTEGER	wachttijd criterium

#### 4.2.20 SLUIS

De representant van een sluis.

ALLE_KOLKEN	REFERENCE TO SET	verzameling van alle kolken in de sluis
BESTKOLK	REFERENCE TO KOLK	
DEB_AANT[7]	INTEGER	aantal debietwaarnemingen per dagtype
DEBIETBEPALING	LOGICAL	TRUE indien debiet te bepalen
GEMENGD_SCHUTTEN	LOGICAL	TRUE indien gemengd schutten
KANT[2]	REFERENCE TO NODE	NODE aan kant <i>i</i> van de sluis
KOLKEN	REFERENCE TO SET	verzameling in bedrijf zijnde kolken
MN_SCHUTREGIME	REFERENCE TO SCHUTREGIME	invulling van de schutstrategie
NRSLUIS	INTEGER	identificatienummer
ONMOGELIJK[2]	REFERENCE TO SET	planningset voor kant <i>i</i> van de sluis
PERIODIEK	LOGICAL	TRUE in geval van periodiek schutten
REMANENT[2]	REFERENCE TO SET	planningset voor kant <i>i</i> van de sluis
S_NRTIJ[2]	INTEGER	per kant <i>i</i> : 0: constant peil >0: nummer van te gebruiken getijdetabel
S_PEILPAR[2]	REAL	S_NRTIJ[I]=0: waterstand t.o.v. gekozen referentievlak S_NRTIJ[I]>0: tijdtranslatie
SCHUTPLANNING	MACRO	verwijzing naar de MACRO die de schutplanningen maakt
SLUISWEKKER	REFERENCE TO WEKKER	wekker die de schutplanning activeert
SSCHIPSTATS	REFERENCE TO SET	verzameling SCHIPSTATS
STRATEGIE	INTEGER	nummer van de planningstrategie
VOORKEUR[2]	REFERENCE TO SET	planningset voor kant <i>i</i> van de sluis
VRIJE_KOLKEN	REFERENCE TO SET	
WACHTRUIMTE[2]	REFERENCE TO SET	verzameling wachtende schepen aan kant <i>i</i> van de sluis
WACHTTIJD_ONMOGELIJK	LOGICAL	TRUE indien het niet <i>kunnen</i> passeren van de sluis door schepen tot de wachttijd wordt gerekend

#### 4.2.21 STROOM

De afbeelding van een verkeersstroom over een brug.

CAT	INTEGER	categorie van de verkeersequivalten: 1 = snelverkeer 2 = langzaam verkeer
CUM_WACHTTIJD	CONTINUOUS (1)	totale wachttijd van alle beïnvloede verkeersequivalenten bij de brugopening
INHOUD	CONTINUOUS (1)	aantal verkeersequivalenten in de file
MIJNBRUG	REFERENCE TO BRUG	de brug
NRINTENS	INTEGER	tebelnummer van het intensiteitspatroon
OPLOSVERMOGEN	REAL	afrijcapaciteit van de file per rijstrook
RICHT	INTEGER	rijrichting
STROKEN	INTEGER	aantal beschikbare rijstroken
TABNR	INTEGER	tabelnummer m.b.t. de intensiteit
TOTAAL	CONTINUOUS (1)	totaal aantal beïnvloede verkeersequivalenten bij de brugopening
WEEKTOT	REAL	aantal verkeersequivalenten dat de brug per week passeert

#### 4.2.22 VAARWEG

Een ononderbroken stuk vaarwater tussen twee knopen. Deze klasse heeft geen eigen proces.

AFSTAND	REAL	lengte van de vaarweg
RAAI [2]	REFERENCE TO NODE	NODE aan het uiteinde van de vaarweg
VAARVAK	REFERENCE TO SET	verzameling schepen die de vaarweg bevaren

#### 4.2.23 VERNAUWING

De representant van een vernauwing.

EBENUT [2]	REAL	
EINDGROEN	REAL	
ENGTE	REFERENCE TO SET	verzameling schepen in de engte
GARANTIESCHEPEN	REFERENCE TO SET	verzameling garantieschepen
INVAREN_VRIJ [2]	LOGICAL	
MAXGROEN [2]	REAL	maximale groentijd per kant
NRVERNAUWING	INTEGER	identificatienummer
ONTMOETING [30, 30]	LOGICAL	ontmoetingstabel
ONTRUIMINGSTIJD	REAL	ontruimingstijd
PLUSTIJD [30]	REAL	
REGELING	CHARACTER (3)	soort verkeersregeling: VL of TAB
SNELSTE_EERST	LOGICAL	volgorde van invaren van wachtende schepen
STAR	LOGICAL	type regeling bij gebruik van verkeerslichten
V_AFSTAND	REAL	lengte van de vernauwing
VERNAUWINGSKANT	INTEGER	
VLICHT [2]	INTEGER	verkeerslichten
VOORRANGSKANT	INTEGER	geldig bij ontmoetingsregeling
VRIJSTELLING [2]	REAL	
VSCHIPSTATS	REFERENCE TO SET	verzameking SCHIPSTATS
VVAARSNELHEID [30, 2]	REAL	vaarsnelheid in de vernauwing per klasse en ladingstoestand
VWACHTRIJ [2]	REFERENCE TO SET	verzameling wachtende schepen aan kant <i>i</i> van de vernauwing
VZIJDE [2]	REFERENCE TO NODE	NODE aan kant <i>i</i> van de vernauwing



#### 4.2.24 WEKKER

Hulpcomponenten ten behoeve van schutstrategieën met meerdere kolken.

WEKSLUIS                      REFERENCE TO SLUIS                      de te activeren sluis

#### 4.2.25 WERKTIJD

Een datacomponent die een operationele periode definieert. Deze klasse heeft geen eigen proces.

BEGIN                      REAL                      tijdstip begin werktijd  
EIND                      REAL                      tijdstip einde werktijd

### 4.3 Table

BRUG_VOLG[30,2]	TABLE(3)	tabel met doorvaartijden bij een brug per scheepsklasse en ladingstoestand voor volgschepen
BRUG_1STE[30,2]	TABLE(3)	tabel met doorvaartijden bij een brug per scheepsklasse en ladingstoestand voor voorste chepen
DOODTIJ[30]	TABLE(26)	dood tij tabel
INTENS[10]	TABLE(170)	wegverkeersintensiteiten
LUSBRUG[2]	TABLE(10)	luskrommes bij een brug
LUSTIJD[2]	TABLE(10)	lustijden bij een sluis
SPRINGTIJ[30]	TABLE(26)	spring tij tabel
TIJDIN_VOLG[30,2]	TABLE(3)	tabel met invaartijden bij een sluis per scheepsklasse en ladingstoestand voor volgschepen
TIJDIN_1STE[30,2]	TABLE(3)	tabel met invaartijden bij een sluis per scheepsklasse en ladingstoestand voor voorste schepen
TIJDUIT_VOLG[30,2]	TABLE(3)	tabel met uitvaartijden bij een sluis per scheepsklasse en ladingstoestand voor volgschepen
TIJDUIT_1STE[30,2]	TABLE(3)	tabel met uitvaartijden bij een sluis per scheepsklasse en ladingstoestand voor voorste schepen
VASTEBRUG[2]	TABLE(10)	doorvaartijden bij vaste bruggen



## 4.4 *Inputstream*

BIBGETIJ	BIBFILE met doodtij- en springtijtabellen.
BIBINTW	BIBFILE met aanbodspatronen voor het wegverkeer.
BIBKLAS	BIBFILE met statistische informatie over de karakteristieken van de verschillende scheepsklassen en de indeling van deze klassen naar categorieën.
BIBKOSTEN	BIBFILE met de verschillende soorten kosten voor het scheepvaartverkeer.
BIBREGIMES	BIBFILE met tabellen met bloktijden en openingsduurbeperingen zoals deze voor kunnen komen bij bruggen.
BIBSCHUTREGIMES	BIBFILE met schutregimes zoals deze voor kunnen komen als schutwaterbesparende maatregelen bij sluizen.
BIBVAART	BIBFILE met informatie op grond waarvan de in- en uitvaartijden bij sluizen en de (onder)doorvaartijden bij bruggen bepaald kunnen worden.
KOPCLUST	MODELFILE met informatie voor model SUBMOD2 met betrekking tot de lay-out van een rapport (aanhef en clustervorming van scheepsklassen).
NETWERK	MODELFILE waarin de gebruiker de configuratie van het te simuleren netwerk aangeeft.

## 4.5 *Outputstream*

ONDERZOEK	OUTPUTSTREAM die de mogelijkheid biedt regels toe te voegen aan een bestand met resultaten; door programmeur in te voorzien, niet door gebruiker
QUERY_INVOER	wordt gebruikt om gegevens op weg te schrijven die als invoer zullen dienen voor het QUERY model
RAPPORT	wordt gebruikt om de door SUBMOD2 samengestelde rapporten op weg te schrijven

## 4.6 External

CASE_STRING	<p>vervangt in een string een deel van of alle letters door hun kapitalen dan wel hun onderkast</p> <p>Parameters</p> <p>&lt;STRING&gt; de te modificeren string</p> <p>START de start van de substring</p> <p>LENGTE de lengte van de substring</p> <p>TYPE het conversiotype</p> <p>1 vervang door kapitalen</p> <p>2 vervang door onderkast</p> <p>Terugkeerwaarde</p> <p>0 Ok</p> <p>-1 Ongeldig type</p>
CHAR_IN_STRING	<p>vervangt een deel van een string door een andere string</p> <p>Parameters</p> <p>&lt;STRING&gt; de te modificeren string</p> <p>TSTART startpositie van de modificatie</p> <p>LENGTE de lengte van de modificatie</p> <p>SOURCE de in te voegen string</p> <p>SSTART de startindex in SOURCE</p> <p>Terugkeerwaarde</p> <p>0 Ok</p> <p>-1 Ongeldige substring specificatie</p>
CHREAD_MATFILE	<p>leest een string uit een MATFILE</p> <p>Parameters</p> <p>&lt;HANDLE&gt; de MATFILE handle</p> <p>POSITIE de positie van de te lezen string in het current record</p> <p>&lt;TARGET&gt; het attribuut van het type CHARACTER waarin het resultaat moet worden opgeslagen</p> <p>AANTAL het aantal te lezen karakters</p> <p>Terugkeerwaarde</p> <p>0 altijd; conversie fouten genereren een PROSIM runtime error</p>
CLEAR_TEXT_BOX	<p>wist een rechthoekig deel van het display</p> <p>Parameters</p> <p>L1 eerste regel</p> <p>L2 laatste regel</p> <p>C1 eerste kolom</p> <p>C2 laatste kolom</p> <p>CLR kleur</p> <p>Terugkeerwaarde</p> <p>0 Ok</p> <p>1 Inconsistente parameters of scherm is niet in textmode</p>
CLEAR_TEXT_SCREEN	<p>wist het gehele display</p> <p>Terugkeerwaarde</p> <p>0 Ok</p> <p>1 scherm is niet in textmode</p>

CREATE_MATFILE	maakt een nieuwe MATFILE aan	
	Parameters	
	<HANDLE>	de MATFILE handle, een integer array van 8 elementen
	FNAAM	de naam van de file; indien het tweede karakter van de naam een dubbele punt is (:) wordt de naam verondersteld een volledige DOS filename te zijn, zoniet dan de filename (zonder extensie, dus maximaal 8 karakters lang) met de current data directory als pad
	RECLEN	het aantal woorden (ban 2 bytes) per record
	Terugkeerwaarde	
	1	Ok
	-1	handle deugt niet
	-2	duplicate name
	-3	disk error
	-4	write error
CLOSE_MATFILE	sluit een MATFILE	
	Parameters	
	<HANDLE>	de MATFILE handle
	Terugkeerwaarde	
	1	Ok
	-1	handle deugt niet
	-2	file niet geopend
	-3	disk error
ERASE_FILE	verwijdert een bestand van disk	
	Parameters	
	FILENAME	de naam van de file
	Terugkeerwaarde	
	0	Ok
	-1	file niet gevonden
ERASE_MATFILE	verwijdert een MATFILE van disk	
	Parameters	
	FILENAME	de naam van de file
	Terugkeerwaarde	
	0	Ok
	-1	file niet gevonden
EXIST_FILE	test of een bestand met gegeven naam op disk bestaat	
	Parameters	
	FILENAME	de naam van de file inclusief DOS-pad
	Terugkeerwaarde	
	0	file bestaat
	-1	file niet gevonden
EXTEND_MATFILE	breidt een MATFILE uit	
	Parameters	
	<HANDLE>	de MATFILE handle
	Terugkeerwaarde	
	>0	nieuwe current record locatie
	-1	handle deugt niet
	-2	file niet geopend
	-3	disk error



GET_COLOURS	vraagt de door PROSIM gebruikte kleurinstellingen op	
	Parameters	de kleur instellingen
	<N[4,2]>	N[* , 1]: voorgrond N[* , 2]: achtergrond N[1, *]: standaard N[2, *]: attentie N[3, *]: invoer N[4, *]: foutmeldingen
	Terugkeerwaarde	
	0	Ok
	1	argument verkeerd gedimensioneerd
GET_DATE	vraagt de actuele datum op	
	Parameters	de datum
	<N[4]>	N[1]: jaar N[2]: maand N[3]: dag N[4]: dagnummer (0=Zondag...)
	Terugkeerwaarde	
	0	Ok
	1	argument verkeerd gedimensioneerd
GET_TIME	vraagt de actuele tijd op	
	Parameters	de tijd
	<N[4]>	N[1]: uur N[2]: minuten N[3]: seconden N[4]: centi-seconden
	Terugkeerwaarde	
	0	Ok
	1	argument verkeerd gedimensioneerd
INFO_STORESTREAM	vraagt een aantal kenmerken betreffende de inhoud van een STORESTREAM op	
	Parameters	de filename
	FNAME	de naam van de STORESTREAM
	SNAME	een DOUBLE array met 7 elementen
	<INFO>	INFO[1]: aantal entries INFO[2]: minimum waarde INFO[3]: maximum waarde INFO[4]: gemiddelde INFO[5]: standaardafwijking INFO[6]: min van de onafhankelijke INFO[7]: max van de onafhankelijke
	Terugkeerwaarde	
	0	Ok
	-1	buffer allocatie mislukt
	-2	write error
	-3	file niet gevonden
	-4	open error
	-5	read error
	-6	file is leeg
	-7	file verouderd
	-8	stream niet gevonden
	-9	INFO verkeerd gedimensioneerd

LOCATE_MATFILE	positioneert de lees/schrijf pointer in een MATFILE
Parameters	
<HANDLE>	de MATFILE handle
LOKATIE	gewenste record nummer
Terugkeerwaarde	
>0	current recod indien Ok
-1	handle deugt niet
-2	file niet open
MOVE_BLOCK_MATFILE	leest of schrijft een blok records van/naar een MATFILE
Parameters	
<HANDLE>	de MATFILE handle
REC1ST	start record
NUMREC	aantal records
<VECTOR>	PROSIM container, een REAL array van voldoende grootte
VEC1ST	start index in VECTOR
TYPE	type van de operatie
	0: lezen
	1: schrijven
Terugkeerwaarde	
>0	aantal verplaatste records indien Ok
-1	handle deugt niet of file niet open
-2	TYPE ongeldig
-3	VEC1ST ongeldig
-4	VECTOR te klein
-5	write error
-6	read error
-7	file is read only
NUM_IN_STRING	converteert een numerieke waarde naar een (sub)string
Parameters	
<TARGET>	de te modificeren string
TSTART	start van de substring in TARGET
LENGTE	de lengte van de substring
SOURCE	de te converteren numerieke waarde
CODE	het type van de conversie
	<0: scientific
	0: integer
	>0: fixed point met CODE decimalen
Terugkeerwaarde	
0	Ok
-1	de substring valt niet binnen TARGET
-2	substring te kort om geconverteerde waarde te bevatten
OPEN_MATFILE	opent een MATFILE
Parameters	
<HANDLE>	de MATFILE handle
FNAME	de filename
Terugkeerwaarde	
1	Ok
-1	handle deugt niet
-2	file bestaat niet
-3	file in gebruik
-4	read error

READ_MATFILE	leest een numerieke waarde uit een MATFILE
	Parameters
	<HANDLE> de MATFILE handle
	POSITIE offset in current record
	TYPE type van de te lezen waarde
	1: integer
	2: real
	4: double
	Terugkeerwaarde
	De gelezen waarde. Conversie fouten resulteren in een PROSIM runtime error
STREAMS_FILENAME	vraagt de met een gegeven STREAM geassocieerde bestandsnaam op
	Parameters
	STREAM de naam van de INPUTSTREAM of OUTPUTSTREAM in hoofdletters
	<FNAME> de filename zonder pad en extensie
	<PNAME> het pad van de filename
	Terugkeerwaarde
	0 Ok
	-1 STREAM niet gevonden
	-2 STREAM niet verbonden met bestand
	-3 FNAME onjuist gedimensioneerd
	-4 PNAME onjuist gedimensioneerd
STRING_LENGTH	levert de lengte van een aangeboden string
	Parameters
	STRING de gegeven string
	Terugkeerwaarde
	De lengte van de string
UPGRADE_MATFILE	verjongt een verouderde MATFILE
	Parameters
	FNAME de naam van de file
	<BUFFER> een integer array van ten minste 256 elementen
	Terugkeerwaarde
	0 Ok
	-1 ongeldige buffer
	-2 file bestaat niet
	-3 open error
	-4 read error
	-5 fout bij aanmaken tijdelijke bestand
	-6 fout bij renamen tijdelijk bestand
	-7 write error
	-8 ongeldige file
WAIT_KEY	wacht tot de gebruiker een toets aanslaat
	Terugkeerwaarde
	Code van aangeslagen toets
WRITE_MATFILE	schrijft informatie naar een MATFILE
	Parameters
	<HANDLE> de MATFILE handle
	POSITIE offset in current record
	WAARDE de toe te kennen waarde
	TYPE type van die waarde
	1: integer
	2: real
	4: double
	Terugkeerwaarde
	0 altijd; conversie fouten genereren een PROSIM runtime error

## 4.7 Storestreams

Passt.S..s	passeertijden van schepen bij sluis <i>s</i> in minuten
Wachtt.S.s	wachttijden van schepen bij sluis <i>s</i> in minuten
Overlt.S.s	overligttijden van schepen bij sluis <i>s</i> in minuten
Schutt.S.s	schutttijden van schepen bij sluis <i>s</i> in minuten
Schutten.s	type schuttingen van sluis <i>s</i> type = kolknummer + 0.5 voor een volschutting type = -kolknummer voor een leegom schutting type = kolknummer in alle andere gevallen
z.Debietsk	waterverplaatsing in 1000m <sup>3</sup> per dag naar kant <i>z</i> door kolk <i>k</i> van sluis <i>s</i>
z.Debiet.s	waterverplaatsing in 1000m <sup>3</sup> per dag naar kant <i>z</i> door sluis <i>s</i>
Passt.B..b	passeertijden van schepen bij brug <i>b</i> in minuten
Wachtt.B.b	wachttijden van schepen bij brug <i>b</i> in minuten
Overlt.B.b	overligttijden van schepen bij brug <i>b</i> in minuten
Doorvt.B.b	doorvaartijden van schepen bij brug <i>b</i> in minuten
Roodlt.B.b	roodlichttijden bij brug <i>b</i> in minuten
#Sch\o.B.b	aantal schepen dat de geopende doorgang(en) van brug <i>b</i> is gepasseerd per brugopening
#Open\u.Bb	aantal brugopeningen per uur van brug <i>b</i>
Passt.V..v	passeertijden van schepen bij vernauwing <i>v</i> in minuten
Wachtt.V.v	wachttijden van schepen bij vernauwing <i>v</i> in minuten
Overlt.V.v	overligttijden van schepen bij vernauwing <i>v</i> in minuten
Doorvt.V.v	doorvaartijden van schepen bij vernauwing <i>v</i> in minuten
ZEngBenutv	periode dat de vernauwing <i>v</i> bezet is met schepen van kant <i>z</i>



## 5. Layout databestanden

Voor zover van toepassing wordt in de formele structuur definities gebruik gemaakt van de volgende typografische elementen:

{ en } omsluiten een blok dat niet aanwezig hoeft te zijn of 1 of meer keer herhaald mag worden

*cursieve* tekst is verklarend en komt als zodanig niet voor in de bestanden (tenzij daar opgenomen als commentaar, dus omsloten met een commentaarsymbool @)

NIET-PROPORTIONELE tekst staat op de plaats van de in te vullen informatie.

MATRIX files zijn binair gecodeerde bestanden die niet met een gewone editor bekeken of gewijzigd kunnen worden. Ze zijn record georiënteerd en alleen de layout van de gebruikte records zal worden beschreven, waarbij de gegeven offset in bytes zijn..

## 5.1 BIBGETIJ

De formele structuur van dit bestand is als volgt

```
{tabelnummer T
  springtij curve
  {tijdstip HH.hh waarde A
  }
  -1
  doodtij curve
  {tijdstip HH.hh waarde A
  }
  -1
}
```

Hierin staat hh voor decimale uren (dus geen minuten):

## 5.2 **BIBINTW**

De formele structuur van dit bestand is:

```
{patroonnummer P
  metingen
  {tijdstip H intensiteit I
  }
  -1
-1
```

Hierin is het tijdstip H uitgedrukt in uren vanaf Maandag 0.00 uur en is maximaal 168 (Zondag 24.00 uur).



### 5.3 **BIBKLAS**

De formele specificatie van het relevante deel van dit bestand is:

```
{categorienummer C
  {scheepsklasse S
  }
  -1
}
-1
```

De resterende informatie in dit bestand is niet van belang voor model SUBMOD2 en wordt daarom niet beschreven.

## 5.4 BIBKOSTEN

De formele structuur van dit bestand is:

```
verhouding vaarkosten:ligkosten
  sluis
  vaarkosten  V  ligkosten  L
  brug
  vaarkosten  V  ligkosten  L
  vernauwing
  vaarkosten  V  ligkosten  L
{klasse S
  motorvrachtschepen
  percentage P  vaarkosten  V  ligkosten  L
  motortankschepen
  percentage P  vaarkosten  V  ligkosten  L
}
```

Het blok met gegevens over de scheepsklassen moet precies 30 keer aanwezig zijn.

## 5.5 BIBRREGIMES

De formele structuur van dit bestand is:

```
{regime_nummer R
  bloktijden
  {dagnummer D
    periode op de dag
    {begin UU.MM eind UU.MM
      blok in minuten na elk hele uur in periode
      {begin MM eind MM
        }
        -1
      }
      -1
    }
    -1
  duurbeperkingen
  {dagnummer D
    {begin UU.MM eind UU.MM grens MM
      }
      -1
    }
    -1
  }
  -1
```

Bloktijden en duurbeperving mogen elkaar niet overlappen.

## 5.6 BIBSCHUTREGIMES

De formele structuur van dit bestand is

```
{regime-nummer R
  minimale kolkbenuttinsgraad
    open zijde  O
    gesloten zijde  G
    vergelijkingtype [LINEAIR / ELLIPS / BLOK]
  maximale te veroorzaken wachttijd (min)  W
  wachttijdgewichtsfactor v/d recreatieklasse (cat III)  F
  voorkennistijd beroepsvaart (cat I & cat II)
    kant 1  V1
    kant 2  V2
  periodiek schutten
    {dagnummer D
      {tijdstip HH.MM}
      -1
    }
  -1
}
```



## 5.7 BIBVAART

De formele structuur van dit bestand is:

```

volgafstand vernauwingen
  achterste schip categorie 1
    volgafstand categorie 1 C1 categorie 2 C2 categorie 3 C3
  achterste schip categorie 2
    volgafstand categorie 1 C1 categorie 2 C2 categorie 3 C3
  achterste schip categorie 3
    volgafstand categorie 1 C1 categorie 2 C2 categorie 3 C3
vaartijden sluisen
{scheepsklasse S
  invaartijd eerste schip ongeladen
    {tabelpositie X tabelwaarde Y}
    -1
  invaartijd eerste schip geladen
    {tabelpositie X tabelwaarde Y}
    -1
  invaartijd volgende schip ongeladen
    {tabelpositie X tabelwaarde Y}
    -1
  invaartijd volgende schip geladen
    {tabelpositie X tabelwaarde Y}
    -1
  uitvaartijd eerste schip ongeladen
    {tabelpositie X tabelwaarde Y}
    -1
  uitvaartijd eerste schip geladen
    {tabelpositie X tabelwaarde Y}
    -1
  uitvaartijd volgende schip ongeladen
    {tabelpositie X tabelwaarde Y}
    -1
  uitvaartijd volgende schip geladen
    {tabelpositie X tabelwaarde Y}
    -1
}
-1
lustijd correctie kromme ongeladen
{tabelpositie X tabelwaarde Y}
-1
lustijd correctie kromme geladen
{tabelpositie X tabelwaarde Y}
-1

```

vervolg ...

... vervolg

```
vaartijden bruggen
{scheepsklasse S
  invaartijd eerste schip ongeladen
    {tabelpositie X tabelwaarde Y}
    -1
  invaartijd eerste schip geladen
    {tabelpositie X tabelwaarde Y}
    -1
  doorvaartijd volgende schip ongeladen
    {tabelpositie X tabelwaarde Y}
    -1
  doorvaartijd volgende schip geladen
    {tabelpositie X tabelwaarde Y}
    -1
}
-1
lustijd correctie kromme ongeladen
{tabelpositie X tabelwaarde Y}
-1
lustijd correctie kromme geladen
{tabelpositie X tabelwaarde Y}
-1
vaartijden vaste brug ongeladen
{tabelpositie X tabelwaarde Y}
-1
vaartijden vaste brug geladen
{tabelpositie X tabelwaarde Y}
-1
```

## 5.8 KOPCLUST

De formele structuur van dit bestand is:

```
hoofd van het rapport
{regel TEXT}
END
aantal regels per pagina N
definitie clusters
{clusternummer C
    {scheepsklasse S}
    -1
}
-1
```

## 5.9 NETWERK

De formele structuur van dit bestand is:

```

simulatieduur S
scheepsklassen met vooraanmelding ivm garantie
    bij sluizen
        {scheepsklasse S}
        -1
    bij vernauwingen
        {scheepsklasse S}
        -1
definitie vaarwegen
    {knoop 1 K1 knoop 2 K2 afstand A}
    -1
definitie sluizen
    {knoop 1 K1
     knoop 2 K2
     peilinformatie (tabelnr+translatie of 0+vast peil)
         kant 1 T P
         kant 2 T P
     strategie bij kolkpreferentie S
     schutmethode [GEMENGD / GESCEIDEN]
     schutwaterbeperking B
     wachttijd [JA / NEE]
     definitie kolken
         {kolknummer K
          breedte kolkhoofd H
          fuiklengte kant 1 F1
          fuiklengte kant 2 F2
          breedte B
          lengte L
          waterlengte W
          drempeldiepte kant 1 D1
          drempeldiepte kant2 D2
          duur deur openen kant 1 O1
          duur deur openen kant 2 O2
          nullast nivelleren 1 → 2 N1
          nullast nivelleren 2 → 1 N2
          factor nivelleren 1 → 2 V1
          factor nivelleren 2 → 1 V2
          duur deur sluiten kant 1 S1
          duur deur sluiten kant 2 S2
          veiligheidsmarges mbt gemengd schutten
              cat 2 op 1: tussenruimte T extra intijd I extra uittijd U
              cat 3 op 1: tussenruimte T extra intijd I extra uittijd U
              cat 3 op 2: tussenruimte T extra intijd I extra uittijd U
          geschiktheid scheepsklassen
              {scheepsklasse S}
              -1
          voorkeursklassen ivm garantie
              {scheepsklasse S}
              -1
          bedrijfstijden
              {van HH.MM tot HH.MM}
              -1
          }
    }
    -1

```

vervolg ...



... vervolg

```

voorwaarde bepalingen
  [AFHANKELIJK
    kolnummer K1
    kolnummer K2
    -1
    parameters
      {scheepsklasse S}
      0
  /HEVELEND
    kolnummer K1
    kolnummer K2
    -1
    parameters
      hevelfactor F
      debietreductie kolk 1 R1
      debietreductie kolk 2 R2
  /HEVELEN
    kolnummer K1
    kolnummer K2
    -1
    parameters
      hevelfactor F
      debietreductie kolk 1 R1
      debietreductie kolk 2 R2
  ]
}
-1
definitie bruggen
{knoop 1 K1
knoop 2 K2
type [DRAAI / ENKEL / DUBBEL]
peilinformatie (tabelnr+translatie of 0+vast peil) T P
doorgang kant 1: breedte B hoogte H
doorgang kant 2: breedte B hoogte H
fuiklengte kant1 F1 kant 2 F2
duur openen MM
duur sluiten MM
veiligheidsmarges cat 2 op 1 V1 cat 3 op 1 V2 cat 3 op 2 V3
wegveerkeer
  categorie 1
    kosten K
    afrijcapaciteit A
    richting 1
      tabelnr intensiteit I
      weektotaal W
      aantal stroken N
  categorie 2
    kosten K
    afrijcapaciteit A
    richting 1
      tabelnr intensiteit I
      weektotaal W
      aantal stroken N
  bedrijfstijden
    {van HH.MM tot HH.MM}
    -1
  bedieningsregime R
}
-1

```

vervolg ...

... vervolg

```
definitie vernauwingen
{knoop 1 K1
knoop 2 K2
verkeersregeling [VL / TAB]
vaargegevens voor precies alle 30 klassen
    {snelheid ongeladen V1
    snelheid geladen V2
    plustijd P
    }
verkeerslichten
max groentijd kant 1 G1
max groentijd kant 2 G2
ontruimingstijd O
invaarvolgorde [FIFO / SE]
ontmoetingstabel
    {klasse K1
    versus
        {klasse K2}
        -1
    }
    -1
voorrangsrichting R
}
-1
definitie generatoren
{trjactnaam T
(heen)route
    {knoop K}
    -1
}
END
```

### **5.10 ONDERZOEK**

Dit is een bestand dat niet wordt gebruikt. Er valt dus ook niets over de structuur te zeggen.

## 5.11 QUERY\_INVOER

Dit bestand bevat de noodzakelijke gegevens voor model QUERY. De structuur van dit bestand mag dus niet worden veranderd, omdat dan het model QUERY niet meer werkt. De hier beschreven layout is dan ook alleen bedoeld ter informatie.

```

tijdstip van aanmaak
jaar JJJJ maand MM dag DD uur UU min MM sec SS
definitie kunstwerken
sluizen
{knoop 1 K1 knoop 2 K2
  kolken
    {kolnummer KN kolklengte KL kolkbreedte KB
    }
  -1
}
-1
bruggen
{knoop 1 K1 knoop 2 K2 type T
}
-1
vernuwningen
{knoop 1 K1 knoop 2 K2 regeling "[VL/TAB] [STAR] [AFH] [FIFO/SE]"
}
-1
filename registrator FILENAME.MA0
info betreffende generatoren
{routefile naam ROUTE.SHP
timestamp JJJJ MM DD HH MM SS
historyfile naam FILENAME.Mai
}
END
geaggregeerde info over kunstwerken
sluizen
{#schepen richting 1 N1 #schepen richting 2 N2
totale passeertijd P totale kosten K
}
bruggen
{#schepen richting 1 N1 #schepen richting 2 N2
totale passeertijd P totale kosten K
#brugopeningen O
totaal aantal gepasseerde schepen doorgang 1 D1 doorgang 2 D2
totale kosten wegverkeer W
}
vernuwningen
{#schepen richting 1 N1 #schepen richting 2 N2
totale passeertijd P totale kosten K
}
simulatieduur NOW

```



## 5.12 RAPPORT

De structuur van dit bestand is als volgt te beschrijven, waarbij NIET-CURSIEVE tekst letterlijk voorkomt en op de plaats van CURSIEVE TEKST variabele informatie wordt ingevuld.

RIJKSWATERSTAAT SIVAK VERSIE	DATUM TIJD																																																																																										
TEKST ALS GEDEFINIEERD IN KOPCLUST																																																																																											
ALGEMENE GEGEVENS																																																																																											
TIJDEN IN MINUTEN OPPERVLAKTES IN M <sup>2</sup> KOSTEN IN GULDENS																																																																																											
<u>DEFINITIE CLUSTERS:</u> { Ci : { Sj } }																																																																																											
<u>OPBOUW VAN HET NETWERK:</u> { Ki-Kj TYPE }																																																																																											
<u>SCHEEPVAART-ROUTES (IN KNOOPPUNTEN):</u> { ROUTE: { Ki } }																																																																																											
{																																																																																											
Ki SLUIS Kj OPERATIONELE GEGEVENS																																																																																											
<u>OVERZICHT PER KOLK</u> { <u>KOLK i</u> TC = XXXXX.YY + XXXXX.YY * G1/1000 + XXXXX.YY * G2/1000 TC = XXXXX.YY + XXXXX.YY * N1 + XXXXX.YY * N2  <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">RICHTING</th> <th style="text-align: center;">Ki</th> <th style="text-align: center;">Kj</th> <th style="text-align: center;">Kj</th> <th style="text-align: center;">Ki</th> <th style="text-align: left;">SOM</th> </tr> </thead> <tbody> <tr> <td>GEMIDDELD GMAX</td> <td>=</td> <td>XXXXX.YY</td> <td></td> <td>XXXXX.YY</td> <td></td> </tr> <tr> <td>GEMIDDELD NMAX</td> <td>=</td> <td>XXXXX.YY</td> <td></td> <td>XXXXX.YY</td> <td></td> </tr> <tr> <td>AANTAL SCHUT</td> <td>=</td> <td>XXXXXXXX</td> <td></td> <td>XXXXXXXX</td> <td>XXXXXXXX</td> </tr> <tr> <td>WAARVAN VOL</td> <td>=</td> <td>XXXXXXXX</td> <td></td> <td>XXXXXXXX</td> <td>XXXXXXXX</td> </tr> <tr> <td>WAARVAN LEEGOM</td> <td>=</td> <td>XXXXXXXX</td> <td></td> <td>XXXXXXXX</td> <td>XXXXXXXX</td> </tr> <tr> <td>TOTAAL GESCHUT</td> <td>=</td> <td>XXXXXXXX</td> <td></td> <td>XXXXXXXX</td> <td>XXXXXXXX</td> </tr> <tr> <td>DEBIET GEM. MA</td> <td>=</td> <td>XXXX.YY</td> <td></td> <td>XXXX.YY</td> <td>IN 1000 M3 PER DAG</td> </tr> <tr> <td style="padding-left: 20px;">" DI</td> <td>=</td> <td>XXXX.YY</td> <td></td> <td>XXXX.YY</td> <td></td> </tr> <tr> <td style="padding-left: 20px;">" WO</td> <td>=</td> <td>XXXX.YY</td> <td></td> <td>XXXX.YY</td> <td></td> </tr> <tr> <td style="padding-left: 20px;">" DO</td> <td>=</td> <td>XXXX.YY</td> <td></td> <td>XXXX.YY</td> <td></td> </tr> <tr> <td style="padding-left: 20px;">" VR</td> <td>=</td> <td>XXXX.YY</td> <td></td> <td>XXXX.YY</td> <td></td> </tr> <tr> <td style="padding-left: 20px;">" ZA</td> <td>=</td> <td>XXXX.YY</td> <td></td> <td>XXXX.YY</td> <td></td> </tr> <tr> <td style="padding-left: 20px;">" ZO</td> <td>=</td> <td>XXXX.YY</td> <td></td> <td>XXXX.YY</td> <td></td> </tr> <tr> <td>TOTAAL</td> <td>=</td> <td>XXXX.YY</td> <td></td> <td>XXXX.YY</td> <td>IN 1000 M3 PER 7 DAGEN</td> </tr> </tbody> </table> }		RICHTING	Ki	Kj	Kj	Ki	SOM	GEMIDDELD GMAX	=	XXXXX.YY		XXXXX.YY		GEMIDDELD NMAX	=	XXXXX.YY		XXXXX.YY		AANTAL SCHUT	=	XXXXXXXX		XXXXXXXX	XXXXXXXX	WAARVAN VOL	=	XXXXXXXX		XXXXXXXX	XXXXXXXX	WAARVAN LEEGOM	=	XXXXXXXX		XXXXXXXX	XXXXXXXX	TOTAAL GESCHUT	=	XXXXXXXX		XXXXXXXX	XXXXXXXX	DEBIET GEM. MA	=	XXXX.YY		XXXX.YY	IN 1000 M3 PER DAG	" DI	=	XXXX.YY		XXXX.YY		" WO	=	XXXX.YY		XXXX.YY		" DO	=	XXXX.YY		XXXX.YY		" VR	=	XXXX.YY		XXXX.YY		" ZA	=	XXXX.YY		XXXX.YY		" ZO	=	XXXX.YY		XXXX.YY		TOTAAL	=	XXXX.YY		XXXX.YY	IN 1000 M3 PER 7 DAGEN
RICHTING	Ki	Kj	Kj	Ki	SOM																																																																																						
GEMIDDELD GMAX	=	XXXXX.YY		XXXXX.YY																																																																																							
GEMIDDELD NMAX	=	XXXXX.YY		XXXXX.YY																																																																																							
AANTAL SCHUT	=	XXXXXXXX		XXXXXXXX	XXXXXXXX																																																																																						
WAARVAN VOL	=	XXXXXXXX		XXXXXXXX	XXXXXXXX																																																																																						
WAARVAN LEEGOM	=	XXXXXXXX		XXXXXXXX	XXXXXXXX																																																																																						
TOTAAL GESCHUT	=	XXXXXXXX		XXXXXXXX	XXXXXXXX																																																																																						
DEBIET GEM. MA	=	XXXX.YY		XXXX.YY	IN 1000 M3 PER DAG																																																																																						
" DI	=	XXXX.YY		XXXX.YY																																																																																							
" WO	=	XXXX.YY		XXXX.YY																																																																																							
" DO	=	XXXX.YY		XXXX.YY																																																																																							
" VR	=	XXXX.YY		XXXX.YY																																																																																							
" ZA	=	XXXX.YY		XXXX.YY																																																																																							
" ZO	=	XXXX.YY		XXXX.YY																																																																																							
TOTAAL	=	XXXX.YY		XXXX.YY	IN 1000 M3 PER 7 DAGEN																																																																																						

vervolg ...

... vervolg

Ki SLUIS Kj OPERATIONELE GEGEVENS

CAPACITEIT SLUIS:

RICHTING	TON/UUR	AANTAL/UUR
Ki-Kj	XXXXX.YY	XXXXX.YY
Kj-Ki	XXXXX.YY	XXXXX.YY
TOTAAL	XXXXX.YY	XXXXX.YY

DEBIET IN 1000 M3 PER 7 DAGEN:

Ki-Kj	:	XXXXXXXX
Kj-Ki	:	XXXXXXXX

GEMIDDELDE PASSEERTIJD:

Ki-Kj	:	XXXXX.YY
Kj-Ki	:	XXXXX.YY
BEIDE RI:		XXXXX.YY

GEMIDDELDE PASSEERKOSTEN:

Ki-Kj	:	XXXXXXXX.YY
Kj-Ki	:	XXXXXXXX.YY
BEIDE RI:		XXXXXXXX.YY

TOTALE PASSEERKOSTEN OVER DE SIMULATIETIJD  
(IN DUIZENDEN GULDENS): XXXXXXX.YYY

Ki SLUIS Kj SCHEEPVAART-STATISTIEKEN

CLU- KLAS	KOLK 1				...
	TP	KP	N	NO	
{Si	XXXX	XXXXXX	XXXX	XX	...}
...					...

CLU- KLAS	TOTAAL				...
	TP	KP	N	NO	
{Si	XXXXX	XXXXXXX	XXXXX	XXXX	...}
...					...

vervolg ...

... vervolg

{

<i>Ki</i> BRUG <i>Kj</i> OPERATIONELE GEGEVENS
--

TOTAAL AANTAL BRUGOPENINGEN : XXXXX

	MIN	GEM	MAX	AFW
BEDIENINGSTIJD (ROODLICHTTIJD)	: XX.YY	XX.YY	XX.YY	XX.YY
AANTAL SCHEPEN PER BEWEEGBARE BRUGOPENING	: XXXX.Y	XXXX.Y	XXXX.Y	XXXX.Y
AANTAL BRUGOPENINGEN PER UUR	: XXXX.Y	XXXX.Y	XXXX.Y	XXXX.Y

	<i>Ki-Kj</i>	<i>Kj-Ki</i>	SOM
AANTAL SCHEPEN DOOR DOORGANG 1	: XXXXXXXXXXXX	XXXXXXXXXX	XXXXXXXX
AANTAL SCHEPEN DOOR DOORGANG 2	: XXXXXXXXXXXX	XXXXXXXXXX	XXXXXXXX

WEGVERKEER

	WEGKANT 1	WEGKANT 2	SOM
TOTAANTAL BEINVL. EENH. CAT.1	: XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXXXX
TOTAANTAL BEINVL. EENH. CAT.2	: XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXXXX

	MIN	GEM	MAX	AFW
AANTAL BEINVL. EENH. PER BR.OPENING CAT.1,WEGK.1:	XX.Y	XXXX.Y	XXXX.Y	XXXX.Y
AANTAL BEINVL. EENH. PER BR.OPENING CAT.1,WEGK.2:	XX.Y	XXXX.Y	XXXX.Y	XXXX.Y
AANTAL BEINVL. EENH. PER BR.OPENING CAT.2,WEGK.1:	XX.Y	XXXX.Y	XXXX.Y	XXXX.Y
AANTAL BEINVL. EENH. PER BR.OPENING CAT.2,WEGK.2:	XX.Y	XXXX.Y	XXXX.Y	XXXX.Y
LANGSTE FILE-OPLOSTIJD PER BRUGOPENING	: XX.Y	XXXX.Y	XXXX.Y	XXXX.Y
GESOMMEERDE WACHTTIJD PER OPENING CAT.1 (UREN)	: XX.Y	XXXX.Y	XXXX.Y	XXXX.Y
GESOMMEERDE WACHTTIJD PER OPENING CAT.2 (UREN)	: XX.Y	XXXX.Y	XXXX.Y	XXXX.Y

SCHEEPVAART, GEMIDDELDE PASSEERTIJD

*Ki-Kj* : XXXXX.YY  
*Kj-Ki* : XXXXX.YY  
BEIDE RI: XXXXX.YY

SCHEEPVAART, GEMIDDELDE PASSEERKOSTEN

*Ki-Kj* : XXXXX.YY  
*Kj-Ki* : XXXXX.YY  
BEIDE RI: XXXXX.YY

TOTALE KOSTEN (IN DUIZENDEN GULDENS)

WEGVERKEER CAT.1: XXXXXXXX.YYY  
WEGVERKEER CAT.2: XXXXXXXX.YYY  
SCHEEPVAART : XXXXXXXX.YYY  
SOM: XXXXXXXX.YYY

vervolg ...

... vervolg

$K_i$	BRUG	$K_j$	SCHEEPVAART-STATISTIEKEN
-------	------	-------	--------------------------

CLU- KLAS	DOORGANG 1				DOORGANG 2			
	$\overline{TP}$	$\overline{KP}$	N	NO	$\overline{TP}$	$\overline{KP}$	N	NO
{ Si	XXXX	XXXXXX	XXXX	XXX	XXXX	XXXXXX	XXXX	XXX
...								

CLU- KLAS	TOTAAL			
	$\overline{TP}$	$\overline{KP}$	N	NO
{ Si	XXXXX	XXXXXXX	XXXXX	XXXX
...				
}				

vervolg ...



... vervolg

{

$K_i$	ENGTE	$K_j$	OPERATIONELE GEGEVENS
-------	-------	-------	-----------------------

TYPE VERKEERSAFHANDELING: *REGELING*

ENGTE BENUTTING	$K_i-K_j$	$K_j-K_i$	SOM
GEMIDDELDE TIJD:	XXXXX.YY	XXXXX.YY	XXXXX.YY
AANTAL KEREN	: XXXXX	XXXXX	XXXXX

GEMIDDELDE PASSEERTIJD:

$K_i-K_j$	:	XXXXX.YY
$K_j-K_i$	:	XXXXX.YY
BEIDE RI:		XXXXX.YY

GEMIDDELDE PASSEERKOSTEN:

$K_i-K_j$	:	XXXXX.YY
$K_j-K_i$	:	XXXXX.YY
BEIDE RI:		XXXXX.YY

TOTALE PASSEERKOSTEN OVER DE SIMULATIETIJD  
(IN DUIZENDEN GULDENS): XXXXXXX.YYY

$K_i$	ENGTE	$K_j$	SCHEEPVAART-STATISTIEKEN
-------	-------	-------	--------------------------

CLU- KLAS	VAN KANT $K_i$				VAN KANT $K_j$			
	$\overline{TP}$	$\overline{KP}$	N	NO	$\overline{TP}$	$\overline{KP}$	N	NO
{ Si	XXXX	XXXXXX	XXXX	XXX	XXXX	XXXXXX	XXXX	XXX }

...

CLU- KLAS	TOTAAL			
	$\overline{TP}$	$\overline{KP}$	N	NO
{ Si	XXXXXX	XXXXXXX	XXXXXX	XXXX }

...

}

[EINDE RAPPORT]

### 5.13 Schepen MATRIX file

Deze file bevat de scheepsgegevens zoals die zijn gegenereerd door SUBMOD1.

Record 1 is een header met de volgende layout:

<i>Offset</i>	<i>type</i>	<i>inhoud</i>
0	integer	jaar
2	integer	maand
4	integer	dag
6	integer	uur
8	integer	minuten
10	integer	seconden
12	integer	eerste schiprecord
14	char(8)	BIBKLS filename
22	char(8)	BIBINTS filename

Record 2 legt de deelnemende scheepsklassen vast.

<i>Offset</i>	<i>type</i>	<i>inhoud</i>
0	char(30)	string met een √ in de posities van deelnemende klassen.

Record 3 bevat de brondata.

<i>Offset</i>	<i>type</i>	<i>inhoud</i>
0	real	seed van multi-normale verdeling
4	real	seed van aankomsten verdeling
8	real	seed van multi-geladen/ongeladen verdeling
12	real	seed van multi-kegelschepen verdeling
16	real	seed van multi-snelheid verdeling
20	real	verdichtingsfactor heen
24	real	verdichtingsfactor terug
28	integer	aantal simulatie weken

Record 4 t/m n+3 (als n het aantal deelnemende scheepsklassen is)

<i>Offset</i>	<i>type</i>	<i>inhoud</i>
0	integer	klassenummer
Informatie betreffende de heenroute:		
2	real	% geladen schepen of -1 indien niet aanwezig
6	real	beladingsgraad
10	real	% kegelschepen
14	real	weektotaal
18	integer	aantal
Informatie betreffende de terugroute:		
20	real	% geladen schepen of -1 indien niet aanwezig
24	real	beladingsgraad
28	real	% kegelschepen
32	real	weektotaal
36	integer	aantal

Alle volgende records zijn schiprecords met de volgende layout:

<i>Offset</i>	<i>type</i>	<i>inhoud</i>
0	real	eta ( >0 = heen, <0 = terug)
4	real	klasse ( >0 = ongeladen, <0 = geladen)
8	real	laadvermogen
12	real	breedte
16	real	lengte
20	real	diepgang
24	real	strijkhogte
28	real	snelheid
32	real	kielspeling
36	real	schrikhogte
40	real	breedte incl planspeling ( >0 = geen, <0 = wel kegelschip)
44	real	lengte incl planspeling

## 5.14 Schuttingen MATRIX file

Deze file bevat informatie betreffende het gebruik van de kolken.

Record 1 is een timestamp

<i>Offset</i>	<i>type</i>	<i>inhoud</i>
0	integer	jaar
2	integer	maand
4	integer	dag
6	integer	uur
8	integer	minuten
10	integer	seconden

Record 2 en volgende hebben de volgende layout:

<i>Offset</i>	<i>type</i>	<i>inhoud</i>
0	real	nummer knoop aan kant 1
4	real	kolk nummer
8	real	tijdstip van buiten bedrijfstelling of duur van de schutting
12	real	tijdstip van entry
16	real	oppervlaste rendement van schutting of -1
20	real	idle time
24	real	wachttijd op voorwaarde igv schutting
28	real	debiet igv schutting

## 5.15 Tijdregistratie MATRIX file

Deze file bevat informatie betreffende het leven van schepen.

Record 1 is een timestamp

<i>Offset</i>	<i>type</i>	<i>inhoud</i>
0	integer	jaar
2	integer	maand
4	integer	dag
6	integer	uur
8	integer	minuten
10	integer	seconden

Record 2 en volgende zijn schiprecords. Voor ieder schip is er zo'n record. Een record bestaat uit een aaneenschakeling van subrecords (van variabele lengte) voor iedere mijlpaal. De lengte van een subrecord is afhankelijk van het type mijlpaal:

- Gewone knoop, lengte 4

<i>Offset</i>	<i>type</i>	<i>inhoud</i>
0	real	passagemoment

- Sluis, lengte 20

<i>Offset</i>	<i>type</i>	<i>inhoud</i>
0	real	start overliggen
4	real	start schutten
8	real	kolknummer
12	real	naar opstelruimte
16	real	uit opstelruimte

- Brug, lengte 12

<i>Offset</i>	<i>type</i>	<i>inhoud</i>
0	real	start overliggen
4	real	start doorvaren
8	real	nummer doorgang

- Vernauwing, lengte 8

<i>Offset</i>	<i>type</i>	<i>inhoud</i>
0	real	start overliggen
4	real	start doorvaren

## 6. Bronteksten

### 6.1 AFGELEIDE

Deze MACRO wordt gebruikt als rechterlid in de PRECEPT van het continue attribuut INHOUD van een STROOM. De terugkeerwaarde is de lengteverandering van de file die door de betreffende STROOM wordt beschreven.

```
@=====
@
@ AFGELEIDE
@
@=====
```

```
@=====
@ Afgeleide van de STROOM @
@=====
```

```
y ← VALUE OF intens[tabnr] AT (CT-168.0 * FLOOR (CT/168))
```

```
x ← weektot*y
```

Bereken eerst de aangroei van de file  
De intensiteitstabel bevat entries voor 1 week, dus  
het zoekargument moet modulo 1 week genomen.  
De tabel is genormeerd, dus:

```
y ← oplosvermogen * licht OF mijnbrug="GROEN"
```

```
y ← stroken*y
```

```
x ← x - y
```

```
RETURN x
```

Bepaal de krimp van de file. Als het licht op rood  
staat neemt de lengte niet af  
Oplosvermogen is per rijstrook, dus:  
De verandering in de stroom is dan:



## 6.2 AFHANKELIJK

Deze MACRO wordt aan het attribuut VOORWAARDE van een KOLK toegevoegd als de kolk afhankelijk geschut wordt.

Deze voorwaarde verbiedt een kolk met invaren of uitvaren te beginnen zolang de buurkolk nog bezig is met het doen invaren of uitvaren van schepen aan dezelfde kant. Deze voorwaarde kan gerelaxeerd worden door het opgeven van klassen. Dit houdt dan in dat een kolk met het in- of uitvaren kan beginnen zodra er geen schepen van die klassen bij dat in- en/of uitvaren (kolk en buurkolk) betrokken zijn.

Terugkeerwaarde:

TRUE (=1): voorwaarde vervuld of niet relevant.

FALSE (=0): voorwaarde niet vervuld.

```
@=====
@
@ AFHANKELIJK
@
@=====
```

@ afhakelijk voorwaarden, aangeroepen door THIS KOLK @

LOCAL:

INTEGER : tel

Ga na of er sprake is van een te conditioneren toestand

RETURN TRUE IF (kolk\_wens ≠ invaren) & kolk\_wens ≠ uitvaren

RETURN TRUE IF side OF buurkolk ≠ side

RETURN TRUE IF (kolk\_status OF buurkolk ≠ invaren) &  
kolk\_status OF buurkolk ≠ uitvaren

RETURN TRUE IF (kolk\_wens = uitvaren) & schutruimte IS EMPTY

RETURN TRUE IF (kolk\_wens = invaren) & opstelplan[side] IS EMPTY

Nu is er sprake van een onvervulde voorwaarde als er geen relaxatie naar klassen plaats vindt.

RETURN FALSE IF afh\_klas[1] = -1

Onderzoek nu of de relevante belemmering van toepassing is op de deelverzameling scheepsklassen.

IF kolk\_wens = invaren

FOR EACH schip IN opstelplan[side]

tel ← tel + afh\_klas[klas] = 1

END

RETURN FALSE IF tel > 0

END

IF kolk\_wens = uitvaren

FOR EACH schip IN schutruimte

tel ← tel + afh\_klas[klas] = 1

END

RETURN FALSE IF tel > 0

END

IF kolk\_status OF buurkolk = invaren

FOR EACH schip IN opstelplan[side] OF buurkolk

THAT BELONGS NOT TO schutruimte OF buurkolk

tel ← tel + afh\_klas[klas] = 1

END

RETURN FALSE IF tel > 0

END

IF kolk\_status OF buurkolk = uitvaren

FOR EACH schip IN schutruimte OF buurkolk

tel ← tel + afh\_klas[klas] = 1

END

RETURN FALSE IF tel > 0

END

RETURN TRUE

### 6.3 AFHANKELIJK\_INIT

Deze MACRO verzorgt de initialisatie van het afhankelijk schutten.

Terugkeerwaarde:

0: Ok.

-1: er is iets mis.

```

@=====
@
@ AFHANKELIJK_INIT
@
@=====

@ macro ter initialisatie van het afhankelijk schutten van kolken @
@ wordt aangeroepen in LEES_SLUIZEN en betreft THIS SLUIS      @
LOCAL :
  INTEGER: nr
  REAL    : h_fac

                                Ga na of afhankelijk schutten mag.
GOTO fout0 IF schutplanning IS NOT geen_schutstrategie
ref[1] ← FIRST kolk IN kolken WITH voorwaarde IS afhankelijk
ref[2] ← LAST  kolk IN kolken WITH voorwaarde IS afhankelijk
GOTO fout1 IF ref[1] IS ref[2]
buurkolk OF ref[1] ← ref[2]
buurkolk OF ref[2] ← ref[1]

FOR nr ← 1 TO 30
  afh_klas[nr] ← 0
END
nr ← READ FROM netwerk
IF nr = 0

                                Geen relaxatie mbt scheepsklassen.
  afh_klas[1] ← -1
  RETURN 0
END
WHILE nr ≠ 0

                                Volgende klasse.
  GOTO fout2 IF (nr < 0) | nr > 30
  afh_klas[nr] ← 1
  nr ← READ FROM netwerk
END

RETURN 0

fout0: WRITE "voorwaarden samen met een schutstrategie is te bar!"
        WITH IMAGE ΔΔΔa
RETURN -1
fout1: WRITE "kolk";nrkolk OF ref[1];"is afhankelijk van zichzelf"
        WITH IMAGE ΔΔΔa**Δa
RETURN -1
fout2: WRITE "onmogelijk klassenr";nr;" in afhankelijkheid specificatie"
        WITH IMAGE ΔΔΔa**Δa
RETURN -1

```

## 6.4 BEPAAL\_UITSTEL

Deze MACRO bepaalt het maximale uitstel van het schutten voor de sluis als geheel.

Terugkeerwaarde:

>0: het maximale uitstel in uren

<=0: geen grens te bepalen.

```

@=====
@
@ BEPAAL_UITSTEL
@
@=====

@=====
@ Bepaal het maximale uitstel van schutten voor THIS SLUIS @
@=====
LOCAL:
  INTEGER: k i
  REAL    : x uitstel

                                Beperk de berekening tot de kolk als dat toch niet
                                uit maakt.
RETURN bepaal_uitstel_kolk IF schutplanning IS schutstrategie_gelyk

@-----@
@   Bepaal voor alle nog in te plannen schepen het kritieke tijdstip   @
@ op basis van de 'kleinste' beschikbare kolk waar het schip in 'past' @
@-----@
                                Stel deze sets veilig.
FOR k ← 1 TO 2
  origineel[k] ← remanent[k]
END
n ← 0
uitstel ← 1e6
FOR EACH kolk IN kolken WITH bezig=FALSE
                                Bekijk alleen schepen waarvoor THIS KOLK de
                                kleinste kolk is.
  FOR k ← 1 TO 2
    remanent[k] ← NEW SET
    JOIN EACH schip IN origineel[k] WITH kleinste_kolk IS THIS kolk
      TO remanent[k]
  END
                                Zijn er zulke schepen ?
  GOTO volgende_kolk IF remanent[1] IS EMPTY & remanent[2] IS EMPTY
    & opstelplan[1] IS EMPTY & opstelplan[2] IS EMPTY
  @-----@
  @ plan de kolk aan beide kanten vol met wachtende schepen @
  @-----@
  FOR k ← 1 TO 2
                                Even veilig stellen
    origineel[2+k] ← opstelplan[k]
    origineel[4+k] ← indeling[k]
                                Maak een indeling met de geselecteerde schepen
    opstelplan[k] ← NEW SET
    CALL initdeel(k)
    FOR EACH schip IN origineel[2+k]
      CALL herdeel(k)
    END
    FOR EACH schip IN remanent[k]
      IF passen(k) < 2
        CALL herdeel(k)
        REMOVE THIS schip FROM remanent[k]
      END
    END
  END
END

```

```
END
x ← bepaal_uitstel_kolk
IF x < uitstel
    n ← nrkolk
    uitstel ← x
END
FOR k ← 1 TO 2
    opstelplan[k] ← origineel[2+k]
    indeling[k] ← origineel[4+k]
END
volgende_kolk:
END
FOR k ← 1 TO 2
    remanent[k] ← origineel[k]
END
RETURN -1000 IF n=0
RETURN uitstel
```

Hoelang kan de kolk nu wachten ?

Neem de kleinste waarde

Terugzetten originele sets

Terugzetten originele sets

Er valt geen grens te bepalen

Het maximale uitstel



## 6.5 BEPAAL\_UITSTEL\_KOLK

Deze MACRO bepaalt het maximale uitstel van het schutten voor de kolk.

Terugkeerwaarde:

>0: het maximale uitstel in uren

<=0: geen grens te bepalen.

```
@=====
@
@ BEPAAL_UITSTEL_KOLK
@
@=====
```

```
@=====
@ Bepaal maximale uitstel van de schutting van THIS KOLK @
@=====
```

LOCALS:

INTEGER: k

REAL : x y

DOUBLE : tijd tsluitdeur speling speling2

Veilig stellen van 2 references

ref[1] ← preschip

ref[2] ← postschip

tijd ← NOW

speling ← 1e6

THIS sluis ← zijnsluis

THIS schutregime ← mn\_schutregime OF zijnsluis

```
@-----
@ geplande schepen deze kant @
@-----
```

Bepaal benodigde uitvaartijd.

preschip ← NONE

FOR EACH schip IN schutruimte WITH richting ≠ side

    tijd ← tijd + uittijd(side)

    preschip ← THIS schip

END

Kijk of er een lusschip is.

preschip ← FIRST schip IN opstelplan[side] WITH richting ≠ side

postschip ← NONE

postschip ← THIS schip IF preschip IS NOT NONE @ lustijd

Beschouw alle nog niet ingevaren schepen

FOR EACH schip IN opstelplan[side] THAT BELONGS NOT TO schutruimte

    IF tijd < eta

Schip zal in lopende vaart binnenvaren, het is dus een eerste schip en geen lusschip

        preschip ← NONE

        postschip ← NONE

    END

Zolang hebben we in ieder geval nodig

    tijd ← MAX(tijd, eta)

Nu nog invaren

    tijd ← tijd + intijd(side)

    preschip ← THIS schip

END

Nu kan de deur dicht

tsluitdeur ← tijd

refschip ← NONE

Bepaal de minimale uiterste speling

FOR EACH schip IN opstelplan[side]

    y ← tbijsluis

@ y= aankomsttijdstip van THIS SCHIP

    y ← eta IF tbijsluis=0

    x ← xtra\_wachttijd MINUTES

@ x= uiterste tijdstip van THIS SCHIP



```

x ← xtra_wachttijd MINUTES / wacht_gewicht_cat3 IF categorie[klas]=3
x ← y + x - tsluitdeur @ x= uiterste speling van THIS SCHIP
IF speling > x
  refschip ← THIS schip
  speling ← x
END
END

Minder dan 5 sec wordt als 0 verwerkt
geen_uitstel[1] ← speling < 5 SECONDS

@-----@
@ geplande schepen overkant @
@-----@

Eerst even schutten
x ← waterstand(s_nrtij[side] ,s_peilpar[side])
y ← waterstand(s_nrtij[3-side],s_peilpar[3-side])
Sluiten deur
tijd ← tijd + tsluit[side] MINUTES
Niveleren
tijd ← tijd + (nulnivel[side] + ABS(x-y) * facnivel[side]) MINUTES
Openen aan andere zijde
tijd ← tijd + topen[3-side] MINUTES
Uitvaren
preschip ← NONE
FOR EACH schip IN opstelplan[side]
  tijd ← tijd + uittijd(3-side)
  preschip ← THIS schip
END

Invaren vanaf andere kant
postschip ← preschip
preschip ← FIRST OF opstelplan[3-side]
FOR EACH schip IN opstelplan[3-side]
  IF tijd < eta
    preschip ← NONE
    postschip ← NONE
  END
  tijd ← MAX(tijd,eta)
  tijd ← tijd + intijd(3-side)
  preschip ← THIS schip
END
tsluitdeur ← tijd

Bepaal de minimale uiterste speling
refschip ← NONE
speling2 ← 1e6
FOR EACH schip IN opstelplan[3-side]
  y ← tbijsluis @ y= aankomsttijdstip van THIS SCHIP
  y ← eta IF tbijsluis=0
  x ← xtra_wachttijd MINUTES @ x= uiterste tijdstip van THIS SCHIP
  x ← xtra_wachttijd MINUTES / wacht_gewicht_cat3 IF categorie[klas]=3
  x ← y + x - tsluitdeur @ x= uiterste speling van THIS SCHIP
  IF speling2 > x
    refschip ← THIS schip
    speling2 ← x
  END
END
geen_uitstel[2] ← speling2 < 5 SECONDS
refschip ← NONE IF speling < speling2

Minimale overall speling
speling ← MIN(speling,speling2)

Hebben we het zo gehad ?
GOTO einde IF (speling ≤ 0) | remanent[1] IS EMPTY & remanent[1] IS
EMPTY

```

```

@-----@
@ remanente schepen deze kant @
@-----@
x ← waterstand(s_nrtij[side] ,s_peilpar[side])
y ← waterstand(s_nrtij[3-side],s_peilpar[3-side])
tijd ← tijd + tsluit[3-side] MINUTES
tijd ← tijd + (nulnivel[3-side] + ABS(x-y) * facnivel[3-side]) MINUTES
tijd ← tijd + topen[side] MINUTES
preschip ← NONE
FOR EACH schip IN opstelplan[3-side]
    tijd ← tijd + uittijd(side)
    preschip ← THIS schip
END

postschip ← preschip
preschip ← FIRST schip IN remanent[side] WITH passen(10)=0
FOR EACH schip IN remanent[side] WITH passen(10)=0
    IF tijd < eta
        preschip ← NONE
        postschip ← NONE
    END
    tijd ← MAX(tijd,eta)
    tijd ← tijd + intijd(side)
    preschip ← THIS schip
END
tsluitdeur ← tijd

refschip ← NONE
FOR EACH schip IN remanent[side] WITH passen(10)=0
    y ← tbijsluis @ y= aankomsttijdstip van THIS SCHIP
    y ← eta IF tbijsluis=0
    x ← xtra_wachttijd MINUTES @ x= uiterste tijdstip van THIS SCHIP
    x ← xtra_wachttijd MINUTES / wacht_gewicht_cat3 IF categorie[klas]=3
    x ← y + x - tsluitdeur @ x= uiterste speling van THIS SCHIP
    IF speling > x
        refschip ← THIS schip
        speling ← x
    END
END
GOTO einde IF speling ≤ 0
@-----@
@ remanente schepen overkant @
@-----@
x ← waterstand(s_nrtij[side] ,s_peilpar[side])
y ← waterstand(s_nrtij[3-side],s_peilpar[3-side])
tijd ← tijd + tsluit[side] MINUTES
tijd ← tijd + (nulnivel[side] + ABS(x-y) * facnivel[side]) MINUTES
tijd ← tijd + topen[3-side] MINUTES
preschip ← NONE
FOR EACH schip IN remanent[side] WITH passen(10)=0
    tijd ← tijd + uittijd(3-side)
    preschip ← THIS schip
END

postschip ← preschip
preschip ← FIRST schip IN remanent[3-side] WITH passen(10)=0
FOR EACH schip IN remanent[3-side] WITH passen(10)=0
    IF tijd < eta
        preschip ← NONE
        postschip ← NONE
    END
    tijd ← MAX(tijd,eta)
    tijd ← tijd + intijd(3-side)
    preschip ← THIS schip
END
tsluitdeur ← tijd

```

```

refschip ← NONE
FOR EACH schip IN remanent[3-side] WITH passen(10)=0
  y ← tbijsluis @ y= aankomsttijdstip van THIS SCHIP
  y ← eta IF tbijsluis=0
  x ← xtra_wachttijd MINUTES @ x= uiterste tijdstip van THIS SCHIP
  x ← xtra_wachttijd MINUTES / wacht_gewicht_cat3 IF categorie[klas]=3
  x ← y + x - tsluitdeur @ x= uiterste speling van THIS SCHIP
  IF speling > x
    refschip ← THIS schip
    speling ← x
  END
END
einde:

                                Terugzetten

preschip ← ref[1]
postschip ← ref[2]

                                Rekening houden met voorkennis ?
IF 0 < voorkennistijd[side]
  IF speling ≤ voorkennistijd[side] MINUTES + 1 SECOND
    Ja
    x ← voorkennis(NOW+speling)
    Uitsel heeft geen zin
    RETURN 0 IF x=0
    speling ← 0 IF speling < 1 SECOND
    RETURN speling
  END
  IF speling > voorkennistijd[side] MINUTES
    Alleen echte speling
    speling ← speling-voorkennistijd[side] MINUTES
  END
END
speling ← 0 IF speling < 1 SECOND
RETURN speling

```

## 6.6 BRUGREGIME

Deze MACRO zoekt uit of er een bloktijd of duurbepanking geldt voor een brug.

Terugkeerwaarde:

>=0: Resttijd tot einde bloktijd indien van toepassing

-1: geen bloktijd beperking

Bijproduct:

DEADLINE, zijnde het uiterste tijdstip waarop de brug gesloten moet zijn in verband met de bloktijdenregeling of de brugopeningsduurbepanking.

```
@=====
@
@ BRUGREGIME
@
@=====
```

```
@-----@
@ Geldt er nu een BLOKTIJD voor THIS BRUG ? @
@-----@
```

LOCAL:

REAL: x

```
deadline ← 10E10
THIS brug ← zijnbrug
RETURN 0 IF regimes IS EMPTY
```

Zitten we nu in een bloktijd ?

```
THIS regime ← FIRST regime IN regimes WITH (geldvan ≤ NOW) &
(NOW < geldtot) & soort = 1
RETURN 1 SECOND + geldtot - NOW IF THIS regime IS NOT NONE
```

Nee, bepaal tijd die nodig is om schip te laten passeren

```
n ← 1
x ← NOW + tbopen
IF x < eta @ THIS SCHIP IS IN LOPENDE VAART BIJ OPEN BRUG
    x ← eta
    n ← 4
END
x ← x + tonderdoor(n) + tbsluit
```

Is er een volgende bloktijd ?

```
THIS regime ← FIRST regime IN regimes WITH (NOW ≤ geldvan) & soort=1
deadline ← geldvan IF THIS regime IS NOT NONE
```

Wachten tot einde volgende bloktijd

```
RETURN 1 SECOND + geldtot - NOW IF deadline ≤ x
```

Is er een duurbepanking ?

```
THIS regime ← FIRST regime IN regimes WITH (geldvan ≤ NOW)
& (NOW < geldtot) & soort=2
IF THIS regime IS NOT NONE
```

Ja, bepaal deadline

```
y ← 1 SECOND + NOW + param MINUTES
deadline ← y IF y < deadline
WRITE NOW;NAME OF THIS schip;"kan";NAME OF zijnbrug;
    "niet passeren i.v.m. DUURBEPANKING!" WITH IMAGE ****.**ΔaΔaΔaΔa
    IF y ≤ x
END
RETURN -1
```



## 6.7 BRUG\_REGISTRATIE

Deze MACRO registreert informatie mbt brugopeningen

```

@=====
@
@ BRUG_REGISTRATIE
@
@=====

@=====
@ registratie van de brugopening @
@=====

FOR n ← 1 TO 9
  g[n] ← 0
END
g[1] ← bopen*60
g[2] ← number OF brughulp[1] + number OF brughulp[2]
STORE g[1] AS "Roodlt.B."|nrbrug
STORE g[2] AS "#Sch\o.B "|nrbrug
j ← 3
k ← 7
FOR m ← 1 TO 2      @ 1: snelverkeer (cat 1) 2: langzaamverkeer (cat 2)
  FOR n ← 1 TO 2      @ richting 1 & 2
    THIS stroom ← FIRST stroom IN wegverkeer WITH (cat=m) & (richt=n)
    IF THIS stroom IS NOT NONE
      g[j] ← g[j] + totaal                                @*** g[3,4,5,6] ***
      g[k] ← g[k] + cum_wachttijd                        @*** g[7,8]      ***
    END
    j ← j+1
  END
  k ← k + 1
END
g[9] ← 60*oplos
totopen ← totopen + 1
FOR k ← 1 TO 9
  bmin[k] ← g[k] IF bmin[k] > g[k]
  bmax[k] ← g[k] IF bmax[k] < g[k]
  bgem[k] ← bgem[k] + g[k]
  bafw[k] ← bafw[k] + g[k]_2
END
RETURN

```



## 6.8 BUITEN\_SCHUTPERIODE

Deze MACRO gaat na of THIS KOLK buiten de schutperiode zit en levert een indicatieve terugkeerwaarde af.

Terugkeerwaarde:

WACHTTIJD indien buiten gedefinieerde periode  
 -1 buiten periode terwijl andere kolk(en) nog bezig zijn  
 0 overige gevallen

```

@=====
@
@ BUITEN_SCHUTPERIODE
@
@=====

@=====
@
@   Bekijk of THIS KOLK buiten de schutperiode zit.
@   Wanneer de kolk net aan het einde gekomen is
@wordt deze geactiveerd vanaf het begin van de volgende periode
@   returnwaarde: • WACHTTIJD als periode gedefinieerd is
@                  & voorbij of komende
@                  • -1      als de periode afgelopen zou zijn,
@                  als de andere kolk(en) niet nog
@                  bezig zouden zijn.
@                  • 0      anderszids
@=====
LOCAL:
  INTEGER: dag n
  REAL    : uur wachttijd

  dag ← 1 + FLOOR(NOW /24) MODULO(7)
  uur ← NOW MODULO(24)

  THIS sluis ← zijnsluis
  @-----@
  @ Er werd niet periodiek geschut @
  @-----@

start:
  IF periodiek=FALSE
    @*** geen 'periodiek schutten'-dag ***
    THIS regime ← FIRST regime IN schutperiodes WITH param = dag
    RETURN FALSE IF THIS regime IS NONE

    @*** laatste periode is al geweest ***
    THIS regime ← FIRST regime IN schutperiodes WITH (param = dag)
    & uur ≤ geldvan + 1 MINUTE
    GOTO wacht_tot_morgen IF THIS regime IS NONE

    @***** dit is een nieuwe periode *****
    IF uur ≥ geldvan - 1 MINUTE
      periodiek ← TRUE
      GOTO is_periode_afgelopen
    END
    ENTER vrije_kolken
    RETURN -1 IF LENGTH OF vrije_kolken < LENGTH OF alle_kolken
    GOTO zet_kolk_uit
  END

  @-----@
  @ Er werd wel periodiek geschut @
  @-----@
  THIS regime ← LAST regime IN schutperiodes WITH (param = dag)
  & geldvan < uur + 1 SECOND
  IF THIS regime IS NONE @*** al het begin van de volgende dag ***

```

```

    vrije_kolken ← NEW SET CALLED "vrije kolk"
    periodiek ← FALSE
    REPEAT FROM start
END
is_periode_afgelopen:
    THIS regime ← SUCC OF THIS regime IN schutperiodes

    IF THIS regime IS NOT NONE
        IF param=dag @*** er komt nog een periode ***
            RETURN FALSE IF voldoende_benutting = TRUE
            ENTER vrije_kolken
            RETURN -1 IF LENGTH OF vrije_kolken < LENGTH OF alle_kolken
            GOTO zet_kolk_uit
        END
    END

    @-----@
    @ laatste periode van de dag @
    @-----@
    RETURN FALSE IF opstelplan[1] IS NOT EMPTY |
                    opstelplan[2] IS NOT EMPTY
    ENTER vrije_kolken
    RETURN -1 IF wachtruimte[1] IS NOT EMPTY |
                wachtruimte[2] IS NOT EMPTY
    RETURN -1 IF LENGTH OF vrije_kolken < LENGTH OF alle_kolken
    periodiek ← FALSE

    @-----@
    @ aktiveer THIS KOLK bij het begin van de volgende periode @
    @ op de volgende dag @
    @-----@

wacht_tot_morgen:
    dag ← 1 + dag MODULO(7)
    THIS regime ← FIRST regime IN schutperiodes WITH param = dag
    wachttijd ← 1 SECOND + 24 HOURS - uur
    wachttijd ← 1 DAY + (geldvan - uur) HOURS IF THIS regime IS NOT NONE
    wachttijd ← MAX(0,wachttijd)
    bezig ← TRUE
    n ← nrkolk
    FOR EACH kolk IN vrije_kolken WITH nrkolk#n
        bezig ← TRUE
        CANCEL THIS kolk IF THIS kolk IS ACTIVE
        ACTIVATE THIS kolk WITH DELAY wachttijd FROM beschikbaar IN kolkmod
    END
    vrije_kolken ← NEW SET CALLED "vrije kolk"
    RETURN wachttijd

    @-----@
    @ aktiveer THIS KOLK bij het begin van de volgende periode @
    @-----@

zet_kolk_uit:
    bezig ← TRUE
    wachttijd ← MAX(0,(geldvan - uur) HOURS)
    n ← nrkolk
    FOR EACH kolk IN vrije_kolken WITH nrkolk#n
        bezig ← TRUE
        CANCEL THIS kolk IF THIS kolk IS ACTIVE
        ACTIVATE THIS kolk WITH DELAY wachttijd FROM beschikbaar IN kolkmod
    END
    vrije_kolken ← NEW SET CALLED "vrije kolk"
    RETURN wachttijd

```

## 6.9 CHECK\_NETWORK

Deze MACRO gaat na of een juist ingelezen vaarweg of kunstwerk twee knopen verbindt die reeds verbonden zijn door een vaarweg of kunstwerk.

Parameters:

N type van te testen waterweg  
 1 = vaarweg  
 2 = sluis  
 3 = brug  
 4 = vernauwing

Terugkeerwaarde:

1: Ok  
 -1: duplicaat waterweg

```
@=====
@
@ CHECK_NETWORK
@
@=====
```

```
@=====
@ Ga na of een waterweg reeds is gedefinieerd @
@=====
```

PARAMETER:

INTEGER: n

LOCAL:

INTEGER: k

```
IF n = 1 @ THIS = VAARWEG @
  plaats[3] ← THIS vaarweg
  FOR k ← 1 TO 2
    plaats[k] ← raai[k]
  END
END
```

```
IF n = 2 @ THIS = SLUIS @
  plaats[3] ← THIS sluis
  FOR k ← 1 TO 2
    plaats[k] ← kant[k]
  END
END
```

```
IF n = 3 @ THIS = BRUG @
  plaats[3] ← THIS brug
  FOR k ← 1 TO 2
    plaats[k] ← zijde[k]
  END
END
```

```
IF n = 4 @ THIS = VERNAUWING @
  plaats[3] ← THIS vernauwing
  FOR k ← 1 TO 2
    plaats[k] ← vzijde[k]
  END
END
```

```
@-----@
@ check VAARWEGEN @
@-----@
```

```
plek ← FIRST vaarweg IN vaarwegen WITH raai[1] IS plaats[1]
      & raai[2] IS plaats[2]
GOTO fout IF plek IS NOT NONE
```

```

RETURN 1 IF n=1

@-----@
@ check SLUIZEN @
@-----@

plek ← FIRST sluis IN sluizen WITH kant[1] IS plaats[1]
      & kant[2] IS plaats[2]
GOTO fout IF plek IS NOT NONE
RETURN 1 IF n=2

@-----@
@ check BRUGGEN @
@-----@

plek ← FIRST brug IN bruggen WITH zijde[1] IS plaats[1]
      & zijde[2] IS plaats[2]
GOTO fout IF plek IS NOT NONE
RETURN 1 IF n=3

@-----@
@ check BRUGGEN @
@-----@

plek ← FIRST vernauwing IN vernauwingen WITH vzijde[1] IS plaats[1]
      & vzijde[2] IS plaats[2]
GOTO fout IF plek IS NOT NONE
RETURN 1

fout:
WRITE ". "; "Tussen" WITH IMAGE a20a
FOR k ← 1 TO 2
  WRITE "en" WITH IMAGE |Δa IF k=2
  WRITE "knoop";nrnode OF plaats[k] WITH IMAGE |Δa|***
END
WRITE "bevinden zich twee waterwegen," WITH IMAGE |ΔΔa
WRITE " "; "te weten: "; NAME OF plek; "en"; NAME OF plaats[3]
  WITH IMAGE a20aΔaΔaΔa
RETURN -1

```



## 6.10 DISPLAY\_VERNAUWING

Deze MACRO verzorgt de tekst-animatie van een vernauwing. Dat gebeurt alleen als er maar 1 vernauwing is gedefinieerd en het daarvoor bestemde attribuut TOON\_VERNAUWING een waarde ongelijk 0 heeft gekregen, wat alleen kan door gebruik te maken van STATE ANALYSIS.

```
@=====
@
@ DISPLAY_VERNAUWING
@
@=====

LOCALS:
  INTEGER: k lin pos lin1 lin2 k1 kleur

RETURN IF toon_vernauwing = FALSE
RETURN IF LENGTH OF vernauwingen ≠ 1

CALL clear_text_box(4,4,1,80,sb)
CALL clear_text_box(5,19,1,21,sb)
CALL clear_text_box(5,19,58,80,sb)
CALL clear_text_box(20,20,1,80,sb)

FOR k ← 1 TO 2
  lin ← 4
  pos ← 2 + 62*k-1
  FOR EACH schip IN vwachtrij[k]
    IF lin ≤ 20
      schipnaam ← klas|" :....."
      CALL char_in_string(<schipnaam>,4,5,NAME OF THIS schip,6)
      DISPLAY schipnaam;60*NOW-tw AT LINE lin POSITION pos
      WITH IMAGE aΔ****.* IF wachtend
      DISPLAY schipnaam;60*tw+NOW-tl AT LINE lin POSITION pos
      COLOUR af ON ab WITH IMAGE aΔ****.* IF wachtend=FALSE
      lin ← lin + 1
    END
  END
END

CALL clear_text_box(5,19,22,57,0)
refnode ← vzijde[2]

lin1 ← 19
lin2 ← 5

FOR EACH schip IN engte
  lin ← lin1
  lin ← lin2 IF richting=2

  IF (lin ≤ 19) & lin ≥ 5
    pos ← 23 + 18*richting-1

    schipnaam ← klas|" :....."
    CALL char_in_string(<schipnaam>,4,5,NAME OF THIS schip,6)
```

Veeg wat delen van het scherm schoon

Voor iedere kant

Begin op regel 4

Kant 1 links, kant 2 rechts op scherm

Voor alle schepen in de wachtrijen

Meer dan 17 worden er niet afgebeeld  
Stel identificatiestring samen

Kleurtje afhankelijk van wachtend

Richting 1 vult van onder naar boven, 2 andersom

Alle schepen in de engte

Alleen wat binnen dit schermdeel valt wordt afgebeeld

Stel identificatie samen



## Kleurtje afhankelijk van garantie

```

kleur ← wit
kleur ← 12 IF garantie

```

## Gegevens afhankelijk van invaarvolgorde

```

DISPLAY schipnaam;eta AT LINE lin POSITION pos COLOUR kleur ON 0
    WITH IMAGE aΔ****.** IF snelste_eerst = FALSE
DISPLAY schipnaam;vvaarsnelheid[klas,ge1] AT LINE lin POSITION pos
    COLOUR kleur ON 0 WITH IMAGE aΔ****.** IF snelste_eerst
lin1 ← lin1-1 IF richting = 1
lin2 ← lin2+1 IF richting = 2

```

```

END

```

```

END

```

## Invaartoestand

```

DISPLAY "VRIJ" AT LINE 4 POSITION 23 WITH IMAGE a IF invaren_vrij[1]
DISPLAY "VRIJ" AT LINE 20 POSITION 53 WITH IMAGE a IF invaren_vrij[2]

```

## De afbeelding van de verkeerslichten

```

IF vlicht[1]=groen
    DISPLAY " GROEN " AT LINE 4 POSITION 28 COLOUR 11 ON 0 WITH IMAGE a
    DISPLAY 60*eindgroen-NOW AT LINE 4 POSITION 35 COLOUR 11 ON 0
        WITH IMAGE **.** IF (eindgroen < 9e5)

```

```

END

```

```

DISPLAY " ROOD " AT LINE 4 POSITION 28 COLOUR 12 ON 0 WITH IMAGE a
    IF vlicht[1]=rood
DISPLAY " GARANTIE! " AT LINE 4 POSITION 41 COLOUR 12 ON 0 WITH IMAGE a
    IF garantieschepen IS NOT EMPTY

```

```

IF vlicht[2]=groen
    DISPLAY " GROEN " AT LINE 20 POSITION 45 COLOUR 11 ON 0 WITH IMAGE a
    DISPLAY 60*eindgroen-NOW AT LINE 20 POSITION 40 COLOUR 11 ON 0
        WITH IMAGE **.** IF (eindgroen < 9e5)

```

```

END

```

```

DISPLAY " ROOD " AT LINE 20 POSITION 46 COLOUR 12 ON 0 WITH IMAGE a
    IF vlicht[2]=rood

```

## Wacht eventueel op een toetsaanslag

```

CALL wait_key IF toon_vernauwing=2
RETURN

```

## 6.11 FYSIEK

Dit is een MODULE dat de procesbeschrijvingen bevat van GENERATOR, STROOM, OBSERVATOR en BRUG.

```
@=====
@
@ FYSIEK
@
@=====
```

```
@=====
@ Procesbeschrijving van de GENERATOR @
@=====
```

genese:

Voor elke bladzij

```
WHILE r_schepen IS NOT EMPTY
  schipregel ← 0
```

Voor elke regel

```
WHILE schipregel < regels OF FIRST OF r_schepen
```

Maak volgende schip

```
  CALL gemaakt
  schipregel ← schipregel+1
  x ← eta - NOW
  x ← 0 IF x < 0
```

Activeer dat schip op zijn ETA

```
  ACTIVATE THIS schip WITH DELAY x FROM vaar IN schipmod
  WAIT x
```

END

```
  REMOVE FIRST OF r_schepen FROM r_schepen
```

END

TERMINATE

```
@=====
@ Procesbeschrijving van de STROOM @
@=====
```

stopop:

Laat het verkeer maar razen

```
  INTEGRATE WHILE (0 < inhoud) | licht OF mijnbrug = "ROOD"
  vrijverkeer OF mijnbrug ← vrijverkeer OF mijnbrug + 1
  TERMINATE
```

```
@=====
@ Procesbeschrijving van de OBSERVATOR @
@=====
```

observeer :

Eindigen als dat handmatig is ingesteld via STATE  
ANALYSIS

```
  IF eindigen
    CANCEL MAIN
    REACTIVATE MAIN FROM afsluiten IN MAINMOD
    TERMINATE
```

END

Slaap 1 uur

```
  WAIT 1
```

```
@-----
@ het updaten van de REGIMES @
@-----
```

Verschuif niet actuele regimes een week

```
  FOR EACH regimeblok IN regimeblokken
    FOR EACH regime IN r_inhoud WITH geldtot ≤ NOW - 2 DAYS
```

```

    REMOVE THIS regime FROM r_inhoud
    geldvan ← geldvan + 168
    geldtot ← geldtot + 168
    JOIN THIS regime TO r_inhoud
  END
END

```

#### Datacollectie mbt bruggen

```

FOR EACH brug IN bruggen @*** # brugopeningen/uur ***
  STORE nbopen AS "#Open\u.B "|nrbrug
  bmin[10] ← nbopen IF bmin[10] > nbopen
  bmax[10] ← nbopen IF bmax[10] < nbopen
  bgem[10] ← bgem[10] + nbopen
  bafw[10] ← bafw[10] + nbopen_2
  n_uren ← n_uren + 1
  nbopen ← 0
END

```

```

n ← NOW

```

#### Is er precies een dag om ?

```

REPEAT FROM observeer IF n MODULO(24) ≠ 0

```

#### Ja, bereken debieten

```

n ← FLOOR((n/24) MODULO(7))
n ← 7 IF n=0
FOR EACH sluis IN sluizen WITH debietbepaling
  FOR k ← 1 TO 2
    w[k] ← 0
  END
  deb_aant[n] ← deb_aant[n] + 1
  FOR EACH kolk IN alle_kolken
    FOR k ← 1 TO 2
      tot_debiet[n,k] ← tot_debiet[n,k] + debiet[k]
      STORE debiet[k] VERSUS NOW/24
      AS k|". Debiet ."|nrkolk + 10*nrsluis
      w[k] ← w[k] + debiet[k]
      debiet[k] ← 0
    END
  END
  FOR k ← 1 TO 2
    STORE w[k] VERSUS NOW/24 AS k|". Debiet ."|nrsluis
  END
END
REPEAT FROM observeer

```

```

@=====
@ Procesbeschrijving van de BRUG @
@=====

```

startbrug:

#### Wacht zolang er geen meldingen zijn

```

WAIT WHILE meldingen OF zijde[1] IS EMPTY &
      meldingen OF zijde[2] IS EMPTY

```

```

n ← 1
n ← 2 IF meldingen OF zijde[1] IS EMPTY

```

#### Eerst gemelde schip

```

THIS schip ← FIRST OF meldingen OF zijde[n]
REMOVE THIS schip FROM meldingen OF zijde[n]
instructie ← THIS brug
richting ← n

```

#### Bepaal DOORGANG en ONDERDOOR

```

IF type = "DUBBEL"
  onderdoor ← FALSE

```

```
doorgang ← 3
FOR j ← 1 TO nrdoorgangen
  IF hoogte[j] ≥ strijkh + schrikh + waterstand(b_nrtij,b_peilpar)
    IF wijdte[j] > breedte
      onderdoor ← TRUE
      doorgang ← j IF j < doorgang
    END
  END
END
doorgang ← nrdoorgangen IF doorgang = 3
END
IF type ≠ "DUBBEL"
  doorgang ← richting
  doorgang ← 1 IF type = "ENKEL"
  onderdoor ← hoogte[doorgang] ≥ strijkh + schrikh +
    waterstand(b_nrtij,b_peilpar)
END

                                Schuif door naar brugwachter als de brug open
                                moet
JOIN THIS schip TO brugplan[n] OF zijnwachter IF onderdoor = FALSE
REPEAT FROM startbrug
```



## 6.12 GEEN\_SCHUTSTRATEGIE

Pas de schutplanning voor een sluis aan indien er geen schutstrategie is.

Parameters:

TYPE            de reden voor de vraag naar een schutplanning  
                  VOOR\_AANMELDING dwz er heeft zich een garantieschip gemeld  
                  SCHIP\_BIJ\_MELDRAAI dwz een gewone aanmelding  
                  NIEUW\_OPSTELPLAN dwz er moet een nieuw opstelplan gemaakt  
                  VRIJE\_KOLK dwz er komt een kolk vrij en er moet geverifieerd worden of de kolk door kan  
                  gaan met schutten

Terugkeerwaarde:

-1: in geval van een vrije kolk en leeg opstelplan  
 0: overigens.

```

=====
@
@ GEEN_SCHUTSTRATEGIE
@
=====

@=====
@           Pas de schutplanning aan voor THIS SLUIS           @
@           Er is geen schutstrategie                       @
@=====

PARAMETER:
  INTEGER: type
  INTEGER: sluiskant

LOCALS:
  INTEGER : k m
  REAL    : x y

                                Switch naar gelang het type
GOTO melding_schip IF type = voor_aanmelding
GOTO melding_schip IF type = schip_bij_meldraai
GOTO vul_aan       IF type = nieuw_opstelplan
GOTO vrije_kolk    IF type = vrije_kolk
RETURN

@-----@
@ De CENTRALIST behandelt een melding van een schip @
@-----@

melding_schip:
  RETURN IF instructie IS NOT NONE           @ THIS SCHIP is al ingepast
  THIS sluis ← plek
  THIS node ← doel

@-----@
@ Negeer MELDING als het een vooraanmelding-schip betreft @
@ terwijl geen enkele KOLK in de SLUIS garantie verleent @
@-----@
IF type = voor_aanmelding
  THIS kolk ← FIRST kolk IN kolken WITH garkolk[klas]=1
  RETURN IF THIS kolk IS NONE
  THIS node ← SUCC OF doel IN route
END
sluiskant ← 1 + THIS node IS kant[2]
richting ← sluiskant

@-----@
@ Bepaal de beste KOLK @
@-----@

```



```

@ Bij een vooraanmelding slechts die KOLKEN bekijken, die garantie @
@                               voor deze SCHEEPSKLASSE verlenen. @
@-----@
m ← 3
y ← 1e6
bestkolk ← NONE

FOR EACH kolk IN kolken WITH (garkolk[klas]=1) | type ≠ voor_aanmelding
    k ← passen(sluiskant)
    m ← MIN(m, k)
    IF k = 0
        x ← kolk_preferentie(sluiskant)
        IF x < y
            y ← x
            bestkolk ← THIS kolk
        END
    END
END
THIS kolk ← bestkolk

Loop de kolken af
Bekijk of schip ingepast kan worden
Begerenswaardigste optie heeft laagste waarde
Schip kan mee met deze kolk,
Begerenswaardigste optie heeft laagste waarde

@-----@
@ SCHIP kan geplaatst worden in OPSTELRUIMTE van BESTKOLK @
@-----@
IF m=0
    CALL herdeel(sluiskant)
    instructie ← THIS kolk
    garantie ← speciaal & garkolk[klas] = 1
    GOTO aktiveer_kolk
END
@-----@
@ Het SCHIP kan niet in opstelruimte en @
@ moet een voorkeursbehandeling krijgen @
@-----@
IF type = voor_aanmelding
    JOIN THIS schip TO voorkeur[sluiskant] RANKED BY 4-m
    instructie ← THIS sluis
    RETURN
END
@-----@
@ SCHIP kan niet in OPSTELRUIMTE en moet naar WACHTRUIMTE @
@-----@
IF m=2
    JOIN THIS schip TO remanent[sluiskant] RANKED BY 45 * eta
    instructie ← THIS sluis
END
@-----@
@ Fysieke beperkingen SCHIP voor deze SLUIS @
@-----@
IF m=3
    JOIN THIS schip TO onmogelijk[sluiskant] RANKED BY 45 * eta
    instructie ← NONE
END
RETURN

@-----@
@ De CENTRALIST moet een nieuw OPSTELPLAN maken voor this KOLK @
@-----@
vul_aan:
    THIS sluis ← zijnsluis

```

```

                                Is de kolk operationeel ?
RETURN IF THIS kolk BELONGS NOT TO kolken

@-----@
@ Gegarandeerde SCHEPEN gaan voor @
@-----@
FOR EACH schip IN voorkeur[sluiskant] WITH (garkolk[klas]=1)
    | (SUCC OF cur_plek IN vaarplan IS THIS sluis
    | cur_plek IS THIS sluis)
    IF passen(n) ≤ 1
        Het schip past
        REMOVE THIS schip FROM voorkeur[sluiskant]
        garantie ← garkolk[klas]=1
        Herindelen
        CALL herdeel(sluiskant)
        instructie ← THIS kolk
        Schip wekken indien nodig
        REACTIVATE THIS schip
        IF THIS schip BELONGS TO wachtruimte[sluiskant]
    END
END

                                Beschouw alle schepen die eerder niet konden
                                worden ingepast
FOR EACH schip IN onmogelijk[sluiskant]
    m ← passen(sluiskant)
    IF m=0
        Nu lukt het wel
        REMOVE THIS schip FROM onmogelijk[sluiskant]
        CALL herdeel(sluiskant)
        instructie ← THIS kolk
    END
    IF m=2
        Schip kan naar wachtruimte, maar dat is nog niet
        beslist
        REMOVE THIS schip FROM onmogelijk[sluiskant]
        JOIN THIS schip TO remanent[sluiskant] RANKED BY 45 * eta
        instructie ← THIS sluis
    END
END

                                Onderzoek alle inpasbare schepen
FOR EACH schip IN remanent[sluiskant]
    IF passen(sluiskant) ≤ 1
        Schip kan mee
        REMOVE THIS schip FROM remanent[sluiskant]
        CALL herdeel(sluiskant)
        REACTIVATE THIS schip
        IF THIS schip BELONGS TO wachtruimte[sluiskant]
            instructie ← THIS kolk
        END
    END
END

                                Is er iets te schutten ?
RETURN IF opstelplan[1] IS EMPTY & opstelplan[2] IS EMPTY

aktiveer_kolk:
    IF bezig = FALSE
        bezig ← TRUE
        CANCEL THIS kolk IF THIS kolk IS ACTIVE
        REACTIVATE THIS kolk FROM vaarin IN kolkmod
    END
    RETURN

@-----@
@ De CENTRALIST bekijkt of THIS KOLK door kan gaan met schutten @
@ in verband met een niet leeg opstelplan @
@-----@

```

vrije\_kolk:

Opstelplan leeg ?

RETURN IF opstelplan[1] IS NOT EMPTY | opstelplan[2] IS NOT EMPTY

Ja

RETURN -1

### 6.13 GEMAAKT

Deze MACRO maakt een nieuw SCHIP op grond van een regel in de schepenfile die door SUBMOD1 is gemaakt.

```

@=====
@
@ GEMAAKT
@
@=====

@=====
@ Initialisatie van SCHEPEN @
@=====
LOCAL:
    INTEGER: offset

    Actuele bladzij in bestand

    THIS bladzij ← FIRST OF r_schepen
    Start van de informatie over het volgende schip

    offset      ← s_recl * schipregel

    THIS schip ← NEW schip
    creatietijd ← bladinhoud[offset+1]
    klas        ← bladinhoud[offset+2]
    route       ← route_heen
    vaarplan    ← weg_heen
    mpalen      ← palen_heen
    IF creatietijd < 0
        route      ← route_terug
        vaarplan    ← weg_terug
        mpalen      ← palen_terug
        creatietijd ← ABS(creatietijd)
    END
    eta         ← creatietijd
    laadverm    ← bladinhoud[offset+3]
    breedte     ← bladinhoud[offset+4]
    lengte      ← bladinhoud[offset+5]
    diepgang    ← bladinhoud[offset+6]
    strijkh     ← bladinhoud[offset+7]
    snelheid    ← bladinhoud[offset+8]
    gel         ← 1      @ ongeladen
    IF klas < 0
        gel ← 2      @ geladen
        klas ← ABS(klas)
    END
    kielspeling ← bladinhoud[offset+9]
    schrikh     ← bladinhoud[offset+10]
    planbreedte ← ABS(bladinhoud[offset+11]) @ negatief betekent 'kegelschip'
    planlengte  ← bladinhoud[offset+12]
    gen         ← THIS generator
    start       ← FIRST OF route
    doel        ← SUCC OF start IN route
    cur_plek    ← FIRST OF vaarplan
    vorige_mijlpaal ← FIRST OF mpalen
    speciaal    ← sgarklas[klas] = TRUE
    vspeciaal   ← vgarklas[klas] = TRUE

    @-----@
    @ Tijdregistratie @
    @-----@

    THIS bladzij ← LAST OF bladen
    CALL wegschrijven_bladen(FALSE) IF regels = regels_per_bladzij
    bladstart ← regellengte * regels
    blad      ← THIS bladzij

```

```
regels ← regels + 1  
RETURN
```



## 6.14 HERDEEL

Deze MACRO herzielt de indeling van een kolk ten gevolge van het moeten toevoegen van een nieuw schip aan het opstelplan. Zie de toelichting bij MACRO PASSEN voor een uitleg over het indelen van een kolk.

Parameter:

SLUISKANT De kant van de sluis waar het nieuwe schip aankomt.

```

@=====
@
@ HERDEEL
@
@=====

@=====
@ Herindeling van THIS KOLK door toevoegen van THIS SCHIP ah OPSTELPLAN @
@=====
PARAMETER:
  INTEGER: sluiskant

  k ← categorie[klas]
  IF opstelplan[sluiskant] IS EMPTY
    Er is nog geen opstelplan
    beroeps_ruimte[sluiskant] ← 0 IF k=3
    GOTO simpel
  END
  x ← eta

  Als een van de volgende voorwaarden geldt moet
  het schip ergens in het opstelplan ingevoegd
  worden. Zo niet dan kan het simpelweg achteraan
  refschip ← FIRST schip IN opstelplan[sluiskant] WITH categorie[klas] > k
  GOTO reshuffle IF refschip IS NOT NONE & gemengd_schutten = FALSE
  refschip ← FIRST schip IN opstelplan[sluiskant] WITH categorie[klas] = k
  GOTO reshuffle IF refschip IS NONE & gemengd_schutten = FALSE
  refschip ← FIRST schip IN opstelplan[sluiskant] WITH eta > x
  GOTO reshuffle IF refschip IS NOT NONE

simpel:
  JOIN THIS schip TO opstelplan[sluiskant]
  CALL voegtoe(sluiskant)
  RETURN

reshuffle:
  Maak een hulpset waarin het nieuwe schip samen
  met alle schepen uit het oorspronkelijke opstelplan
  zit
  hulp ← NEW SET CALLED sluiskant|.PLAN.....|nrkol
  JOIN THIS schip TO hulp RANKED BY 45 * eta
  FOR EACH schip IN opstelplan[sluiskant]
    REMOVE THIS schip FROM opstelplan[sluiskant]
    JOIN THIS schip TO hulp RANKED BY 45 * eta
  END
  Opstelplan is nu leeg

  IF gemengd_schutten = FALSE
    Groepeer alle schepen naar categorie als er niet
    gemengd geschut wordt

    FOR j ← 1 TO 3
      JOIN EACH schip IN hulp WITH categorie[klas] = j
      TO opstelplan[sluiskant]
    END
  END
  Laat de volgorde ongemoeid als er toch gemengd
  geschut wordt
  opstelplan[sluiskant] ← hulp IF gemengd_schutten

```

```

CALL initdeel(sluiskant)

Maak de basis indeling

Categorie van het eerste schip (alleen van belang
als er niet gemengd geschut wordt)
catwis ← categorie[klas OF FIRST OF opstelplan[sluiskant]]
beroeps_ruimte[sluiskant] ← 0

Beschouw alle schepen in het opstelplan
FOR EACH schip IN opstelplan[sluiskant]
  IF (catwis ≠ categorie[klas]) & gemengd_schutten = FALSE
    Nu moet er een veiligheidszone komen
    Onthoudt eerste en laatste punt van contour
    plaats[1] ← FIRST OF indeling[sluiskant]
    REMOVE plaats[1] FROM indeling[sluiskant]
    plaats[2] ← LAST OF indeling[sluiskant]
    REMOVE plaats[2] FROM indeling[sluiskant]
    k ← categorie[klas]

    Achterkant van verst naar achter liggende schip
    THIS punt ← FIRST punt IN indeling[sluiskant] WITH GREATEST xx
    x ← xx

    Veiligheidszone
    x ← x + cat12 IF (1=catwis) & 2 = k
    x ← x + cat13 IF (1=catwis) & 3 = k
    x ← x + cat23 IF (2=catwis) & 3 = k
    catwis ← k
    beroeps_ruimte[sluiskant] ← brdtkolk * x IF catwis=3

    Nieuwe indeling voor resterende kolkruimte met
    volledige breedte
    indeling[sluiskant] ← NEW SET
    JOIN plaats[1] TO indeling[sluiskant]
    THIS punt ← NEW punt
    xx ← x
    yy ← brdtkolk
    JOIN THIS punt TO indeling[sluiskant]
    THIS punt ← NEW punt
    xx ← x
    yy ← 0
    JOIN THIS punt TO indeling[sluiskant]
    JOIN plaats[2] TO indeling[sluiskant]
  END

Voeg nu het schip toe
CALL voegtoe(sluiskant)
END

RETURN

```

## 6.15 HEVELEN

Deze MACRO wordt aan het attribuut VOORWAARDE van een KOLK toegevoegd als de kolk gesloten wordt met schutwaterbesparende maatregelen gerelateerd aan het nivelleren.

De voorwaarde is dat een kolk pas met nivelleren mag beginnen als de hiertoe bestemde buurkolk (aan de andere kant) daar ook aan toe is. Om een dergelijke situatie ooit te doen starten, dienen de betrokken kolken aan het begin van de simulatie hun deuren in tegenovergestelde richting geopend te hebben. Hierin wordt voorzien door het in de file NETWERK opgeven kolknummer. Is dit oneven dan is de deur aan kant 1 open. Is dat even dan is de deur aan kant 2 open. De gebruiker zal er voor moeten zorgen dat twee kolken die samen hevelen een even en een oneven nummer hebben. Uiteraard zal dit hevelen invloed hebben op de snelheid van het nivelleren en het daarbij optredende debiet.

Terugkeerwaarde:

TRUE (=1): voorwaarde vervuld of niet relevant.

FALSE (=0): voorwaarde niet vervuld.

```
@=====
@
@ HEVELEN
@
@=====

@ hevel voorwaarden, aangeroepen door THIS KOLK @

IF kolk_wens = nivelleren
  IF buurkolk IS NOT ACTIVE
    ACTIVATE buurkolk FROM vaarin IN kolkmod
    RETURN FALSE
  END
  RETURN FALSE IF side          OF buurkolk = side
  RETURN TRUE  IF kolk_status OF buurkolk = nivelleren
  RETURN TRUE  IF kolk_wens   OF buurkolk = nivelleren
  RETURN FALSE
END
RETURN TRUE
```

## 6.16 HEVELEND

Deze MACRO wordt aan het attribuut VOORWAARDE van een KOLK toegevoegd als de kolk gesloten wordt met schutwaterbesparende maatregelen gerelateerd aan het sluiten van de deur.

Het verschil met hevelen zit in de letter d van deur. De voorwaarde is nu gevoegd aan het sluiten van de deur. Dat mag pas als de buurkolk ook zo ver is. Het essentiële verschil met HEVELEN is dat gedurende die wachttijd de deur nog open is en er dus nog schepen de kolk in kunnen varen. Met betrekking tot het gebruik van parameters is er geen verschil met de afhankelijkheid HEVELEN.

Terugkeerwaarde:

TRUE (=1): voorwaarde vervuld of niet relevant.

FALSE (=0): voorwaarde niet vervuld.

```
@=====
@
@ HEVELEND
@
@=====

@ hevel voorwaarden, aangeroepen door THIS KOLK @

IF kolk_wens = d_dicht
  IF buurkolk IS NOT ACTIVE
    ACTIVATE buurkolk FROM vaarin IN kolkmod
    RETURN FALSE
  END
  RETURN FALSE IF side      OF buurkolk = side
  RETURN FALSE IF kolk_wens OF buurkolk ≠ d_dicht
  CANCEL buurkolk
  REACTIVATE buurkolk
  RETURN TRUE
END
IF kolk_wens = wachten
  RETURN FALSE IF kolk_wens OF buurkolk = d_dicht
END
RETURN TRUE
```



## 6.17 HEVELEND\_INIT

Deze MACRO verzorgt de initialisatie van het schutten met voorwaarde hevelen.

Terugkeerwaarde:

0: Ok.

-1: er is iets mis.

```
@=====
@
@ HEVELEND_INIT
@
@=====

@ macro ter initialisatie van het hevelend schutten van twee kolken @
@ wordt aangeroepen in LEES_SLUIZEN en betreft THIS SLUIS @
LOCAL :
  INTEGER: nr
  REAL    : h_fac

                                Ga na of hevelend schutten mag
GOTO fout0 IF schutplanning IS NOT geen_schutstrategie
ref[1] ← FIRST kolk IN kolken WITH voorwaarde IS hevelend
ref[2] ← LAST  kolk IN kolken WITH voorwaarde IS hevelend
h_fac ← READ FROM netwerk

GOTO fout1 IF ref[1] IS ref[2]
nr ← nrkolk OF ref[1] + nrkolk OF ref[2]
GOTO fout2 IF nr = 2*FLOOR(nr/2)

hevel_factor OF ref[1] ← h_fac
hevel_factor OF ref[2] ← h_fac
buurkolk     OF ref[1] ← ref[2]
buurkolk     OF ref[2] ← ref[1]
deb_reductie OF ref[1] ← READ FROM netwerk
deb_reductie OF ref[2] ← READ FROM netwerk

RETURN 0
fout0: WRITE "voorwaarden samen met een schutstrategie is te bar!"
      WITH IMAGE ΔΔΔa
RETURN -1
fout1: WRITE "kolk";nrkolk OF ref[1];"hevelt met zichzelf"
      WITH IMAGE ΔΔΔa**Δa
RETURN -1
fout2:
      WRITE "deuren van de hevelende kolken staan aan dezelfde kant open"
      WITH IMAGE ΔΔΔa
RETURN -1
```



## 6.18 HEVELEN\_INIT

Deze MACRO verzorgt de initialisatie van het hevelend schutten.

Terugkeerwaarde:

- 0: Ok.
- 1: er is iets mis.

```

@=====
@
@ HEVELEN_INIT
@
@=====

@ macro ter initialisatie van het hevelend schutten van twee kolken @
@ wordt aangeroepen in LEES_SLUIZEN en betreft THIS SLUIS          @
LOCAL :
  INTEGER: nr
  REAL    : h_fac

                                Ga na of hevelend schutten mag
GOTO fout0 IF schutplanning IS NOT geen_schutstrategie
ref[1] ← FIRST kolk IN kolken WITH voorwaarde IS hevelen
ref[2] ← LAST  kolk IN kolken WITH voorwaarde IS hevelen
h_fac ← READ FROM netwerk

GOTO fout1 IF ref[1] IS ref[2]
nr ← nrkolk OF ref[1] + nrkolk OF ref[2]
GOTO fout2 IF nr = 2*FLOOR(nr/2)

hevel_factor OF ref[1] ← h_fac
hevel_factor OF ref[2] ← h_fac
buurkolk     OF ref[1] ← ref[2]
buurkolk     OF ref[2] ← ref[1]
deb_reductie OF ref[1] ← READ FROM netwerk
deb_reductie OF ref[2] ← READ FROM netwerk

RETURN 0
fout0: WRITE "voorwaarden samen met een schutstrategie is te bar!"
      WITH IMAGE ΔΔΔa
RETURN -1
fout1: WRITE "kolk";nrkolk OF ref[1];"hevelt met zichzelf"
      WITH IMAGE ΔΔΔa**Δa
RETURN -1
fout2:
      WRITE "deuren van de hevelende kolken staan aan dezelfde kant open"
      WITH IMAGE ΔΔΔa
RETURN -1

```

## 6.19 HULPJE

Dit is een hulp-MACRO om een serie getallenparen in te lezen.

```
@=====
@
@ HULPJE
@
@=====
```

De paren komen als punten in een hulpset.

```
hulp ← NEW SET
x ← READ FROM data
WHILE x ≥ 0
  THIS punt ← NEW punt
  xx      ← x
  yy      ← READ FROM data
  JOIN THIS punt TO hulp RANKED BY 10*xx
  x      ← READ FROM data
END
x ← -1
RETURN
```

## 6.20 IDLE\_STOP\_WATCH

Deze MACRO verzorgt de administratie betreffende de idle-time van een kolk.

Parameter:

START_STOP	te nemen actie
	1: administreer het begin van een idle periode
	2: administreer het eind van een idle periode

```

@=====
@
@ IDLE_STOP_WATCH
@
@=====

@=====
@   Start (1) of stop (2) de idle-time meting van THIS KOLK
@=====
PARAMETER:
    INTEGER: start_stop

GOTO stop IF start_stop=2
    IF start_idle = -1
        start_idle ← NOW
        idle_time ← 0
    END
    RETURN

stop:
    idle_time ← NOW-start_idle IF start_idle ≠ -1
    start_idle ← -1
    RETURN

```

## 6.21 INITDEEL

Deze MACRO initialiseert een indeling voor een kolk. Zie de toelichting bij MACRO PASSEN voor een uitleg over het indelen van een kolk.

```
@=====
@
@ INITDEEL
@
@=====

@=====@
@ Basis INDELING van THIS KOLK aan SLUISKANT @
@=====@

PARAMETER:
    INTEGER: sluiskant

indeling[sluiskant] ← NEW SET CALLED sluiskant|" indeling "|nrkolk

THIS punt ← NEW punt
xx         ← lengkolk
yy         ← brdtkolk
JOIN THIS punt TO indeling[sluiskant]

THIS punt ← NEW punt
xx         ← 0
yy         ← brdtkolk
JOIN THIS punt TO indeling[sluiskant]

THIS punt ← NEW punt
xx         ← 0
yy         ← 0
JOIN THIS punt TO indeling[sluiskant]

THIS punt ← NEW punt
xx         ← lengkolk
yy         ← 0
JOIN THIS punt TO indeling[sluiskant]

RETURN
```

## 6.22 INITIALISATIE

Deze MACRTO verzorgt de initialisatie van de simulatie.

Terugkeerwaarde:

TRUE: de initialisatie is gelukt

FALSE: de initialisatie is mislukt

```
@=====
@
@ INITIALISATIE
@
@=====
```

### Namen van de weekdays

```
dag[1]      ← NEW dagtype CALLED "maandag"
dag[2]      ← NEW dagtype CALLED "dinsdag"
dag[3]      ← NEW dagtype CALLED "woensdag"
dag[4]      ← NEW dagtype CALLED "donderdag"
dag[5]      ← NEW dagtype CALLED "vrijdag"
dag[6]      ← NEW dagtype CALLED "zaterdag"
dag[7]      ← NEW dagtype CALLED "zondag"
```

### Constanten

```
voor_aanmelding ← 1
schip_bij_meldraai ← 2
nieuw_opstelplan ← 3
vrije_kolk      ← 4
bepaal_deadline ← 5
schip_bij_sluis ← 6
sluis_zelf      ← 7
```

```
passeer_knoop      ← 0
naar_opstelruimte ← 3
uit_opstelruimte ← 4
start_overliggen   ← 0
start_schutten     ← 1
start_doorvaren    ← 1
via_kolk           ← 2
via_doorgang       ← 2
```

```
s_recl            ← 12
```

```
rood    ← 1
oranje  ← 2
groen   ← 3
```

```
invaren    ← 1
d_dicht    ← 2
nivelleren ← 3
d_open     ← 4
uitvaren   ← 5
wachten   ← 6
```

```
@-----@
@      Display schepen bij een vernauwing:      @
@ 0=niet, 1=wel ononderbroken, 2 =wel ononderbroken @
@-----@
toon_vernauwing ← 0
```

### Kleurinstellingen

```
CALL get_colours(<colors>)
sf ← colors[1,1]
sb ← colors[1,2]
af ← colors[2,1]
ab ← colors[2,2]
```



```

ikf ← colors[3,1]
ikb ← colors[3,2]
mf ← colors[4,1]
mb ← colors[4,2]
wit ← 15

```

#### Strepen voor gebruik in rapporten

```

FOR k ← 1 TO 127
  CALL char_in_string(<estreep>,k,1,"-",1)
  CALL char_in_string(<dstreep>,k,1,"=",1)
END

```

#### Datum en tijd

```

CALL get_date(<date>)
CALL get_time(<time>)

```

#### Initialisatie query bestand

```

REWIND query_invoer
WRITE date[1];date[2];date[3] TO query_invoer WITH IMAGE |****Δ**Δ**
FOR k ← 1 TO 3
  WRITE time[k] TO query_invoer WITH IMAGE |Δ**
END
WRITE "" TO query_invoer WITH IMAGE a

WRITE "Inlezen bestanden" WITH IMAGE ΔΔa
WRITE " " WITH IMAGE *

```

#### Globale verzamelingen

```

sluizen      ← NEW SET CALLED "SLUIZEN"
nodes        ← NEW SET CALLED "NODES"
bruggen      ← NEW SET CALLED "BRUGGEN"
vernuwingen  ← NEW SET CALLED "VNAUWINGEN"
vaarwegen    ← NEW SET CALLED "VAARWEGEN"
generatoren  ← NEW SET CALLED "GENERATOREN"

```

#### Vraag DOS-pad op voor lokatie van query MATFILES

```

dospad      ← ""
qfilenaam   ← ""
IF streams_filename("QUERY_INVOER",<qfilenaam>,<dospad>) < 0
  WRITE ". "; "geen file gevonden voor QUERY_INVOER" WITH IMAGE a13a
  RETURN FALSE
END
dospadshp ← dospad
CALL char_in_string(<qfilenaam>,1+string_length(qfilenaam),4,".MAF",1)
CALL char_in_string(<dospad>,1+string_length(dospad),12,qfilenaam,1)

data        ← netwerk
simulatieduur ← READ FROM netwerk

```

```

@-----@
@ Inlezen klassen van schepen die zich eerder melden @
@-----@
@-----@
@ bij een sluis @
@-----@
k ← READ FROM netwerk
WHILE 0 < k
  sgarklas[k] ← TRUE
  k ← READ FROM netwerk
END
@-----@
@ bij een vernauwing @
@-----@
k ← READ FROM netwerk
WHILE 0 < k
  vgarklas[k] ← TRUE

```

```

    k ← READ FROM netwerk
END

@-----@
@ Categorie%n @
@-----@
j ← 0
n ← READ FROM bibklas
WHILE 0 < n
    k ← READ FROM bibklas
    WHILE 0 < k
        IF 0 < categorie[k]
            IF j=0
                WRITE ". "; "scheepsklasse valt onder meer dan één categorie:"
                WITH IMAGE a13a
                WRITE "" WITH IMAGE a14
                j ← 1
            END
            WRITE k WITH IMAGE |Δ**
        END
        categorie[k] ← n
        k ← READ FROM bibklas
    END
    n ← READ FROM bibklas
END

FOR k ← 1 TO 30
    categorie[k] ← 2 IF categorie[k]=0
END
CLOSE bibklas
RETURN FALSE IF j=1

ACTIVATE observator FROM observeer IN fysiek

RETURN FALSE IF lees_vaarwegen < 0 @inlezen vaarwegen @
RETURN FALSE IF lees_sluizen < 0 @ " " sluizen @
RETURN FALSE IF lees_bruggen < 0 @ " " bruggen @
RETURN FALSE IF lees_vernauwingen < 0 @ " " vernauwingen @
RETURN FALSE IF lees_getijden < 0 @ " " getij-krommen @
RETURN FALSE IF lees_wegintens < 0 @ " " wegverkeer-intensiteiten@
RETURN FALSE IF lees_regimes < 0 @ " " brug- & schutregimes @
RETURN FALSE IF lees_generatoren < 0 @ " " generatoren @
RETURN FALSE IF lees_vaarkrommen < 0 @ " " Am/Ac, Bs/Bc KROMMEN @
CLOSE netwerk

@-----@
@ initialisatie DEBIET storestreams @
@-----@
FOR EACH sluis IN sluizen WITH debietbepaling
    FOR k ← 1 TO 2
        STORE 0 AS k|" Debiet "|nrsluis
    END
    FOR EACH kolk IN alle_kolken
        FOR k ← 1 TO 2
            STORE 0 AS k|" Debiet "|nrkolk + 10*nrsluis
        END
    END
END
END

CALL clear_text_screen
RETURN TRUE

```

## 6.23 INTIJD

Deze MACRO berekent de invaartijd van een schip in een kolk.

Parameters:

SLUISKANT de kant van de sluis waarvandaan het schip invaart

```
@=====
@
@ INTIJD
@
@=====

@=====
@ Macro voor de berekening van de INVAAR- en VOLGTIJDEN bij een KOLK @
@=====
PARAMETER:
    INTEGER: sluiskant
LOCAL:
    REAL    : x y z

THIS sluis ← zijnsluis

@-----@
@          correctiefactor          @
@ volproppen & gescherfd invaren van de @
@ recreatievaart bij gescheiden schutten @
@ recr_ruimte is bepaald in MAC HERDEEL @
@-----@
z ← 1
IF (gemengd_schutten = FALSE) & categorie[klas] = 3
    Recreatievaart bij niet-gemengd schutten
    @***** bij volproppen wordt invaartijd met 50% verlengd *****
    y ← beroeps_ruimte[sluiskant] @ wat is de bezette ruimte?
    FOR EACH schip IN schutruimte WITH categorie[klas] = 3
        y ← y + planlengte * planbreedte
    END
    z ← 1.5 IF 0.8 < (y/brdtkolk*lengkolk) @ THIS KOLK is reeds voor meer
                                           @ dan 80% bezet

    @***** gescherfd invaren reduceert invaartijd *****
    IF z=1 @ niet bij volproppen
        z ← 3 * planbreedte/brdhoofd
        z ← 1 IF z > 1
    END
END

@-----@
@ niet recreatievaart of gemengd schutten @
@-----@
x ← breedte / brdhoofd
IF (categorie[klas] ≠ 3) & gel = 2
    y ← waterstand(s_nrtij[sluiskant],s_peilpar[sluiskant])
    x ← breedte * diepgang / brdhoofd * y - drempel[sluiskant]
    x ← 1 IF x < 0
END

IF postschip IS NONE
    @-----@
    @ schip in lopende vaart @
    @-----@
    IF preschip IS NONE
        z ← 1 IF z < 1 @ geen gescherfd invaren in lopende vaart!
        y ← z * VALUE OF tijdin_1ste[klas,gel] AT(x)
    END
END
```

```
@-----@
@ eerste schip @
@-----@
IF preschip IS THIS schip
  y ← z * VALUE OF tijdin_1ste[klas,gel] AT(x)
END

@-----@
@ volgschip @
@-----@
IF preschip IS NOT NONE & preschip IS NOT THIS schip
  y ← z * VALUE OF tijdin_volg[klas,gel] AT(x)
  IF (gemengd_schutten=FALSE) &
    categorie[klas OF preschip] < categorie[klas]
    y ← y + icat12 IF (categorie[klas OF preschip]=1) &
      categorie[klas]=2
    y ← y + icat13 IF (categorie[klas OF preschip]=1) &
      categorie[klas]=3
    y ← y + icat23 IF (categorie[klas OF preschip]=2) &
      categorie[klas]=3
  END
END
END

@-----@
@ lusschip @
@-----@
IF postschip IS NOT NONE
  y ← z * VALUE OF tijdin_1ste[klas,gel] AT(x)
  y ← y + VALUE OF lustijd[gel] AT(lengte + fuiklengte[sluiskant])
END

postschip ← NONE

RETURN y MINUTES
```



## 6.24 KOLKMOD

Dit MODULE bevat de procesbeschrijving van een kolk.

```
@=====
@
@ KOLKMOD
@
@=====

@=====
@ Procesbeschrijving van de kolk/kolkbediende @
@=====
```

vaaruit:

```

                                Uitvaren van de schepen in de kolk
kolk_wens ← uitvaren
IF voorwaarde = FALSE

                                Even wachten op voorwaarden
    vw_start ← NOW
    WAIT WHILE voorwaarde = FALSE
    wacht_op_voorw ← wacht_op_voorw + NOW - vw_start
END

                                Daar gaan we dan
kolk_wens ← 0
kolk_status ← uitvaren
@-----@
@ categorie uitvaarvolgtijd-correctie @
@-----@
IF gemengd_schutten
    x ← 0
    x ← ucat12          IF cat_in_kolk=3
    x ← ucat13          IF cat_in_kolk=4
    x ← ucat23          IF cat_in_kolk=5
    x ← ucat12 + ucat23 IF cat_in_kolk=6
    WAIT x MINUTES IF x > 0
END

postschip ← LAST OF schutruimte
preschip ← NONE

                                Zolang er nog een schip in de schutruimte zit
WHILE schutruimte IS NOT EMPTY

                                Laat dat uitvaren
    THIS schip ← FIRST OF schutruimte
    WAIT uittijd(side)

    THIS schip ← FIRST OF schutruimte
    REMOVE THIS schip FROM schutruimte
    vulgraad ← vulgraad - 100*planbreedte*planlengte/opp_kolk
    REACTIVATE THIS schip
    preschip ← THIS schip
END

                                Kolk is nu leeg
kolk_status ← idle

                                Om voordrangers een kans te geven
WAIT 1 SECOND
duur ← NOW - duur
CALL sluis_registratie(TRUE)

postschip ← NONE IF opstelplan[side] IS EMPTY
schuttende ← FALSE
CALL idle_stop_watch(1) IF THIS kolk BELONGS TO kolken OF zijnsluis
buiten_bedrijf ← NOW
```



```

@-----@
@ wacht tot centralist bepaalt dat er weer geschut moet worden @
@-----@
beschikbaar:
  LEAVE vrije_kolken OF zijnsluis
    IF THIS kolk BELONGS TO vrije_kolken OF zijnsluis
      bezig ← FALSE
    Laat sluismeester bepalen wat er moet gebeuren
  x ← schutplanning(vrije_kolk,0) OF zijnsluis
  IF 0 < x
    Wachten op volgende bedrijfsperiode
    wacht_op_periode ← TRUE
    WAIT x HOURS

    LEAVE vrije_kolken OF zijnsluis
      IF THIS kolk BELONGS TO vrije_kolken OF zijnsluis
        wacht_op_periode ← FALSE
        REPEAT FROM beschikbaar
  END
  IF x = -1 @ kolk zou passief mogen worden
    kolk_wens ← wachten
    IF voorwaarde = FALSE
      Mag toch niet wachten
      kolk_wens ← 0
      GOTO doorwerken
  END
  PASSIVATE IF x=-1 @*** onbepaalde tijd ***
  END
  doorwerken:
  LEAVE vrije_kolken OF zijnsluis
    IF THIS kolk BELONGS TO vrije_kolken OF zijnsluis

vaarin:
  CALL idle_stop_watch(2)
  bezig ← TRUE
  schuttende ← TRUE
  duur ← NOW
  wacht_op_voorw ← 0
  kolk_wens ← invaren
  IF voorwaarde = FALSE
    Even wachten op voorwaarden
    vw_start ← NOW
    WAIT WHILE voorwaarde = FALSE
    wacht_op_voorw ← wacht_op_voorw + NOW - vw_start
  END
  kolk_wens ← 0
  kolk_status ← invaren
  Schepen 1 voor 1 laten invaren
  THIS schip ← FIRST schip IN opstelplan[side]
  THAT BELONGS TO opstelruimte[side]
  preschip ← THIS schip
  vaardoor:
  WHILE THIS schip IS NOT NONE
    y ← intijd(side)
    preschip ← THIS schip
    REMOVE preschip FROM opstelruimte[side]
    CALL query_registratie(uit_opstelruimte)
    Schip heeft opstelruimte verlaten en vaart naar binnen
    WAIT y
    Schip bevindt zich nu in de schutruimte

```

```

JOIN preschip TO schutruimte
THIS schip ← preschip
vulgraad ← vulgraad + 100*planbreedte*planlengte/opp_kolk
                                Volgende schip is eerste in opstelruimte omdat
                                voorganger daar inmiddels uit is
THIS schip ← FIRST schip IN opstelplan[side]
                                THAT BELONGS TO opstelruimte[side]
END

```

lopendevaart:

Als er geen invarende schepen meer zijn zou de deur dicht kunnen

```

GOTO zetover IF zetover_ok = TRUE
WAIT UNTIL preschip BELONGS TO opstelruimte[side]

THIS schip ← preschip
preschip ← NONE
REPEAT FROM vaardoor

```

zetover:

Alle schepen zijn nu binnen

```

@-----@
@      wacht tot centralist bepaalt de deuren dicht mogen      @
@      igv een éénkolkse sluis met een schutstrategie      @
@-----@

```

Vraag om beslissing van sluismeester

```

x ← schutplanning(bepaal_deadline,0) OF zijnsluis
IF 0 < x

```

Te wachten op deadline

```

    bezig ← FALSE
    wacht_op_deadline ← TRUE
    WAIT x HOURS

    bezig ← TRUE
    wacht_op_deadline ← FALSE
    REPEAT FROM zetover
END

```

@\*\*\* ivm voorkennistijd \*\*\*

Deur kan nu dicht

```

kolk_wens ← d_dicht
IF voorwaarde = FALSE

```

Toch nog even op voorwaarden wachten  
Dwz wachten of er nog een schip komt in geval van hevelend schutten

```

vw_start ← NOW
WAIT WHILE zetover_ok = TRUE
kolk_wens ← 0
wacht_op_voorw ← wacht_op_voorw + NOW - vw_start
IF preschip IS NOT NONE

```

Werk ook dit schip nog naar binnen

```

    WAIT UNTIL preschip BELONGS TO opstelruimte[side]
    THIS schip ← preschip
    preschip ← NONE
    REPEAT FROM vaardoor
END

```

```

END
kolk_status ← d_dicht

```

Zoek uit of het een volschutting is

```

refschip ← NONE
FOR EACH schip IN wachtruimte[side] OF zijnsluis
    refschip ← THIS schip IF passen(10) ≤ 1
END
kmax ← refschip IS NOT NONE & schutruimte IS NOT EMPTY
REMOVE EACH schip IN opstelplan[side] FROM opstelplan[side]

```

```

@-----@
@ categorie invaarvolgtijd-correctie @
@-----@
opprend ← 0
cat_in_kolk ← 0
FOR k ← 1 TO 3
    THIS schip ← FIRST schip IN schutruimte WITH categorie[klas]=k
    cat_in_kolk ← cat_in_kolk + k IF THIS schip IS NOT NONE
END
IF gemengd_schutten=FALSE
    opprend ← cat12          IF cat_in_kolk=3
    opprend ← cat13          IF cat_in_kolk=4
    opprend ← cat23          IF cat_in_kolk=5
    opprend ← cat12 + cat23 IF cat_in_kolk=6
END
opprend ← opprend * brdtkolk
IF gemengd_schutten
    x ← 0
    x ← icat12          IF cat_in_kolk=3
    x ← icat13          IF cat_in_kolk=4
    x ← icat23          IF cat_in_kolk=5
    x ← icat12 + icat23 IF cat_in_kolk=6
    WAIT x MINUTES IF x > 0
END

aantal      ← LENGTH OF schutruimte
totlaadverm ← 0
FOR EACH schip IN schutruimte
    opprend ← opprend + planlengte * planbreedte
    totlaadverm ← totlaadverm + laadverm
    tw ← NOW - tw IF wachtend
    t1 ← 0 IF wachtend
    t1 ← NOW - t1 IF wachtend = FALSE
    ts ← NOW
    CALL query_registratie(start_schutten)
END

FOR EACH schip IN wachtruimte[side] OF zijnsluis WITH wachtend
    IF passen(10) ≤ 1
        tw ← NOW - tw
        t1 ← NOW
        wachtend ← FALSE
        CALL query_registratie(start_overliggen)
    END
END

Maak nieuwe indeling

CALL initdeel(side)

Laat nieuw opstelplan maken
CALL schutplanning(nieuw_opstelplan,side) OF zijnsluis
Sluit de deuren

WORK tsluit[side] MINUTES

Nu moeten we nivelleren

kolk_wens ← nivelleren
IF voorwaarde = FALSE

Toch maar even wachten

    vw_start ← NOW
    WAIT WHILE voorwaarde = FALSE
    wacht_op_voorw ← wacht_op_voorw + NOW - vw_start
END
kolk_wens ← 0

Daar gaat ie dan

kolk_status ← nivelleren
THIS sluis ← zijnsluis

```

```

x ← waterstand(s_nrtij[side],s_peilpar[side])
y ← waterstand(s_nrtij[3-side],s_peilpar[3-side])
cur_debiet ← 0
cur_debiet ← .001 * waterlengte * brdtkolk * x-y IF x > y
cur_debiet ← deb_reductie * cur_debiet
                IF voorwaarde IS hevelen | voorwaarde IS hevelend
debiet[3-side] ← debiet[3-side] + cur_debiet IF debietbepaling
WORK (hevel_factor * nulniveau[side] + ABS(x-y) * facniveau[side]) MINUTES

```

Dat is gebeurd

```
side ← 3 - side
```

Kunnen openen aan de andere kant

```

kolk_wens ← d_open
IF voorwaarde = FALSE

```

Toch maar even wachten

```

    vw_start ← NOW
    WAIT WHILE voorwaarde = FALSE
    wacht_op_voorw ← wacht_op_voorw + NOW - vw_start
END
kolk_wens ← 0
kolk_status ← d_open

```

Wacht to de deuren open zijn

```
WORK topen[side] MINUTES
```

En laat de schepen uitvaren

```
REPEAT FROM vaaruit
```



## 6.25 KOLK\_PREFERENTIE

Deze MACRO bepaalt de preferentie voor een kolk op basis van een strategie.

Parameters:

SLUISKANT de kant van de sluis waar het te beschouwen schip aankomt.

```
@=====
@
@ KOLK_PREFERENTIE
@
@=====

@=====@
@ Bepaal de preferentie voor een KOLK op basis van een strategie @
@=====@
PARAMETER:
    INTEGER: sluiskant

Switch naar gelang de strategie

GOTO strategie_1 IF strategie = 1
GOTO strategie_2 IF strategie = 2
GOTO strategie_3 IF strategie = 3

Onbekend strategienummer, kies default 1
WRITE ". "; "Illegaal STRATEGIE NR voor KOLK PREFERENTIE: "; strategie
    WITH IMAGE a20a*****
WRITE " "; "Strategie nr 1 is gekozen" WITH IMAGE a20a

@-----@
@ Preferentie op basis van GROOTTE SCHIP @
@-----@
strategie_1:
    RETURN brdtkolk * lengkolk / planbreedte * planlengte

@-----@
@ Preferentie op basis van service verlening @
@-----@
strategie_2:

    RETURN 0 IF (sluiskant=side) & kolk_status = invaren
    RETURN 1 IF (sluiskant=side) & kolk_status = idle
    RETURN 2 IF (sluiskant=side) & kolk_status = uitvaren
    RETURN 3 IF (sluiskant=side) & kolk_status = d_open
    RETURN 4 IF (sluiskant=3-side) & kolk_status = nivelleren
    RETURN 5 IF (sluiskant=3-side) & kolk_status = d_dicht
    RETURN 6 IF (sluiskant=3-side) & kolk_status = idle
    RETURN 7 IF (sluiskant=3-side) & kolk_status = invaren
    RETURN 8 IF (sluiskant=3-side) & kolk_status = uitvaren
    RETURN 9 IF (sluiskant=3-side) & kolk_status = d_open
    RETURN 10 IF (sluiskant=side) & kolk_status = nivelleren
    RETURN 11

@-----@
@ gemengde strategie @
@-----@
strategie_3:
    RETURN brdtkolk * lengkolk / planbreedte * planlengte
        IF sluiskant ≠ side
    z ← brdtkolk * lengkolk
    FOR EACH schip IN opstelplan[sluiskant]
        z ← z - planbreedte * planlengte
    END
    RETURN z / planbreedte * planlengte
```



## 6.26 KOP

Deze MACRO drukt, indien nodig, de kop van een bladzijde van het rapport af.

```

@=====
@
@ KOP
@
@=====

@=====
@ MACRO KOP: DRUKT DE KOP VAN DE BLADZIJDE AF INDIEN NODIG @
@=====
LOCAL:
  INTEGER: i

                                Start van het rapport ?
GOTO eersteblad IF pagnr = 0
IF nieuw | (linenr + ABS(regeloph)) > regelaantal-2
                                Pagina is vol, dus formfeed als er maar 1 pagina in
                                het rapport zit
  WRITE #12 TO rapport WITH IMAGE |a IF pagnr=1
  IF 1 < pagnr
                                Skip naar einde van de pagina om een
                                paginanummer te kunne printen
    FOR i ← linenr+1 TO regelaantal-2
      WRITE "" TO rapport WITH IMAGE *
    END
    WRITE #13;"-";pagnr;"-" TO rapport WITH IMAGE *A34Δ**Δ*
    IF regeloph < 0
                                Einde rapport
      WRITE #13;"[einde rapport]";#12 TO rapport WITH IMAGE |*A75a
      RETURN
    END
                                En dan uiteindelijk toch de formfeed
    WRITE #12 TO rapport WITH IMAGE |a
  END
  pagnr ← pagnr + 1
eersteblad:
                                Druk de kop-informatie af
  WRITE "[";estreep;"]" TO rapport WITH IMAGE *a73*
  WRITE "[";text      ;"]" TO rapport WITH IMAGE *A58A16
  WRITE "[";estreep;"]" TO rapport WITH IMAGE *a73*
  linenr ← 4
  WRITE "" TO rapport WITH IMAGE *
                                @-----@
                                @ REGELNR VAN DE EERST VOLGENDE @
                                @ TE PRINTEN REGEL                @
                                @-----@

  IF pagnr = 0
    pagnr ← 1
    regeloph ← 4
  END
END
linenr ← linenr + regeloph IF nieuw=FALSE
nieuw ← FALSE
regeloph ← 1
RETURN

```



```

END
FOR k ← 1 TO 2
  wachtrij[k] ← NEW SET WITH STATISTICS
    CALLED i[k] | "..BWACHT.." | i[3-k]
  meldingen OF zijde[k] ← NEW SET CALLED "MELDING.." | i[k]
  waterpad[k] ← FIRST vaarweg IN vaarwegen
    WITH raai[1] IS zijde[k] | raai[2] IS zijde[k]
  IF waterpad[k] IS NONE
    WRITE "• "; "Er grenst geen VAARWEG aan"; NAME OF zijde[k]
    WITH IMAGE a20aΔa
    gelukt ← FALSE
  END
END
END
CALL case_string(<type>, 1, 6, 1)
GOTO fout IF (type ≠ "ENKEL") & (type ≠ "DUBBEL") & (type ≠ "DRAAI")
nrdoorgangen ← 1 + type = "DUBBEL"
bschipstats ← NEW SET
FOR k ← 1 TO 1 + type ≠ "ENKEL"
  wijdte[k] ← READ FROM netwerk
  hoogte[k] ← READ FROM netwerk
  THIS schipstat ← NEW schipstat
  s_volgnr ← k
  JOIN THIS schipstat TO bschipstats
END
FOR k ← 1 TO 10
  bmin[k] ← 1.0E5 @*** minima initialiseren
END
open ← FALSE
licht ← "GROEN"
brfuiklengte[1] ← READ FROM netwerk
brfuiklengte[2] ← READ FROM netwerk
tboopen ← READ FROM netwerk MINUTES
tbsluit ← READ FROM netwerk MINUTES

@-----@
@ Categorie volgtijden @
@-----@
bcat12 ← READ FROM netwerk
bcat13 ← READ FROM netwerk
bcat23 ← READ FROM netwerk

@-----@
@ Inlezen WEGVERKEER @
@-----@
wegverkeer ← NEW SET
FOR k ← 1 TO 2 @ 1=snelverkeer, 2=langzaamverkeer
  wkosten[k] ← READ FROM netwerk @ kosten wegverkeer
  y ← READ FROM netwerk * 60 @ oplosvermogen in # auto's per uur
  FOR l ← 1 TO 2 @ richting
    m ← READ FROM netwerk
    IF m < 0
      x ← READ FROM netwerk
      x ← READ FROM netwerk
    END
    IF 0 ≤ m
      THIS stroom ← NEW stroom CALLED nrbrug|" STROOM "|1 + 10*k
      cat ← k
      richt ← l
      nrintens ← m
      THIS ipunt ← FIRST ipunt IN a_intens WITH nn1 = m
      IF THIS ipunt IS NONE
        THIS ipunt ← NEW ipunt
        nn1 ← m
        JOIN THIS ipunt TO a_intens RANKED BY nn1
      END
      nn2 ← -1
    END
  END
END

```

```

        weektot      ← READ FROM netwerk
        stroken      ← READ FROM netwerk
        oplosvermogen ← y
        mijnbrug     ← THIS brug
        JOIN THIS stroom TO wegverkeer
    END
END
END

```

#### Verwerk de brugwachter

```

THIS brugwachter ← NEW brugwachter
FOR k ← 1 TO 2
    brugplan[k] ← NEW SET CALLED k|" PLAN  "|nrbrug
    THIS regelaar ← NEW regelaar
    watcher     ← THIS brugwachter
    bridge      ← THIS brug
    bside       ← k
    brughulp[k] ← THIS regelaar
END
zijnbrug      ← THIS brug
zijnwachter   ← THIS brugwachter

```

#### Verwerk de klok van de brugwachter

```

THIS klok      ← NEW klok
shift         ← NEW SET
inhibitor     ← THIS brugwachter
x             ← READ FROM netwerk
WHILE x ≥ 0
    y ← READ FROM netwerk
    THIS werktijd ← NEW werktijd
    begin ← x
    einde ← y
    JOIN THIS werktijd TO shift RANKED BY 100*begin
    x ← READ FROM netwerk
END
@-----@
@ Inlezen REGIMEnummer (geen regime = -1) @
@-----@
k ← READ FROM netwerk
THIS regimeblok ← leegblok
IF 0 < k
    THIS regimeblok ← FIRST regimeblok IN regimeblokken
                     WITH bloknr = k
    IF THIS regimeblok IS NONE
        THIS regimeblok ← NEW regimeblok
        bloknr ← k
        r_inhoud ← NEW SET CALLED "REGIMES  "|k
        JOIN THIS regimeblok TO regimeblokken RANKED BY bloknr
    END
END
regimes ← r_inhoud

```

#### Laat alles tot leven komen

```

ACTIVATE THIS klok      FROM brugklok  IN manage_brug
ACTIVATE THIS brug      FROM startbrug IN fysiek
ACTIVATE THIS brugwachter FROM wacht   IN manage_brug
i[1] ← READ FROM netwerk
END

```

```

RETURN 1 IF gelukt
RETURN -1

```

fout:

```

WRITE "• "; "Deze brug is van een onbekend type"; type WITH IMAGE a20aΔa
RETURN -1

```







```

r_schepen      ← NEW SET
aantal_schepen ← 1 + hship[5] - n
max_op_blad    ← FLOOR(10000/s_recl)
rc             ← n
n              ← aantal_schepen
WHILE n > 0
  THIS bladzij ← NEW bladzij
  JOIN THIS bladzij TO r_schepen
  IF n ≥ max_op_blad
    regels ← max_op_blad
    n      ← n - max_op_blad
    GOTO smoven
  END
  regels ← n
  n      ← 0
smoven: rcode ← move_block_matfile(<hship>,rc,regels,<bladinhoud>,1,0)
IF rcode < 0
  WRITE ". "; "fout bij het inlezen van";filenaam;
        "(code=";rcode;)" WITH IMAGE a20aΔaΔaΔ***a
  gelukt ← FALSE
END
rc ← rc + regels
END

```

vervolg:

```

route_heen      ← NEW SET CALLED "ROUTE H "|nrgen
route_terug     ← NEW SET CALLED "ROUTE T "|nrgen
weg_heen        ← NEW SET CALLED "TRAJECT H "|nrgen
weg_terug       ← NEW SET CALLED "TRAJECT T "|nrgen
palen_heen      ← NEW SET CALLED "PALEN H "|nrgen
palen_terug     ← NEW SET CALLED "PALEN T "|nrgen

```

#### Route informatie

#### Eerste knoop

```

m      ← READ FROM netwerk
THIS node ← FIRST node IN nodes WITH m = nrnode
IF THIS node IS NONE
  WRITE ". "; "onbekende NODE";m WITH IMAGE a20aΔΔ*****
  THIS node ← NEW node
  nrnode ← m
  gelukt ← FALSE
END
JOIN THIS node TO route_heen
JOIN THIS node TO route_terug IN FRONT POSITION
THIS mijlpaal ← NEW mijlpaal @ node
m_lengte      ← 1
m_kop         ← nrnode
JOIN THIS mijlpaal TO palen_heen
m ← READ FROM netwerk

```

#### Vervolg knopen

```

WHILE m > 0
  refnode ← THIS node
  THIS node ← FIRST node IN nodes WITH m = nrnode
  IF THIS node IS NONE
    WRITE ". "; "onbekende NODE";m WITH IMAGE a20aΔΔ*****
    THIS node ← NEW node
    nrnode ← m
    gelukt ← FALSE
  END
  JOIN THIS node TO route_heen
  JOIN THIS node TO route_terug IN FRONT POSITION
  Ga na of zich een waterweg tussen deze knopen
  bevindt
  plek ← FIRST vaarweg IN vaarwegen
        WITH (raai[1] IS refnode & raai[2] IS THIS node)
        | raai[1] IS THIS node & raai[2] IS refnode

```

```

GOTO found IF plek IS NOT NONE
plek ← FIRST sluis IN sluizen
      WITH (kant[1] IS refnode & kant[2] IS THIS node)
           | kant[1] IS THIS node & kant[2] IS refnode
GOTO found IF plek IS NOT NONE
plek ← FIRST brug IN bruggen
      WITH (zijde[1] IS refnode & zijde[2] IS THIS node)
           | zijde[1] IS THIS node & zijde[2] IS refnode
GOTO found IF plek IS NOT NONE
plek ← FIRST vernauwing IN vernauwingen
      WITH (vzijde[1] IS refnode & vzijde[2] IS THIS node)
           | vzijde[1] IS THIS node & vzijde[2] IS refnode
IF plek IS NONE
  WRITE "• "; "Er is geen WATERWEG tussen de knopen"
      ;nrnode OF refnode;"en";nrnode WITH IMAGE a20aΔ***ΔaΔ***
  gelukt ← FALSE
  GOTO volgende
END
found: JOIN plek TO weg_heen
      JOIN plek TO weg_terug IN FRONT POSITION

@-----@
@ mijlpalen voor registratie @
@-----@
IF plek BELONGS NOT TO vaarwegen
  THIS mijlpaal ← NEW mijlpaal
  m_type ← 3 @ vernauwing
  m_lengte ← 2
  IF plek BELONGS TO sluizen
    m_type ← 1
    m_lengte ← 5
  END
  IF plek BELONGS TO bruggen
    m_type ← 2
    m_lengte ← 3
  END
  JOIN THIS mijlpaal TO palen_heen
END
THIS mijlpaal ← NEW mijlpaal @ node
m_lengte ← 1
m_kop ← nrnode
JOIN THIS mijlpaal TO palen_heen
volgende: m ← READ FROM netwerk
END

                                Zet generator aan het werk
ACTIVATE THIS generator FROM genese IN fysiek
JOIN      THIS generator TO generatoren
GOTO volg_gen IF gelukt = FALSE

                                Initialiseren klasse informatie met betrekking tot
                                sluizen en bruggen
THIS sluis ← FIRST sluis IN weg_heen
THIS brug ← FIRST brug IN weg_heen

CALL locate_matfile(<hschip>,2)
CALL chread_matfile(<hschip>,1,<regel>,30)
FOR n ← 1 TO 30
  CALL char_in_string(<char>,1,1,regel,n)
  klas_bij_sluis[n] ← TRUE IF THIS sluis IS NOT NONE & char = "√"
  klas_bij_brug[n] ← TRUE IF THIS brug IS NOT NONE & char = "√"
END

@-----@
@ Tijdregistratie @
@-----@
n ← 1

```

```

FOR EACH mijlpaal IN palen_heen
  m_index ← n
  n ← n + m_lengte
  JOIN THIS mijlpaal TO palen_terug IN FRONT POSITION
END
regellengte ← MAX(4,n-1)

tijdregistratie:
  bladen ← NEW SET CALLED "BLADEN."|nrngen
  regels_per_bladzij ← FLOOR(10000/regellengte)
  tijdrecords ← 2
  CALL num_in_string(<dospad>,string_length(dospad),1,nrngen,0)
  CALL erase_file(dospad) IF exist_file(dospad) = TRUE
  hist_fullname ← dospad
  k ← create_matfile(<htijd>,dospad,2*regellengte)
  IF k ≠ 1
    WRITE ". "; "fout bij het maken van:" WITH IMAGE a20a
    WRITE dospad;k WITH IMAGE a77Δ**
    gelukt ← FALSE
  END
  IF THIS generator IS NOT schutregistrator
    CALL locate_matfile(<hschip>,1)
    FOR k ← 1 TO 6
      tschip[k] ← read_matfile(<hschip>,k)
    END
    CALL close_matfile(<hschip>)
  END
  CALL num_in_string(<qfilenaam>,string_length(qfilenaam),1,nrngen,0)
  history_filenaam ← qfilenaam

  CALL wegschrijven_bladen(FALSE) @ maak nieuwe BLADZIJ
  regels ← 1
  volle_regels ← 1
  GOTO eruit IF schutregistrator IS NOT NONE

  FOR EACH mijlpaal IN palen_heen
    bladinhoud[m_index] ← m_kop IF 0 < m_kop
  END

```

```

volg_gen:
  filenaam ← CHREAD FROM netwerk
  CALL case_string(<filenaam>,1,string_length(filenaam),1)
END

```

In model SUBMOD2 is SCHUTREGISTRATOR hier altijd NONE.

```

IF schutregistrator IS NONE & sluizen IS NOT EMPTY
  schutregistrator ← NEW generator
  THIS generator ← schutregistrator
  regellengte ← 8
  REPEAT FROM tijdregistratie
END

```

eruit:

Voeg generator informatie toe aan query\_invoer

```

IF schutregistrator IS NOT NONE
  WRITE history_filenaam OF schutregistrator TO query_invoer
  WITH IMAGE Δa45
  FOR k ← 1 TO 3
    CALL write_matfile(<htijd>,k,date[k])
    CALL write_matfile(<htijd>,k+3,time[k])
  END
  CALL close_matfile(<htijd>)
END
FOR EACH generator IN generatoren
  WRITE route_filenaam TO query_invoer WITH IMAGE a
  WRITE tschip[1] TO query_invoer WITH IMAGE |Δ****

```

```
FOR k ← 2 TO 6
    WRITE tship[k]    TO query_invoer WITH IMAGE |Δ**
END
WRITE history_filenaam TO query_invoer WITH IMAGE |ΔΔa12
FOR k ← 1 TO 3
    CALL write_matfile(<htijd> ,k,date[k])
    CALL write_matfile(<htijd> ,k+3,time[k])
END
CALL close_matfile(<htijd>)
END
WRITE " end";"" TO query_invoer WITH IMAGE aa24
WRITE ""       TO query_invoer WITH IMAGE a

RETURN 1 IF gelukt
RETURN -1
```



## 6.29 LEES\_GETIJDEN

Deze MACRO verzorgt het inlezen van de getijdetabellen.

Terugkeerwaarde:

1 Ok  
-1 Er is iets mis

```
@=====
@
@ LEES_GETIJDEN
@
@=====
```

```
THIS sluis ← FIRST sluis IN sluizen
                WITH (0 < s_nrtij[1]) | 0 < s_nrtij[2]
THIS brug ← FIRST brug IN bruggen WITH 0 < b_nrtij
                Is het nodig deze tabellen in te lezen ?
RETURN 1 IF THIS sluis IS NONE & THIS brug IS NONE
                Ja
WRITE "Getijden" WITH IMAGE ΔΔΔΔΔΔΔΔΔΔA
data ← bibgetij

l ← 0
                Tabel index
n ← READ FROM data
                Tabel nummer
WHILE 0 < n
    m ← 0
    FOR EACH brug IN bruggen WITH b_nrtij = n
        Vervang tabelnummer door verwijzing naar
        volgende tabel index
        b_nrtij ← 0 - (l+1)
        m ← 1
    END
    FOR k ← 1 TO 2
        FOR EACH sluis IN sluizen WITH s_nrtij[k] = n
            Vervang tabelnummer door verwijzing naar
            volgende tabel index
            s_nrtij[k] ← 0 - (l+1)
            m ← 1
        END
    END
    CALL hulpje
                Lees tabel voor springtij
    IF m=1
        l ← l + 1
        FOR EACH punt IN hulp
            TABULATE yy IN springtij[1] AT xx
        END
    END
    CALL hulpje
                Lees tabel voor dood tij
    IF m=1
        FOR EACH punt IN hulp
            TABULATE yy IN doodtij[1] AT xx
        END
    END
    n ← READ FROM data
END
m ← 0
```

Test of alle tabellen waarnaar wordt verwezen ook zijn ingelezen



```

FOR EACH brug IN bruggen WITH 0 < b_nrtij
  WRITE "• ";nrnode OF zijde[1];"brug";nrnode OF zijde[2]" :";
  "verwijst naar onbekende GETIJ TABEL";b_nrtij
  WITH IMAGE a20***ΔaΔaΔaΔ***
  m ← 1
END
FOR EACH sluis IN sluizen WITH (0 < s_nrtij[1]) | (0 < s_nrtij[2])
  WRITE "• ";nrnode OF kant[1];"brug";nrnode OF kant[2]" :";
  "verwijst naar onbekende GETIJ TABEL"
  WITH IMAGE a20***ΔaΔaΔa
  FOR k ← 1 TO 2
    WRITE s_nrtij[k] WITH IMAGE |Δ*** IF 0 < s_nrtij[k]
    m ← 1
  END
END
IF m = 1
  RETURN -1
END

@-----@
@ Zet de tabelverwijzingen weer positief @
@-----@
FOR EACH brug IN bruggen
  b_nrtij ← 0 - b_nrtij
END
FOR EACH sluis IN sluizen
  s_nrtij[1] ← 0 - s_nrtij[1]
  s_nrtij[2] ← 0 - s_nrtij[2]
END
CLOSE bibgetij
RETURN 1

```

### 6.30 LEES\_KOSTEN

Deze MACRO verzorgt het inlezen van de kosten.

Terugkeerwaarde:

1     Ok  
-1    Er is iets mis

```

@=====
@
@ LEES_KOSTEN
@
@=====

@-----@
@ Inlezen kosten scheepvaartverkeer @
@-----@

WRITE "• "; "Inlezen kosten" WITH IMAGE a20a
FOR k ← 1 TO 6
    xxx[k] ← READ FROM bibkosten
END

x ← xxx[1]+xxx[2]
xxx[1] ← xxx[1] / x     @ Vaarkosten% sluis
xxx[2] ← xxx[2] / x     @ Ligkosten% sluis

x ← xxx[3]+xxx[4]
xxx[3] ← xxx[3] / x     @ Vaarkosten% brug
xxx[4] ← xxx[4] / x     @ Ligkosten% brug

x ← xxx[5]+xxx[6]
xxx[5] ← xxx[5] / x     @ Vaarkosten% vernauwing
xxx[6] ← xxx[6] / x     @ Ligkosten% vernauwing

FOR m ← 1 TO 30
    k ← READ FROM bibkosten
    FOR j ← 1 TO 3
        kosten[j,k] ← 0
    END
    FOR n ← 1 TO 2                     @ 1= motorvrachtschepen; 2= motortankchepen
        z ← READ FROM bibkosten / 100
        FOR l ← 1 TO 2                 @ 1= vaarkosten; 2= ligkosten
            x ← z * READ FROM bibkosten
            kosten[1,k] ← kosten[1,k] + xxx[1]     * x
            kosten[2,k] ← kosten[2,k] + xxx[2+1]   * x
            kosten[3,k] ← kosten[3,k] + xxx[4+1]   * x
        END
    END
END
CLOSE bibkosten
RETURN

```



```

      xtra_wachttijd      ← READ FROM bibschutregimes
      wacht_gewicht_cat3 ← MAX(0.01, READ FROM bibschutregimes)
      FOR n ← 1 TO 2
        voorkennistijd[n] ← READ FROM bibschutregimes
      END
      @-----@
      @ Inlezen schutperiodes @
      @-----@
      j ← READ FROM bibschutregimes
      schutperiodes ← NEW SET CALLED "speriod..."|sregimenr
      WHILE 0 < j
        x1 ← READ FROM bibschutregimes
        WHILE 0 ≤ x1
          THIS regime ← NEW regime
          param      ← j
          geldvan     ← FLOOR(x1) + (x1-FLOOR(x1)) / 0.6
          JOIN THIS regime TO schutperiodes
          x1 ← READ FROM bibschutregimes
        END
        j ← READ FROM bibschutregimes
      END
      GOTO volgendet1

sla_over1:
      @-----@
      @ sla SCHUTREGIME over @
      @-----@
      FOR n ← 1 TO 7
        char ← CHREAD FROM bibschutregimes
      END
      j ← READ FROM bibschutregimes
      WHILE 0 < j
        x ← READ FROM bibschutregimes
        WHILE 0 ≤ x
          x ← READ FROM bibschutregimes
        END
        j ← READ FROM bibschutregimes
      END
      volgendet1:
      END
      hulp ← NONE
      eruit1:
      CLOSE bibschutregimes

      @=====
      @ Inlezen OPENINGS-REGIMES voor BRUGGEN @
      @=====

      hulp ← NEW SET
      JOIN EACH regimeblok IN regimeblokken TO hulp
      WHILE hulp IS NOT EMPTY
        k ← READ FROM bibregimes
        IF k ≤ 0
          WRITE "• "; "De volgende regimes ontbreken in BIBREGIMES:"
          WITH IMAGE a20a
          FOR EACH regimeblok IN hulp
            FOR EACH brug IN bruggen WITH regimes IS r_inhoud
              WRITE " "; nrnode OF zijde[1]; "brug"; nrnode OF zijde[2] | "      : "
              ; bloknr WITH IMAGE a25***ΔaΔaΔ***
            END
          END
          gelukt ← FALSE
          GOTO eruit2
        END
        THIS regimeblok ← FIRST regimeblok IN hulp WITH bloknr = k
        GOTO sla_over2 IF THIS regimeblok IS NONE

```

```

@-----@
@ Lees een REGIME behorende bij THIS REGIMEBLOK in @
@-----@
REMOVE THIS regimeblok FROM hulp

@-----@
@ Inlezen BLOKTIJDEN @
@-----@
l ← READ FROM bibregimes
WHILE l > 0
  x1 ← READ FROM bibregimes
  WHILE x1 ≥ 0
    THIS regime ← NEW regime
    geldvan ← FLOOR(x1) + (x1-FLOOR(x1)) / 0.6
    x2 ← READ FROM bibregimes
    geldtot ← FLOOR(x2) + (x2-FLOOR(x2)) / 0.6
    ii ← FLOOR(geldvan)
    jj ← FLOOR(geldtot)
    refregime ← THIS regime
    x1 ← READ FROM bibregimes
    WHILE x1 ≥ 0
      x2 ← READ FROM bibregimes
      FOR j ← ii TO jj
        x ← j + x1/60
        y ← j + x2/60
        x ← geldvan OF refregime IF x < geldvan OF refregime
        y ← geldtot OF refregime IF y > geldtot OF refregime
        IF x ≤ y
          THIS regime ← NEW regime
          geldvan ← x + 24*l-1
          geldtot ← y + 24*l-1
          soort ← 1
          JOIN THIS regime TO r_inhoud RANKED BY geldvan*100
        END
      END
      x1 ← READ FROM bibregimes
    END
    x1 ← READ FROM bibregimes
  END
  l ← READ FROM bibregimes
END

@-----@
@ Inlezen brugopeningsduur-beperking @
@-----@
l ← READ FROM bibregimes
WHILE l > 0
  x1 ← READ FROM bibregimes
  WHILE x1 ≥ 0
    THIS regime ← NEW regime
    geldvan ← FLOOR(x1+24*l-1) + (x1-FLOOR(x1)) / 0.6
    x2 ← READ FROM bibregimes
    geldtot ← FLOOR(x2+24*l-1) + (x2-FLOOR(x2)) / 0.6
    param ← READ FROM bibregimes
    soort ← 2
    x1 ← READ FROM bibregimes
    JOIN THIS regime TO r_inhoud RANKED BY geldvan*100
  END
  l ← READ FROM bibregimes
END
GOTO volgende2

sla_over2:
@-----@
@ sla REGIMEBLOK over @

```



```

@-----@
@-----@
@ bloktijden @
@-----@
x ← READ FROM bibregimes
WHILE 0 < x
  x ← READ FROM bibregimes
  WHILE 0 ≤ x
    x ← READ FROM bibregimes
    x ← READ FROM bibregimes
    WHILE 0 ≤ x
      x ← READ FROM bibregimes
      x ← READ FROM bibregimes
    END
    x ← READ FROM bibregimes
  END
  x ← READ FROM bibregimes
END
@-----@
@ brugopeningsduur-beperking @
@-----@
x ← READ FROM bibregimes
WHILE 0 < x
  x ← READ FROM bibregimes
  WHILE 0 ≤ x
    x ← READ FROM bibregimes
    x ← READ FROM bibregimes
    x ← READ FROM bibregimes
  END
  x ← READ FROM bibregimes
END
volgende2:
  END
  hulp ← NONE
eruit2:
  CLOSE bibregimes

  RETURN -1 IF gelukt=FALSE
  RETURN 1

```

### 6.32 LEES\_SLUIZEN

Deze MACRO verzorgt het inlezen van de gegevens over de sluizen.

Terugkeerwaarde:

1      Ok

-1 Er is iets mis

```

@=====
@ LEES_SLUIZEN
@
@=====
@=====
@ Inlezen sluizen @
@=====
LOCAL:
  LOGICAL: gelukt
  INTEGER: nr
  REAL    : lv1 lv2 bv1 bv2 wv1 wv2

  schutregimes ← NEW SET
  gelukt       ← TRUE

  WRITE "Sluizen" WITH IMAGE ΔΔΔΔΔΔΔΔΔΔ

  kk      ← 0
  i[1]    ← READ FROM network
  WHILE 0 < i[1]
    i[2] ← READ FROM network
    CALL switch @ knoop met het laagste nummer wordt 'kant 1'
    WRITE i[1];i[2] TO query_invoer WITH IMAGE ***Δ***
    kk      ← kk + 1
    THIS sluis ← NEW sluis
    FOR k ← 1 TO 2
      kant[k] ← FIRST node IN nodes WITH nrnode = i[k]
      IF kant[k] IS NONE
        kant[k] ← NEW node CALLED "NODE....."|i[k]
        nrnode OF kant[k] ← i[k]
        JOIN kant[k] TO nodes
      END
    END
    WRITE ";i[1];sluis;i[2]|" " WITH IMAGE a13***ΔaΔa
    gelukt ← FALSE IF check_network(2) < 0
    JOIN THIS sluis TO sluizen
    nrsluis ← kk
    FOR k ← 1 TO 2
      s_nrtij[k] ← READ FROM network
      s_peilpar[k] ← READ FROM network
      IF s_nrtij[k] < 0
        WRITE "• "; "Illegaal GETIJDJE TABEL NR: "; s_nrtij[k]
        WITH IMAGE a20aΔ*****
        gelukt ← FALSE
      END
      wachtruimte[k] ← NEW SET WITH STATISTICS
        CALLED i[k] | "..SWACHT.." | i[3-k]
      meldingen OF kant[k] ← NEW SET CALLED "MELDING..." | i[k]
      voorkeur[k] ← NEW SET WITH STATISTICS CALLED "VOORKEUR..." | i[k]
      remanent[k] ← NEW SET WITH STATISTICS CALLED "REMANENT..." | i[k]
      onmogelijk[k] ← NEW SET WITH STATISTICS CALLED "ONMOGLYK..." | i[k]
    END
    strategie ← READ FROM network
    ch10 ← CHREAD FROM network @*** kolkvulmodule ***@
    CALL case string(<ch10>,1,10,1)

```

```

IF (ch10 ≠ "GEMENGD") & ch10 ≠ "GESCHEIDEN"
  WRITE ". "; "onbekende schutmethode:"; ch10 WITH IMAGE a20aΔa
  gelukt ← FALSE
END
gemengd_schutten ← ch10 = "GEMENGD"
n ← READ FROM netwerk @*** schutregime ***@
debietbepaling ← 0 ≤ n @***** schutwaterregistratie *****@
IF 1 ≤ n
  THIS schutregime ← FIRST schutregime IN schutregimes
  WITH sregimenr = n
  IF THIS schutregime IS NONE
    THIS schutregime ← NEW schutregime CALLED "sstrat...."|n
    sregimenr ← n
    JOIN THIS schutregime TO schutregimes RANKED BY sregimenr
  END
  IF gemengd_schutten = FALSE
    WRITE ". "; "Het gebruik van schutregimes" WITH IMAGE a20a
    WRITE ". "; "in combinatie met het schutten" WITH IMAGE a20a
    WRITE ". "; "met gescheiden categorieën is niet toegestaan!"
    WITH IMAGE a20a
    gelukt ← FALSE
  END
  FOR k ← 1 TO 30
    IF categorie[k] = 1
      WRITE ". "; "Bij gebruik van schutregimes" WITH IMAGE a20a
      WRITE ". "; "mag categorie 1 niet gedefinieerd zijn!"
      WITH IMAGE a20a
      gelukt ← FALSE
      k ← 30
    END
  END
  mn_schutregime ← THIS schutregime
  @ THIS toestand ← NEW toestand
  @ t_tijdstip ← NOW
  @ t_criterium ← onbepaald
  @ mijn_toestand ← THIS toestand
END
ch10 ← CHREAD FROM netwerk @*** wachttijd_onmogelijk ***@
CALL case_string(<ch10>, 1, 10, 1)
IF (ch10 ≠ "JA") & ch10 ≠ "NEE"
  WRITE ". "; "fout antwoord:"; ch10 WITH IMAGE a20aΔa
  WRITE ". "; "bij 'wachttijd fysieke beperking?'" WITH IMAGE a20a
  gelukt ← FALSE
END
wachttijd_onmogelijk ← ch10 = "JA"
schutplanning ← geen_schutstrategie

@-----@
@ Inlezen kolken @
@-----@
kolken ← NEW SET CALLED "kolken"
alle_kolken ← NEW SET CALLED "alle kolk"
vrije_kolken ← NEW SET CALLED "vrije kolk"
sschipstats ← NEW SET
n ← READ FROM netwerk
WHILE n > 0
  THIS kolk ← NEW kolk
  nrkolk ← n
  side ← 1 + n = 2 * FLOOR(n/2)
  zijnsluis ← THIS sluis
  schutruimte ← NEW SET WITH STATISTICS CALLED "SCHUTRUIMT"|nrkolk
  voorwaarde ← voorw_dummy
  hevel_factor ← 1
  FOR k ← 1 TO 2
    opstelruimte[k] ← NEW SET WITH STATISTICS
    CALLED k|.OPSTEL...|nrkolk

```

```

    opstelplan[k] ← NEW SET CALLED k|".PLAN....."|nrkolk
    kolk_stat[k] ← NEW kolkstat
END
bezig ← FALSE
brdhoofd ← READ FROM netwerk
fuiklengte[1] ← READ FROM netwerk
fuiklengte[2] ← READ FROM netwerk
brdtkolk ← READ FROM netwerk
lengkolk ← READ FROM netwerk
waterlengte ← READ FROM netwerk
drempel[1] ← READ FROM netwerk
drempel[2] ← READ FROM netwerk
topen[1] ← READ FROM netwerk
topen[2] ← READ FROM netwerk
nulniveel[1] ← READ FROM netwerk
nulniveel[2] ← READ FROM netwerk
facniveel[1] ← READ FROM netwerk
facniveel[2] ← READ FROM netwerk
tsluit[1] ← READ FROM netwerk
tsluit[2] ← READ FROM netwerk
opp_kolk ← brdtkolk * lengkolk

cat12 ← READ FROM netwerk
icat12 ← READ FROM netwerk
ucat12 ← READ FROM netwerk
cat13 ← READ FROM netwerk
icat13 ← READ FROM netwerk
ucat13 ← READ FROM netwerk
cat23 ← READ FROM netwerk
icat23 ← READ FROM netwerk
ucat23 ← READ FROM netwerk

k ← READ FROM netwerk
WHILE 0 < k
    geschikt[k] ← TRUE
    k ← READ FROM netwerk
END
n ← 0
k ← READ FROM netwerk
WHILE 0 < k
    garkolk[k] ← TRUE
    n ← 1 IF sgarklas[k] = TRUE
    k ← READ FROM netwerk
END
IF mn_schutregime IS NOT NONE & n=1
    WRITE "• "; "Het gebruik van schutregimes in combinatie"
        WITH IMAGE a20a
    WRITE "•"; "met de garantieregeling is niet toegestaan!"
        WITH IMAGE a20a
    gelukt ← FALSE
END
start_idle ← -1
THIS klok ← NEW klok
inhibitor ← THIS kolk
shift ← NEW SET
x ← READ FROM netwerk
WHILE x ≥ 0
    THIS werktijd ← NEW werktijd
    begin ← x
    einde ← READ FROM netwerk
    JOIN THIS werktijd TO shift RANKED BY begin
    x ← READ FROM netwerk
END
WRITE nrkolk;lengkolk;brdtkolk TO query_invoer
    WITH IMAGE |Δ**Δ****.*Δ****.*
JOIN THIS kolk TO alle_kolken

```



```

                                RANKED BY waterlengte * brdtkolk / 10
JOIN THIS kolk TO kolken RANKED BY waterlengte * brdtkolk / 10
ACTIVATE THIS klok FROM kolkklok IN manage_sluis
FOR n ← 1 TO 2
    CALL initdeel(n)
END
THIS schipstat ← NEW schipstat
s_volgnr      ← nrkolk
JOIN THIS schipstat TO sschipsstats RANKED BY s_volgnr
ACTIVATE THIS kolk FROM beschikbaar IN kolkmod
n ← READ FROM netwerk
END
@-----@
@ kolk voorwaarden @
@-----@
WRITE "kolk voorwaarden:" WITH IMAGE ΔΔΔΔΔΔΔΔΔΔΔΔ
ch30 ← CHREAD FROM netwerk
CALL case_string(<ch30>,1,30,1)
WHILE ch30 ≠ "END"
    WRITE ch30 WITH IMAGE ΔΔΔΔΔΔΔΔΔΔΔΔ
    nr ← READ FROM netwerk
    WHILE nr > 0
        THIS kolk ← FIRST kolk IN kolken WITH nrkolk = nr
        IF THIS kolk IS NONE
            WRITE "er is geen kolk met nummer";nr WITH IMAGE aΔ**
            WHILE ch30 ≠ "END"
                ch30 ← CHREAD FROM netwerk
                CALL case_string(<ch30>,1,30,1)
            END
            gelukt ← FALSE
            GOTO voorw_klaar
        END
        voorwaarde ← ch30
        nr ← READ FROM netwerk
    END
    CALL char_in_string(<ch30>,1+string_length(ch30),5,"_INIT",1)
    voorw_init ← ch30
    gelukt ← FALSE IF voorw_init = -1
    ch30 ← CHREAD FROM netwerk
    CALL case_string(<ch30>,1,30,1)
END
voorw_klaar:
IF mn_schutregime IS NOT NONE
    lv2 ← 1000
    bv2 ← 1000
    wv2 ← 1000
    FOR EACH kolk IN kolken
        lv1 ← MAX(lv1,lengkolk)
        bv1 ← MAX(bv1,brdtkolk)
        wv1 ← MAX(wv1,waterlengte)
        lv2 ← MIN(lv2,lengkolk)
        bv2 ← MIN(bv2,brdtkolk)
        wv2 ← MIN(wv2,waterlengte)
    END
    schutplanning ← schutstrategie_gelyk
    schutplanning ← schutstrategie_nkolk IF (lv1 - lv2) ≥ 0.1*lv1
    schutplanning ← schutstrategie_nkolk IF (bv1 - bv2) ≥ 0.1*bv1
    schutplanning ← schutstrategie_nkolk IF (wv1 - wv2) ≥ 0.1*wv1
    strategie ← 2 IF schutplanning IS schutstrategie_gelyk
END
WRITE -1 TO query_invoer WITH IMAGE |Δ**
i[1] ← READ FROM netwerk
END

RETURN 1 IF gelukt
RETURN -1

```



### 6.33 LEES\_VAARKROMMEN

Deze MACRO verzorgt het inlezen van de vaarkrommen.

Terugkeerwaarde:

1     Ok  
-1    Er is iets mis

```

@=====
@
@ LEES_VAARKROMMEN
@
@=====

@=====
@ Inlezen vaartijd-krommen bij BRUGGEN en SLUIZEN @
@=====
LOCAL:
  LOGICAL: gelukt

  gelukt ← TRUE

  Zijn er vaarkrommen nodig ?
  RETURN 1 IF bruggen IS EMPTY & sluizen IS EMPTY & vernauwingen IS EMPTY

  Ja
  WRITE "Vaarkrommen" WITH IMAGE ΔΔΔΔΔΔΔΔΔΔΔΔ
  data ← bibvaart

  @-----@
  @ volgafstanden in VERNAUWINGEN @
  @-----@
  FOR j ← 1 TO 3
    FOR k ← 1 TO 3
      volglengte[j,k] ← READ FROM data
    END
  END

  @-----@
  @ Am/Ac, Bs/Bc krommen voor SLUIS @
  @-----@

  IF sluizen IS EMPTY
    WHILE READ FROM data ≥ 0
      WHILE READ FROM data ≥ 0
        END
      END
    FOR k ← 1 TO 2
      WHILE READ FROM data ≥ 0
        END
      END
    GOTO lees_bruggen
  END

  DISPLAY "sluis" AT LINE 20 POSITION 1 WITH IMAGE a
  FOR k ← 1 TO 30
    categorie[k] ← 0-categorie[k] IF klas_bij_sluis[k]
  END
  k ← READ FROM data
  WHILE 0 < k
    DISPLAY k AT LINE 20 POSITION 10 WITH IMAGE **
    IF klas_bij_sluis[k]=FALSE
      FOR j ← 1 TO 8
        WHILE READ FROM data ≥ 0
          END

```

```

        END
        GOTO volgende_klas
    END
    categorie[k] ← ABS(categorie[k])
    @**** invaartijd - 1ste schip - ongeladen/geladen *****
    FOR j ← 1 TO 2
        CALL hulpje
        FOR EACH punt IN hulp
            TABULATE yy IN tijdin_1ste[k,j] AT xx
        END
    END
    @**** invaartijd - volg schip - ongeladen/geladen *****
    FOR j ← 1 TO 2
        CALL hulpje
        FOR EACH punt IN hulp
            TABULATE yy IN tijdin_volg[k,j] AT xx
        END
    END
    @*** uitvaartijd - 1ste schip - ongeladen/geladen *****
    FOR j ← 1 TO 2
        CALL hulpje
        FOR EACH punt IN hulp
            TABULATE yy IN tijduit_1ste[k,j] AT xx
        END
    END
    @*** uitvaartijd - volg schip - ongeladen/geladen *****
    FOR j ← 1 TO 2
        CALL hulpje
        FOR EACH punt IN hulp
            TABULATE yy IN tijduit_volg[k,j] AT xx
        END
    END
    volgende_klas:
    k ← READ FROM data
    END

    @-----@
    @ LUS-krommen bij SLUIZEN @
    @-----@
    FOR k ← 1 TO 2
        CALL hulpje
        FOR EACH punt IN hulp
            TABULATE yy IN lustijd[k] AT xx
        END
    END

    j ← 0
    FOR k ← 1 TO 30
        j ← j + 1 IF categorie[k] < 0
    END
    IF 0 < j
        WRITE ". "; "voor scheepsklasse" WITH IMAGE a20A
        WRITE "n" WITH IMAGE |A IF j > 1
        FOR k ← 1 TO 30
            IF categorie[k] < 0
                WRITE k WITH IMAGE |Δ**
                categorie[k] ← ABS(categorie[k])
            END
        END
        WRITE "" ; "zijn geen sluisvaartijden" WITH IMAGE a20ΔA
        WRITE "gespecificeerd in BIBVTD!" WITH IMAGE |ΔA
        gelukt ← FALSE
    END

    @-----@
    @ Doorvaartijden bij BRUGGEN @

```

```
        categorie[k] ← ABS(categorie[k])
    END
END
WRITE "";"zijn geen brugvaartijden" WITH IMAGE a20A
WRITE "gespecificeerd in BIBVTD!" WITH IMAGE |ΔA
gelukt ← FALSE
END

einde:
    CLOSE bibvaart
    RETURN 1 IF gelukt
    RETURN -1
```

### 6.34 LEES\_VAARWEGEN

Deze MACRO verzorgt het inlezen van de vaarwegen.

Terugkeerwaarde:

1    Ok  
-1   Er is iets mis

```
@=====
@
@ LEES_VAARWEGEN
@
@=====
```

```
@=====
@ Inlezen vaarwegen @
@=====
```

LOCAL:

LOGICAL: gelukt

```
gelukt ← TRUE
WRITE "Vaarwegen" WITH IMAGE ΔΔΔΔΔΔΔΔΔΔΔΔ
i[1] ← READ FROM netwerk
WHILE 0 < i[1]
  i[2] ← READ FROM netwerk
  CALL switch
  THIS vaarweg ← NEW vaarweg
  FOR k ← 1 TO 2
    raai[k] ← FIRST node IN nodes WITH nrnode = i[k]
    IF raai[k] IS NONE
      raai[k] ← NEW node CALLED "NODE....."|i[k]
      nrnode OF raai[k] ← i[k]
    JOIN raai[k] TO nodes
  END
END
gelukt ← FALSE IF check_netwerk(1) < 0
JOIN THIS vaarweg TO vaarwegen
vaarvak ← NEW SET CALLED i[1]|"...VAAR..."|i[2]
afstand ← READ FROM netwerk
i[1] ← READ FROM netwerk
END

RETURN 1 IF gelukt
RETURN -1
```

### 6.35 LEES\_VERNAUWINGEN

Deze MACRO verzorgt het inlezen van de vernauwingen.

Terugkeerwaarde:

```
1    Ok
-1   Er is iets mis
```

```
@=====
@
@ LEES_VERNAUWINGEN
@
@=====

@=====
@ Inlezen vernauwingen @
@=====

LOCAL:
    LOGICAL: gelukt default

                                                    Sluit sectie in query_invoer af
WRITE -1 TO query_invoer WITH IMAGE **                @** einde bruggen ***

WRITE "Vernauwingen" WITH IMAGE ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ
gelukt ← TRUE

kk ← 0
i[1] ← READ FROM netwerk
WHILE i[1] > 0
    i[2] ← READ FROM netwerk
    CALL switch
    WRITE i[1];i[2] TO query_invoer WITH IMAGE ***Δ***
    THIS vernauwing ← NEW vernauwing
    FOR k ← 1 TO 2
        vzijde[k] ← FIRST node IN nodes WITH nrnode = i[k]
        IF vzijde[k] IS NONE
            vzijde[k] ← NEW node CALLED "NODE....."|i[k]
            nrnode OF vzijde[k] ← i[k]
        JOIN vzijde[k] TO nodes
    END
END
kk ← kk + 1
nrvernauwing ← kk
WRITE "";i[1];"vernauwing";i[2]|" " WITH IMAGE a13***ΔaΔa
gelukt ← FALSE IF check_netwerk(4) < 0
JOIN THIS vernauwing TO vernauwingen
v_afstand      ← READ FROM netwerk
regeling       ← CHREAD FROM netwerk
engte          ← NEW SET CALLED i[1]|" engte "|i[2]
garantieschepen ← NEW SET CALLED "V_GARANTIE"
CALL case_string(<regeling>,1,3,1)
GOTO fout IF (regeling ≠ "VL") & regeling ≠ "TAB"
WRITE #34;regeling TO query_invoer WITH IMAGE |Δaa
FOR j ← 1 TO 30
    vvaarsnelheid[j,1] ← READ FROM netwerk
    vvaarsnelheid[j,2] ← READ FROM netwerk
    plustijd[j]        ← READ FROM netwerk
END

maxgroen[1]      ← READ FROM netwerk
maxgroen[2]      ← READ FROM netwerk
ontruimingstijd ← READ FROM netwerk
chl0             ← CHREAD FROM netwerk @ invaarvolgorde
default          ← READ FROM netwerk=FALSE
```



```

FOR j ← 1 TO 30
  FOR k ← j TO 30
    ontmoeting[j,k] ← default
  END
END
j ← READ FROM netwerk
WHILE 0 < j
  k ← READ FROM netwerk
  WHILE 0 < k
    ontmoeting[j,k] ← default=FALSE
    ontmoeting[k,j] ← ontmoeting[j,k]
    k ← READ FROM netwerk
  END
  j ← READ FROM netwerk
END
voorrangskant ← READ FROM netwerk
IF regeling = "VL"
  CALL case_string(<ch10>,1,10,1)
  GOTO fout2 IF (ch10 ≠ "FIFO") & ch10 ≠ "SE"
  snelste_eerst ← ch10 = "SE"
  vlicht[1] ← rood
  vlicht[2] ← rood
  vernauwingskant ← 1
  star ← ontruimingstijd > 0
  WRITE "STAR" TO query_invoer WITH IMAGE |a IF star
  WRITE "AFH" TO query_invoer WITH IMAGE |a IF star=FALSE
  WRITE ch10 TO query_invoer WITH IMAGE |ΔA4
  ACTIVATE THIS vernauwing
    FROM starre_lichten IN manage_vernauwing IF star
  ACTIVATE THIS vernauwing FROM verkeersafhankelijk
    IN manage_vernauwing IF star=FALSE
END
FOR k ← 1 TO 30
  IF vgarklas[k] = 1
    IF regeling = "TAB"
      WRITE "• "; "Zonder verkeerslichten is er geen"
        WITH IMAGE a20a
      WRITE "• "; "gegarandeerde doorvaart!" WITH IMAGE a20a
      gelukt ← FALSE
      k ← 30
    END
    FOR j ← 1 TO 2
      meldingen OF vzijde[j] ← NEW SET CALLED "MELDING..." |i[j]
    END
  END
END
IF regeling = "TAB"
  snelste_eerst ← FALSE
  vlicht[1] ← oranje
  vlicht[2] ← oranje
END
FOR k ← 1 TO 2
  vwachtrij[k] ← NEW SET WITH STATISTICS
    CALLED i[k] | "..VWACHT.." |i[3-k]
  invaren_vrij[k] ← TRUE
END
vschipstats ← NEW SET
FOR k ← 1 TO 2
  THIS schipstat ← NEW schipstat
  s_volgnr ← k
  JOIN THIS schipstat TO vschipstats
END
WRITE #34 TO query_invoer WITH IMAGE |*
i[1] ← READ FROM netwerk
END

```

```
RETURN 1 IF gelukt
RETURN -1
```

fout:

```
WRITE "• "; "Deze vernauwing heeft een onbekende regeling "; regeling; ""
      WITH IMAGE a20aa*
WRITE   "" ; "De mogelijkheden zijn 'VL' en 'TAB'" WITH IMAGE a20a
RETURN -1
```

fout2:

```
WRITE "• "; "Deze vernauwing heeft een onbekende invaarvolgorde"; chl0
      WITH IMAGE a20aΔa4
WRITE   "" ; "de mogelijkheden zijn 'FIFO' (First In First Out)"
      WITH IMAGE a20a
WRITE   "" ; "                en 'SE'   (Snelste Eerst)" WITH IMAGE a20a
RETURN -1
```

### 6.36 LEES\_WEGINTENS

Deze MACRO verzorgt het inlezen van de intensiteiten van het wegverkeer.

Terugkeerwaarde:

1    Ok  
-1   Er is iets mis

```
@=====
@
@ LEES_WEGINTENS
@
@=====

RETURN 1 IF bruggen IS EMPTY
WRITE "Wegintens" WITH IMAGE ΔΔΔΔΔΔΔΔΔΔ

data ← bibintw
hulp2 ← NEW SET
JOIN EACH ipunt IN a_intens TO hulp2

k ← 0                               @ Tabel nummers @
WHILE hulp2 IS NOT EMPTY
  l ← READ FROM bibintw
  GOTO fout IF l < 0

  THIS ipunt ← FIRST ipunt IN hulp2 WITH nn1 = 1
  IF THIS ipunt IS NONE               @-----@
    x ← READ FROM bibintw            @ sla tabel over @
    WHILE x ≥ 0                      @-----@
      y ← READ FROM bibintw
      x ← READ FROM bibintw
    END
  END
  IF THIS ipunt IS NOT NONE
    k ← k + 1
    CALL hulpje
    hulp3[k] ← hulp IF nn2 < 0
    nn2 ← k
    @-----@
    @ Normeren van de tabel @
    @-----@
    tot ← (xx OF FIRST OF hulp * yy OF FIRST OF hulp) / 2
    x1 ← 0
    x2 ← 0
    FOR EACH punt IN hulp
      tot ← tot + (xx-x1)*.5*yy+x2
      x1 ← xx
      x2 ← yy
    END
    tot ← tot + (168-xx)*yy IF xx < 168
    FOR EACH punt IN hulp
      yy ← yy / tot IF tot > 0
      TABULATE yy IN intens[k] AT xx
    END
    REMOVE THIS ipunt FROM hulp2
  END
END
END

@-----@
@ Verbindt de stromen van de bruggen met de tabellen @
@-----@
FOR EACH brug IN bruggen
  FOR EACH stroom IN wegverkeer
```

```
        THIS ipunt ← FIRST ipunt IN a_intens WITH nn1 = nrintens
        tabnr ← nn2
    END
END
hulp2 ← NONE
a_intens ← NONE
FOR k ← 1 TO 10
    hulp3[k] ← NONE
END

CLOSE bibintw
RETURN 1

fout:
WRITE "• "; "Intensiteit patronen:" WITH IMAGE a20a
FOR EACH ipunt IN hulp2
    WRITE nn1 WITH IMAGE |Δ***
END
WRITE "ontbreken in WEGINT" WITH IMAGE |Δa
RETURN -1
```

### 6.37 MAINMOD

Dit is het hoofd module waarin het proces van MAIN wordt beschreven

```

@=====
@
@ MAINMOD
@
@=====

@=====
@ Proces van MAIN @
@=====

SPECIFY inhoud          PRECEPT(inhoud' ← afgeleide)
SPECIFY cum_wachttijd PRECEPT(cum_wachttijd' ← inhoud)
SPECIFY totaal          PRECEPT(totaal' ← weektot*VALUE OF intens[tabnr]
                        AT(CT-168.0*FLOOR(CT/168)))

                                Initialisatie

GOTO einde IF initialisatie=FALSE

                                Gelukt

simuleer:
  WAIT simulatieduur DAYS      @*** laat OBSERVATOR z'n uurtje bereiken ***
  WAIT UNTIL (NOW + 1 SECOND) < EVENTTIME OF observator

afsluiten:
  @-----@
  @ AFSLUITEN VAN DE SIMULATIE @
  @-----@

  @-----@
  @ RAPPORT @
  @-----@
  CALL rapport_initia
  CALL rapport_sluizen
  CALL rapport_bruggen
  CALL rapport_vernauwingen
  regeloph ← -1000
  CALL kop
  CALL clear_text_screen

  @-----@
  @ model QUERY @
  @-----@
  FOR EACH sluis IN sluizen
    FOR EACH kolk IN alle_kolken
      THAT BELONGS NOT TO kolken & schuttende = FALSE
      CALL sluis_registratie(FALSE)
    END
  END

  JOIN schutregistrator TO generatoren IF schutregistrator IS NOT NONE
  FOR EACH generator IN generatoren
    CALL wegschrijven_bladen(TRUE)
  END

einde:
  @-----@
  @ simulatieduur voor model QUERY @
  @-----@
  WRITE NOW TO query_invoer WITH IMAGE *****.*
  CLOSE query_invoer
  CALL resultaat_regel

```



```
FOR k ← 1 TO 4
    date[k] ← time[k]
END
CALL get_time(<time>)
x ← (time[2]-date[2]) MINUTES + (time[3]-date[3]) SECONDS
x ← 60* x MODULO(1)
y ← 60*(x MODULO(1))
z ← 100*(y MODULO(1))
DISPLAY "duur:";FLOOR(x);":00"|FLOOR(y);":00"|FLOOR(z)
        AT LINE 20 POSITION 1 WITH IMAGE aΔ**aa
CANCEL ALL
TERMINATE
```

### 6.38 *MANAGE\_BRUG*

Dit MODULE bevat de procesbeschrijvingen van de BRUGWACHTER, zijn KLOK en zijn REGELAARS.

```

@=====
@
@ MANAGE_BRUG
@
@=====

@-----@
@ Directoraat BRUGGEN @
@-----@

@=====
@ Procesbeschrijving van de KLOK van de BRUG @
@=====

brugklok:
  werkt OF inhibitor ← FALSE
  werk ← FIRST OF shift
  TERMINATE IF werk IS NONE
  x ← begin OF werk - NOW
  WAIT x IF x > 0
  werkt OF inhibitor ← TRUE
  x ← einde OF werk - NOW
  WAIT x IF x > 0
  REMOVE werk FROM shift
  begin OF werk ← begin OF werk + 168
  einde OF werk ← einde OF werk + 168
  JOIN werk TO shift
  REPEAT FROM brugklok

@=====
@ Procesbeschrijving van de BRUGWACHTER @
@=====

wacht:
  WAIT WHILE brugplan[1] IS EMPTY & brugplan[2] IS EMPTY & werkt
  IF werkt = FALSE
    Wacht zolang er niets is te doen

    WAIT UNTIL werkt
    Brug is niet in bedrijf

    Haal rustperiode in
    FOR j ← 1 TO 2
      FOR EACH schip IN wachtrij[j] OF zijnbrug
        ref[1] ← vorige_mijlpaal
        vorige_mijlpaal ← PRED OF ref[1] IN mpalen
        CALL query_registratie(passeer_knoop)
        vorige_mijlpaal ← ref[1]
        bladinhoud[bladstart+m_index OF vorige_mijlpaal+
          start_overliggen] OF blad ← 0
        tw ← NOW
        wachtend ← TRUE
      END
    END
    REPEAT FROM wacht
  END

  Nu gaan we aan de slag
  x ← 1.0E8
  THIS schip ← FIRST schip IN brugplan[1] WITH SMALLEST eta
  x ← eta IF THIS schip IS NOT NONE
  y ← 1.0E8
  THIS schip ← FIRST schip IN brugplan[2] WITH SMALLEST eta
  y ← eta IF THIS schip IS NOT NONE

```

```

brugkant ← 1 + x > y
THIS schip ← FIRST schip IN brugplan[brugkant] WITH SMALLEST eta
x ← eta - (NOW + tbopen OF zijnbrug + 1 SECOND)
WAIT x IF x > 0

```

## Te behandelen schip

```

THIS schip ← FIRST schip IN brugplan[brugkant] WITH SMALLEST eta
x ← brugregime
IF x > 0

```

## Moeten wachten

```

    WAIT x
    REPEAT FROM wacht
END

```

```

@-----@
@ OPENEN BRUG @
@-----@
licht      OF zijnbrug ← "ROOD"
bopen     OF zijnbrug ← NOW
vrijverkeer OF zijnbrug ← 0

```

## Laat files ontstaan

```

FOR EACH stroom IN wegverkeer OF zijnbrug
    cum_wachttijd ← 0
    inhoud ← 0
    inhoud' ← 0
    totaal ← 0
    ACTIVATE THIS stroom FROM stopop IN fysiek
END

```

## Open de brug

```

WORK tbopen OF zijnbrug

THIS brug ← zijnbrug
open      ← TRUE
FOR j ← 1 TO 2
    lastboot OF brughulp[j] ← NONE
END
nbopen ← nbopen + 1

```

## Zet de regelaars aan het werk om de wachtende schepen te laten passeren

```

IF type OF zijnbrug = "DRAAI"
    j ← 3-brugkant
    FOR j ← 1 TO 2
        case OF brughulp[j] ← 1
        number OF brughulp[j] ← 0
        ACTIVATE brughulp[j] FROM regelmaar
    END
    WAIT

    WAIT

    REPEAT FROM sluiten
END
case OF brughulp[brugkant] ← 1
number OF brughulp[brugkant] ← 0
ACTIVATE brughulp[brugkant] FROM regelmaar
WAIT

brugkant ← 3 - brugkant
case OF brughulp[brugkant] ← 3
number OF brughulp[brugkant] ← 0
ACTIVATE brughulp[brugkant] FROM regelmaar
WAIT

```

sluiten:

Nu moet de brug weer dicht

```
open OF zijnbrug ← FALSE
WORK tbsluit OF zijnbrug
```

```
THIS brug ← zijnbrug
licht      ← "GROEN"
```

Laat de schepen die dat kunnen en mogen onder de brug door varen

```
FOR j ← 1 TO 2
  FOR EACH schip IN wachtrij[j] WITH onderdoor
    REMOVE THIS schip FROM wachtrij[j]
    REMOVE THIS schip FROM brugplan[j]
    ACTIVATE THIS schip FROM onderbrugdoor IN schipmod
  END
  FOR EACH schip IN wachtrij[j] WITH wachtend
    tw      ← NOW - tw
    tl      ← NOW
    wachtend ← FALSE
  END
END
bopen ← NOW - bopen
oplos ← NOW
```

Wacht totdat de files zijn opgelost

```
WAIT UNTIL vrijverkeer OF zijnbrug = LENGTH OF wegverkeer OF zijnbrug
```

```
THIS brug ← zijnbrug
oplos     ← NOW - oplos
CALL brug_registratie
```

En dan maar weer wachten

```
REPEAT FROM wacht
```

```
@=====
@ Procesbeschrijving van de REGELAAR @
@=====
regelmaar:
```

```
  THIS brug ← bridge
```

Eerste schip

```
nxtboot ← FIRST schip IN brugplan[bside] OF watcher WITH SMALLEST eta
IF nxtboot IS NONE & type OF bridge ≠ "DRAAI"
```

Niets aan deze kant en draaibrug

```
  REACTIVATE watcher AFTER THIS
  TERMINATE
END
IF nxtboot IS NONE
```

Niets aan deze kant

```
  case ← 10 + LENGTH OF brugplan[bside] OF watcher
  Wacht tot andere regelaar is uitgeregeld
  WAIT WHILE (case = 10 + LENGTH OF brugplan[bside] OF watcher) &
    case OF brughulp[3-bside] OF watcher ≤ 4
```

```
  IF case ≠ 10+LENGTH OF brugplan[bside] OF watcher
    Inmiddels nog een schip van deze kant; dat kan
    dus in lopende vaart
```

```
    case ← 4
    REPEAT FROM regelmaar
  END
```

Uitgeregeld

```
  REACTIVATE watcher AFTER THIS
  TERMINATE
END
```

Nu was er een schip aan deze kant

```
@-----
@ Bepaal op het SCHIP nog mee kan ivm REGIMES @
@-----
```

```

THIS schip ← nxtboot
THIS brug ← bridge
laatsteboot ← lastboot
z ← NOW
IF NOW < eta @ NEXTBOOT komt in LOPENDE VAART @
  z ← eta
  case ← 4
END
z ← z + tonderdoor(case) + tbsluit OF bridge
IF z > deadline OF watcher
  case ← 10
  REACTIVATE watcher AFTER THIS
  TERMINATE
END

```

Schip kan niet meer mee

Wacht to het schip echt is aangekomen

```

WAIT UNTIL nxtboot BELONGS TO wachtrij[bside] OF bridge

```

Laat het schip de brug passeren

```

REMOVE nxtboot FROM brugplan[bside] OF watcher
REMOVE nxtboot FROM wachtrij[bside] OF bridge
THIS schip ← nxtboot
THIS brug ← bridge
laatsteboot ← lastboot
tw ← NOW - tw IF wachtend
tl ← 0 IF wachtend
tl ← NOW - tl IF wachtend = FALSE
number ← number + 1
ts ← tonderdoor(case)
CALL query_registratie(0-doorgang)
CALL query_registratie(start_doorvaren)
WORK ts @*** help THIS SCHIP er onderdoor @

```

Laat het schip nu weer zelf zijn weg vervolgen

```

REACTIVATE nxtboot
lastboot ← nxtboot
case ← 2
REPEAT FROM regelmaar

```



### 6.39 **MANAGE\_SLUIS**

Dit MODULE bevat de procesbeschrijvingen van de SLUIS, zijn WEKKER alsmede van de KLOK van zijn kolken.

```

@=====
@
@ MANAGE_SLUIS
@
@=====

@=====
@ Procesbeschrijving van de KLOK van de KOLK @
@=====
kolkklok:
  werk ← FIRST OF shift
  THIS kolk ← inhibitor
  THIS sluis ← zijnsluis

                                Kolk gaat buiten bedrijf
  REMOVE inhibitor FROM kolken IF inhibitor BELONGS TO kolken
  IF schuttende = FALSE

                                Kolk had al niets te doen
  CALL idle_stop_watch(2)
  buiten_bedrijf ← NOW
END

                                Wachtende schepen worden overliggers
  FOR k ← 1 TO 2
    FOR EACH schip IN wachtruimte[k] WITH wachtend
      THIS kolk ← FIRST kolk IN kolken WITH passen(10) ≤ 1
      IF THIS kolk IS NONE
        tw ← NOW-tw
        tl ← NOW
        wachtend ← FALSE
      END
    END
  END
  TERMINATE IF werk IS NONE

                                Wacht op volgende bedrijfsperiode
  WAIT begin OF werk - NOW

  THIS sluis ← zijnsluis OF inhibitor

                                Registreer begintijdstip voor eventuele wachttijd
  FOR k ← 1 TO 2
    FOR EACH schip IN wachtruimte[k]
      THIS kolk ← FIRST kolk IN kolken WITH passen(10) ≤ 1
      tl ← NOW IF THIS kolk IS NONE
    END
  END
  THIS kolk ← inhibitor
  IF schuttende = FALSE

                                Idle
    CALL sluis_registratie(FALSE)
    CALL idle_stop_watch(1)
  END

                                Laat de kolk weer meedoen
                                Laat een nieuw opstelplan maken
  CALL schutplanning(nieuw_opstelplan,1)
                                En een voor de andere kant ?
  CALL schutplanning(nieuw_opstelplan,2)
  IF schutplanning OF zijnsluis IS schutstrategie_nkolk
  REMOVE werk FROM shift

                                Wacht tot einde werktijd
  WAIT einde OF werk - NOW

```

## Verschuif een week

```
begin OF werk ← begin OF werk + 168
einde OF werk ← einde OF werk + 168
JOIN werk TO shift
REPEAT FROM kolkklok
```

```
@=====
@ Procesbeschrijving van de WEKKER van de SLUIS @
@   bij schutstrategie%n met meerdere kolken   @
@=====
```

```
trriinngg:
  ACTIVATE weksluis FROM beschouw_situatie
  TERMINATE
```

```
@=====
@ Procesbeschrijving van de SLUIS bij               @
@ schutstrategie%n met meerdere kolken           @
@=====
```

```
beschouw_situatie:
  CALL schutplanning(sluiss_zelf,0)
  TERMINATE
```

## 6.40 **MANAGE\_VERNAUWING**

Dit MODULE bevat de procesbeschrijving van een vernauwing, dwz van zijn verkeersregeling

```
@=====
@
@ MANAGE_VERNAUWING
@
@=====

@=====
@ Procesbeschrijving van de VERKEERSREGELING door de VERNAUWING @
@=====
```

starre\_lichten:

```

                                Regeling is star
                                Licht gaat op groen

vlicht[vernuwingskant] ← groen
                                Bepaal eindgroen
eindgroen ← NOW + maxgroen[vernuwingskant] MINUTES
CALL display_vernuwing
IF garantieschepen IS NOT EMPTY
                                Laat garantieschepen voorgaan
    WAIT UNTIL garantieschepen IS EMPTY
    GOTO andere_kant
END
WAIT maxgroen[vernuwingskant] MINUTES
```

andere\_kant:

```

                                Nu de andere kant

vlicht[vernuwingskant] ← rood
FOR EACH schip IN vwachtrij[vernuwingskant] WITH wachtend
    tw      ← NOW - tw
    tl      ← NOW
    wachtend ← FALSE
    CALL query_registratie(start_overliggen)
END
CALL display_vernuwing
                                Wacht op ontruiming
WAIT ontruimingstijd MINUTES

                                Andere kant
vernuwingskant ← 3-vernuwingskant
                                En dan gaat het symmetrisch verder
REPEAT FROM starre_lichten
```

verkeersafhankelijk:

```

                                Verkeersafhankelijke regeling

vlicht[vernuwingskant] ← groen
IF vwachtrij[3-vernuwingskant] IS EMPTY
                                Aan de andere kant is er geen schip

    eindgroen ← le6
    CALL display_vernuwing
                                Wacht dus totdat er aan de andere kant wel een
                                schip arriveert
    WAIT WHILE vwachtrij[3-vernuwingskant] IS EMPTY
END

                                Bepaal maximale groentijd
eindgroen ← NOW + maxgroen[vernuwingskant] MINUTES
beperkt_wachten:
    CALL display_vernuwing
                                Laat zoveel mogelijke wachtende- en
                                garantieschepen passeren
```

```
WAIT WHILE vwachtrij[vernuwingskant] IS NOT EMPTY
| garantieschepen IS NOT EMPTY
```

op\_rood:

```
FOR EACH schip IN vwachtrij[vernuwingskant] WITH wachtend
    tw      ← NOW - tw
    tl      ← NOW
    wachtend ← FALSE
    CALL query_registratie(start_overliggen)
END
vlicht[vernuwingskant] ← rood
CALL display_vernuwing
WAIT UNTIL engte IS EMPTY
vernuwingskant ← 3-vernuwingskant
REPEAT FROM verkeersafhankelijk
```

Nu moet het licht op rood  
Wachtende schepen worden overliggers  
Daar verkleurt het licht  
Wacht tot de engte leeg is  
Naar de andere kant  
En daar gaat het symmetrisch verder

## 6.41 PASSAGE\_VERNAUWING

Deze MACRO wordt door een schip aangeroepen om vat te stellen of het een vernauwing kan/mag passeren.

Terugkeerwaarde:

TRUE            het mag  
FALSE          het mag niet

```
@=====
@
@ PASSAGE_VERNAUWING
@
@=====
```

LOCAL:

INTEGER: n

THIS vernauwing ← instructie

Mag er van deze kant ingevaren worden ?

RETURN FALSE IF invaren\_vrij[richting]=FALSE

Ja, staat het licht op groen ?

GOTO varen IF vlicht[richting] = groen

Nee

refnode ← doel

IF vlicht[richting] = oranje

Tabelgestuurde vernauwing

n ← klas

IF richting = 3-voorrangskant @\*\*\* evt. voorrang geven \*\*\*

refschip ← FIRST schip IN vwachtrij[voorrangskant]

WITH ontmoeting[klas,n]=FALSE

RETURN FALSE IF refschip IS NOT NONE

END

refschip ← FIRST schip IN engte WITH refnode IS start

& ontmoeting[klas,n]=FALSE

RETURN refschip IS NONE

END

RETURN FALSE

varen:

IF garantieschepen IS NOT EMPTY

Eventueel voorrang geven

refschip ← FIRST schip IN garantieschepen WITH SMALLEST eta

Zelf garantieschip ?

RETURN TRUE IF refschip IS THIS schip

Nee, voor laten gaan ?

RETURN FALSE IF eta OF refschip < NOW + 2 MINUTES

Nee

END

IF snelste\_eerst

refschip ← FIRST schip IN vwachtrij[richting]

WITH GREATEST vvaarsnelheid[klas,gel]

Volgende test is TRUE als het schip zelf de snelste is

RETURN refschip IS THIS schip | refschip IS NONE

END

RETURN TRUE



## 6.42 PASSEN

Deze MACRO berekent voor THIS SCHIP of het ingepland kan worden in het opstelplan van THIS KOLK.

Parameter:

SLUISKANT de kant vanwaar het schip komt.

Terugkeerwaarde:

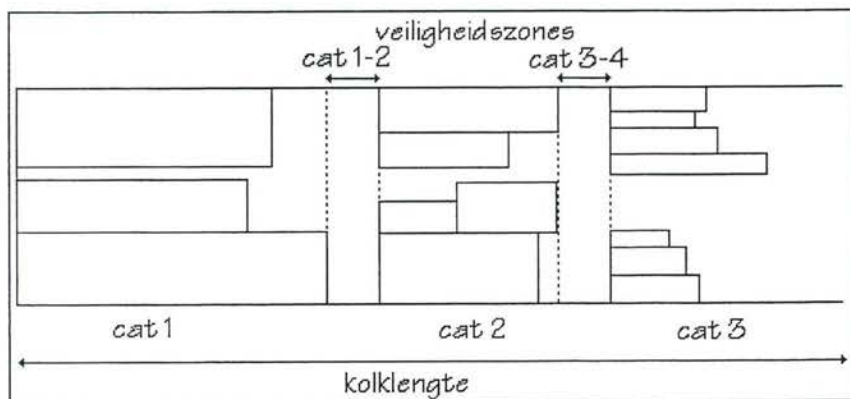
0: THIS SCHIP kan met de volgende schutting van THIS KOLK mee.

2: THIS SCHIP kan naar de wachtruimte.

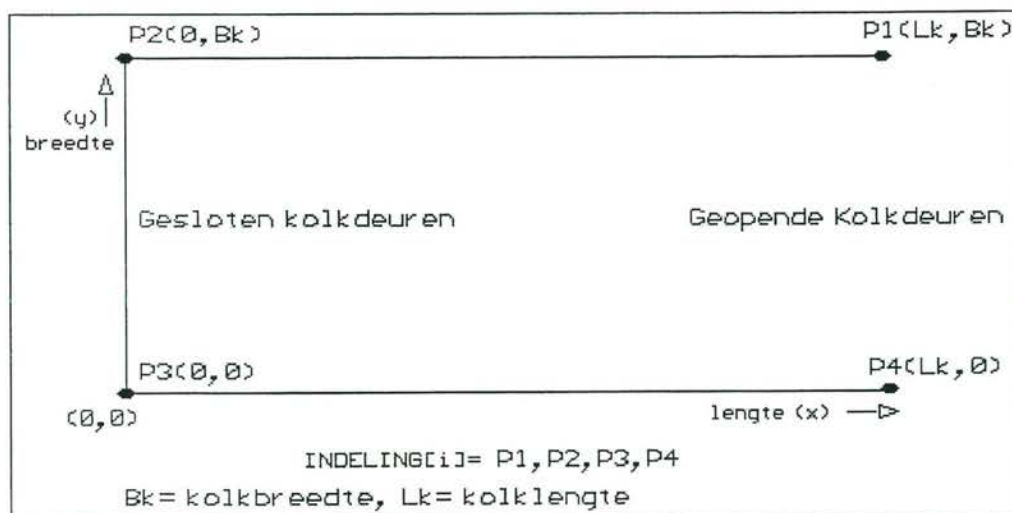
3: THIS SCHIP mag niet of kan (fysiek gezien), niet mee met een schutting van THIS KOLK.

Toelichting:

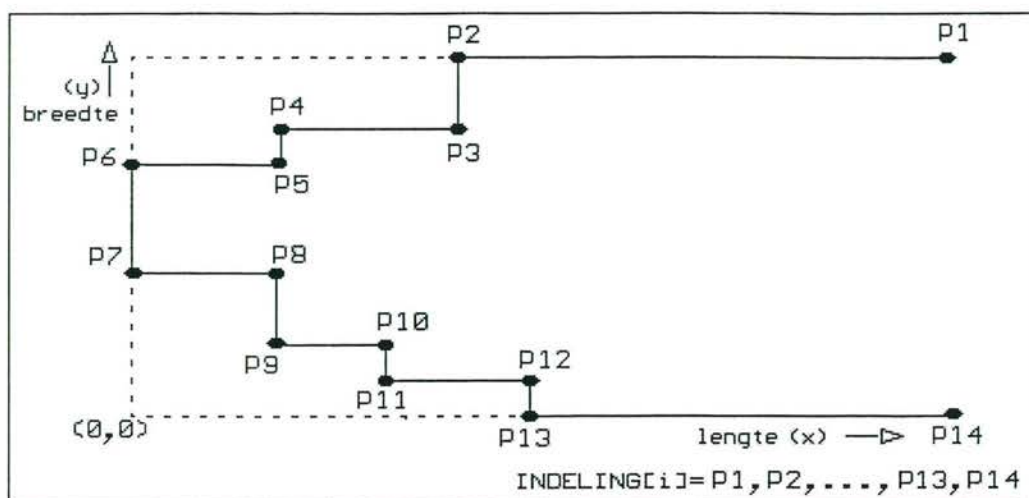
- Wanneer gebruik gemaakt wordt van schutten met gescheiden categorieën, kan de categorie van THIS SCHIP het hele opstelplan opnieuw doen laten maken omdat categorieën gescheiden moeten liggen in de schutruimte. Tussen de categorieën bevindt zich nog een veiligheidszone zoals te zien is in de volgende figuur



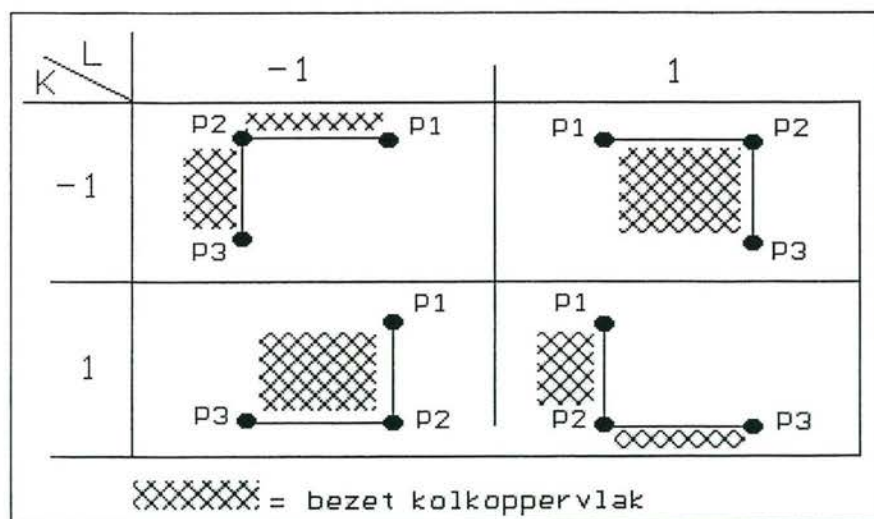
- Wanneer de categorie van het schip daartoe aanleiding geeft maken we een tweede opstelplan (vanaf label RESHUFFLE) en delen alle schepen die met de schutting mee zouden moeten opnieuw in; schepen die reeds ingedeeld waren blijven ingedeeld. THIS SCHIP "past" in het opstelplan als deze nieuwe indeling slaagt.
- Wanneer de categorie van THIS SCHIP geen aanleiding geeft tot het herzien van het opstelplan, wordt er een ligplaats voor het schip gezocht in de bestaande indeling.
- Bij het plannen maakt de sluismeester gebruik van de sets INDELING[I] ( $i = 1, 2$ ) waarin door middel van PUNTen een omtrek wordt beschreven. Deze omtrek omvat de vrije ruimte in een kolkindeling. Elk PUNT heeft twee coördinaten: lengtecoördinaat XX en breedtecoördinaat YY.
- In de volgende figuur is te zien dat de punten in INDELING[I] zodanig gerangschikt zijn, dat zij van rechtsboven tot rechtsonder tegen de wijzers van de klok in de omtrek van het vrije kolkoppervlak beschrijven.



- Een voorbeeld van een vrij kolkoppervlak, wanneer reeds enkele schepen ingepland zijn, zou als volgt kunnen zijn

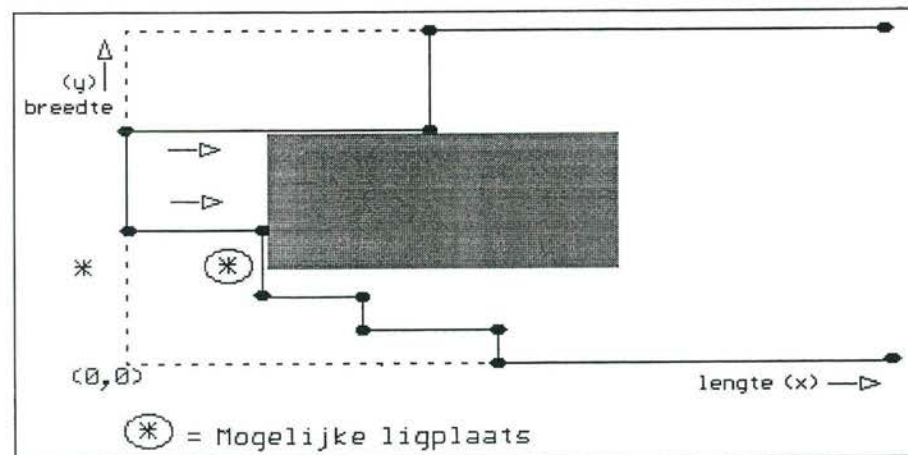
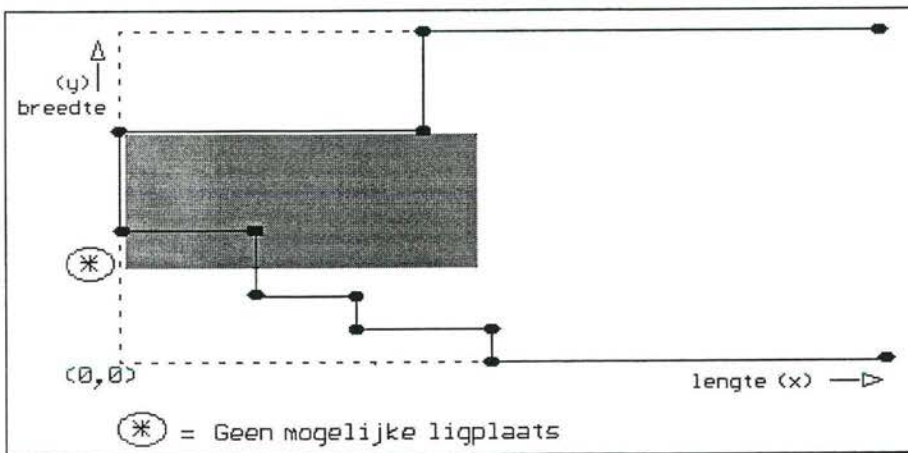
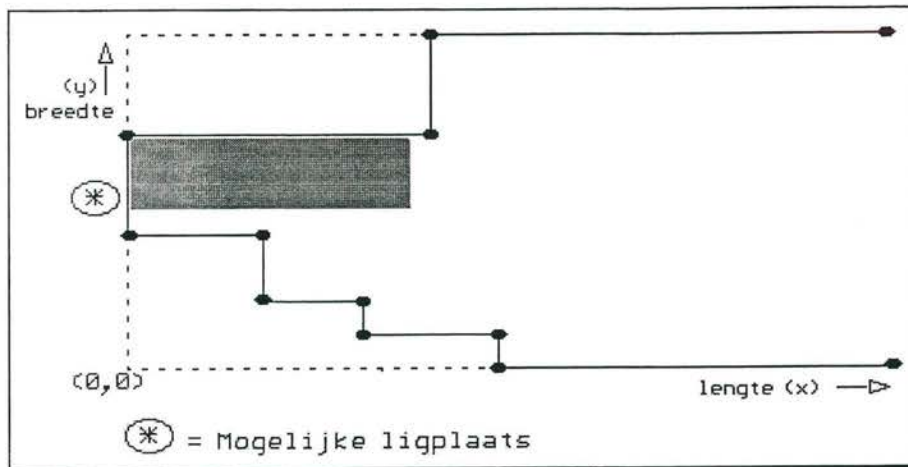


- Met behulp van de MACRO PASSEN wordt bekeken of een schip bij de indeling past, gebruikmakend van de MACRO VOEGTOE.
- De MACRO VOEGTOE heeft tot doel een schip zo mogelijk in te voegen in de INDELING[N] van THIS KOLK. Het schip wordt gerepresenteerd door een rechthoek met lengte PLANLENGTE en breedte PLANBREEDTE. Er wordt een PUNT gezocht waar de linker onderhoek van de rechthoek moet komen te liggen.
- Bepaling van alle mogelijke PUNTEN  
Alle hoeken van de omtrek worden afgezocht naar ligplaatsen. Een hoek wordt bepaald door drie PUNTEN, waarvan het tweede het PUNT in de hoek voorstelt. L en K geven aan van welk type de hoek is. De volgende mogelijkheden doen zich voor:



Als L gelijk is aan K gaat het om een hoek waar het schip eventueel in kan liggen (een binnenhoek). Als L van K verschilt kan een schip zich alleen aan zo'n hoek stoten.

Voor elke binnenhoek wordt gekeken of het schip qua breedte past en het PUNT wordt zonodig naar rechts verplaatst tot het wel past. Zie ook de volgende illustraties.



Vervolgens wordt gekeken of bij het zo verkregen PUNT, het schip nog binnen de kolkgrenzen ligt. Zo ja dan is het gevonden PUNT een mogelijke oplossing

- Selectie van het beste PUNT.

Van de mogelijke oplossingen worden slechts die PUNTen bekeken die het diepst in de kolk liggen (die PUNTen met de kleinste XX).

Van deze selectie wordt dat PUNT gekozen, waarbij het schip met een van z'n lengtezijden aan een rand van de omtrek ligt, die het "verst naar het begin van de kolk doorloopt" om restlengte tussen het schip en het begin van de kolk optimaal te benutten. Er is dus "voorkeur voor lengtezijde aan een rand", om zoveel mogelijk vastmaakpunten tussen schip en kolkwand te verkrijgen.

Bij twee gelijke oplossingen wordt het schip zo dicht mogelijk bij de bovenste wand van de kolk gelegd (voorkeur voor stuurboordaanleggen).



```

@=====
@
@ PASSEN
@
@=====

@=====
@ returnwaarde: 0) THIS SCHIP kan nog mee in THIS KOLK
@                2) THIS SCHIP kan naar de WACHTRUIMTE
@                3) fysieke beperking
@=====

PARAMETER:      @ 1/2: kant vd sluis waar THIS SCHIP aankomt
INTEGER: sluiskant @ 10: schip ligt in wachtrij
LOCAL:
  INTEGER: k

                                Mag deze klasse in deze kolk ?
RETURN 3 IF geschikt[klas] = FALSE

                                Ja
THIS sluis ← zijnsluis
FOR k ← 1 TO 2
  x ← waterstand(s_nrtij[k],s_peilpar[k])
                                Staat het water hoog genoeg ?
  RETURN 3 IF (diepgang + kielspeling) > x - drempel[k]
                                Ja
END

                                Test nog wat fysieke beperkingen
RETURN 3          IF planbreedte > brdhoofd
RETURN 3          IF planlengte > lengkolk
RETURN 0          IF sluiskant > 2 @*** zie wachtende schepen ****
                                @***** in kolkmodule *****
RETURN 0 IF opstelplan[sluiskant] IS EMPTY

@-----@
@ Als THIS KOLK gereserveerd is door een gegarandeerd schip @
@ mag een uitgesloten schip niet in het plan meegenomen worden @
@-----@

refschip ← FIRST schip IN opstelplan[side] WITH garantie
refschip ← FIRST schip IN opstelplan[3-side] WITH garantie
          IF refschip IS NONE
RETURN 2  IF refschip IS NOT NONE & garkolk[klas]=0

@-----@
@ Categorie I schepen en categorie III @
@ schepen mogen niet door elkaar worden geschut @
@-----@
IF gemengd_schutten
  k ← categorie[klas]
  refschip ← FIRST schip IN opstelplan[sluiskant]
                                WITH 3 = categorie[klas] * k
  RETURN 2 IF refschip IS NOT NONE
END

k ← categorie[klas]
x ← eta

                                Ga na of er reshuffeld moet worden
refschip ← FIRST schip IN opstelplan[sluiskant]
                                WITH categorie[klas] > k
GOTO reshuffle IF refschip IS NOT NONE & gemengd_schutten=FALSE

refschip ← FIRST schip IN opstelplan[sluiskant]

```

```

                                WITH categorie[klas] = k
GOTO reshuffle IF refschip IS NONE      & gemengd_schutten=FALSE

refschip ← FIRST schip IN opstelplan[sluiskant] WITH eta > x
GOTO reshuffle IF refschip IS NOT NONE

@-----@
@ ligplaatsen in de bestaande indeling @
@-----@
plaats[1] ← FIRST OF indeling[sluiskant]
plaats[2] ← SUCC OF plaats[1] IN indeling[sluiskant]
plaats[3] ← SUCC OF plaats[2] IN indeling[sluiskant]
k        ← -1
WHILE plaats[3] IS NOT NONE
  l ← 1
  IF yy OF plaats[1] = yy OF plaats[2]
    l ← -1 IF xx OF plaats[2] < xx OF plaats[1]
  END
  IF yy OF plaats[2] = yy OF plaats[3]
    l ← -1 IF xx OF plaats[3] < xx OF plaats[2]
  END
  IF l = k
    plaats[1] ← NEW punt
    xx OF plaats[1] ← xx OF plaats[2]
    yy OF plaats[1] ← yy OF plaats[2]
    yy OF plaats[1] ← yy OF plaats[2] - planbreedte IF l = -1
    FOR EACH punt IN indeling[sluiskant]
      WITH (xx > xx OF plaats[1]) & (yy > yy OF plaats[1])
        & (yy < planbreedte + yy OF plaats[1])
      xx OF plaats[1] ← xx IF xx ≥ xx OF plaats[1]
    END
    IF (planlengte ≤ lengkolk - xx OF plaats[1])
      & (yy OF plaats[1] ≥ 0)
      & brdtkolk ≥ planbreedte + yy OF plaats[1]
      THIS punt ← FIRST punt IN indeling[sluiskant] WITH
        (ABS(yy - yy OF plaats[1]) ≤ 0.001)
        | (ABS(yy - planbreedte + yy OF plaats[1]) ≤ 0.001)
      RETURN 0 IF THIS punt IS NOT NONE
    END
  END
  plaats[1] ← plaats[2]
  plaats[2] ← plaats[3]
  plaats[3] ← SUCC OF plaats[2] IN indeling[sluiskant]
  k        ← 0 - k
END

RETURN 2

reshuffle:
  IF gemengd_schutten=FALSE
    IF refschip IS NOT NONE
      refschip ← FIRST schip IN opstelplan[sluiskant]
        THAT BELONGS TO schutruimte
      RETURN 2 IF refschip IS NOT NONE
    END
  END
  help[1]      ← opstelplan[sluiskant]
  opstelplan[sluiskant] ← NEW SET
  JOIN EACH schip IN help[1] TO opstelplan[sluiskant]
  help[2]      ← NEW SET
  JOIN EACH punt IN indeling[sluiskant] TO help[2]
  hulp        ← NEW SET CALLED sluiskant|" kladplan "|nrkolk
  JOIN THIS schip TO hulp RANKED BY 45 * eta
  FOR EACH schip IN opstelplan[sluiskant]
    JOIN THIS schip TO hulp RANKED BY 45 * eta
    REMOVE THIS schip FROM opstelplan[sluiskant]

```



```

END
IF gemengd_schutten = FALSE
  FOR j ← 1 TO 3
    JOIN EACH schip IN hulp WITH categorie[klas]=j
      TO opstelplan[sluiskant]
  END
END
opstelplan[sluiskant] ← hulp IF gemengd_schutten

CALL initdeel(sluiskant)
refschip ← FIRST OF opstelplan[sluiskant]
catwis ← categorie[klas OF refschip]
WHILE refschip IS NOT NONE
  IF (catwis ≠ categorie[klas OF refschip]) & gemengd_schutten = FALSE
    plaats[1] ← FIRST OF indeling[sluiskant]
    REMOVE plaats[1] FROM indeling[sluiskant]
    plaats[2] ← LAST OF indeling[sluiskant]
    REMOVE plaats[2] FROM indeling[sluiskant]
    k ← categorie[klas OF refschip]
    THIS punt ← FIRST punt IN indeling[sluiskant] WITH GREATEST xx
    x ← xx
    x ← x + cat12 IF (1=catwis) & 2 = k
    x ← x + cat13 IF (1=catwis) & 3 = k
    x ← x + cat23 IF (2=catwis) & 3 = k
    GOTO kanniet IF x ≥ lengkolk
    catwis ← k
    indeling[sluiskant] ← NEW SET
    JOIN plaats[1] TO indeling[sluiskant]
    THIS punt ← NEW punt
    xx ← x
    yy ← brdtkolk
    JOIN THIS punt TO indeling[sluiskant]
    THIS punt ← NEW punt
    xx ← x
    yy ← 0
    JOIN THIS punt TO indeling[sluiskant]
    JOIN plaats[2] TO indeling[sluiskant]
  END
  refbrug ← THIS schip @*** REFBRUG wordt hier gebruikt als ***
  THIS schip ← refschip @*** REFERENCE TO schip! ***
  x ← voegtoe(sluiskant)
  THIS schip ← refbrug
  GOTO kanniet IF x ≤ 0
  refschip ← SUCC OF refschip IN opstelplan[sluiskant]
END

opstelplan[sluiskant] ← help[1]
indeling[sluiskant] ← help[2]
RETURN 0

kanniet:
opstelplan[sluiskant] ← help[1]
indeling[sluiskant] ← help[2]
RETURN 2

```

### 6.43 PLAN\_THIS\_KOLK\_IN

Deze MACRO vervolmaakt de schutplanning van een kolk voor alle reeds aanwezige schepen.

Parameters:

SLUISKANT de kant van de kolk.

```

@=====
@
@ PLAN_THIS_KOLK_IN
@
@=====

@=====
@           Maak de schutplanning (af) voor THIS KOLK           @
@           voor alle reeds aanwezige schepen                   @
@=====
PARAMETER:
    INTEGER: sluiskant
LOCAL:
    INTEGER: m

THIS sluis ← zijnsluis

                                Onderzoek alle onmogelijke schepen
FOR EACH schip IN onmogelijk[sluiskant] WITH cur_plek IS THIS sluis
    m ← passen(sluiskant)
    IF (m=0) | m=1

                                Schip past nu wel
        REMOVE THIS schip FROM onmogelijk[sluiskant]
        CALL herdeel(sluiskant)
        instructie ← THIS kolk
    END
    IF m=2

                                Schip kan wachten
        REMOVE THIS schip FROM onmogelijk[sluiskant]
        instructie ← THIS sluis
    END
END

                                Onderzoek alle andere nog niet opgestelde
                                schepen
FOR EACH schip IN remanent[sluiskant]
    IF passen(sluiskant) ≤ 1
        REMOVE THIS schip FROM remanent[sluiskant]
        CALL herdeel(sluiskant)
        REACTIVATE THIS schip
        IF THIS schip BELONGS TO wachtruimte[sluiskant]
            instructie ← THIS kolk
    END
END
RETURN

```

## 6.44 QUERY\_REGISTRATIE

Deze MACRO verzorgt de registratie tbv het QUERY model.

Parameters:

TYPE            te registreren toestandsverandering

```
@=====
@
@ QUERY_REGISTRATIE
@
@=====
```

PARAMETER:

INTEGER: type

LOCAL:

INTEGER: k

```
RETURN IF vorige_mijlpaal IS FIRST OF mpalen
IF type < 0                                @*** welke doorgang van de brug ***
    bladinhoud[bladstart+m_index OF vorige_mijlpaal+via_doorgang]
    OF blad ← 0-type
    RETURN
END
    @*** tijdregistratie ***
    bladinhoud[bladstart+m_index OF vorige_mijlpaal+type]
    OF blad ← NOW - creatietijd
IF type = naar_opstelruimte               @*** welke kolk van de sluis ***
    bladinhoud[bladstart+m_index OF vorige_mijlpaal+via_kolk]
    OF blad ← nrkolk OF instructie
    RETURN
END
volle_regels OF blad ← volle_regels OF blad + 1
    IF vorige_mijlpaal IS LAST OF mpalen
RETURN
```

## 6.45 RAPPORT\_BRUGGEN

Deze MACRO verzorgt de rapportage betreffende bruggen.

```
@=====
@
@ RAPPORT_BRUGGEN
@
@=====

@=====
@ rapport BRUGGEN @
@=====

LOCALS:
  INTEGER: knoop1 knoop2 i j k l m n kl ri
  REAL    : x y z

FOR EACH brug IN bruggen
  FOR k ← 1 TO 9
    bgem[k] ← bgem[k] / totopen IF totopen > 0
    IF 1 < totopen
      bafw[k] ← (bafw[k] / totopen) - bgem[k]_2
      bafw[k] ← totopen * bafw[k] / (totopen-1)
      bafw[k] ← 0 IF bafw[k] < 0
      bafw[k] ← SQRT(bafw[k])
    END
    bafw[k] ← 0 IF totopen ≤ 1
  END
  bgem[10] ← bgem[10] / n_uren IF 0 < n_uren
  IF 1 < n_uren
    bafw[10] ← (bafw[10] / n_uren) - bgem[10]_2
    bafw[10] ← n_uren * bafw[10] / (n_uren-1)
    bafw[10] ← 0 IF bafw[10] < 0
    bafw[10] ← SQRT(bafw[10])
  END
  bafw[10] ← 0 IF n_uren ≤ 1
  knoop1 ← nrnode OF zijde[1]
  knoop2 ← nrnode OF zijde[2]
  text ← knoop1|". BRUG      ."|knoop2
  CALL char_in_string(<text>,30,21,"OPERATIONELE GEGEVENS",1)
  nieuw ← TRUE
  CALL kop
  regeloph ← 11
  CALL kop
  WRITE totopen TO query_invoer WITH IMAGE *****
  WRITE #10;"TOTAAL AANTAL BRUGOPENINGEN :";totopen TO rapport
    WITH IMAGE *AΔ*****
  WRITE #13;"                " TO rapport
    WITH IMAGE |*A
  WRITE #10;" "; "MIN      GEM      MAX      AFW" TO rapport WITH IMAGE *A47A
  WRITE "BEDIENINGSTIJD (roodlichttijd)";" ";":" TO rapport
    WITH IMAGE AA12*
  WRITE bmin[1];bgem[1];bmax[1];bafw[1] TO rapport
    WITH IMAGE |ΔΔ**.*ΔΔ**.*ΔΔ**.*ΔΔ**.*
  WRITE "AANTAL SCHEPEN PER BEWEEGBARE BRUGOPENING : " TO rapport
    WITH IMAGE A
  WRITE bmin[2];bgem[2];bmax[2];bafw[2] TO rapport
    WITH IMAGE |Δ****.*Δ****.*Δ****.*Δ****.*
  WRITE "AANTAL BRUGOPENINGEN per UUR";" ";":" TO rapport WITH IMAGE AA14*
  WRITE bmin[10];bgem[10];bmax[10];bafw[10] TO rapport
    WITH IMAGE |Δ****.*Δ****.*Δ****.*Δ****.*
  WRITE #10;" ";knoop1;"-";knoop2|"      ";knoop2;"-";knoop1|"      "; "SOM"
    TO rapport WITH IMAGE *A37***|*|***ΔΔΔ***|*|***ΔΔΔΔ***
```



```

cap[1,1] ← 0
cap[1,2] ← 0
cap[2,1] ← 0
cap[2,2] ← 0
FOR EACH schipstat IN bschipstats
  FOR k1 ← 1 TO 30
    FOR ri ← 1 TO 2
      cap[s_volgnr,ri] ← cap[s_volgnr,ri] + s_aantal[ri,k1]
    END
  END
END
FOR k ← 1 TO 2
  WRITE "AANTAL SCHEPEN DOOR DOORGANG";k;":" TO rapport
  WITH IMAGE AΔ*ΔΔ*
  WRITE cap[k,1];cap[k,2];cap[k,1]+cap[k,2] TO rapport
  WITH IMAGE |Δ*****Δ*****Δ*****
  WRITE cap[k,1]+cap[k,2] TO query_invoer WITH IMAGE |Δ*****
END

regeloph ← 7
CALL kop
WRITE #10;#10;"WEGVERKEER" TO rapport WITH IMAGE aaA
WRITE #13;" " TO rapport WITH IMAGE |*a
WRITE #10;" "; "WEGKANT 1   WEGKANT 2           SOM" TO rapport
  WITH IMAGE *A33A
FOR k ← 1 TO 2
  WRITE "TOTAANTAL BEINVL. EENH. CAT.";k;":" TO rapport
  WITH IMAGE A|*Δ*
  l ← 1 + 2*k
  WRITE bgem[l]*totopen;bgem[l+1]*totopen;totopen*bgem[l]+bgem[l+1]
  TO rapport WITH IMAGE |Δ*****Δ*****Δ*****
END

regeloph ← 10
CALL kop
WRITE #10;#10;" "; "MIN      GEM      MAX      AFW" TO rapport
  WITH IMAGE aaA51A
FOR k ← 1 TO 2
  FOR l ← 1 TO 2
    m ← (2*k) + 1
    WRITE "AANTAL BEINVL. EENH. PER BR.OPENING CAT." TO rapport
    WITH IMAGE A
    WRITE k;","; "WEGK.";l;":" TO rapport WITH IMAGE |*|*****|*|*
    WRITE bmin[m];bgem[m];bmax[m];bafw[m] TO rapport
    WITH IMAGE |Δ**.*Δ****.*Δ****.*Δ****.*
  END
END

WRITE "LANGSTE FILE-OPLOSTIJD PER BRUGOPENING";":" TO rapport
  WITH IMAGE AΔΔΔΔΔΔΔΔΔΔ*
WRITE bmin[9];bgem[9];bmax[9];bafw[9] TO rapport
  WITH IMAGE |Δ**.*Δ****.*Δ****.*Δ****.*
FOR k ← 1 TO 2
  l ← 6 + k
  WRITE "GESOMMEERDE WACHTTIJD PER OPENING CAT.";k; "(uren)  :"
  TO rapport WITH IMAGE A|*Δ*****
  WRITE bmin[l];bgem[l];bmax[l];bafw[l] TO rapport
  WITH IMAGE |Δ**.*Δ****.*Δ****.*Δ****.*
END

FOR k ← 1 TO 4
  xxx[k] ← 0
  cc[k] ← 0
END
FOR EACH schipstat IN bschipstats
  FOR k1 ← 1 TO 30

```



```

        FOR ri ← 1 TO 2
            xxx[ri] ← xxx[ri] + s_cumpastijd[ri,kl]
            xxx[ri+2] ← xxx[ri+2] + s_cumpastijd[ri,kl]*kosten[2,kl]/60
            cc[ri] ← cc[ri] + s_aantal[ri,kl]
        END
    END
END

regeloph ← 6
CALL kop
WRITE #10;#10;"SCHEEPVAART, GEMIDDELDE PASSEERTIJD" TO rapport
    WITH IMAGE aaA
WRITE #13;" " TO rapport
    WITH IMAGE |*A
IF cc[1] > 0
    WRITE knoop1;"-";knoop2|" " :";xxx[1]/cc[1] TO rapport
        WITH IMAGE ***aaΔΔΔ*****.**
END
IF cc[2] > 0
    WRITE knoop2;"-";knoop1|" " :";xxx[2]/cc[2] TO rapport
        WITH IMAGE ***aaΔΔΔ*****.**
END
IF (cc[1]+cc[2]) > 0
    WRITE "BEIDE RI: ";(xxx[1]+xxx[2])/cc[1]+cc[2] TO rapport
        WITH IMAGE AΔΔΔ*****.**
END

regeloph ← 5
CALL kop
WRITE #10;"SCHEEPVAART, GEMIDDELDE PASSEERKOSTEN" TO rapport
    WITH IMAGE *A
WRITE #13;" " TO rapport
    WITH IMAGE |*A
IF cc[1] > 0
    WRITE knoop1;"-";knoop2|" " :";xxx[3]/cc[1] TO rapport
        WITH IMAGE ***aaΔΔΔ*****.** IF 0 < cc[1]
END
IF cc[2] > 0
    WRITE knoop2;"-";knoop1|" " :";xxx[4]/cc[2] TO rapport
        WITH IMAGE ***aaΔΔΔ*****.**
END
IF (cc[1]+cc[2]) > 0
    WRITE "BEIDE RI: ";(xxx[3]+xxx[4])/cc[1]+cc[2] TO rapport
        WITH IMAGE AΔΔΔ*****.**
END

regeloph ← 7
CALL kop
WRITE #10;#10;"TOTALE KOSTEN (in duizenden guldens)" TO rapport
    WITH IMAGE aaA
WRITE #13;" " TO rapport
    WITH IMAGE |*a
x ← totopen * bgem[7] * wkosten[1]
y ← totopen * bgem[8] * wkosten[2]
z ← 0
z ← xxx[3]+xxx[4]
WRITE x+y TO query_invoer WITH IMAGE |Δ*****
WRITE "wegverkeer cat.1: ";x/1000 TO rapport WITH IMAGE AΔ*****.**
WRITE "wegverkeer cat.2: ";y/1000 TO rapport WITH IMAGE AΔ*****.**
WRITE "scheepvaart " :";z/1000 TO rapport WITH IMAGE AΔ*****.**
WRITE " " som: ";(x+y+z)/1000 TO rapport
    WITH IMAGE AΔ*****.**
CALL rapport_scheepvaart(2)
END

RETURN

```

## 6.46 RAPPORT\_INITIA

Deze MACRO verzorgt het algemene gedeelte van de rapportage.

```
@=====
@
@ RAPPORT_INITIA
@
@=====

WRITE "Rapport" WITH IMAGE ΔΔΔΔΔΔΔΔΔΔ
REWIND rapport
WRITE #13;" Rijkswaterstaat SIVAK v2.01, juli'96" TO rapport WITH IMAGE |*A
WRITE ";date[3];"-00"|date[2];"-";date[1] TO rapport
    WITH IMAGE |a20**aa****
WRITE time[1];":00"|time[2] TO rapport WITH IMAGE |Δ**a

WRITE "• "; "Inlezen clusters" WITH IMAGE a20a
tekst ← CHREAD FROM kopclust
linenr ← 3
WHILE tekst ≠ "end"
    WRITE tekst TO rapport WITH IMAGE A
    tekst ← CHREAD FROM kopclust
    linenr ← linenr+1
END

WRITE "" TO rapport WITH IMAGE *
text      ← "                ALGEMENE GEGEVENS"
pagnr     ← 0

regelaantal ← READ FROM kopclust
CALL kop
CALL lees_kosten

clusters ← NEW SET
n        ← READ FROM kopclust
WHILE n > 0
    THIS cluster ← NEW cluster
    nrcluster    ← n
    m ← READ FROM kopclust
    WHILE 0 < m
        klassen[m] ← TRUE
        m          ← READ FROM kopclust
    END
    JOIN THIS cluster TO clusters RANKED BY nrcluster
    n ← READ FROM kopclust
END
WRITE "• "; "schrijven" WITH IMAGE a20a
regeloph ← 3
CALL kop
WRITE "TIJDEN in MINUTEN" TO rapport WITH IMAGE A
WRITE "OPPERVLAKTES in mý" TO rapport WITH IMAGE A
WRITE "KOSTEN in GULDENS" TO rapport WITH IMAGE A

regeloph ← 2+LENGTH OF clusters
CALL kop
WRITE #10;"DEFINITIE CLUSTERS:" TO rapport WITH IMAGE *A
WRITE #13;"                " TO rapport WITH IMAGE |*A
FOR EACH cluster IN clusters
    WRITE nrcluster;": " TO rapport WITH IMAGE *****Δ**
    FOR l ← 1 TO 30
        WRITE l TO rapport WITH IMAGE |*** IF klassen[l]
    END
END
END
```

```
regeloph ← 4 + LENGTH OF sluizen + LENGTH OF bruggen
           + LENGTH OF vaarwegen + LENGTH OF vernauwingen
CALL kop
WRITE #10;#10;"OPBOUW VAN HET NETWERK:" TO rapport WITH IMAGE aaA
WRITE #13;      "                " TO rapport WITH IMAGE |aA
FOR EACH sluis IN sluizen
  WRITE nrnode OF kant[1];"-";nrnode OF kant[2]|"      sluis" TO rapport
  WITH IMAGE ΔΔ***aa
END
FOR EACH brug IN bruggen
  WRITE nrnode OF zijde[1];"-";nrnode OF zijde[2]|"      brug" TO rapport
  WITH IMAGE ΔΔ***aa
END
FOR EACH vernauwing IN vernauwingen
  WRITE nrnode OF vzijde[1];"-";nrnode OF vzijde[2]|"      vernauwing"
  TO rapport WITH IMAGE ΔΔ***aa
END
FOR EACH vaarweg IN vaarwegen
  WRITE nrnode OF raai[1];"-";nrnode OF raai[2]|"      vaarweg"
  TO rapport WITH IMAGE ΔΔ***aa
END

regeloph ← 3 + LENGTH OF generatoren
CALL kop
WRITE #10;#10;"SCHEEPVAART-ROUTES (in knooppunten):" TO rapport
  WITH IMAGE AAA
WRITE #13;"                " TO rapport WITH IMAGE |*A
FOR EACH generator IN generatoren
  WRITE route_filenaam;": " TO rapport WITH IMAGE ΔAA
  FOR EACH node IN route_heen
    WRITE nrnode TO rapport WITH IMAGE |Δ***
  END
END

RETURN
```

## 6.47 RAPPORT\_SCHEEPVAART

Deze MACRO verzorgt de rapportage mbt de scheepvaart bij een kunstwerk

Parameters:

SOORT        de soort van het kunstwerk  
              1 = sluis  
              2 = brug  
              3 = vernauwing

```
@=====
@
@ RAPPORT_SCHEEPVAART
@
@=====

@=====
@ Scheepvaartgegevens voor een kunstwerk bij de set SCHIPSTATS @
@=====

PARAMETER:
    INTEGER: soort
LOCAL:
    INTEGER: vn kl ri tel
    REAL    : pastijd kost aantal nover

WRITE cc[1];cc[2];xxx[1]+xxx[2];xxx[3]+xxx[4] TO query_invoer
    WITH IMAGE *****Δ*****Δ*****Δ*****
CALL char_in_string(<text>,28,24,"SCHEEPVAART-STATISTIEKEN",1)
nieuw ← TRUE
CALL kop

schipstats ← sschipstats IF soort = 1
schipstats ← bschipstats IF soort = 2
schipstats ← vschipstats IF soort = 3

regeloph ← 5
CALL kop
WRITE #10;#10;" " TO rapport WITH IMAGE AA****
FOR EACH schipstat IN schipstats
    WRITE "|      KOLK";s_volgnr;" " TO rapport
        WITH IMAGE |*****Δ**Δ**** IF soort = 1
    WRITE "|      DOORGANG";s_volgnr;" " TO rapport WITH IMAGE |AΔ**Δ***
                                                IF soort = 2
    IF soort = 3
        WRITE "|      van KANT ";nrnode OF vzijde[s_volgnr]|"      "
            TO rapport WITH IMAGE |AA
    END
END

WRITE "CLU-" TO rapport WITH IMAGE ****
FOR k ← 1 TO LENGTH OF schipstats
    WRITE "|      —      " TO rapport WITH IMAGE |A
    WRITE " " —      TO rapport WITH IMAGE |* IF soort ≠ 1
END
WRITE "KLAS" TO rapport WITH IMAGE ****
FOR k ← 1 TO LENGTH OF schipstats
    WRITE "| Tp | Kp | N |" TO rapport WITH IMAGE |*****
    WRITE " " TO rapport WITH IMAGE |* IF soort ≠ 1
    WRITE "No" TO rapport WITH IMAGE |**
END
FOR EACH cluster IN clusters
    IF tel MODULO(5)=0
        CALL kop
        WRITE "-----" TO rapport WITH IMAGE ****
        FOR k ← 1 TO LENGTH OF schipstats
```



```

        WRITE "+---+---+---+---" TO rapport WITH IMAGE |A
        WRITE "-" TO rapport WITH IMAGE |*                IF soort ≠ 1
    END
END
tel ← tel+1
WRITE nrcluster;" " TO rapport WITH IMAGE ***|*
c_aantal ← 0
c_nover ← 0
c_pastijd ← 0
c_kosten ← 0
CALL kop
FOR EACH schipstat IN schipstats
    aantal ← 0
    nover ← 0
    pastijd ← 0
    kost ← 0
    FOR ri ← 1 TO 2
        FOR kl ← 1 TO 30
            IF klassen[kl]
                aantal ← aantal + s_aantal[ri,kl]
                nover ← nover + s_nover[ri,kl]
                pastijd ← pastijd + s_cumpastijd[ri,kl]
                kost ← kost + s_cumpastijd[ri,kl] * kosten[soort,kl]
            END
        END
    END
    c_aantal ← c_aantal + aantal
    c_nover ← c_nover + nover
    c_pastijd ← c_pastijd + pastijd
    c_kosten ← c_kosten + kost
    IF 0 < aantal
        pastijd ← pastijd / aantal
        kost ← kost / aantal*60.0
    END
    WRITE "|";pastijd;"|";kost;"|";aantal;"|" TO rapport
    WITH IMAGE |A****A*****A*****A
    WRITE nover TO rapport WITH IMAGE |**                IF soort = 1
    WRITE nover TO rapport WITH IMAGE |***                IF soort ≠ 1
END
IF 0 < c_aantal
    c_pastijd ← c_pastijd / c_aantal
    c_kosten ← c_kosten / c_aantal*60.0
END
END

regeloph ← 5 + LENGTH OF clusters + CEIL(LENGTH OF clusters/5)

CALL kop
WRITE #10;#10;"      |          TOTAAL          " TO rapport WITH IMAGE aaa
WRITE "CLU-|          " TO rapport WITH IMAGE a
WRITE "KLAS|  Tp |  Kp |  N |  No" TO rapport WITH IMAGE a
tel ← 0
FOR EACH cluster IN clusters
    IF tel MODULO(5)=0
        WRITE "-----+-----+-----+-----+-----" TO rapport WITH IMAGE A
    END
    tel ← tel+1
    WRITE nrcluster;"|";c_pastijd;"|";c_kosten;"|";c_aantal;"|";c_nover
    TO rapport WITH IMAGE ***a2*****a*****a*****a*****a*****
END
RETURN

```

## 6.48 RAPPORT\_SLUIZEN

Deze MACRO verzorgt de rapportage mbt de sluizen

```

@=====
@
@ RAPPORT_SLUIZEN
@
@=====

@=====
@ rapport SLUIZEN @
@=====

LOCALS:
  INTEGER: knoop1 knoop2 i j k l m n kl ri
  REAL    : x y z

FOR EACH sluis IN sluizen
  FOR EACH kolk IN alle_kolken
    REMOVE THIS kolk FROM alle_kolken
    JOIN THIS kolk TO alle_kolken RANKED BY nrkolk
    FOR j ← 1 TO 2 @ kant
      FOR m ← 1 TO 7 @ dag
        tot_debiet[8,j] ← tot_debiet[8,j] + tot_debiet[m,j] @ totaal
        tot_debiet[m,j] ← tot_debiet[m,j] / deb_aant[m]
        IF 0 < deb_aant[m]
          tot_debiet[m,j] ← -1.11E-12 IF deb_aant[m] = 0 @ ivm sterretjes
      END
    END
    FOR j ← 1 TO 2
      THIS kolkstat ← kolk_stat[j]
      FOR k ← 1 TO 3
        kgem[k] ← kgem[k] / volschut IF volschut > 0
      END

      @*** factor & constante voor capaciteitsfunctie ***
      IF 0 < r[4]
        fact[1] ← 1
        x ← (r[1]*r[3]) - r[2]*r[2]
        fact[1] ← ((r[1]*r[4]) - r[2]*r[5]) / x IF x ≠ 0
        intc[1] ← (r[4] - fact[1]*r[3]) / r[2]
      END
      IF 0 < s[4]
        fact[2] ← 1
        x ← (s[1]*s[3]) - s[2]*s[2]
        fact[2] ← ((s[1]*s[4]) - s[2]*s[5]) / x IF x ≠ 0
        intc[2] ← (s[4] - fact[2]*s[3]) / s[2]
      END
    END
  END
  FOR m ← 2 TO 7
    deb_aant[1] ← deb_aant[1] + deb_aant[m]
  END
END

FOR EACH sluis IN sluizen
  knoop1 ← nrnode OF kant[1]
  knoop2 ← nrnode OF kant[2]
  text ← nrnode OF kant[1] | ".. SLUIS .." | nrnode OF kant[2]
  CALL char_in_string(<text>, 30, 21, "OPERATIONELE GEGEVENS", 1)
  nieuw ← TRUE
  CALL kop
  regeloph ← 2
  CALL kop

```

```

WRITE #10;"OVERZICHT PER KOLK";#13;"_____ " TO rapport
  WITH IMAGE *Aaa
cap[1,1] ← 0
cap[1,2] ← 0 @ CAP[i,j]: i=1 TONNAGE, i=2 AANTAL, j = RICHTING @
cap[2,1] ← 0
cap[2,2] ← 0
FOR EACH kolk IN alle_kolken
  regeloph ← 13 + 8 * debietbepaling
  CALL kop
  THIS kolkstat ← kolk_stat[1]
  nextstat ← kolk_stat[2]
  WRITE #10;"KOLK";nrkolk|" " ;#13;"_____ " TO rapport
    WITH IMAGE *AΔa*a
  x ← intc[1] + intc[1] OF nextstat
  y ← fact[1]
  z ← fact[1] OF nextstat
  WRITE #10;"Tc =" ;60*x;"+" ;60000*y;"* G1/1000 +" ;60000*z;"* G2/1000"
    TO rapport WITH IMAGE *aΔΔ*****.ΔΔΔ*****.ΔaΔΔ*****.Δa
  t ← x + (y*kgem[3]) + z*kgem[3] OF nextstat
  IF 0 < t
    cap[1,1] ← cap[1,1] + kgem[3] / t
    cap[1,2] ← cap[1,2] + kgem[3] OF nextstat / t
  END
  x ← intc[2] + intc[2] OF nextstat
  y ← fact[2]
  z ← fact[2] OF nextstat
  WRITE "Tc =" ;60*x;"+" ;60*y;"* N1 "+" ;60*z;"* N2" TO rapport
    WITH IMAGE aΔΔ*****.ΔΔΔ*****.ΔaΔΔ*****.Δa
  t ← x + (y*kgem[1]) + z*kgem[1] OF nextstat
  IF 0 < t
    cap[2,1] ← cap[2,1] + kgem[1] / t
    cap[2,2] ← cap[2,2] + kgem[1] OF nextstat / t
  END
  WRITE #10;"RICHTING";knoop1;knoop2;knoop2;knoop1;"SOM" TO rapport
    WITH IMAGE *AΔ*****Δ*****Δ*****Δ*****Δ*****
  WRITE "GEMIDDELD Gmax =" ;kgem[3];kgem[3] OF nextstat TO rapport
    WITH IMAGE AΔ*****.ΔΔ*****.ΔΔ
  WRITE "GEMIDDELD Nmax =" ;kgem[1];kgem[1] OF nextstat TO rapport
    WITH IMAGE AΔ*****.ΔΔ*****.ΔΔ

  WRITE "AANTAL SCHUT =" ;totschut;totschut OF nextstat;
    totschut+totschut OF nextstat TO rapport
    WITH IMAGE AΔ*****Δ*****Δ*****
  WRITE "waarvan VOL =" ;volschut;volschut OF nextstat;
    volschut+volschut OF nextstat TO rapport
    WITH IMAGE AΔ*****Δ*****Δ*****
  WRITE "waarvan LEEGOM =" ;leegom;leegom OF nextstat;
    leegom+leegom OF nextstat TO rapport
    WITH IMAGE AΔ*****Δ*****Δ*****
  WRITE "TOTAAL GESCHUT =" ;totover;totover OF nextstat;
    totover+totover OF nextstat TO rapport
    WITH IMAGE AΔ*****Δ*****Δ*****
  IF debietbepaling
    FOR k ← 1 TO 8
      WRITE "DEBIET gem." TO rapport WITH IMAGE A IF k=1
      WRITE " "; "totaal =" TO rapport WITH IMAGE A8A IF k=8
      IF k ≤ 7
        WRITE " ";#34;" " TO rapport WITH IMAGE A8*A2
          IF k≠1
        WRITE NAME OF dag[k];" =" TO rapport WITH IMAGE |ΔA2A
      END
      FOR l ← 1 TO 2
        WRITE tot_debiet[k,l];" " TO rapport
          WITH IMAGE |ΔΔ*****.ΔA
      END
      WRITE " in 1000 m3 per dag" TO rapport WITH IMAGE |A IF k=1

```



```

        WRITE " in 1000 m3 per ";deb_aant[1];" dagen" TO rapport
        WITH IMAGE |A***A IF k=8
    END
END
regeloph ← 22 + 5*debietbepaling
CALL kop
WRITE #10;#10;"CAPACITEIT SLUIS:" TO rapport WITH IMAGE a*A
WRITE #13;" " TO rapport WITH IMAGE |*A
WRITE #10;"RICHTING ton/uur aantal/uur" TO rapport WITH IMAGE *A
WRITE #13;" " TO rapport WITH IMAGE |*A
WRITE knoop1;"-";knoop2|" " ;cap[1,1];cap[2,1] TO rapport
WITH IMAGE ****aaΔΔ*****.**ΔΔ*****.**
WRITE knoop2;"-";knoop1|" " ;cap[1,2];cap[2,2] TO rapport
WITH IMAGE ****aaΔΔ*****.**ΔΔ*****.**
WRITE "totaal";cap[1,1]+cap[1,2];cap[2,1]+cap[2,2] TO rapport
WITH IMAGE ΔaΔΔΔ*****.**ΔΔ*****.**

IF debietbepaling
    FOR l ← 1 TO 2
        xxx[l] ← 0
        FOR EACH kolk IN alle_kolken
            xxx[l] ← xxx[l] + tot_debiet[8,l]
        END
    END
    WRITE #10;#10;"DEBIET in 1000 m3 per ";deb_aant[1];" dagen:"
    TO rapport WITH IMAGE aaa***A
    WRITE #13;" " TO rapport
    WITH IMAGE |*A
    WRITE knoop1;"-";knoop2|" " :";xxx[1] TO rapport
    WITH IMAGE ***aaΔΔΔ*****
    WRITE knoop2;"-";knoop1|" " :";xxx[2] TO rapport
    WITH IMAGE ***aaΔΔΔ*****
END

FOR k ← 1 TO 4
    xxx[k] ← 0
    cc[k] ← 0
END
FOR EACH schipstat IN sschipstats
    FOR kl ← 1 TO 30
        FOR ri ← 1 TO 2
            xxx[ri] ← xxx[ri] + s_cumpastijd[ri,kl]
            xxx[ri+2] ← xxx[ri+2] + s_cumpastijd[ri,kl]*kosten[1,kl]/60
            cc[ri] ← cc[ri] + s_aantal[ri,kl]
        END
    END
END

WRITE #10;#10;"GEMIDDELDE PASSEERTIJD:" TO rapport WITH IMAGE aaA
WRITE #13;" " TO rapport WITH IMAGE |*A
WRITE knoop1;"-";knoop2|" " :";xxx[1]/cc[1] TO rapport
WITH IMAGE ***aaa3*****.** IF cc[1] > 0
WRITE knoop2;"-";knoop1|" " :";xxx[2]/cc[2] TO rapport
WITH IMAGE ***aaa3*****.** IF cc[2] > 0
WRITE "BEIDE RI:"; " " ;(xxx[1]+xxx[2])/cc[1]+cc[2] TO rapport
WITH IMAGE AA2*****.** IF (cc[1]+cc[2]) > 0

WRITE #10;"GEMIDDELDE PASSEERKOSTEN:" TO rapport WITH IMAGE *A
WRITE #13;" " TO rapport WITH IMAGE |*A
WRITE knoop1;"-";knoop2|" " :";xxx[3]/cc[1] TO rapport
WITH IMAGE ***aaΔ*****.** IF cc[1] > 0
WRITE knoop2;"-";knoop1|" " :";xxx[4]/cc[2] TO rapport
WITH IMAGE ***aaΔ*****.** IF cc[2] > 0
WRITE "BEIDE RI:";(xxx[3]+xxx[4])/cc[1]+cc[2] TO rapport
WITH IMAGE AΔ*****.** IF (cc[1]+cc[2]) > 0

```



```
WRITE #10;"TOTALE PASSEERKOSTEN OVER DE SIMULATIETIJD" TO rapport
      WITH IMAGE *A
WRITE " "; " (in duizenden guldens):" TO rapport WITH IMAGE A19A
WRITE .001*(xxx[3]+xxx[4]) TO rapport WITH IMAGE |Δ*****.***
CALL rapport_scheepvaart(1)
END

RETURN
```

## 6.49 RAPPORT VERNAUWINGEN

Deze MACRO voorziet in de rapportage mbt vernauwingen.

```

=====
@
@ RAPPORT_VERNAUWINGEN
@
=====
@=====
@ rapport VERNAUWINGEN @
@=====
LOCALS:
  INTEGER: knoop1 knoop2 i j k l m n kl ri
  REAL    : x y z

FOR EACH vernauwing IN vernauwingen
  knoop1 ← nrnode OF VZIJDE[1]
  knoop2 ← nrnode OF VZIJDE[2]
  text ← nrnode OF vzijde[1]|".  ENGTE  ."|nrnode OF vzijde[2]
  CALL char_in_string(<text>,30,21,"OPERATIONELE GEGEVENS",1)
  nieuw ← TRUE
  CALL kop
  regeloph ← 2
  CALL kop

  WRITE #10;"type verkeersafhandeling:" TO rapport WITH IMAGE *A
  WRITE "ontmoetingstabel" TO rapport
    WITH IMAGE |ΔA IF regeling = "TAB"
  IF regeling = "VL"
    WRITE "verkeerslichten," TO rapport WITH IMAGE |ΔA
    WRITE "star" TO rapport WITH IMAGE |ΔA IF star
    WRITE "verkeersafhankelijk" TO rapport WITH IMAGE |ΔA IF star=FALSE
  END

  IF star=FALSE & regeling = "VL"
    i ← info_storestream(" ",1|"_Engbenut_"|nrvernauwing,<info1>)
    j ← info_storestream(" ",2|"_Engbenut_"|nrvernauwing,<info2>)

    i ← info1[1] * i = 0 @ # waarnemingen kant 1
    j ← info2[1] * j = 0 @ " " " 2

    regeloph ← 4
    CALL kop
    WRITE #10;"ENGTE BENUTTING";knoop1;"-";knoop2|" " ;knoop2;"-"
      ;knoop1|" " ;"SOM" TO rapport
    WITH IMAGE *AΔΔΔΔ***aaΔΔ***aaΔΔΔΔa
    WRITE #13;" " TO rapport
    WITH IMAGE |*A
    WRITE "gemiddelde tijd:" TO rapport WITH IMAGE A
    WRITE info1[4] TO rapport WITH IMAGE |Δ*****.** IF i > 0
    WRITE " - " TO rapport WITH IMAGE |Δa IF i = 0
    WRITE info2[4] TO rapport WITH IMAGE |Δ*****.** IF j > 0
    WRITE " - " TO rapport WITH IMAGE |Δa IF j = 0
    WRITE ((info1[4]*i) + (info2[4]*j))/i+j TO rapport
    WITH IMAGE |ΔΔ*****.** IF 0 < i+j
    WRITE "aantal keren :";i;j;i+j TO rapport
    WITH IMAGE AΔ*****ΔΔΔΔ*****ΔΔΔΔΔ*****
  END

  FOR k ← 1 TO 4
    xxx[k] ← 0
    cc[k] ← 0

```

```

END
FOR EACH schipstat IN vschipstats
  FOR kl ← 1 TO 30
    FOR ri ← 1 TO 2
      xxx[ri] ← xxx[ri] + s_cumpastijd[ri,kl]
      xxx[ri+2] ← xxx[ri+2] + s_cumpastijd[ri,kl]*kosten[3,kl]/60
      cc[ri] ← cc[ri] + s_aantal[ri,kl]
    END
  END
END

regeloph ← 14
CALL kop
WRITE #10;#10;"GEMIDDELDE PASSEERTIJD:" TO rapport WITH IMAGE aaA
WRITE #13; " " TO rapport WITH IMAGE |*A
WRITE knoop1;"-";knoop2|" " :";" " ;xxx[1]/cc[1] TO rapport
  WITH IMAGE ***aaa3*****.** IF cc[1] > 0
WRITE knoop2;"-";knoop1|" " :";" " ;xxx[2]/cc[2] TO rapport
  WITH IMAGE ***aaa3*****.** IF cc[2] > 0
WRITE "BEIDE RI:";" " ;(xxx[1]+xxx[2])/cc[1]+cc[2] TO rapport
  WITH IMAGE AA2*****.** IF (cc[1]+cc[2]) > 0

WRITE #10;"GEMIDDELDE PASSEERKOSTEN:" TO rapport WITH IMAGE *A
WRITE #13;" " TO rapport WITH IMAGE |*A
WRITE knoop1;"-";knoop2|" " :";xxx[3]/cc[1] TO rapport
  WITH IMAGE ***aaΔ*****.** IF cc[1] > 0
WRITE knoop2;"-";knoop1|" " :";xxx[4]/cc[2] TO rapport
  WITH IMAGE ***aaΔ*****.** IF cc[2] > 0
WRITE "BEIDE RI:";(xxx[3]+xxx[4])/cc[1]+cc[2] TO rapport
  WITH IMAGE AΔ*****.** IF (cc[1]+cc[2]) > 0

WRITE #10;"TOTALE PASSEERKOSTEN OVER DE SIMULATIETIJD" TO rapport
  WITH IMAGE *A
WRITE " " ;" (in duizenden guldens):" TO rapport WITH IMAGE A19A
WRITE .001*(xxx[3]+xxx[4]) TO rapport WITH IMAGE |Δ*****.**
CALL rapport_scheepvaart(3)
END

RETURN

```

## 6.50 RESULTAAT\_REGEL

Deze MACRO is bestemd voor de uitvoer naar het bestand dat gekoppeld is aan ONDERZOEK.  
Momenteel is dit een dummy MACRO.

```
@=====
@
@ RESULTAAT_REGEL
@
@=====
```

```
@=====
@      Voeg een zelf samen te stellen regel
@      met kenmerken & resultaten van deze simulatie toe
@      aan het bestand, dat aan ONDERZOEK is gekoppeld (#9=tab)
@=====
```

RETURN



## 6.51 SCHEPEN\_NAAR\_SLUIS

Deze MACRO onderzoekt of er nog schepen onderweg zijn naar een knoop die vóór een gegeven tijdstip bij de sluis zullen zijn.

Parameters:

AF_TE_LEGGEN	De in-zicht afstand
KNOOPNR	Het nummer van de knoop
AANKOMSTTIJD	De tijdhorizon

```

@=====
@
@ SCHEPEN_NAAR_SLUIS
@
@=====

@=====
@ Zoek naar schepen die op een vaarweg naar KNOOPNR varen @
@ en voor AANKOMSTTIJD bij THIS SLUIS zijn @
@=====
PARAMETER:
  REAL : af_te_leggen
  INTEGER: knoopnr
  REAL : aankomsttijd
LOCAL:
  INTEGER: k
  REAL : x

start:
  THIS node ← FIRST node IN nodes WITH nrnode=knoopnr
                                     Is er nog een niet onderzochte vaarweg ?
  THIS vaarweg ← FIRST vaarweg IN vaarwegen
                                     THAT BELONGS NOT TO voorkennisboom
                                     & raai[1] IS THIS node | raai[2] IS THIS node
  RETURN 0 IF THIS vaarweg IS NONE
                                     Ja

  JOIN THIS vaarweg TO voorkennisboom
  THIS schip ← FIRST schip IN vaarvak WITH doel IS THIS node
               & THIS sluis BELONGS TO vaarplan
               & ((eta + af_te_leggen/snelheid*1000) ≤ aankomsttijd)
               & (passen(10)=0) & doel IS kant[side] | categorie[klas]≠3
                                     Is er een schip dat aan de voorwaarden voldoet ?
  RETURN (eta + af_te_leggen/snelheid*1000) - NOW
               IF THIS schip IS NOT NONE

                                     Nee, zoek recursief verder
  k ← nrnode OF raai[1]
  k ← nrnode OF raai[2] IF raai[1] IS THIS node
  x ← schepen_naar_sluis(af_te_leggen+afstand,k,aankomsttijd)
  RETURN x IF 0 < x
  REPEAT FROM start

```

## 6.52 SCHIPMOD

Dit MODULE bevat de procesbeschrijving van een schip.

```
@=====
@
@ SCHIPMOD
@
@=====

@=====
@ Procesbeschrijving van het SCHIP @
@=====

vaar:
                                Werk de hele route af
WHILE doel IS NOT NONE
                                Switch naar gelang de plaats
    GOTO varen                IF cur_plek BELONGS TO vaarwegen
    GOTO bij_brug             IF cur_plek BELONGS TO bruggen
    GOTO bij_vernauwing       IF cur_plek BELONGS TO vernauwingen
    GOTO bij_sluis

    @-----@
    @ schip in vrije vaart @
    @-----@

varen:
    CALL query_registratie(passeer_knoop)
    THIS vaarweg ← cur_plek
    cumafst      ← cumafst + afstand
    eta          ← NOW + afstand / snelheid * 1000

    @-----@
    @ Test of het een SPECIAAL schip is, dat zich eerder moet melden @
    @-----@
    IF speciaal
                                Speciaal schip, kijk wat verder vooruit
        plek ← SUCC OF cur_plek IN vaarplan
        IF plek IS NOT NONE
            plek ← SUCC OF plek IN vaarplan
            IF plek IS NOT NONE
                                Schip moet voor-aanmelding doen
                CALL schutplanning(voor_aanmelding,0) OF plek
                IF plek BELONGS TO sluizen
                    END
            END
        END
    IF meldingen OF doel IS NOT NONE
                                Schip moet zich melden
        plek ← SUCC OF cur_plek IN vaarplan
        CALL schutplanning(schip_bij_meldraai,0) OF plek
                                IF plek BELONGS TO sluizen
        CALL vernauwingsgarantie IF plek BELONGS TO vernauwingen & vspeciaal
        ENTER meldingen OF doel IF plek BELONGS TO bruggen
        @-----@
        @ Ga na of het SCHIP geregistreerd moet worden @
        @-----@
        IF plek BELONGS NOT TO vernauwingen
            refnode ← doel
            refschip ← LAST schip IN vaarvak WITH refnode IS doel
            IF refschip IS NOT NONE
                eta ← eta OF refschip IF eta ≤ eta OF refschip
            END
        END
    END
```

```

END
ENTER vaarvak
x ← eta - NOW
WORK x IF x > 0
LEAVE vaarvak OF cur_plek
GOTO vervolg

@=====
@ SCHIP komt bij een SLUIS @
@=====
bij_sluis:
    @*** wachten op instructies ***@
    WAIT WHILE instructie IS NONE
        IF wachttijd_onmogelijk OF cur_plek = FALSE

    CALL query_registratie(passeer_knoop)
    vorige_mijlpaal ← SUCC OF vorige_mijlpaal IN mpalen
    tw ← NOW
    wachtend ← TRUE

    WAIT WHILE instructie IS NONE IF wachttijd_onmogelijk OF cur_plek
    tbijsluis ← NOW
    GOTO kolkproces IF instructie BELONGS NOT TO sluizen
    @-----@
    @ SCHIP gaat de wachtruimte van de SLUIS in @
    @-----@
    ENTER wachtruimte[richting] OF instructie
    CALL schutplanning(schip_bij_sluis,0) OF instructie
    PASSIVATE

    LEAVE wachtruimte[richting] OF zijnsluis OF instructie

    @-----@
    @ SCHIP kan naar een KOLK @
    @-----@
kolkproces:
    CALL query_registratie(naar_opstelruimte)
    ENTER opstelruimte[richting] OF instructie
    @-----@
    PASSIVATE @ De controle over het schip wordt overgenomen @
    @ door het sluispersoneel @
    @-----@

vaar_verder:
    garantie ← FALSE @ geef het schip een schone lei @
    tbijsluis ← 0
    ts ← NOW - ts
    tw ← tw * 60
    tl ← tl * 60
    ts ← ts * 60
    k ← nrsluis OF zijnsluis OF instructie
    x ← tw + tl + ts
    STORE x AS "Passt.S "|k
    STORE tw AS "Wachtt.S "|k
    STORE tl AS "Overlt.S "|k
    STORE ts AS "Schutt.S "|k

    THIS schipstat ← FIRST schipstat
        IN sschipstats OF zijnsluis OF instructie
        WITH s_volgnr = nrkolk OF instructie
GOTO noteren

@=====
@ SCHIP komt bij een BRUG @
@=====
bij_brug:

```

```

WAIT WHILE instructie IS NONE

tw ← NOW
wachtend ← TRUE
CALL query_registratie(passeer_knoop)
vorige_mijlpaal ← SUCC OF vorige_mijlpaal IN mpalen

IF onderdoor & ((open OF instructie = FALSE) |
                (doorgang < nrdoorgangen OF instructie))
    @-----@
    @ SCHIP gaat onder een VASTE DOORGANG door @
    @-----@
onderbrugdoor:
    tw ← NOW - tw
    tl ← 0
    ts ← VALUE OF vastebrug[gel]
        AT(breedte/wijdte[doorgang] OF instructie) MINUTES
    CALL query_registratie(0-doorgang+10)
    CALL query_registratie(start_doorvaren)
    WORK ts

    GOTO noteer
END
ENTER wachtrij[richting] OF instructie
ENTER brugplan[richting] OF zijnwachter OF instructie
    IF onderdoor & (doorgang ≥ nrdoorgangen OF instructie)
PASSIVATE

noteer:
    tw ← tw * 60
    tl ← tl * 60
    ts ← ts * 60
    k ← nrbrug OF instructie
    x ← tw + tl + ts
    STORE x AS "Passt.B "|k
    STORE tw AS "Wachtt.B "|k
    STORE tl AS "Overlt.B "|k
    STORE ts AS "Doorvt.B "|k
    THIS schipstat ← FIRST schipstat IN bschipstats OF instructie
        WITH s_volgnr = doorgang
GOTO noteren

@-----@
@ SCHIP komt bij een VERNAUWING @
@-----@
bij_vernauwing:
    tw ← NOW
    wachtend ← TRUE
    instructie ← cur_plek
    richting ← 1 + start IS vzijde[2] OF instructie

    CALL query_registratie(passeer_knoop)
    vorige_mijlpaal ← SUCC OF vorige_mijlpaal IN mpalen

    @-----@
    @ SCHIP gaat na ofhet de VERNAUWING kan/mag passeren @
    @-----@
    IF passage_vernauwing = FALSE
        ENTER vwachtrij[richting]
        CALL display_vernauwing
        WAIT WHILE passage_vernauwing = FALSE
    END

    @-----@
    @ SCHIP passeert VERNAUWING @
    @-----@

```



```

THIS vernauwing ← instructie
@*** x: tijd nodig om met achterkant in de vernauwing te komen ***
x ← 0
IF THIS schip BELONGS TO vwachtrij[richting]
  x ← plustijd[klas] MINUTES IF NOW > vrijstelling[richting] + 1 SECOND
  LEAVE vwachtrij[richting]
END
x ← x + lengte / vvaarsnelheid[klas,gel] * 1000
LEAVE garantieschepen IF THIS schip BELONGS TO garantieschepen
tw ← NOW - tw IF wachtend
tl ← 0 IF wachtend
tl ← NOW - tl IF wachtend=FALSE
ts ← NOW
cumafst ← cumafst + v_afstand
eta ← NOW + x + v_afstand / vvaarsnelheid[klas,gel] * 1000
refnode ← doel
refschip ← LAST schip IN engte WITH refnode IS doel
IF refschip IS NOT NONE
  y ← volglengte[categorie[klas],categorie[klas OF refschip]]
    * lengte OF refschip
    @-----@
    @ wanneer is voorganger ver genoeg weg @
    @ en dus dit schip over de streep? @
    @-----@
    z ← v_afstand/eta OF refschip-vrijstelling[richting] @ gem.snelh.
    z ← vrijstelling[richting] + (lengte+y)/z @ meters/uur
    x ← MAX(x,z-NOW) @ tijdstip
    @-----@
    @ de laatste lengte+volgafstand meters @
    @ kan het schip met eigen snelheid @
    @-----@
    eta ← NOW + x + v_afstand / vvaarsnelheid[klas,gel] * 1000
    eta ← MAX(eta,eta OF refschip + (lengte+y)
      / vvaarsnelheid[klas,gel] * 1000)
    @-----@
END
ebenut[richting] ← NOW IF engte IS EMPTY
ENTER engte
invaren_vrij[richting] ← FALSE
CALL query_registratie(start_doorvaren)
CALL display_vernauwing
@-----@
@ Na THIS SCHIP gaat het licht op 'rood' ivm MAXGROEN @
@ bij verkeersafhankelijke lichten @
@-----@
IF (regeling = "VL") & star=FALSE
  IF garantieschepen IS EMPTY & eindgroen < NOW + x
    CANCEL instructie
    REACTIVATE instructie WITH DELAY MAX(0,eindgroen-NOW)
    FROM op_rood IN manage_vernauwing
  END
END
WORK x

invaren_vrij[richting] ← TRUE
vrijstelling[richting] ← NOW @ tijdstip schip achterstevan in vernauwing
CALL display_vernauwing
WORK eta-NOW IF NOW < eta

THIS vernauwing ← instructie
LEAVE engte
CALL display_vernauwing
k ← nrvernauwing
ts ← NOW - ts
tw ← tw * 60
tl ← tl * 60
ts ← ts * 60

```

```

STORE 60 * NOW-ebenut[richting] AS richting|"_Engbenut_"|k
        IF engte IS EMPTY & (star=FALSE) & regeling = "VL"
garantie ← FALSE
x ← tw + tl + ts
STORE x AS "Passt.V "|k
STORE tw AS "Wachtt.V "|k
STORE tl AS "Overlt.V "|k
STORE ts AS "Doorvt.V "|k

THIS schipstat ← FIRST schipstat IN vschipstats OF instructie
        WITH s_volgnr = richting
GOTO noteren @ voor de duidelijkheid

noteren:
    s_cumpastijd[richting,klas] ← s_cumpastijd[richting,klas] + x
    s_aantal[richting,klas] ← s_aantal[richting,klas] + 1
    s_nover[richting,klas] ← s_nover[richting,klas] + 1 IF 0 < tl
instructie ← NONE

@=====
@ SCHIP vervolgt zijn route @
@=====

vervolg:
    start ← doel
    doel ← SUCC OF start IN route
    cur_plek ← SUCC OF cur_plek IN vaarplan
    vorige_mijlpaal ← SUCC OF vorige_mijlpaal IN mpalen
END
CALL query_registratie(passeer_knoop)

TERMINATE @**** einde van de route ****@

```

### 6.53 SCHUTSTRATEGIE\_GELYK

Deze MACRO voorziet in een schutplanning voor een sluis met schutstrategie en kolken van gelijke grootte.

Parameters:

TYPE           Reden voor aanpassing van de schutplanning  
SLUISKANT De kant van de sluis

```
@=====
@
@ SCHUTSTRATEGIE_GELYK
@
@=====
```

```
@=====
@           Pas de schutplanning aan voor THIS SLUIS           @
@Er is een schutstrategie & de sluis heeft kolken van gelijke grootte@
@zodat er geen reden is schepen buiten de kolken te laten wachten   @
@=====
```

PARAMETER:

INTEGER: type  
INTEGER: sluiskant

LOCAL:

INTEGER: k m  
REAL     : x y

Switch naar gelang de aanleiding

```
GOTO melding_schip   IF type = schip_bij_meldraai
GOTO vul_aan         IF type = nieuw_opstelplan
GOTO vrije_kolk       IF type = vrije_kolk
GOTO bepaal_deadline IF type = bepaal_deadline
RETURN
```

```
@-----@
@ De CENTRALIST behandelt een melding van een schip @
@-----@
```

melding\_schip:

```
THIS sluis ← plek
sluiskant ← 1 + doel IS kant[2]
richting ← sluiskant
m ← 3
y ← 1e6
bestkolk ← NONE
```

Onderzoek de kolken

FOR EACH kolk IN kolken

Past het schip ?

```
k ← passen(sluiskant)
m ← MIN(m, k)
IF k = 0
```

Schip kan mee

```
    x ← kolk_preferentie(sluiskant)
    IF x < y
        bestkolk ← THIS kolk
        y ← x
    END
```

END

END

THIS kolk ← bestkolk

```
@-----@
@ Fysieke beperkingen SCHIP voor deze SLUIS @
@-----@
```

IF m=3

```
    JOIN THIS schip TO onmogelijk[sluiskant] RANKED BY 45 * eta
    instructie ← NONE
```

```

    RETURN
END

@-----@
@ SCHIP kan geplaatst worden in OPSTELRUIMTE van THIS KOLK @
@-----@
IF m=0
    CALL herdeel(sluiskant)
    instructie ← THIS kolk
    GOTO aktiveer_kolk
END
@-----@
@ SCHIP kan niet in OPSTELRUIMTE en moet naar WACHTRUIMTE @
@-----@
IF m=2
    JOIN THIS schip TO remanent[sluiskant] RANKED BY 45 * eta
    instructie ← THIS sluis
    FOR EACH kolk IN kolken WITH bezig = FALSE
        CANCEL THIS kolk IF THIS kolk IS ACTIVE
        REACTIVATE THIS kolk
    END
END
RETURN

@-----@
@ De CENTRALIST moet een nieuw OPSTELPLAN maken voor this KOLK @
@               aan sluis kant                               @
@-----@
vul_aan:
    THIS sluis ← zijnsluis
    RETURN IF THIS kolk BELONGS NOT TO kolken

    FOR EACH schip IN onmogelijk[sluiskant]
        m ← passen(sluiskant)
        IF m=0
            REMOVE THIS schip FROM onmogelijk[sluiskant]
            CALL herdeel(sluiskant)
            instructie ← THIS kolk
        END
        IF m=2
            REMOVE THIS schip FROM onmogelijk[sluiskant]
            JOIN THIS schip TO remanent[sluiskant] RANKED BY 45 * eta
            instructie ← THIS sluis
        END
    END
END
FOR EACH schip IN remanent[sluiskant]
    IF passen(sluiskant) ≤ 1
        REMOVE THIS schip FROM remanent[sluiskant]
        CALL herdeel(sluiskant)
        REACTIVATE THIS schip
        IF THIS schip BELONGS TO wachtruimte[sluiskant]
            instructie ← THIS kolk
        END
    END
END
RETURN IF opstelplan[1] IS EMPTY & opstelplan[2] IS EMPTY

aktiveer_kolk:
    IF bezig = FALSE
        RETURN IF wacht_op_periode
        bezig ← TRUE
        wacht_op_deadline ← FALSE
        CANCEL THIS kolk IF THIS kolk IS ACTIVE
        REACTIVATE THIS kolk FROM vaarin IN kolkmod
    END

```

Nu mag de kolk de regie weer overnemen

Als dat tenminste kan ...



RETURN

```
@-----@
@ De CENTRALIST bekijkt of BEST_KOLK door mag gaan met schutten ivm @
@       • periodiek schutten @
@       • minstens ,,n schip in een opstelplan @
@-----@
```

vrije\_kolk:

x ← buiten\_schutperiode

RETURN x IF 0 < x

RETURN -1 IF opstelplan[1] IS EMPTY & opstelplan[2] IS EMPTY

RETURN 0

```
@-----@
@ De CENTRALIST bepaalt wanneer THIS KOLK uiterlijk overgezet moet worden @
@-----@
```

bepaal\_deadline:

THIS\_sluis ← zijns\_luis

RETURN 0 IF voldoende\_benutting = TRUE

x ← bepaal\_uitstel

RETURN 0 IF x ≤ 0

RETURN x HOURS

## 6.54 SCHUTSTRATEGIE\_NKOLK

Deze MACRO voorziet in een nieuwe schutplanning voor een sluis met strategie en meerdere, ongelijke, kolken.

Parameters:

TYPE Reden voor aanpassing van de schutplanning

SLUISKANT De kant van de sluis

```
@=====
@
@ SCHUTSTRATEGIE_NKOLK
@
@=====
```

```
@=====
@ kritieke tijdstip alleen voor schepen @
@ waarvan de kleinste kolk beschikbaar is @
@=====
@ Pas de schutplanning aan voor THIS SLUIS @
@ Er is een schutstrategie & de sluis heeft méérdere kolken @
@=====
```

PARAMETER:

INTEGER: type

INTEGER: sluiskant

LOCAL:

INTEGER: k i

REAL : x uitstel

Switch naar gelang de aanleiding

GOTO melding\_schip IF type = schip\_bij\_meldraai

GOTO beslisboom IF type = sluis\_zelf

IF type = schip\_bij\_sluis

THIS sluis ← instructie

GOTO bij\_sluis

END

IF (type = nieuw\_opstelplan) | type = vrije\_kolk

THIS sluis ← zijnsluis

GOTO beslisboom

END

RETURN

```
@=====
@ THIS SCHIP meldt zich bij de meldraai @
@=====
```

melding\_schip:

THIS sluis ← plek

sluiskant ← 1 + doel IS kant[2]

richting ← sluiskant

Is er een passende kolk ?

THIS kolk ← FIRST kolk IN kolken WITH passen(10)=0

IF THIS kolk IS NOT NONE

Ja

instructie ← THIS sluis

RETURN

END

Nee, onmogelijk toe te wijzen

instructie ← NONE

ENTER onmogelijk[sluiskant] AND TAKE PLACE RANKED BY 45 \* eta

RETURN

```
@=====
@ THIS SCHIP komt bij sluis aan @
@=====
bij_sluis:
```

```

ENTER remanent[richting] AND TAKE PLACE RANKED BY 45 * eta
kleinste_kolk ← FIRST kolk IN alle_kolken WITH passen(10)=0

```

Waarschuw de sluismeester

```

CALL zet_wekker(0)
RETURN

```

```

@=====
@ STAP 1: Valt er wat te beslissen? @
@=====

```

beslisboom:

```

stack ← NEW SET CALLED "stack"
JOIN THIS schip TO stack IF THIS schip IS NOT NONE
JOIN THIS kolk TO stack IF THIS kolk IS NOT NONE
JOIN THIS sluis TO stack IF THIS sluis IS NOT NONE

```

```

@- A -----
@ Verwerk wachtende schepen indien mogelijk @
@-----

```

```

FOR k ← 1 TO 2
  FOR EACH kolk IN alle_kolken WITH opstelplan[k] IS NOT EMPTY
    & (bezig=FALSE) | side ≠ k
    CALL plan_this_kolk_in(k)
  END
  FOR EACH schip IN onmogelijk[k]
    THIS kolk ← FIRST kolk IN kolken WITH passen(10) ≤ 2
    IF THIS kolk IS NOT NONE
      REMOVE THIS schip FROM onmogelijk[k]
      instructie ← THIS sluis
    END
  END
END

```

```

@- B -----
@ Zijn er nog beschikbare kolken @
@-----

```

stap\_1b:

```

THIS kolk ← FIRST kolk IN kolken WITH bezig=FALSE
GOTO einde IF THIS kolk IS NONE @*** buiten periodiek zijn
@*** alle kolken BEZIG

```

```

@=====
@ STAP 2: Kan er op basis van het benuttingscriterium geschut worden? @
@=====

```

```

@- A1 -----
@ Bepaal van alle nog in te plannen schepen @
@ de 'kleinste' kolk waar elk schip in 'past' @
@-----

```

```

THIS kolk ← FIRST OF kolken
n ← SORTINGPAR OF THIS kolk IN kolken
FOR k ← 1 TO 2
  FOR EACH schip IN remanent[k]
    ref[1] ← FIRST kolk IN kolken WITH passen(10) = 0
    IF ref[1] IS NOT NONE
      m ← SORTINGPAR OF ref[1] IN kolken
      IF n < m
        THIS kolk ← ref[1]
        @ 'kleinste kolk' waar elk schip in 'past'
      n ← m
      x ← 1e6
    END
    IF eta < x
      ref[2] ← THIS schip @*** kolkbepalende schip ***
      x ← eta
    END
  END

```

```

        END
    END
END
GOTO stap_2b IF THIS kolk IS FIRST OF kolken
@-----@
@ nieuwe kolk moet gebruikt worden! @
@-----@
GOTO stap_2b IF opstelplan[richting OF ref[2]] IS NOT EMPTY

@- A2 -----@
@   Kan het kolkbepalende schip mee met de   @
@ eerst volgende schutting v/d betreffende kolk @
@-----@

hulp2 ← NEW SET
i ← richting OF ref[2]
JOIN EACH punt IN indeling[i] TO hulp2
THIS schip ← FIRST OF remanent[i]
WHILE THIS schip IS NOT NONE
    IF THIS schip IS ref[2]
        x ← passen(i)
        indeling[i] ← hulp2
        GOTO stap_2b IF 2 ≤ x
        GOTO het_schip_kan_mee
    END
    CALL voegtoe(i) IF passen(i) < 2
    THIS schip ← SUCC OF THIS schip IN remanent[i]
END

het_schip_kan_mee:
    CALL plan_this_kolk_in(i)
    REPEAT FROM stap_1b

@- B1 -----@
@   Is er een 'kleinste' beschikbare kolk   @
@ waarvan het benuttingscriterium is overschreden? @
@-----@
stap_2b:
    THIS kolk ← FIRST kolk IN kolken WITH (bezig = FALSE)
                                         & voldoende_benutting = TRUE
    IF THIS kolk IS NOT NONE
        CALL plan_this_kolk_in(side)
        bezig ← TRUE

        Activeer kolk
        REACTIVATE THIS kolk IF THIS kolk IS NOT ACTIVE
        REPEAT FROM stap_1b
    END

@=====
@ STAP 3: Moet er geschut worden? @
@=====

@-----@
@   Bepaal voor alle nog in te plannen schepen het kritieke tijdstip   @
@ op basis van de 'kleinste' beschikbare kolk waar het schip in 'past' @
@-----@
uitstel ← bepaal_uitstel

GOTO einde IF uitstel=-1000    @ niets helpt!
IF uitstel < 5 SECONDS
    THIS kolk ← FIRST kolk IN kolken WITH nrkolk=n
    CALL plan_this_kolk_in(side)
    CALL plan_this_kolk_in(3-side) IF geen_uitstel[3-side]
    bezig ← TRUE
    REACTIVATE THIS kolk IF THIS kolk IS NOT ACTIVE
    REPEAT FROM stap_1b

```



```
END
CALL zet_wekker(uitstel)

einde:
THIS schip ← FIRST schip IN stack
THIS kolk ← FIRST kolk IN stack
THIS sluis ← FIRST sluis IN stack
IF type = sluis_zelf
    FOR EACH kolk IN alle_kolken THAT BELONGS NOT TO kolken
        & THIS kolk IS NOT ACTIVE
        & opstelplan[1] IS NOT EMPTY
        | opstelplan[2] IS NOT EMPTY
        REACTIVATE THIS kolk
    END
END
IF type=vrije_kolk
    x ← buiten_schutperiode
    RETURN x IF 0 < x
    RETURN -1 IF x = -1
    IF THIS kolk BELONGS NOT TO kolken
        & opstelplan[1] IS NOT EMPTY | opstelplan[2] IS NOT EMPTY
        RETURN 0
    END
    RETURN -1 IF bezig=FALSE
    RETURN 0
END
RETURN
```

## 6.55 SLUIS\_REGISTRATIE

Deze MACRO registreert de toestand van de sluis.

Parameters:

SCHUTTING TRUE als het een schutting betreft, anders FALSE.

```

@=====
@
@ SLUIS_REGISTRATIE
@
@=====

PARAMETER:
    LOGICAL: schutting
LOCAL:
    INTEGER: offset
    REAL    : x

    @-----@
    @ registratie @
    @-----@
    THIS generator ← schutregistrator
    THIS bladzij   ← LAST OF bladen
    CALL wegschrijven_bladen(FALSE) IF regels = regels_per_bladzij
    offset         ← regellengte * regels
    regels         ← regels + 1
    volle_regels   ← regels
    bladinhoud[offset+1] ← nrnode OF kant[1] OF zijnsluis
    bladinhoud[offset+2] ← nrkolk
IF schutting
    bladinhoud[offset+3] ← NOW-duur
    bladinhoud[offset+5] ← opprend
    bladinhoud[offset+7] ← wacht_op_voorw
    bladinhoud[offset+8] ← cur_debiet
END
IF schutting=FALSE
    bladinhoud[offset+3] ← buiten_bedrijf
    bladinhoud[offset+5] ← -1
END
    bladinhoud[offset+4] ← NOW
    bladinhoud[offset+6] ← idle_time IF 0 < idle_time
    RETURN IF schutting=FALSE

    THIS kolkstat ← kolk_stat[3-side]
    r[1] ← r[1] + 1
    r[2] ← r[2] + totlaadverm
    r[3] ← r[3] + totlaadverm*totlaadverm
    r[4] ← r[4] + totlaadverm*duur
    r[5] ← r[5] + duur
    s[1] ← s[1] + 1
    s[2] ← s[2] + aantal
    s[3] ← s[3] + aantal*aantal
    s[4] ← s[4] + aantal*duur
    s[5] ← s[5] + duur
    leegom ← leegom + 1 IF aantal < 1
    totschut ← totschut + 1
    totover ← totover + aantal
    IF kmax = TRUE
        volschut ← volschut + 1
        kgem[1] ← kgem[1] + aantal
        kgem[2] ← kgem[2] + duur
        kgem[3] ← kgem[3] + totlaadverm
    END
    x ← nrkolk + .5*kmax
    x ← 0-nrkolk IF aantal=0

```

```
STORE x AS "Schutten. "|nrsluis  
STORE 0 AS "Schutten. "|nrsluis  
RETURN
```

## 6.56 SWITCH

Deze MACRO verwisselt twee knopen opdat de laagstgenummerde voorop staat.

```
@=====
@
@ SWITCH
@
@=====

@=====
@           Deze MACRO zorgt ervoor dat i[1] < i[2]           @
@           en wordt aangeroepen in:                         @
@           MOD MAINMOD                                       @
@           MAC LEES_VAARWEGEN                                @
@           MAC LEES_VERNAUWINGEN                             @
@           MAC LEES_SLUIZEN                                  @
@           MAC LEES_BRUGGEN                                  @
@=====

IF i[2] < i[1]
    k    ← i[1]
    i[1] ← i[2]
    i[2] ← k
END
RETURN
```



## 6.57 TONDERDOOR

Deze MACRO bepaalt de doorvaartijd van een bij een brug.

Parameters:

KEES            aard van het schip  
                  1: eerste schip  
                  2: volgschip  
                  3: lusschip  
                  4: schip in lopende vaart

```
@=====
@
@ TONDERDOOR
@
@=====

@=====
@ Macro voor de bepaling van de DOORVAARTIJDEN van een schip bij een BRUG @
@=====

@-----@
@ KEES = 1 : eerste schip @
@       2 : volgschip     @
@       3 : lusschip      @
@       4 : lopende vaart @
@-----@

PARAMETER:
  INTEGER: kees

LOCALS:
  REAL: x
  REAL: k

                                Dat moet dus niet kunnen
WRITE NOW;"Macro TONDERDOOR is aangeroepen met parameter:";kees
  WITH IMAGE ****.****ΔΔaΔ***** IF (kees < 1) & (4 < kees)

IF (kees = 1) | kees = 3
  x ← VALUE OF brug_1ste[klas,gel] AT(breedte/wijdte[doorgang])
END

IF kees = 2
  x ← VALUE OF brug_volg[klas,gel] AT(breedte/wijdte[doorgang])
  x ← x + bcat12 IF (categorie[klas OF laatsteboot]=1) & categorie[klas]=2
  x ← x + bcat13 IF (categorie[klas OF laatsteboot]=1) & categorie[klas]=3
  x ← x + bcat23 IF (categorie[klas OF laatsteboot]=2) & categorie[klas]=3
END

IF kees = 3
  x ← x + VALUE OF lusbrug[gel] AT(brfuiklengte[richting] + lengte)
END

IF kees = 4
  x ← VALUE OF vastebrug[gel] AT(breedte/wijdte[doorgang])
END

RETURN x MINUTES
```

## 6.58 UITTIJD

Deze MACRO berekent de uitvaart- en volgtijden van een schip bij een kolk.

Parameters:

SLUISKANT De kant van de sluis.

```

@=====
@
@ UITTIJD
@
@=====

@=====
@ Macro voor de berekening van de UITVAAR- en VOLGTIJDEN bij een KOLK @
@=====
PARAMETER:
    INTEGER: sluiskant
LOCAL:
    REAL    : x y

THIS sluis ← zijnsluis

@-----@
@ niet recreatievaart of gemengd schutten @
@-----@
x ← breedte / brdhoofd
IF (categorie[klas] ≠ 3) & gel = 2
    y ← waterstand(s_nrtij[sluiskant],s_peilpar[sluiskant])
    x ← breedte * diepgang / brdhoofd * (y - drempel[sluiskant])
    x ← 1 IF x < 0
END

@-----@
@ eerste schip dat uitvaart @
@-----@
IF preschip IS NONE
    y ← VALUE OF tijduit_1ste[klas,gel] AT(x)
END

@-----@
@ volgschip @
@-----@
IF preschip IS NOT NONE
    y ← VALUE OF tijduit_volg[klas,gel] AT(x)
    IF (gemengd_schutten=FALSE)
        & categorie[klas OF preschip] < categorie[klas]
        y ← y + ucat12 IF (categorie[klas OF preschip]=1) & categorie[klas]=2
        y ← y + ucat13 IF (categorie[klas OF preschip]=1) & categorie[klas]=3
        y ← y + ucat23 IF (categorie[klas OF preschip]=2) & categorie[klas]=3
    END
END
RETURN y MINUTES

```

## 6.59 VERNAUWINGSGARANTIE

Deze MACRO handelt de aanvraag voor invaargarantie bij een vernauwing af.

```

@=====
@
@ VERNAUWINGSGARANTIE
@
@=====

@=====
@ THIS SCHIP vraagt invaartgarantie bij de VERNAUWING @
@=====

LOCAL:
  LOGICAL: eerste

  THIS vernauwing ← plek
  richting ← 1 + doel IS vzijde[2]
  IF garantieschepen IS NOT EMPTY
    Garantie in 2 richtingen is te bar
    RETURN IF richting ≠ richting OF FIRST OF garantieschepen
  END
  garantie ← TRUE
  eerste ← garantieschepen IS EMPTY
  ENTER garantieschepen

  RETURN IF eerste = FALSE

@-----@
@ verkeersafhankelijk @
@-----@

  IF star = FALSE
    Nu kan er ingegrepen worden
    IF (vernuwingskant = richting) & vlicht[richting] = rood
      vlicht[richting] ← groen
      CANCEL plek
      REACTIVATE plek FROM beperkt_wachten IN manage_vernuwing
    END
    IF vlicht[3-richting] = groen
      CANCEL plek
      REACTIVATE plek FROM op_rood IN manage_vernuwing
    END
    RETURN
  END

@-----@
@ starre lichten @
@-----@

  @**** lichten gaan al om naar goede kant ****
  RETURN IF (vernuwingskant ≠ richting) & vlicht[vernuwingskant] = rood

  CANCEL plek

  @**** lichten kunnen meteen op groen als dit nog niet het geval was ****
  REACTIVATE plek FROM starre_lichten IN manage_vernuwing
    IF vernauwingskant = richting

  @**** lichten moeten om naar goede kant ****
  REACTIVATE plek IF vernauwingskant ≠ richting
  RETURN

```

## 6.60 VOEGTOE

Deze MACRO voegt een schip toe aan de indeling van een kolk.

Parameters:

SLUISKANT De kant van de sluis.

Terugkeerwaarde:

1: Gelukt.

-1: Niet gelukt.

Zie ook de toelichting bij de MACRO PASSEN.

```
@=====
@
@ VOEGTOE
@
@=====

@=====
@ THIS SCHIP wordt toegevoegd aan INDELING(sluiskant) @
@ returnwaarde: -1= niet gelukt @
@ 1= wel gelukt @
@=====
PARAMETER:
    INTEGER: sluiskant

@=====
@ Bepaal mogelijke ligplaatsen @
@=====

hulp      ← NEW SET
plaats[1] ← FIRST OF indeling[sluiskant]
plaats[2] ← SUCC OF plaats[1] IN indeling[sluiskant]
plaats[3] ← SUCC OF plaats[2] IN indeling[sluiskant]
k         ← -1
WHILE plaats[3] IS NOT NONE
    l ← 1
    IF yy OF plaats[1] = yy OF plaats[2]
        l ← -1 IF xx OF plaats[2] < xx OF plaats[1]
    END
    IF yy OF plaats[2] = yy OF plaats[3]
        l ← -1 IF xx OF plaats[3] < xx OF plaats[2]
    END
    IF l = k
        plaats[1] ← NEW punt
        xx OF plaats[1] ← xx OF plaats[2]
        yy OF plaats[1] ← yy OF plaats[2]
        yy OF plaats[1] ← yy OF plaats[2] - planbreedte IF l = -1
        FOR EACH punt IN indeling[sluiskant] WITH (xx > xx OF plaats[1])
            & (yy > yy OF plaats[1])
            & (yy < planbreedte + yy OF plaats[1])
            xx OF plaats[1] ← xx IF xx ≥ xx OF plaats[1]
        END
        IF (planlengte ≤ lengkolk - xx OF plaats[1]) & (yy OF plaats[1] ≥ 0)
            & (brdtkolk ≥ planbreedte + yy OF plaats[1])
            THIS punt ← FIRST punt IN indeling[sluiskant]
                WITH (ABS(yy-yy OF plaats[1]) ≤ 0.001)
                | (ABS(yy-planbreedte+yy OF plaats[1]) ≤ 0.001)
            JOIN plaats[1] TO hulp RANKED BY 100*xx OF plaats[1] - 320
                IF THIS punt IS NOT NONE
        END
    END
    plaats[1] ← plaats[2]
    plaats[2] ← plaats[3]
    plaats[3] ← SUCC OF plaats[2] IN indeling[sluiskant]
    k         ← 0 - k
```



END

RETURN -1 IF hulp IS EMPTY

```
@=====
@ Selektie van het beste punt @
@=====
```

```
@-----
@ PLAATS[3] WORDT DE OPLOSSING @
@-----
```

plaats[1] ← FIRST OF hulp  
REMOVE EACH punt IN hulp WITH xx > xx OF plaats[1] FROM hulp

```
z ← xx OF plaats[1]
plaats[3] ← NONE
FOR EACH punt IN hulp
  plaats[1] ← THIS punt
  @-----
  @ raakvlak ONDER @
  @-----
  FOR EACH punt IN indeling[sluiskant]
    WITH (ABS(yy-yy OF plaats[1]) < 0.001) & (z < xx)
      z ← xx
      plaats[3] ← plaats[1]
  END
  @-----
  @ raakvlak BOVEN @
  @-----
  FOR EACH punt IN indeling[sluiskant]
    WITH (ABS(yy - planbreedte+yy OF plaats[1]) < 0.001)
      & (z < xx)
      z ← xx
      plaats[3] ← plaats[1]
  END
END
```

END  
plaats[3] ← FIRST OF hulp IF plaats[3] IS NONE

```
@=====
@ Toevoegen van het schip @
@=====
```

```
@-----
@ 1: voer de coördinaten van de achterkant vh schip in (=vertikale lijn) @
@ 2: verwijder alle punten links van deze lijn @
@ 3: zorg dat de indeling in rechte hoeken blijft lopen @
@-----
```

THIS punt ← NEW punt  
xx ← planlengte + xx OF plaats[3]  
yy ← planbreedte + yy OF plaats[3]  
plaats[1] ← THIS punt @ rechtsboven = rechts achter op het schip

```
THIS punt ← NEW punt
xx ← planlengte + xx OF plaats[3]
yy ← yy OF plaats[3]
plaats[2] ← THIS punt @ rechtsonder = links achter op het schip
REMOVE EACH punt IN indeling[sluiskant]
  WITH (yy ≤ yy OF plaats[1])
    & (yy ≥ yy OF plaats[2])
    & (xx ≤ xx OF plaats[1]) FROM indeling[sluiskant]
THIS punt ← FIRST punt IN indeling[sluiskant] WITH yy ≤ yy OF plaats[2]
JOIN plaats[2] TO indeling[sluiskant] IF THIS punt IS NONE
IF THIS punt IS NOT NONE
  JOIN plaats[2] TO indeling[sluiskant] BEFORE THIS punt
  IF yy < yy OF plaats[2] @***** lijn loopt schuin *****
```

```
      x ← xx
      THIS punt ← NEW punt
      xx      ← x
      yy      ← yy OF plaats[2]
      JOIN THIS punt TO indeling[sluiskant] AFTER plaats[2]
    END
  END
  JOIN plaats[1] TO indeling[sluiskant] BEFORE plaats[2]
  THIS punt ← PRED OF plaats[1] IN indeling[sluiskant]
  IF THIS punt IS NOT NONE
    IF yy > yy OF plaats[1] @***** lijn loopt schuin *****
      x ← xx
      THIS punt ← NEW punt
      xx      ← x
      yy      ← yy OF plaats[1]
      JOIN THIS punt TO indeling[sluiskant] BEFORE plaats[1]
    END
  END
RETURN 1
```

## 6.61 VOLDOENDE\_BENUTTING

Deze MACRO gaat na of de benuttingsgraad van een kolk aan de gestelde minimum voorwaarde voldoet.

Terugkeerwaarde:

TRUE            benuttingsgraad voldoende  
FALSE          benuttingsgraad onvoldoende.

```
@=====
@
@ VOLDOENDE_BENUTTING
@
@=====

@=====
@ minimale benuttingsgraad voor THIS KOLK @
@=====
LOCAL:
  INTEGER: k
  REAL   : x y
  LOGICAL: voldoende

  THIS sluis ← zijnsluis

  RETURN IF mn_schutregime IS NONE
  THIS schutregime ← mn_schutregime

  FOR k ← 1 TO 2
    w[k] ← 0
    FOR EACH schip IN remanent[k] WITH passen(10)=0
      w[k] ← w[k] + planlengte*planbreedte
    END
    FOR EACH schip IN opstelplan[k]
      w[k] ← w[k] + planlengte*planbreedte
    END
    w[k] ← 100*w[k]/lengkok*brdtkolk
  END
  x ← w[side] / benuttingsperc[1]
  y ← w[3-side] / benuttingsperc[2]

  voldoende ← (1 ≤ x) | (1 ≤ y) IF combitype = "BLOK"
  voldoende ← 1 ≤ x + y IF combitype = "LINEAIR"
  voldoende ← 1 ≤ (x_2) + (y_2) IF combitype = "ELLIPS"
  RETURN voldoende
```

Indien er geen schutregime is wordt default de waarde FALSE teruggegeven omdat deze conditie dan niet doorslaggevend mag zijn

Loop alle passende schepen af

Eind test afhankelijk van type

## 6.62 VOORKENNIS

Deze MACRO onderzoekt of er nog interessante voorkennis aanwezig is voor een kolk. De sluiskant wordt gegeven door het attribuut `SIDE` van `THIS KOLK`.

Parameters:

AANKOMSTTIJD      de horizon

=====

@

@ VOORKENNIS

@

@

=====

```

=====
@      Bekijk of voor SLUIKANT nog voorkennis aanwezig is      @
@      en of deze interessant is voor THIS KOLK                  @
@=====

```

PARAMETER:

REAL : aankomsttijd

RETURN IF voorkennistijd[side]=0

Ja

```

refschip      ← THIS schip
refvaarweg    ← THIS vaarweg
voorkennisboom ← NEW SET

```

Onderzoek alle schepen die onderweg zijn naar deze sluis en waarvan de ETA voor AANKOMSTTIJD valt

```
x ← schepen_naar_sluis(0,nrnode OF kant[side],aankomsttijd)
```

```
voorkennisboom ← NONE
THIS schip      ← refschip
THIS vaarweg    ← refvaarweg
RETURN x
```



### 6.63 VOORW\_DUMMY

Dit is de MACRO die als voorwaarde wordt gebruikt indien geen voorwaarden aanwezig zijn.

Terugkeerwaarde:

TRUE

```
@=====
@
@ VOORW_DUMMY
@
@=====
```

RETURN TRUE

## 6.64 WACHTTIJD\_SCHIP

Deze MACRO berekent de wachttijd van een schip bij een sluis.

Terugkeerwaarde

De wachttijd

```
@=====
@
@ WACHTTIJD_SCHIP
@
@=====
```

LOCAL:

REAL: tijd

```
tijd ← tbijsluis
tijd ← eta IF tbijsluis=0
RETURN IF NOW ≤ tijd
tijd ← 60*NOW-tijd
RETURN tijd IF categorie[klas]≠3
tijd ← tijd*wacht_gewicht_cat3
RETURN tijd
```

## 6.65 WATERSTAND

Dit is een MACRO ter bepaling van de waterstand bij een kunstwerk.

Parameters:

NRTIJ      Nummer van te gebruiken getijdetabel of 0 in geval van constant peil.  
 PEILPAR    als NRTIJ>0 de tijdtranslatie, anders de waterstand tov gekozen referentievlak.

```
@=====
@
@ WATERSTAND
@
@=====
```

```
@=====
@ Macro ter bepaling van de waterstand bij een kunstwerk @
@=====
```

```
PARAMETERS:
  INTEGER: nrtij
  REAL    : peilpar
```

```
LOCALS:
  REAL: tijd x y z
```

```
RETURN peilpar IF nrtij = 0
```

```
tijd ← NOW + peilpar
z ← tijd - 12.4 * FLOOR(tijd/12.4)
x ← 0.5 + 0.5 * COS(0.018767*tijd)
y ← x * VALUE OF doodtij[nrtij] AT(z)
```

```
RETURN y + (1-x) * VALUE OF springtij[nrtij] AT(z)
```

## 6.66 WEGSCHRIJVEN\_BLADEN

Deze MACRO schrijft een bladzij informatie weg met betrekking tot tijdregistratie door THIS GENERATOR.

Parameters:

ALLES TRUE als de informatie compleet is, FALSE als er nog meer informatie te verwachten is.

```

@=====
@
@ WEGSCHRIJVEN_BLADEN
@
@=====

@ =====@
@ Wegschrijven van volle bladzijden van de @
@ tijdregistratie (THIS GENERATOR) @
@=====@

PARAMETER:
  LOGICAL: alles

                                Laatste geheel gevulde bladzij
  ref[1] ← LAST bladzij IN bladen WITH regels=volle_regels
                                Als ALLES hoeft de laatste niet vol te zijn
  ref[1] ← LAST OF bladen IF alles
  IF ref[1] IS NOT NONE

                                Er is zo'n bladzij, open de file
  CALL open_matfile(<htijd>,hist_fullname,2)
  THIS bladzij ← FIRST OF bladen
volgende_blad_weg:

                                Schrijf hele bladzij in 1 klap weg
  n ← move_block_matfile(<htijd>,st_rec,regels,<bladinhoud>,1,1)
  IF n < 0

                                Er is iets mis
  WRITE "probleem bij het schrijven naar matfile voor route " |nrgen
    WITH IMAGE a
  GOTO nieuw_blad
END
IF st_rec < 0 @ deze bladzijde was nog niet eerder weggeschreven
  st_rec ← tijdrecords + 1 IF (regels≠volle_regels)
  tijdrecords ← tijdrecords + regels
END
st_rec ← 0 IF regels=volle_regels
IF THIS bladzij IS ref[1]
  CALL close_matfile(<htijd>)
  GOTO nieuw_blad
END
THIS bladzij ← SUCC OF THIS bladzij IN bladen
REPEAT FROM volgende_blad_weg
END
nieuw_blad:

                                Was dat alles ?
  RETURN IF alles

                                Nee, cleanup door verwijdering van alle
                                weggeschreven bladen
  REMOVE EACH bladzij IN bladen WITH st_rec=0 FROM bladen

                                Maak een vervolgblad
  THIS bladzij ← NEW bladzij
  st_rec ← -1
  JOIN THIS bladzij TO bladen
  RETURN

```



## 6.67 ZETOVER\_OK

Deze MACRO test of er nog een schip in een kolk geschoven moet worden.

Terugkeerwaarde:

TRUE        er is niet meer zo' schip, dan is ook PRESCHIP NONE.  
FALSE       er is nog wel zo'n schip en PRESCHIP verwijst daarnaar

```
@=====
@
@ ZETOVER_OK
@
@=====
@ wordt aangeroepen door THIS KOLK om bevestigd te krijgen dat er geen @
@ schepen meer de kolk in geschoven hoeven te worden @
@ PRESCHIP is dan NONE @
@ anders moet PRESCHIP naar binnen worden gewerkt. @
@-----@
    preschip ← FIRST schip IN opstelplan[side]
               THAT BELONGS NOT TO opstelruimte[side]
               & BELONGS NOT TO schutruimte

RETURN preschip IS NONE
```

## 6.68 ZET\_WEKKER

Deze MACRO zorgt ervoor de sluismeester gewekt wordt om aan de slag te gaan.

Parameters:

VERTRAGING            de tijd vanaf now totdat de wekker af moet gaan

```
@=====
@
@ ZET_WEKKER
@
@=====
```

PARAMETER:

REAL: vertraging

Als er geen wekker (meer) is, maak er dan een

```
IF sluiswekker IS NONE
    sluiswekker ← NEW wekker
    weksluis OF sluiswekker ← THIS sluis
END
```

Als er al een wekker loopt, stop die dan als zijn wektijd te ver weg ligt

```
IF sluiswekker IS ACTIVE
    CANCEL sluiswekker IF EVENTTIME OF sluiswekker > NOW + vertraging
END
```

Als er geen wekker is gezet doe dat dan alsnog

```
ACTIVATE sluiswekker WITH DELAY vertraging FROM trriinngg IN manage_sluis
    IF sluiswekker IS NOT ACTIVE
RETURN
```

## 7. Duplicator

Met behulp van de PROSIM MODEL DUPLICATOR kan een verzameling van 1 of meer PROSIM modellen worden voorzien van een schil die als zelfstandige applicatie kan worden gebruikt op een computer waarop PROSIM niet is geïnstalleerd. Afgezien van mogelijkheden de modellen te wijzigen zijn in principe alle faciliteiten van PROSIM beschikbaar, tenzij die mogelijkheden bij het genereren van de applicatie zijn beperkt.

Om zo'n applicatie te maken moet in ieder geval van de op te nemen modellen een RUNFILE zijn aangemaakt. Daarna moeten de volgende stappen worden uitgevoerd:

- Start PROSIM
- Kies in het hoofdmenu DUPLICATOR
- Nu moeten de volgende gegevens worden opgegeven:
  - De naam van de applicatie. Die is vrij te kiezen en is de naam waaronder de applicatie moet worden aangeroepen.
  - De locatie van de applicatie. Dit moet een pad zijn naar een bestaande (sub)directory. Neem even aan dat die leeg is.
  - De titel van de applicatie. Dat is de tekst die in het bovenste kader van de schermen wordt afgebeeld.
- Druk op F2 om door te gaan
- Nu wordt de vraag gesteld of de applicatie voorzien moet worden van een security check. Als de keuze NEE is kan de applicatie overal onbeveiligd draaien. Is de keuze JA dan wordt de applicatie beveiligd. Dan zal, bij eerste activering van de applicatie, een access-code (een code van 4 letters) worden gepresenteerd en worden gevraagd om een installatiecode. Die kan de leverancier genereren met de in zijn bezit zijnde software genaamd PROCODE. Na ingeven van de installatiecode zal de applicatie van start gaan. Deze procedure is eenmalig.
- Nu wordt een lijst van beschikbare modellen gepresenteerd.
- Selecteer een model dat in de applicatie moet worden opgenomen.
- Wijzig eventueel de naam en beschrijving van dat model als die in de applicatie anders moeten zijn dan in de PROSIM omgeving.
- Kies nu voor dit model uit de volgende opties:
  - *Standard*. Dit is de minst beperkende vorm. Na keuze van dit model in de applicatie wordt het runtime maintenance menu gepresenteerd. De gebruiker kan dan maintenance doen of het model runnen.
  - *Autorun*. In dit geval wordt, na selectie van dit model in de applicatie, dat model direct gestart. Na beëindiging van de run wordt het RUN CONTROL menu gepresenteerd
  - *Runonly*. Na laden van het model wordt het direct gestart en na beëindiging ook definitief verlaten via de EXIT optie. Er is geen mogelijkheid voor maintenance.
- Vervolgens wordt de vraag gesteld of er STORE FILES, PICTURE FILES, FIGURE FILES en COMPILATION FILES moeten worden meegecopieerd. Als op zo'n vraag JA wordt gekozen wordt een lijst met beschikbare files van het betreffende type afgebeeld waaruit een selectie kan worden gemaakt.
- Nadat op deze wijze een compleet model aan de applicatie is toegevoegd kan een volgend model worden gekozen, of de selectie afgesloten worden middels F10.
- Nadat de modellen zijn toegevoegd kunnen nog USER DATA FILES in de applicatie worden opgenomen.
- Tenslotte wordt nog de mogelijkheid geboden de GRAPHICS DESIGNER en/of de STATISTICS SUPPLEMENT aan de applicatie toe te voegen.
- Indien er niet nog een applicatie moet worden gegenereerd wordt de DUPLICATOR afgesloten door F10 te drukken.

Na bovenstaande procedure bevat de daar genoemde directory alle files die samen de applicatie vormen. Dit pakket kan dan in zijn geheel worden gecopieerd naar een andere machine waar dan door aanroep van de EXE-file de applicatie kan worden gestart.

## A Attribuutlijst

Hierna volgt een alfabetische lijst met alle entiteiten en attributenen.

<i>Attribuut</i>	<i>van</i>
A_INTENS	MAIN
AANTAL	KOLK
AANTAL_SCHEPEN	GENERATOR
AB	MAIN
AF	MAIN
AFH_KLAS	MAIN
AFSTAND	VAARWEG
ALLE_KOLKEN	SLUIS
B_NRTIJ	BRUG
B_PEILPAR	BRUG
BAFW	BRUG
BCATij	BRUG
BEGIN	WERKTIJD
BENUTTINGSPERC	SCHUTREGIME
BEPAAL_DEADLINE	MAIN
BEROEPS_RUIMTE	KOLK
BESTKOLK	SLUIS
BEZIG	KOLK
BGEM	BRUG
BIBGETIJ	INPUTSTREAM
BIBINTW	INPUTSTREAM
BIBKLAS	INPUTSTREAM
BIBKOSTEN	INPUTSTREAM
BIBREGIMES	INPUTSTREAM
BIBSCHUTREGIMES	INPUTSTREAM
BIBVAART	INPUTSTREAM
BLAD	SCHIP
BLADEN	GENERATOR
BLADINHOUD	BLADZIJ
BLADSTART	SCHIP
BLOKNR	REGIMEBLOK
BMAX	BRUG
BMIN	BRUG
BOPEN	BRUG
BRDHOOFD	KOLK
BRDTKOLK	KOLK
BREEDTE	SCHIP
BRFUIKLENGTE	BRUG
BRIDGE	REGELEAAR
BRUG_1STE	TABLE
BRUG_VOLG	TABLE
BRUGGEN	MAIN
BRUGHULP	BRUGWACHTER
BRUGKANT	BRUGWACHTER
BRUGPLAN	BRUGWACHTER
BSCHIPSTATS	BRUG
BSIDE	REGELEAAR
BUF	MAIN
BUITEN_BEDRIJF	KOLK
BUURKOLK	KOLK
C_AANTAL	CLUSTER
C_KOSTEN	CLUSTER
C_NOVER	CLUSTER
C_PASTIJD	CLUSTER
CAP	MAIN
CASE	REGELEAAR
CAT	STROOM
CAT_IN_KOLK	KOLK
CATEGORIE	MAIN
CATij	KOLK



CATWIS	MAIN
CC	MAIN
CH10	MAIN
CH30	MAIN
CH4	MAIN
CH8	MAIN
CHAR	MAIN
CLUSTERS	MAIN
COLORS	MAIN
COMBITYPE	SCHUTREGIME
CREATIETIJD	SCHIP
CUM_WACHTTIJD	STROOM
CUMAFST	SCHIP
CUR_DEBIET	KOLK
CUR_PLEK	SCHIP
D_DICHT	MAIN
D_OPEN	MAIN
DAG	MAIN
DATA	MAIN
DATE	MAIN
DEADLINE	BRUGWACHTER
DEB_AANT	SLUIS
DEB_REDUCTIE	KOLK
DEBIET	KOLK
DEBIETBEPALING	SLUIS
DIEPGANG	SCHIP
DOEL	SCHIP
DOODTIJ	TABLE
DOORGANG	SCHIP
DOSPAD	MAIN
DOSPADSHP	MAIN
DREMPEL	KOLK
DSTREEP	MAIN
DUUR	KOLK
EBENUT	VERNAUWING
EIND	WERKTIJD
EINDGROEN	VERNAUWING
EINDIGEN	MAIN
ENGTE	VERNAUWING
ESTREEP	MAIN
ETA	SCHIP
FACNIVEL	KOLK
FACT	KOLKSTAT
FILENAAM	MAIN
FUIKLENGTE	KOLK
G	MAIN
GARANTIE	SCHIP
GARANTIESCHEPEN	VERNAUWING
GARKOLK	KOLK
GEEN_UITSTEL	KOLK
GEL	SCHIP
GELDTOT	REGIME
GELDVAN	REGIME
GEMENGD_SCHUTTEN	SLUIS
GEN	SCHIP
GENERATOREN	MAIN
GESCHIKT	KOLK
GETA	GENERATOR
GROEN	MAIN
HELP	MAIN
HEVEL_FACTOR	KOLK
HIST_FULLNAME	GENERATOR
HISTORY_FILENAAM	GENERATOR
HOOGTE	BRUG
HSCHIP	GENERATOR
HTIJD	GENERATOR

HULP	MAIN
HULP2	MAIN
HULP3	MAIN
I	MAIN
ICATij	KOLK
IDLE	MAIN
IDLE_TIME	KOLK
II	MAIN
IKB	MAIN
IKF	MAIN
INDELING	KOLK
INFO1	MAIN
INFO2	MAIN
INHIBITOR	KLOK
INHOUD	STROOM
INSTRUCTIE	SCHIP
INTC	KOLKSTAT
INTENS	TABLE
INVAREN	MAIN
INVAREN_VRIJ	VERNAUWING
J	MAIN
JJ	MAIN
K	MAIN
KANT	SLUIS
KGEM	KOLKSTAT
KIELSPELING	SCHIP
KK	MAIN
KLAS	SCHIP
KLAS_BIJ_BRUG	MAIN
KLAS_BIJ_SLUIS	MAIN
KLASSEN	CLUSTER
KLEINSTE_KOLK	SCHIP
KMAX	KOLK
KOLK_STAT	KOLK
KOLK_STATUS	KOLK
KOLK_WENS	KOLK
KOLKEN	SLUIS
KOPCLUST	INPUTSTREAM
KOSTEN	MAIN
L	MAIN
LAADVERM	SCHIP
LAATSTEBOOT	MAIN
LASTBOOT	REGELEAAR
LEGBLOK	MAIN
LEEGOM	KOLKSTAT
LENGKOLK	KOLK
LENGTE	SCHIP
LICHT	BRUG
LINENR	MAIN
LL	MAIN
LUSBRUG	TABLE
LUSTIJD	TABLE
M	MAIN
M_INDEX	MIJLPAAL
M_KOP	MIJLPAAL
M_LENGTE	MIJLPAAL
M_TYPE	MIJLPAAL
MAXGROEN	VERNAUWING
MB	MAIN
MELDINGEN	NODE
MF	MAIN
MIJNBRUG	STROOM
MN_SCHUTREGIME	SLUIS
MPALEN	SCHIP
N	MAIN
N_UREN	BRUG

NAAR_OPSTELRUIMTE	MAIN
NBOPEN	BRUG
NETWERK	INPUTSTREAM
NEXTSTAT	MAIN
NIEUW	MAIN
NIEUW_OPSTELPLAN	MAIN
NIVELLEREN	MAIN
NN1	PUNT
NN2	PUNT
NODES	MAIN
NR_GEN	GENERATOR
NRBRUG	BRUG
NRCLUSTER	CLUSTER
NRDOORGANGEN	BRUG
NRINTENS	STROOM
NRKOLK	KOLK
NRNODE	NODE
NRSLUIS	SLUIS
NRVERNAUWING	VERNAUWING
NULNIVEL	KOLK
NUMBER	REGELEAAR
NXTBOOT	REGELEAAR
ONDERDOOR	SCHIP
ONDERZOEK	OUTPUTSTREAM
ONMOGELIJK	SLUIS
ONTMOETING	VERNAUWING
ONTRUIMINGSTIJD	VERNAUWING
OPEN	BRUG
OPLOS	BRUG
OPLOSVERMOGEN	STROOM
OPP_KOLK	KOLK
OPPREND	KOLK
OPSTELPLAN	KOLK
OPSTELRUIMTE	KOLK
ORANJE	MAIN
ORIGINEEL	MAIN
PAGNR	MAIN
PALEN_HEEN	GENERATOR
PALEN_TERUG	GENERATOR
PARAM	REGIME
PASSEER_KNOOP	MAIN
PERIODIEK	SLUIS
PLAATS	MAIN
PLANBREEDTE	SCHIP
PLANLENGTE	SCHIP
PLEK	MAIN
PLUSTIJD	VERNAUWING
POSTSCHIP	KOLK
PRESCHIP	KOLK
QFILENAAM	MAIN
QUERY_INVOER	OUTPUTSTREAM
R	KOLKSTAT
R_INHOUD	REGIMEBLOK
R_SCHEPEN	GENERATOR
RAAI	VAARWEG
RAPPORT	OUTPUTSTREAM
REF	MAIN
REFBRUG	MAIN
REFKOLK	MAIN
REFNODE	MAIN
REFREGIME	MAIN
REFSHIP	MAIN
REFSLUIS	MAIN
REFVAARWEG	MAIN
REGEL	MAIN
REGELAANTAL	MAIN

REGELING	VERNAUWING
REGELLENGTE	GENERATOR
REGELOPH	MAIN
REGELS	BLADZIJ
REGELS_PER_BLADZIJ	GENERATOR
REGIMEBLOKKEN	MAIN
REGIMES	BRUG
REMANENT	SLUIS
RICHT	STROOM
RICHTING	SCHIP
ROOD	MAIN
ROUTE	SCHIP
ROUTE_FILENAAM	GENERATOR
ROUTE_HEEN	GENERATOR
ROUTE_TERUG	GENERATOR
S	KOLKSTAT
S_AANTAL	SCHIPSTAT
S_CUMKOSTEN	SCHIPSTAT
S_CUMPASTIJD	SCHIPSTAT
S_NOVER	SCHIPSTAT
S_NRTIJ	SLUIS
S_PEILPAR	SLUIS
S_RECL	MAIN
S_VOLGNR	SCHIPSTAT
SB	MAIN
SCHIP_BIJ_MELDRAAI	MAIN
SCHIP_BIJ_SLUIS	MAIN
SCHIPNAAM	MAIN
SCHIPREGEL	GENERATOR
SCHIPSTATS	MAIN
SCHRIKH	SCHIP
SCHUTPERIODES	SCHUTREGIME
SCHUTPLANNING	SLUIS
SCHUTREGIMES	MAIN
SCHUTREGISTRATOR	MAIN
SCHUTRUIMTE	KOLK
SCHUTTENDE	KOLK
SF	MAIN
SGARKLAS	MAIN
SHIFT	KLOK
SIDE	KOLK
SIMULATIEDUUR	MAIN
SLUIS_ZELF	MAIN
SLUISWEKKER	SLUIS
SLUIZEN	MAIN
SNELHEID	SCHIP
SNELSTE_EERST	VERNAUWING
SOORT	REGIME
SPECIAAL	SCHIP
SPRINGTIJ	TABLE
SREGIMENR	SCHUTREGIME
SSCHIPSTATS	SLUIS
ST_REC	BLADZIJ
STÄCK	MAIN
STAR	VERNAUWING
START	SCHIP
START_DOORVAREN	MAIN
START_IDLE	KOLK
START_OVERLIGGEN	MAIN
START_SCHUTTEN	MAIN
STRATEGIE	SLUIS
STRIJKH	SCHIP
STROKEN	STROOM
T	MAIN
TABNR	STROOM
TBIJSLUIS	SCHIP



TBOPEN	BRUG
TBSLUIT	BRUG
TEKST	MAIN
TEXT	MAIN
TEXT10	MAIN
TIJDIN_1STE	TABLE
TIJDIN_VOLG	TABLE
TIJDRECEORDS	GENERATOR
TIJDUIT_1STE	TABLE
TIJDUIT_VOLG	TABLE
TIME	MAIN
TL	SCHIP
TOON_VERNAUWING	MAIN
TOPEN	KOLK
TOT	MAIN
TOT_DEBIET	KOLK
TOTAAL	STROOM
TOTLAADVERM	KOLK
TOTOPEN	BRUG
TOTOVER	KOLKSTAT
TOTSCHUT	KOLKSTAT
TS	SCHIP
TSCHIP	GENERATOR
TSLUIT	KOLK
TW	SCHIP
TYPE	BRUG
UCATij	KOLK
UIT_OPSTELRUIMTE	MAIN
UITVAREN	MAIN
V_AFSTAND	VERNAUWING
VAARPLAN	SCHIP
VAARVAK	VAARWEG
VAARWEGEN	MAIN
VASTEBRUG	TABLE
VERNAUWINGEN	MAIN
VERNAUWINGSKANT	VERNAUWING
VGARKLAS	MAIN
VIA_DOORGANG	MAIN
VIA_KOLK	MAIN
VLIJCHT	VERNAUWING
VOLGLENTE	MAIN
VOLLE_REGELS	BLADZIJ
VOLSCHUT	KOLKSTAT
VOOR_AANMELDING	MAIN
VOORKENNISBOOM	MAIN
VOORKENNISTIJD	SCHUTREGIME
VOORKEUR	SLUIS
VOORRANGSKANT	VERNAUWING
VOORW_INIT	MAIN
VOORWAARDE	KOLK
VORIGE_MIJLPAAL	SCHIP
VRIJE_KOLK	MAIN
VRIJE_KOLKEN	SLUIS
VRIJSTELLING	VERNAUWING
VRIJVERKEER	BRUG
VSCHIPSTATS	VERNAUWING
VSPECIAAL	SCHIP
VULGRAAD	KOLK
VVAARSNELHEID	VERNAUWING
VW_START	KOLK
VWACHTRIJ	VERNAUWING
VZIJDE	VERNAUWING
W	MAIN
WACHT_GEWICHT_CAT3	SCHUTREGIME
WACHT_OP_DEADLINE	KOLK
WACHT_OP_PERIODE	KOLK

WACHT_OP_VOORW	KOLK
WACHTEN	MAIN
WACHTEND	SCHIP
WACHTRIJ	BRUG
WACHTRUIMTE	SLUIS
WACHTTIJD_ONMOGELIJK	SLUIS
WATCHER	REGELEAAR
WATERLENGTE	KOLK
WATERPAD	BRUG
WEEKTOT	STROOM
WEG_HEEN	GENERATOR
WEG_TERUG	GENERATOR
WEGVERKEER	BRUG
WEKSLUIS	WEKKER
WERK	KLOK
WERKT	BRUGWACHTER
WIJDTE	BRUG
WIT	MAIN
WKOSTEN	BRUG
X	MAIN
X1	MAIN
X2	MAIN
XTRA_WACHTTIJD	SCHUTREGIME
XX	PUNT
XXX	MAIN
Y	MAIN
YY	PUNT
Z	MAIN
ZIJDE	BRUG
ZIJNBRUG	BRUGWACHTER
ZIJNSLUIS	KOLK
ZIJNWACHTER	BRUG

## B Cross-reference

Op de volgende bladzijden is een cross-reference tabel afgedrukt. Langs de verticale as staan alle MODULE en MACRO namen, langs de horizontale as alle globale variabelen en attributen zoals in hoofdstuk 3 reeds opgesomd, maar nu in alfabetische volgorde. Een R in een veld betekent dat de waarde van de variabele alleen wordt uitgelezen, een W dat er aan die variabele of dat attribuut een nieuwe waarde wordt toegekend.

[illegible]



[illegible]

	CATWIS	CC	CH10	CH30	CH4	CH8	CHAR	CLUSTERS	COLORS	COMBITYPE	CREATETIJD	CUM_WACHTTIJD	CUMAFST	CUR_DEBIET	CUR_PLEK	D_DICHT	D_OPEN	DAG	DATA	DATE	DEADLINE	DEB_AANT	DEB_REDUCTIE	DEBIET	DEBIETBEPALING	DIEPGANG	DOEL	DOODTIJ	DOORGANG	DOSPAD
AFGELEIDE																														
AFHANKELIJK																														
AFHANKELIJK INIT																														
BEPAAL UITSTEL																														
BEPAAL UITSTEL KOLK																														
BRUGREGIME																						W								
BRUG REGISTRATIE											R																			
BUITEN SCHUTPERIODE																														
CHECK NETWERK																														
DISPLAY VERNAUWING																														
FYSIEK																							W		W	R				W
GEEN SCHUTSTRATEGIE															R													R		
GEMAAKT										W					W												W	W		
HERDEEL	W																													
HEVELEN																														
HEVELEND																R														
HEVELEND INIT																							W							
HEVELEN INIT																							W							
HULPJE																				R										
IDLE STOP WATCH																														
INITDEEL																														
INITIALISATIE									W							W	W	W	W	W						R				W
INTIJD																											R			
KOLKMOD													W			R	R							R	W	R				
KOLK PREFERENTIE																R	R													
KOP																														
LEES BRUGGEN																														
LEES GENERATOREN							W														R									W
LEES GETIJDEN																				W								W		
LEES KOSTEN																														
LEES REGIMES							W			W																				
LEES SLUIZEN			W	W																						W				
LEES VAARKROMMEN																					W									
LEES VAARWEGEN																														
LEES VERNAUWINGEN			W																											
LEES WEGINTENS																					W									
MAINMOD												W									W									
MANAGE BRUG												W										R								R
MANAGE SLUIS																														
MANAGE VERNAUWING																														
PASSAGE VERNAUWING																												R		
PASSEN	W																									R				
PLAN THIS KOLK IN															R															
QUERY REGISTRATIE											R																			
RAPPROT BRUGGEN		W																												
RAPPORT INITIA								W													R									
RAPPORT SCHEEPVAART		R						R																						
RAPPORT SLUIZEN		W																R					W		R					
RAPPORT VERNAUWINGEN		W																												
RESULTAAT REGEL																														
SCHEPEN NAAR SLUIS																												R		
SCHIPMOD												W		W														R		R
SCHUTSTRATEGIE GELYK																												R		
SCHUTSTRATEGIE NKOLK																												R		
SLUIS REGISTRATIE														R																
SWITCH																														
TONDERDOOR																														R
UITTIJD																											R			
VERNAUWINGSGARANTIE																												R		
VOEGTOE																														
VOLDOENDE BENUTTING											R																			
VOORKENNIS																														
VOORW DUMMY																														
WACHTTIJD SCHIP																														
WATERSTAND																													R	
WEGSCHRIJVEN BLADEN																														
ZETOVER OK																														
ZET WEKKER																														



	DOSPADSHP	DREMP	DSTREEP	DUUR	EBENUT	EINDE	EINDGROEN	EINDIGEN	ENGTE	ESTREEP	ETA	FACNIVEL	FACT	FILENAAM	FUIKLENGTE	G	GARANTIE	GARANTIESCHEPEN	GARKOLK	GEEN UITSTEL	GEL	GELDTOT	GELDVAN	GEMENGD_SCHUTTEN	GEN	GENERATOREN	GESCHIKT	GETA	GROEN	HELP
AFGELEIDE																														
AFHANKELIJK																														
AFHANKELIJK INIT																														
BEPAAAL UITSTEL																														
BEPAAAL UITSTEL KOLK											R	R								W										
BRUGREGIME											R											R	R							
BRUG REGISTRATIE																W														
BUITEN SCHUTPERIODE																							R							
CHECK NETWERK																														
DISPLAY VERNAUWING							R		R		R						R	R			R									R
FYSIEK								R			R										R		W	W						
GEEN SCHUTSTRATEGIE											R						W		R				W							
GEMAAKT											W										W				W					
HERDEEL											R													R						
HEVELEN																														
HEVELEND																														
HEVELEND INIT																														
HEVELEN INIT																														
HULPJE																														
IDLE STOP WATCH																														
INITDEEL																														
INITIALISATIE	W		W							W																W				W
INTIJD		R														R						R			R					
KOLKMOD				W								R												R						
KOLK PREFERENTIE																														
KOP										R																				
LEES BRUGGEN						W																								
LEES GENERATOREN	R													E												R				
LEES GETIJDEN																														
LEES KOSTEN																														
LEES REGIMES																						W	W							
LEES SLUIZEN		W				W					W			W					W					W			W			
LEES VAARKROMMEN																														
LEES VAARWEGEN																														
LEES VERNAUWINGEN									W										W											
LEES WEGINTENS																														
MAINMOD																											R			
MANAGE BRUG						W					R																			
MANAGE SLUIS						W																								
MANAGE VERNAUWING							W		R									R												R
PASSAGE VERNAUWING									R		R								R			R								R
PASSEN		R									R						R		R					R			R			W
PLAN THIS KOLK IN																														
QUERY REGISTRATIE																														
RAPPROT BRUGGEN																														
RAPPORT INITIA																											R			
RAPPORT SCHEEPVAART																														
RAPPORT SLUIZEN													W																	
RAPPORT VERNAUWINGEN																														
RESULTAAT REGEL																														
SCHEPEN NAAR SLUIS											R																			
SCHIPMOD					W		R		R		W						W	R				R								
SCHUTSTRATEGIE GELYK											R																			
SCHUTSTRATEGIE NKOLK											R										R									
SLUIS REGISTRATIE				R																										
SWITCH																														
TONDERDOOR																						R								
UITTIJD		R																				R			R					
VERNAUWINGSGARANTIE																	W	R												R
VOEGTOE																														
VOLDOENDE BENUTTING																														
VOORKENNIS																														
VOORW DUMMY																														
WACHTTIJD SCHIP											R																			
WATERSTAND																														
WEGSCHRIJVEN BLADEN																														
ZETOVER OK																														
ZET WEKKER																														

	HEVEL_FACTOR	HIST_FULLNAME	HISTORY_FILNAAM	HOOGTE	HSCHIP	HTIJD	HULP	HULP2	HULP3	I	ICATIJ	IDLE	IDLE_TIME	II	IKB	IKF	INDELING	INFO1	INFO2	INHIBITOR	INHOUD	INSTRUCTIE	INTC	INTENS	INVAREN	INVAREN_VRIJ	J	JJ	K	KANT	
AFGELEIDE																							R								
AFHANKELIJK																								R							
AFHANKELIJK INIT																															
BEPAAAL UITSTEL																	W														
BEPAAAL UITSTEL KOLK																															
BRUGREGIME																															
BRUG REGISTRATIE																										W		W			
BUITEN SCHUTPERIODE																															
CHECK NETWERK																														R	
DISPLAY VERNAUWING																									R						
FYSIEK				R																	R	W				W		W			
GEEN SCHUTSTRATEGIE																						W								R	
GEMAAKT																															
HERDEEL							W										W									W		W			
HEVELEN																															
HEVELEND																															
HEVELEND INIT	W																														
HEVELEN INIT	W																														
HULPJE							W																								
IDLE STOP WATCH													W																		
INITDEEL																		W													
INITIALISATIE															W	W									W		W		W		
INTIJD											R																				
KOLKMOD	R										R	R													R				W		
KOLK PREFERENTIE												R													R						
KOP																															
LEES BRUGGEN				W							W									W										W	
LEES GENERATOREN		W	W		W	W																							W	R	
LEES GETIJDEN							R																						W	R	
LEES KOSTEN																											W		W		
LEES REGIMES							W							W													W	W	W	R	
LEES SLUIZEN	W										W	W								W								W	W		
LEES VAARKROMMEN							R																				W		W		
LEES VAARWEGEN											W																		W		
LEES VERNAUWINGEN											W															W	W	W	W		
LEES WEGINTENS							R	W	W															W					W		
MAINMOD																						W		R					W		
MANAGE BRUG																				R	W						W				
MANAGE SLUIS																				R									W		
MANAGE VERNAUWING																															
PASSAGE VERNAUWING																							R			R					
PASSEN							W										W										W				
PLAN THIS KOLK IN																							W								
QUERY REGISTRATIE																							R								
RAPPROT BRUGGEN																															
RAPPORT INITIA																														R	
RAPPORT SCHEEPVAART																												W			
RAPPORT SLUIZEN																								W						R	
RAPPORT VERNAUWINGEN																			W	W											
RESULTAAT REGEL																															
SCHEPEN NAAR SLUIS																														R	
SCHIPMOD																							W			W			W		
SCHUTSTRATEGIE GELYK																							W							R	
SCHUTSTRATEGIE NKOLK								W									W						W							R	
SLUIS REGISTRATIE													R																	R	
SWITCH											W																		W		
TONDERDOOR																															
UITTIJD																															
VERNAUWINGSGARANTIE																															
VOEGTOE							W										R												W		
VOLDOENDE BENUTTING																															
VOORKENNIS																														R	
VOORW DUMMY																															
WACHTTIJD SCHIP																															
WATERSTAND																															
WEGSCHRIJVEN BLADEN		R				W																									
ZETOVER OK																															
ZET WEKKER																															



	KGEM	KIELSELING	KK	KLAS	KLAS_BIJ_BRUG	KLAS_BIJ_SLUIS	KLASSEN	KLEINSTE_KOLK	KMAX	KOLK_STAT	KOLK_STATUS	KOLK_WENS	KOLKEN	KOPCLUST	KOSTEN	L	LAADVERM	LAATSTEBOOT	LASTBOOT	LEGBLOK	LEEGOM	LENGKOLK	LENHTE	LICHT	LINENR	LL	LUSBRUG	LUSTIJD	M	M_INDEX
AFGELEIDE																							R							
AFHANKELIJK				R						R	R																			
AFHANKELIJK INIT												R																		
BEPAAAL UITSTEL								R				R																		
BEPAAAL UITSTEL KOLK				R																										
BRUGREGIME																														
BRUG REGISTRATIE																													W	
BUITEN SCHUTPERIODE																														
CHECK NETWERK																														
DISPLAY VERNAUWING				R																				R						
FYSIEK																														
GEEN SCHUTSTRATEGIE				R								R																		
GEMAAKT		W		W													W						W							
HERDEEL				R																										
HEVELEN										R	R																			
HEVELEND											R	R																		
HEVELEND INIT												R																		
HEVELEN INIT												R																		
HULPJE																														
IDLE STOP WATCH																														
INITDEEL																						R								
INITIALISATIE																														
INTIJD				R																		R	R					R		
KOLKMOD			R						W		W	W	R				R													
KOLK PREFERENTIE											R											R								
KOP																									W					
LEES BRUGGEN			W														W			W				W					W	
LEES GENERATOREN					W	W																							W	W
LEES GETIJDEN																	W												W	
LEES KOSTEN															W	W													W	
LEES REGIMES																W														
LEES SLUIZEN			W							W		W										W								
LEES VAARKROMMEN					R	R																					W	W		
LEES VAARWEGEN																														
LEES VERNAUWINGEN			W																											
LEES WEGINTENS																	W													
MAINMOD													R																	
MANAGE BRUG																		W	W					W						R
MANAGE SLUIS													R																	
MANAGE VERNAUWING																														
PASSAGE VERNAUWING				R																										
PASSEN		R		R													W					R								
PLAN THIS KOLK IN																														
QUERY REGISTRATIE																														R
RAPPROT BRUGGEN														R																
RAPPORT INITIA							W						R	W											W				W	
RAPPORT SCHEEPVAART						R							R									R								
RAPPORT SLUIZEN	W									R			R																	
RAPPORT VERNAUWINGEN													R																	
RESULTAAT REGEL																														
SCHEPEN NAAR SLUIS				R																										
SCHIPMOD				R																			R							
SCHUTSTRATEGIE GELYK													R																	
SCHUTSTRATEGIE NKOLK								W				R																	W	
SLUIS REGISTRATIE	W									R	R										W									
SWITCH																														
TONDERDOOR				R															R				R				R			
UITTIJD				R																										
VERNAUWINGSGARANTIE																														
VOEGTOE																W						R								
VOLDOENDE BENUTTING																						R								
VOORKENNIS																														
VOORW DUMMY																														
WACHTTIJD SCHIP				R																										
WATERSTAND																														
WEGSCHRIJVEN BLADEN																														
ZETOVER OK																														
ZET WEKKER																														

	M_KOP	M_LENGTE	M_TYPE	MAXGROEN	MB	MELDINGEN	MF	MIJNBRUG	MN_SCHUTREGIME	MPALEN	N	N_UREN	NAAR_OPSTELRUIMTE	NBOPEN	NETWERK	NEXTSTAT	NIEUW	NIEUW_OPSTELPLAN	NIVELLEREN	NN1	NN2	NODES	NRGEN	NRBRUG	NRCLUSTER	NRDOORGANGEN	NRINTENS	NRKOLK	NRNODE	NRSLUIS
AFGELEIDE								R																						
AFHANKELIJK																														
AFHANKELIJK INIT															R													R		
BEPAAAL UITSTEL											W																	R		
BEPAAAL UITSTEL KOLK									R																					
BRUGREGIME											W																			
BRUG REGISTRATIE											W													R						
BUITEN SCHUTPERIODE																												R		
CHECK NETWERK																													R	
DISPLAY VERNAUWING																														
FYSIEK						R		R			W	W		W										R		R		R		R
GEEN SCHUTSTRATEGIE											R							R												
GEMAAKT										W																				
HERDEEL																												R		
HEVELEN																			R											
HEVELEND																														
HEVELEND INIT															R													R		
HEVELEN INIT															R													R		
HULPJE																														
IDLE STOP WATCH																														
INITDEEL																												R		
INITIALISATIE					W	W					W	W		R				W	W			W						R		R
INTIJD																														
KOLKMOD																		R	R											
KOLK PREFERENTIE																			R											
KOP																	W													
LEES BRUGGEN						W		W							R					W	W	R		W		W	W		W	
LEES GENERATOREN	W	W	W								W				R							R	W						W	
LEES GETIJDEN											W																		R	
LEES KOSTEN											W																			
LEES REGIMES									R		W																		R	
LEES SLUIZEN						W			W		W				R							R					W	W	W	
LEES VAARKROMMEN																														
LEES VAARWEGEN															R							R							W	
LEES VERNAUWINGEN					W	W									R								R						W	
LEES WEGINTENS																				R	R						R			
MAINMOD																														
MANAGE BRUG										R				W																
MANAGE SLUIS																			R											
MANAGE VERNAUWING					R																									
PASSAGE VERNAUWING																														
PASSEN																													R	
PLAN THIS KOLK IN																														
QUERY REGISTRATIE									R				R															R		
RAPPROT BRUGGEN												R					W												R	
RAPPORT INITIA											W														W				R	
RAPPORT SCHEEPVAART																	W								R				R	
RAPPORT SLUIZEN																W	W											R	R	
RAPPORT VERNAUWINGEN																	W												R	
RESULTAAT REGEL																														
SCHEPEN NAAR SLUIS																							R						R	
SCHIPMOD						R				R			R											R		R		R		R
SCHUTSTRATEGIE GELYK																			R											
SCHUTSTRATEGIE NKOLK											W								R									R		
SLUIS REGISTRATIE																												R	R	R
SWITCH																														
TONDERDOOR																														
UITTIJD																														
VERNAUWINGSGARANTIE																														
VOEGTOE																														
VOLDOENDE BENUTTING									R																					
VOORKENNIS																													R	
VOORW DUMMY																														
WACHTTIJD SCHIP																														
WATERSTAND																														
WEGSCHRIJVEN BLADEN											W												R							
ZETOVER OK																														
ZET WEKKER																														



	NRVERNAUWING	NULNIVEL	NUMBER	NXTBOOT	ONDERDOOR	ONDERZOEK	ONMOGELIJK	ONTMOETING	ONTRUIMINGSTIJD	OPEN	OPLOS	OPLOSVERMOGEN	OPP_KOLK	OPPREND	OPSTELPLAN	OPSTELRUIMTE	ORANJE	ORIGINEEL	PAGNR	PALEN_HEEN	PALEN_TERUG	PARAM	PASSEER_KNOOP	PERIODIEK	PLAATS	PLANBREEDTE	PLANLENGTE	PLEK	PLUSTIJD	POSTSCHIP
AFGELEIDE												R																		
AFHANKELIJK															R															
AFHANKELIJK INIT																														
BEPAAAL UITSTEL															W		W													
BEPAAAL UITSTEL KOLK		R													R															W
BRUGREGIME																						R								
BRUG REGISTRATIE			R								R																			
BUITEN SCHUTPERIODE															R							R		W						
CHECK NETWERK																									W			W		
DISPLAY VERNAUWING																														
FYSIEK					W																									
GEEN SCHUTSTRATEGIE							R								R														R	
GEMAAKT																				R	R					W	W			
HERDEEL														W											W					
HEVELEN																														
HEVELEND																														
HEVELEND INIT																														
HEVELEN INIT																														
HULPJE																														
IDLE STOP WATCH																														
INITDEEL																														
INITIALISATIE																W							W							
INTIJD																										R	R		W	
KOLKMOD		R											R	W	R	R										R	R		W	
KOLK PREFERENTIE															R											R	R			
KOP																		W												
LEES BRUGGEN										W	W																			
LEES GENERATOREN																			W	W								W		
LEES GETIJDEN																														
LEES KOSTEN																														
LEES REGIMES																						W								
LEES SLUIZEN		W					W						W	W	W															
LEES VAARKROMMEN																														
LEES VAARWEGEN																														
LEES VERNAUWINGEN	W							W	W							R													W	
LEES WEGINTENS																														
MAINMOD																														
MANAGE BRUG			W	W	R					W	W												R							
MANAGE SLUIS																														
MANAGE VERNAUWING									R																					
PASSAGE VERNAUWING																	R													
PASSEN															W										W	R	R			
PLAN THIS KOLK IN							R																							
QUERY REGISTRATIE																														
RAPPROT BRUGGEN																														
RAPPORT INITIA																			W											
RAPPORT SCHEEPVAART																														
RAPPORT SLUIZEN																														
RAPPORT VERNAUWINGEN	R																													
RESULTAAT REGEL																														
SCHEPEN NAAR SLUIS																														
SCHIPMOD	R				R					R						R							R					W	R	
SCHUTSTRATEGIE GELYK							R								R													R		
SCHUTSTRATEGIE NKOLK							R								R													R		
SLUIS REGISTRATIE														R																
SWITCH																														
TONDERDOOR																														
UITTIJD																													R	
VERNAUWINGSGARANTIE																									W	R	R			
VOEGTOE																														
VOLDOENDE BENUTTING															R												R	R		
VOORKENNIS																														
VOORW DUMMY																														
WACHTTIJD SCHIP																														
WATERSTAND																														
WEGSCHRIJVEN BLADEN																														
ZETOVER OK															R	R														
ZET WEKKER																														

	PRESCHIP	QFILENAAM	QUERY_INVOER	R	R_INHOUD	R_SCHEPEN	RAAI	RAPPORT	REF	REFBRUG	REFKOLK	REFNODE	REFREGIME	REFSCHIP	REFSLUIS	REFVAARWEG	REGEL	REGELAANTAL	REGELING	REGELLENGTE	REGELOPH	REGELS	REGELS PER BLADZIJ	REGIMEBLOKKEN	REGIMES	REMANENT	RICHT	RICHTING	ROOD	ROUTE
AFGELEIDE																														
AFHANKELIJK																														
AFHANKELIJK INIT									W																					
BEPAAAL UITSTEL																									W					
BEPAAAL UITSTEL KOLK	W								W				W												R		R			
BRUGREGIME																								R						
BRUG REGISTRATIE																										R				
BUITEN SCHUTPERIODE																											R			
CHECK NETWERK							R																							
DISPLAY VERNAUWING											W																	R	R	
FYSIEK					R	R																R		R				W		
GEEN SCHUTSTRATEGIE																										R		W		
GEMAAKT						R														R		W	R				W			W
HERDEEL														W																
HEVELEN																														
HEVELEND																														
HEVELEND INIT									W																					
HEVELEN INIT									W																					
HULPJE																														
IDLE STOP WATCH																														
INITDEEL																														
INITIALISATIE		W	R																										W	
INTIJD	R																													
KOLKMOD	W												W																	
KOLK PREFERENTIE																														
KOP								R										R			W									
LEES BRUGGEN			R		W		R																	W	W		W			
LEES GENERATOREN		W	R			W	R				W					W			W		W	W								
LEES GETIJDEN																														
LEES KOSTEN																														
LEES REGIMES					R							W												R	R					
LEES SLUIZEN			R																							W				
LEES VAARKROMMEN																														
LEES VAARWEGEN						W																								
LEES VERNAUWINGEN			R																W										R	
LEES WEGINTENS																														
MAINMOD			R																		W									
MANAGE BRUG								W																						
MANAGE SLUIS																														
MANAGE VERNAUWING																													R	
PASSAGE VERNAUWING											W		W															R		
PASSEN									W			W																		
PLAN THIS KOLK IN																														
QUERY REGISTRATIE																														
RAPPROT BRUGGEN			R					R														W								
RAPPORT INITIA						R		R										W				W								
RAPPORT SCHEEPVAART			R					R														W								
RAPPORT SLUIZEN				R				R														W								
RAPPORT VERNAUWINGEN								R											R		W									
RESULTAAT REGEL																														
SCHEPEN NAAR SLUIS						R																								
SCHIPMOD											W		W						R									W		R
SCHUTSTRATEGIE GELYK										W																R		W		
SCHUTSTRATEGIE NKOLK																										R		W		
SLUIS REGISTRATIE				W																	R	W	R							
SWITCH																														
TONDERDOOR																												R		
UITTIJD	R																													
VERNAUWINGSGARANTIE																												W	R	
VOEGTOE																														
VOLDOENDE BENUTTING																										R				
VOORKENNIS													W		W															
VOORW DUMMY																														
WACHTTIJD SCHIP																														
WATERSTAND																														
WEGSCHRIJVEN BLADEN									W													R								
ZETOVER OK	W																													
ZET WEKKER																														



	ROUTE_FILNAAM	ROUTE_HEEN	ROUTE_TERUG	S	S_AANTAL	S_CUMKOSTEN	S_CUMPASTIJD	S_NOVER	S_NRTIJD	S_PEILPAR	S_RECL	S_VOLGNR	SB	SCHIP_BIJ_MELDRAAI	SCHIP_BIJ_SLUIS	SCHIPNAAM	SCHIPREGEL	SCHIPSTATS	SCHRIKH	SCHUTPERIODES	SCHUTPLANNING	SCHUTREGIMES	SCHUTREGISTRATOR	SCHUTRUIMTE	SCHUTTENDE	SF	SGARKLAS	SHIFT	SIDE	SIMULATIEDUUR	
AFGELEIDE																															
AFHANKELIJK																								R						R	
AFHANKELIJK INIT																					R										
BEPAAAL UITSTEL																					R										
BEPAAAL UITSTEL KOLK									R	R														R						R	
BRUGREGIME																															
BRUG REGISTRATIE																					R										
BUITEN SCHUTPERIODE																															
CHECK NETWERK																															
DISPLAY VERNAUWING													R			W															
FYSIEK																	W		R												
GEEN SCHUTSTRATEGIE														R																	
GEMAAKT		R	R								R						R		W									R			
HERDEEL																															
HEVELEN																														R	
HEVELEND																														R	
HEVELEND INIT																					R										
HEVELEN INIT																					R										
HULPJE																															
IDLE STOP WATCH																															
INITDEEL																															
INITIALISATIE												W		W	W	W											W	W			W
INTIJD									R	R														R							
KOLKMOD									R	R												R		R	W					W	
KOLK PREFERENTIE																														R	
KOP																															
LEES BRUGGEN													W																W		
LEES GENERATOREN	W	W	W									R											W								
LEES GETIJDEN									W																						
LEES KOSTEN																															
LEES REGIMES																				W		R									
LEES SLUIZEN									W	W		W									W	W		W				R	W	W	
LEES VAARKROMMEN																															
LEES VAARWEGEN																															
LEES VERNAUWINGEN													W																		
LEES WEGINTENS																															
MAINMOD																							R		R					R	
MANAGE BRUG																														R	
MANAGE SLUIS																					R				R				R		
MANAGE VERNAUWING																															
PASSAGE VERNAUWING																															
PASSEN										R	R													R						R	
PLAN THIS KOLK IN																															
QUERY REGISTRATIE																															
RAPPROT BRUGGEN						R		R					R																		
RAPPORT INITIA	R	R																													
RAPPORT SCHEEPVAART						R		R	R				R						W												
RAPPORT SLUIZEN					R	R		R																							
RAPPORT VERNAUWINGEN						R		R																							
RESULTAAT REGEL																															
SCHEPEN NAAR SLUIS																														R	
SCHIPMOD						W		W	W				R		R	R						R									
SCHUTSTRATEGIE GELYK															R																
SCHUTSTRATEGIE NKOLK															R	R														R	
SLUIS REGISTRATIE					W																		R							R	
SWITCH																															
TONDERDOOR																															
UITTIJD									R	R																					
VERNAUWINGSGARANTIE																															
VOEGTOE																															
VOLDOENDE BENUTTING																														R	
VOORKENNIS																														R	
VOORW DUMMY																															
WACHTTIJD SCHIP																															
WATERSTAND																															
WEGSCHRIJVEN BLADEN																															
ZETOVER OK																								R					R		
ZET WEKKER																															

	SLUIS_ZELF	SLUISWEKKER	SLUIZEN	SNELHEID	SNELSTE_EERST	SOORT	SPECIAL	SPRINGTIJ	SREGIMENR	SSCHIPSTATS	ST_REC	STACK	STAR	START	START DOORVAREN	START_IDLE	START_OVERLIGGEN	START_SCHUTTEN	STRATEGIE	STRIJKH	STROKEN	T	TABNR	TBIJSLUIS	TBOPEN	TBSLUIT	TEKST	TEKST10	TEXT	TIJDIN_1STE
AFGELEIDE																					R		R							
AFHANKELIJK																														
AFHANKELIJK INIT																														
BEPAAAL UITSTEL																														
BEPAAAL UITSTEL KOLK																								R						
BRUGREGIME						R																			R	R				
BRUG REGISTRATIE																														
BUITEN SCHUTPERIODE																														
CHECK NETWERK			R																											
DISPLAY VERNAUWING					R																									
FYSIEK			R																	R										
GEEN SCHUTSTRATEGIE							R																							
GEMAAKT				W			W							W							W									
HERDEEL																														
HEVELEN																														
HEVELEND																														
HEVELEND INIT																														
HEVELEN INIT																														
HULPJE																														
IDLE STOP WATCH																W														
INITDEEL																														
INITIALISATIE	W		W											W		W	W													
INTIJD																														
KOLKMOD																	R	R											R	
KOLK PREFERENTIE																				R										
KOP																												R		
LEES BRUGGEN																					W				W	W				
LEES GENERATOREN			R																											
LEES GETIJDEN			R					W																						
LEES KOSTEN																														
LEES REGIMES					W				R																					
LEES SLUIZEN			R						W	W						W			W											
LEES VAARKROMMEN			R																										W	
LEES VAARWEGEN																														
LEES VERNAUWINGEN					W								W																	
LEES WEGINTENS																								W						
MAINMOD			R																					R						
MANAGE BRUG															R		R									R	R			
MANAGE SLUIS	R																													
MANAGE VERNAUWING																	R													
PASSAGE VERNAUWING					R									R																
PASSEN																														
PLAN THIS KOLK IN																														
QUERY REGISTRATIE																														
RAPPROT BRUGGEN																													W	
RAPPORT INITIA			R																									W	W	
RAPPORT SCHEEPVAART										R																			W	
RAPPORT SLUIZEN			R							R												W							W	
RAPPORT VERNAUWINGEN													R																W	
RESULTAAT REGEL																														
SCHEPEN NAAR SLUIS				R																										
SCHIPMOD			R	R			R			R			R	W	R										W					
SCHUTSTRATEGIE GELYK																														
SCHUTSTRATEGIE NKOLK	R											W																		
SLUIS REGISTRATIE																														
SWITCH																														
TONDERDOOR																														
UITTIJD																														
VERNAUWINGSGARANTIE														R																
VOEGTOE																														
VOLDOENDE BENUTTING																														
VOORKENNIS																														
VOORW DUMMY																														
WACHTTIJD SCHIP																										R				
WATERSTAND								R																						
WEGSCHRIJVEN BLADEN											W																			
ZETOVER OK																														
ZET WEKKER		W																												



	TIJDIN_VOLG	TIJDRECORDS	TIJDUIT_1STE	TIJDUIT_VOLG	TIME	TL	TOON_VERNAUWING	TOPEN	TOT	TOT_DEBIET	TOTAAL	TOTLAADVERM	TOTOPEN	TOTOVER	TOTSCHUT	TS	TSHIP	TSLUIT	TW	TYPE	UCATIJ	UIT_OPSTELRUIMTE	UITVAREN	V_AFSTAND	VAARPLAN	VAARVAK	VAARWEGEN	VASTEBRUG	VERNAUWINGEN	VERNAUWINGSKANT	
AFGELEIDE																															
AFHANKELIJK																							R								
AFHANKELIJK INIT																															
BEPAAAL UITSTEL																															
BEPAAAL UITSTEL KOLK								R										R													
BRUGREGIME																															
BRUG REGISTRATIE										R		W																			
BUITEN SCHUTPERIODE																															
CHECK NETWERK																											R		R		
DISPLAY VERNAUWING						R	R												R										R		
FYSIEK									W											R											
GEEN SCHUTSTRATEGIE																									R						
GEMAAKT																									W						
HERDEEL																															
HEVELEN																															
HEVELEND																															
HEVELEND INIT																															
HEVELEN INIT																															
HULPJE																															
IDLE STOP WATCH																															
INITDEEL																															
INITIALISATIE				W		W																W	W				W		W		
INTIJD	R																														
KOLKMOD				W		W		R			W				W			R	W		R	R	R								
KOLK_PREFERENTIE																							R								
KOP																															
LEES BRUGGEN																					W						R				
LEES GENERATOREN		W		R													W										R		R		
LEES GETIJDEN																															
LEES KOSTEN																															
LEES REGIMES																															
LEES SLUIZEN								W										W			W										
LEES VAARKROMMEN	W		W	W																								W	R		
LEES VAARWEGEN																										W	R				
LEES VERNAUWINGEN																								W					R	W	
LEES WEGINTENS									W																						
MAINMOD				W							W																				
MANAGE BRUG						W					W				W				W	R											
MANAGE SLUIS						W													W												
MANAGE VERNAUWING						W													W											W	
PASSAGE VERNAUWING																															
PASSEN																															
PLAN THIS KOLK IN																															
QUERY REGISTRATIE																															
RAPPROT BRUGGEN													R																		
RAPPORT INITIA				R																							R		R		
RAPPORT SCHEEPVAART																															
RAPPORT SLUIZEN									W					R	R																
RAPPORT VERNAUWINGEN																													R		
RESULTAAT REGEL																															
SCHEPEN NAAR SLUIS																									R	R	R				
SCHIPMOD					W										W			W						R	R	R	R	R	R		
SCHUTSTRATEGIE GELYK																															
SCHUTSTRATEGIE NKOLK																															
SLUIS REGISTRATIE											R		W	W																	
SWITCH																															
TONDERDOOR																												R			
UITTIJD			R	R																		R									
VERNAUWINGSGARANTIE																														R	
VOEGTOE																															
VOLDOENDE BENUTTING																															
VOORKENNIS																															
VOORW DUMMY																															
WACHTTIJD SCHIP																															
WATERSTAND																															
WEGSCHRIJVEN BLADEN		W																													
ZETOVER OK																															
ZET WEKKER																															

	VGARKLAS	VIA_DOORGANG	VIA_KOLK	VLICHT	VOLGLENTE	VOLLE_REGELS	VOLSCHUT	VOOR_AANMELDING	VOORKENNISBOOM	VOORKENNISTIJD	VOORKEUR	VOORRANGSKANT	VOORW_INIT	VOORWAARDE	VORIGE_MIJLPAAL	VRIJE_KOLK	VRIJE_KOLKEN	VRIJSTELLING	VRIJVERKEER	VSCHIEFSTATS	VSPECIAAL	VULGRAAD	VVAARSNELHEID	VW_START	VWACHTRIJ	VZIJDE	W	WACHT_GEWICHT_CAT3	WACHT_OP_DEADLINE	WACHT_OP_PERIODE	WACHT_OP_VOORW
AFGELEIDE																															
AFHANKELIJK																															
AFHANKELIJK INIT														R																	
BEPAAAL UITSTEL																															
BEPAAAL UITSTEL KOLK										R																			R		
BRUGREGIME																															
BRUG REGISTRATIE																															
BUITEN SCHUTPERIODE																	W														
CHECK NETWERK																											R				
DISPLAY VERNAUWING				R																			R		R	R					
FYSIEK																			W									W			
GEEN SCHUTSTRATEGIE								R		R						R															
GEMAAKT	R														W							W									
HERDEEL																															
HEVELEN																															
HEVELEND																															
HEVELEND INIT															R																
HEVELEN INIT															R																
HULPJE																															
IDLE STOP WATCH																															
INITDEEL																															
INITIALISATIE	W	W	W					W								W															
INTIJD																															
KOLKMOD														R		R	R					W		W					W	W	W
KOLK PREFERENTIE																															
KOP																															
LEES BRUGGEN																															
LEES GENERATOREN						W																					R				
LEES GETIJDEN																															
LEES KOSTEN																															
LEES REGIMES										W																			W		
LEES SLUIZEN										W	W	W	W			W															
LEES VAARKROMMEN						W																									
LEES VAARWEGEN																															
LEES VERNAUWINGEN	R			W								W								W		W		W	W						
LEES WEGINTENS																															
MAINMOD																															
MANAGE_BRUG															W				W												
MANAGE_SLUIS																															
MANAGE_VERNAUWING				W																						R					
PASSAGE_VERNAUWING				R								R											R		R						
PASSEN																															
PLAN_THIS_KOLK_IN																															
QUERY REGISTRATIE		R	R			W										R															
RAPPROT_BRUGGEN																															
RAPPORT_INITIA																											R				
RAPPORT_SCHEEPVAART																					R						R				
RAPPORT_SLUIZEN							R																								
RAPPORT_VERNAUWINGEN																					R						R				
RESULTAAT_REGEL																															
SCHEPEN_NAAR_SLUIS									R																						
SCHIPMOD					R			R							W		W			R	R		R		R	R					
SCHUTSTRATEGIE_GELYK																R														W	R
SCHUTSTRATEGIE_NKOLK																R															
SLUIS REGISTRATIE						W	W																								R
SWITCH																															
TONDERDOOR																															
UITTIJD																															
VERNAUWINGSGARANTIE				W																							R				
VOEGTOE																															
VOLDOENDE BENUTTING																												W			
VOORKENNIS										W	R																				
VOORW DUMMY																															
WACHTTIJD SCHIP																													R		
WATERSTAND																															
WEGSCHRIJVEN BLADEN						R																									
ZETOVER OK																															
ZET WEKKER																															



	WACHTEN	WACHTEND	WACHTRIJ	WACHTRUIMTE	WACHTTIJD_ONMOGELIJK	WATCHER	WATERLENGTE	WATERPAD	WEEKTOT	WEG_HEEN	WEG_TERUG	WEGVERKEER	WEGSLUIS	WERK	WERKT	WIJDTE	WIT	WKOSTEN	X	X1	X2	XTRA_WACHTTIJD	XX	XXX	Y	YY	Z	ZIJDE	ZIJNBRUG	ZIJNSLUIS	ZIJNWACHTER
AFGELEIDE									R										W						W						
AFHANKELIJK																															
AFHANKELIJK INIT																															
BEPAAAL UITSTEL																															
BEPAAAL UITSTEL KOLK																						R								R	
BRUGREGIME																								W					R		
BRUG REGISTRATIE												R																			
BUITEN SCHUTPERIODE				R																											R
CHECK NETWERK																												R			
DISPLAY VERNAUWING		R															R														
FYSIEK																R			W									R			R
GEEN SCHUTSTRATEGIE				R																											R
GEMAAKT										R	R																				
HERDEEL																			W				W			W					
HEVELEN																															
HEVELEND	R																														
HEVELEND INIT																															
HEVELEN INIT																															
HULPJE																			W				W			W					
IDLE STOP WATCH																															
INITDEEL																							W			W					
INITIALISATIE	W																W														
INTIJD																															R
KOLKMOD	R	W		R			R												W						W						R
KOLK PREFERENTIE																											W				
KOP																															
LEES BRUGGEN			W			W		W	W			W						W	W	W					W			W	W		W
LEES GENERATOREN										W	W																				
LEES GETIJDEN																							R			R		R			
LEES KOSTEN																			W					W			W				
LEES REGIMES																			W	W	W	W			W			R			
LEES SLUIZEN				W	W		W												W											W	
LEES VAARKROMMEN																							R			R					
LEES VAARWEGEN																															
LEES VERNAUWINGEN																			W	W	W		R		W	R					
LEES WEGINTENS												R							W						W		W				
MAINMOD									R										W						W		W				
MANAGE BRUG		W	R			R						R		W	W				W						W		W		R		
MANAGE SLUIS		W		R			R						R	W																	R
MANAGE VERNAUWING		W																													
PASSAGE VERNAUWING																															
PASSEN																			W				W			W					R
PLAN THIS KOLK IN				R																											R
QUERY REGISTRATIE																															
RAPPROT BRUGGEN																		R						W				R			
RAPPORT INITIA																												R			
RAPPORT SCHEEPVAART																								R							
RAPPORT SLUIZEN																								W							
RAPPORT VERNAUWINGEN																								W							
RESULTAAT REGEL																															
SCHEPEN NAAR SLUIS																															
SCHIPMOD		W	R	R	R												R		W						W		W			R	R
SCHUTSTRATEGIE GELYK				R																											R
SCHUTSTRATEGIE NKOLK																															R
SLUIS REGISTRATIE																															R
SWITCH																															
TONDERDOOR																	R														
UITTIJD																															R
VERNAUWINGSGARANTIE																															
VOEGTOE																			W				W			W	W				
VOLDOENDE BENUTTING																															R
VOORKENNIS																			W												
VOORW DUMMY																															
WACHTTIJD SCHIP																															
WATERSTAND																															
WEGSCHRIJVEN BLADEN																															
ZETOVER OK																															
ZET WEKKER													W																		

