

IVERA-TLC
Verkeersregelininstallaties
Versie 4.00
4 april 2018

Een uitgave van
Stichting Beheer IVERA protocol
Zoetermeer, Nederland

Pub. No.: IVERA OD-VRI 4.00

Datum: 4 april 2018

Titel: IVERA TLC (versie 4.00)

Mocht u fouten of onvolledigheden ontdekken, of heeft u suggesties voor verbetering, dan stellen wij het zeer op prijs dat u deze stuurt naar:

Stichting Beheer IVERA protocol
Postbus 190
2700 AD Zoetermeer

© Copyright 2012-2020 Stichting Beheer IVERA protocol.

Alle rechten voorbehouden.

Niets uit deze uitgave mag worden gekopieerd, verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie of op welke andere wijze dan ook, zonder voorafgaande schriftelijke toestemming van de Stichting Beheer IVERA protocol.

Voorwoord

Nederland kent een groot aantal geregelde kruispunten voorzien van verkeersregelinstallaties. De verkeersregelinstallaties zijn in beheer bij Rijkswaterstaat, provincies en gemeentes. Voor een adequaat beheer van de verkeersregelinstallatie is uniformiteit in beheer een noodzaak, vooral voor beheerders met verkeersregelinstallaties van verschillende fabrikanten in hun beheersarsenaal.

Het IVER en de ASTRIN hebben de noodzaak tot standaardisatie onderkend en de wens uitgesproken in de toekomst alle nieuwe verkeersregelinstallaties te voorzien van een gestandaardiseerde communicatie interface voor de communicatie met een beheerscentrale.

De eerste versie van deze standaard is enkele jaren geleden opgesteld door een werkgroep bestaande uit afgevaardigden van het IVER, de vier grote gemeentes en het ASTRIN (industrie).

Wegens de gebleken behoefte bij wegbeheerders daaraan is door de Stichting Beheer IVERA protocol besloten een tweede versie van de communicatiestandaard te definiëren. Deze nieuwe versie bevat verbeteringen en uitbreidingen waarvan wegbeheerders en leveranciers te kennen gegeven hadden, dat zij gewenst waren.

In juni 2015 is opdracht verstrekt door het Ministerie van Infrastructuur en Milieu via het Beter Benutten Vervolg (BBV) programma aan de VRA leveranciers om te komen tot een gezamenlijke definitie van VRA standaarden ten behoeve van connected en coöperatieve functionaliteit.

De specificatie van versie 4 van het IVERA protocol is vastgelegd in 2 documenten:

- IVERA-APP 4
- IVERA-TLC 4

IVERA 4 gebruikt dezelfde syntax als voorgaande versies.

Het is echter niet backwards compatible.

Oudere beheerscentrales zullen geen verbinding kunnen maken met IVERA 4.

Het login bericht dusdanig is veranderd dat oudere beheerscentrales geen verbinding kunnen maken.

Inhoudsopgave

1.	Samenvatting.....	9
2.	Inleiding	10
2.1	Afkortingen.....	11
3.	Systeem overzicht	12
3.1	IVERA verbindingen	12
3.1.1	Master to Slave (object management).....	12
3.1.2	Slave to Master (event- en logboek afhandeling).....	12
4.	IVERA-protocol.....	14
4.1	Inleiding.....	14
4.2	Master-Slave	14
4.3	OSI-model.....	15
4.4	Toepassingen, Applicaties en Automaten.....	15
4.5	IVERA objecten.....	16
4.6	Objectdefinitie	21
4.7	Objectsoorten.....	27
4.8	Gebruikers	29
4.9	Berichtdefinitie.....	30
4.10	Object Elementbereik	31
4.11	Data context diagrammen	33
4.11.1	Lezen van objecten.....	33
4.11.2	Schrijven van objecten	33
4.11.3	Schrijven van meerdere elementen.....	34
4.11.4	Lezen van objectattributen	34
4.11.5	Wijziging van objectattributen (optioneel).....	35
4.11.6	Master-slave synchronisatie	35
4.11.7	Gebeurtenis in de slave (trigger)	36
5.	Beheer van objecten	37
5.1	Homoniemen.....	37
5.2	Algemene objecten.....	38
5.3	Toepassings specifieke objecten	38
5.4	Automaatspecifieke objecten.....	38
6.	Beveiliging en gebruikers.....	39
6.1	Gebruikers	39
6.2	Lokale wijziging van gebruikersnaam en password	39
6.3	Wijziging gebruikersnaam en password	39
6.4	Beheersaspecten	40
7.	Functionele omschrijving	43
7.1	Functionele modificatie van parameters (FMP)	43
7.2	Monitoring	43
7.3	Data verzameling	44
7.4	Toevoegen extra elementen.....	44
7.5	Naamconventie objecten.....	44
8.	Dataformaten	47
8.1	Object type 0	47
8.2	Object type 1	48
9.	Management Interface TLC Faciliteiten	49
9.1	Object ITSAPPLIC	50
9.1.1	Object ITSAPP.I.....	51

9.1.2	Object ITSAPP	51
9.1.3	Object ITSSTAT	52
9.2	IVERA Gebruiker Beheer	53
9.2.1	Object USER	53
10.	Object definitie	55
10.1	Gebruikersgroepen	55
10.2	Overzicht alle objecten	55
11.	Datum en tijd	56
11.1	Algemeen datum en tijd	56
11.2	Objecten datum en tijd	57
11.2.1	KTIJD	57
11.2.2	TIJD	58
11.2.3	DATUM	59
11.2.4	JAAR	59
11.2.5	WEEK	60
11.2.6	DAG.I	61
11.2.7	DAG	62
11.2.8	BIJZDAG	63
11.2.9	WKZB	64
11.2.10	WKZE	64
12.	Identificatie	65
12.1	Algemeen identificatie	65
12.2	Objecten identificatie	65
12.2.1	VRIID.I	65
12.2.2	VRIID	67
12.2.3	VRIVER.I	68
12.2.4	VRIVER	69
12.2.5	TID	70
12.2.6	XID	70
12.2.7	Automaat specifieke events	70
13.	Toegang	71
13.1	Algemeen toegang	71
13.2	Objecten toegang	73
13.2.1	FTPUSER.I	73
13.2.2	FTPPASS	74
13.2.3	FTPLOCATION	76
13.2.4	DATA.COM.I	77
13.2.5	DATA.COM	79
13.2.6	AUTHOG	80
13.2.7	AUTHOP	81
13.2.8	LOGINNIVEAU	82
13.2.9	LOGIN	82
13.2.10	Datacommunicatie events	83
14.	VRI algemeen	84
14.1	Algemeen VRI algemeen (procesbesturing)	84
14.2	Objecten VRI algemeen (procesbesturing)	87
14.2.1	TP.I	87
14.2.2	TP	88
14.2.3	TP.A	89
14.2.4	SP.I	89
14.2.5	SP	90
14.2.6	EXTRAINFO.I	91
14.2.7	EXTRAINFO.A	92
14.2.8	EXTRAINFOEXT	93
14.2.9	VRI.C	94

14.2.10	Commando-events	95
14.2.11	VRI.A	97
14.2.12	PAR.LB	100
14.2.13	PAR.LA	101
14.2.14	Programma-events	102
14.2.15	Commando's en events	103
15.	Protocol	105
15.1	Algemeen protocol	105
15.2	Objecten protocol	106
15.2.1	ERROR.CODE	106
15.2.2	ERROR.INFO	106
15.2.3	ERROR.CMD	107
15.2.4	ABON	108
15.2.5	BB0	110
15.2.6	BB1	110
15.2.7	BBA0	111
15.2.8	BBA1	111
15.2.9	PING	112
16.	VRI status	113
16.1	Algemeen VRI status	113
16.2	Objecten VRI status	118
16.2.1	VRISTAT.I	118
16.2.2	VRISTAT	119
16.2.3	VRIPROG.I	120
16.2.4	VRIPROG	122
16.2.5	VRISUBPROG	123
16.2.6	VRIPROGLYST	124
16.2.7	VRIPROGLYSTEXT	125
16.2.8	BEDRIJF.I	126
16.2.9	BEDRIJF	126
16.2.10	BEDRIJF events	127
16.2.11	Resetevents	128
17.	Klokperioden	130
17.1	Algemeen klokperioden	130
17.2	Objecten klokperioden	132
17.2.1	KLA1	132
17.2.2	KLU1	132
17.2.3	KLA2	133
17.2.4	KLU2	133
17.2.5	KLA3	133
17.2.6	KLU3	133
17.2.7	KLA4	133
17.2.8	KLU4	133
17.2.9	KLA5	133
17.2.10	KLU5	133
17.2.11	KLOKPER.I	134
17.2.12	KLOKPER	134
17.2.13	KLOKPER.A	136
17.2.14	KLOKPROG.I	137
17.2.15	KLOKPROG	138
17.2.16	KLOKPROG.A	139
18.	Detectie	140
18.1	Algemeen detectie	140
18.2	Objecten detectie	143
18.2.1	D.I	143
18.2.2	D.A	144

18.2.3	TD.A	146
18.2.4	SWD	147
18.2.5	D.LB	148
18.2.6	D.LA	149
18.2.7	TDBG	150
18.2.8	TDOG	151
18.2.9	TDFL	152
18.2.10	CDFL	152
18.2.11	TDBP1	152
18.2.12	TDBP2	153
18.2.13	DC.I	154
18.2.14	DC.A	155
18.2.15	LD	157
18.3	Detectie fout event	158
19.	In- en uitgangen	160
19.1	Algemeen in- en uitgangen	160
19.2	Objecten in- en uitgangen	161
19.2.1	U.I	161
19.2.2	U.A	162
19.2.3	TU.A	162
19.2.4	U.LB	163
19.2.5	U.LA	164
19.2.6	I.I	165
19.2.7	I.A	166
19.2.8	TI.A	166
19.2.9	SWI	167
19.2.10	I.LB	168
19.2.11	I.LA	169
19.2.12	LI	170
19.2.13	LU	171
20.	Signaalgroepen	172
20.1	Algemeen signaalgroepen	172
20.2	Objecten signaalgroepen	176
20.2.1	SG.I	176
20.2.2	SGE.A	177
20.2.3	TSGE.A	177
20.2.4	SGE.LB	178
20.2.5	SGE.LA	178
20.2.6	TGOR1	179
20.2.7	TGG1	180
20.2.8	TGGL1	180
20.2.9	TMGL1	180
20.2.10	TGR1	181
20.2.11	LAMP.I	181
20.2.12	LAMP.A	183
20.2.13	LAMPINFO	184
20.2.14	DIMINST.I	186
20.2.15	DIMINST	188
20.2.16	DIMMEN.I	189
20.2.17	DIMMEN.A	191
20.2.18	AKOESTISCH.I	192
20.2.19	AKOESTISCH.A	193
20.2.20	AKOESTISCH.F	194
20.2.21	KLA_AKOEST	194
20.2.22	KLU_AKOEST	196
20.2.23	KLA_HARD	196
20.2.24	KLU_HARD	197
20.2.25	PAKOESTISCH.I	198

20.2.26	PAKOESTISCH	199
20.2.27	LSGE	200
20.2.28	Events	201
21.	Openbaar vervoer	202
21.1	Algemeen Openbaar vervoer	202
21.2	Objecten openbaar vervoer	204
21.2.1	OVDEVICE	204
21.2.2	OVFILTER	205
21.2.3	OV.LB	206
21.2.4	OV.LA	208
22.	Noodstroom	209
22.1	Algemeen noodstroom	209
22.2	Objecten noodstroom	210
22.2.1	NOODSTROOM.I	210
22.2.2	NOODSTROOM	212
22.2.3	NOODSTROOM.A	213
22.2.4	NOODSTROOM.LA	214
22.2.5	NOODSTROOM.LB	216
23.	Telprogramma	217
23.1	Algemeen telprogramma	217
23.2	Objecten telprogramma	218
23.2.1	TELINST	218
23.2.2	TELDATA	221
23.2.3	TELMON	223
24.	Events en Alarms	224
24.1	Algemeen Events en Alarms	224
24.2	Objecten Events en Alarms	225
24.2.1	EVENTLYST.I	225
24.2.2	EVENTLYST.INFO	226
24.2.3	VRIFOUT.I	227
24.2.4	VRIFOUT	228
24.2.5	VRIFSUB.I	229
24.2.6	VRIFSUB	230
24.2.7	VRI.LB	231
24.2.8	VRI.LA	232
25.	Bijlage: BNF-notatie	234

1. Samenvatting

Het IVERA protocol is ontstaan naar aanleiding van de door het IVER en ASTRIN onderkende noodzaak tot standaardisatie van de communicatie tussen verkeersregelininstallatie en beheercentrale. Het IVERA protocol definieert daartoe objecten in de verkeersregelininstallaties die overeenkomen met functionaliteit. De beheercentrales kan deze objecten (functies) manipuleren door te lezen en te schrijven. Door deze opzet realiseert het IVERA protocol niet alleen standaardisatie van de communicatie maar ook verdergaande standaardisatie van de functionaliteit van een verkeersregelaar; een standaardisatie die werd ingezet door de CVN C-interface.

De opzet van het IVERA-protocol gaat zoveel mogelijke uit van het gebruik van standaard communicatie faciliteiten voor zowel infrastructuur als communicatie software. Daartoe bevindt het IVERA-protocol zich op laag 7 “applicatie laag” van het OSI-model. Tevens wordt alleen gebruik gemaakt van ASCII tekst in de communicatie. Verder is gekozen voor een master/slave protocol waarin de beheerscentrale master is en de VRI slave.

Functioneel levert het IVERA-protocol de mogelijkheid te lezen van en schrijven naar zogeheten objecten in de slave. De objecten hebben een unieke naam en worden door een instantie beheerd zodat namen niet dubbel uitgegeven kunnen worden. Elk object heeft een aantal attributen die de technische aspecten van een object beschrijven zoals omschrijving, type (soort gegevens), het aantal elementen, enzovoort. De data van het object is in de slave opgeslagen en kan benaderd worden via de object definitie. Op deze wijze kan de beheerscentrale gegevens uit de VRI lezen of gegevens naar de VRI schrijven.

De uiteindelijke functionaliteit van de slave wordt bepaald door de aanwezigheid van objecten. Om deze functionaliteit vast te kunnen stellen zijn standaard objecten gedefinieerd waarmee de beheerscentrale kan vaststellen welke objecten ondersteund worden en wat het type van de slave is.

De objecten (functies) van een verkeersregelininstallatie kunnen in verschillende groepen gecategoriseerd worden, te weten:

- Events; meldingen die altijd aan de beheercentrale gemeld moeten worden; er kan bijvoorbeeld gedacht worden aan storingen.
- Monitoring; gegevens met tot doel het meekijken met de actuele toestanden; vooral bij signaalgroepstoestanden is dit van belang.
- Dataverzameling; gegevens die (op basis van programmering) worden verzameld voor verdere verwerking door de beheerscentrale.
- FMP; parameters in de verkeersregelininstallatie.
- Service; (fabrikantafhankelijke) parameters ten behoeve van service en onderhoud.

Georganiseerd naar data formaat bestaan twee typen objecten:

- type 0 voor informatie die met behulp van getallen weergegeven kan worden en
- type 1 voor informatie die met behulp van tekst (ASCII string) weergegeven kan worden.

Met betrekking tot de naamgeving geldt dat het IVERA-protocol richtlijnen geeft. Daarbij zijn (groepen) namen gereserveerd. Alle namen worden vastgelegd met een beschrijving van de werking. Hierdoor kan het IVERA-protocol inclusief de objecten tot in lengte van dagen compatibel blijven en een eenduidige werking garanderen.

Ter beveiliging zijn gebruikers groepen gedefinieerd met behulp van gebruikersnamen en wachtwoorden.

2. Inleiding

Dit document bevat een beschrijving van alle objecten die relevant zijn voor de communicatie tussen een verkeersregelinstallatie (VRI) en een beheercentrale. Het IVERA-protocol is een initiatief van het IVER, CVN en de ASTRIN. De naam IVERA is een samenvoeging van IVER en ASTRIN.

Voor de definitie van een object en de manier waarop de beheercentrale objecten kan manipuleren wordt verwezen naar hoofdstuk 10 en verder. Bij de beschrijving van de objecten wordt uitgegaan van een standaard VRI. De standaard VRI is beschreven in hoofdstuk **Error! Reference source not found.**

Het IVERA-protocol is in eerste instantie ontwikkeld voor communicatie tussen verkeersregelinstallaties en een beheerscentrale. Naast deze toepassing is het IVERA-protocol ook geschikt voor andere toepassing zoals:

- Communicatie met toeritdoozeertoestellen
- Aansturing van borden in een parkeerverwijzingssysteem

Bij de opzet van het IVERA-protocol wordt zoveel mogelijk uitgegaan van het gebruik van standaard communicatie faciliteiten voor zowel de communicatie infrastructuur als de communicatiesoftware. De voordelen van deze aanpak zijn:

- Ondersteuning van diverse communicatienetwerken.
- Ondersteuning van gangbare protocollen.
- Fabrikantonafhankelijke oplossing.
- Minimale ontwikkelingsinspanning door het gebruik van standaard hardware en software componenten.

De technische keuzen ten aanzien van communicatie netwerken vallen buiten de scope van dit document. Hiervoor wordt verwezen naar het technisch ontwerp. Binnen de werkgroep is echter wel de voorkeur uitgesproken voor protocollen uit de TCP/IP suite.

De voorbeelden die in dit document voorkomen hebben betrekking op de toepassing van IVERA-protocol voor de communicatie tussen een verkeersregelinstallatie en een beheerscentrale.

Het IVERA-protocol wordt beheerd door de Stichting Beheer ASTRIN/IVERA-protocol. Voor het gebruik van het IVERA-protocol tussen een verkeersregelinstallatie en een beheerscentrale worden door de stichting licenties uitgegeven. Door Ko Hartog Verkeerstechiek is een testsuite worden ontwikkeld voor de certificering van de implementaties in zowel beheerscentrales als verkeersregelinstallaties.

Vanaf IVERA versie 4 is de IVERA specificatie in 2 documenten gesplitst:

- IVERA-APP
- IVERA-TLC

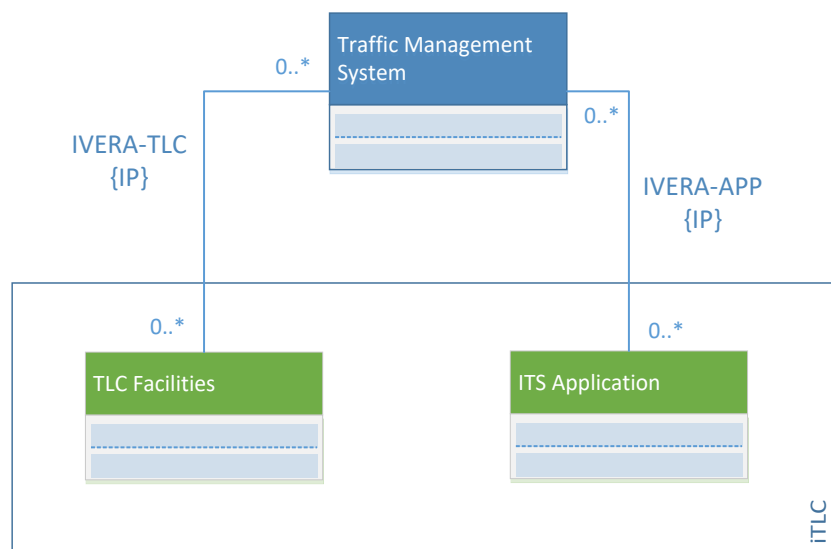
Voor beide IVERA versies is een aparte specificatie beschikbaar.

2.1 Afkortingen

ASTRIN	ASsociation of TRaffic Industries in the Netherlands
BNF	Backus-Naur Form
CIF	CVN C-interface
CVN	Contactgroep Verkeersregeltechnici Nederland
FMP	Functionele Modificatie van Parameters
GUS	Gewenste uitgangssturing
IVER	Initiatiefgroep Verkeersregeltechnici Rijkswaterstaat en Provincies
IVERA	IVER/ASTRIN
OSI	Open Systems Interface
pincode	Persoonlijke Indentificatie Nummer-code
PSTN	Public Switched Telephone Network
SG	Signaalgroep
SWICO	Software Input Commando
TCP/IP	Transport Communicatie Protocol / Internet Protocol
UIC	User Identification Control
UNIX	Computer Operating System
VRI	Verkeersregelinstallatie
WUS	Werkelijke uitgangssturing

3. Systeem overzicht

Deze sectie beschrijft de functionele blokken van een iTLC met IVERA interfaces.



Afbeelding 1 Functioneel model

Error! Reference source not found. afbeelding 1 maken de IVERA-APP en IVERA-TLC verbinding het mogelijk voor het Traffic Management System (TMS) om ITS Applicaties en TLC Faciliteiten te beheren.

De TLC Faciliteiten gedragen zich als een IVERA slave en elke ITS Applicatie kan zich als een IVERA slave gedragen. Elke IVERA slave verschaft een IVERA interface met zijn eigen set IVERA objecten.

Het IVERA-protocol is gebaseerd op objecten, waarbij een object overeenkomt met een bepaalde functie in de VRI. De beschrijving van al deze objecten bij elkaar vormt een abstracte beschrijving van de functionaliteit van een VRI. Hierdoor leidt het IVERA protocol niet alleen tot standaardisatie van de communicatie tussen de VRI en de beheercentrale maar ook tot een verdere standaardisatie van de functionaliteit van een VRI. Een eerste stap tot standaardisatie was de CVN C-interface en applicatiepakketten zoals CCOL en RWSC. Dit hoofdstuk bevat een functionele beschrijving van onderdelen van de standaard VRI. Bij de beschrijving van de objecten wordt hier naar verwezen.

3.1 IVERA verbindingen

3.1.1 Master to Slave (object management)

De IVERA interface is gedefinieerd als een TCP/IP socket verbinding met een berichten syntax.

De TCP/IP server poort voor toegang tot de IVERA-TLC interface is 5200 voor unsafe verbindingen (zonder TLS) en 5300 voor safe verbindingen (met TLS).

Toegang tot een IVERA-APP interface (ook een TCP/IP server port) werkt iets anders. Om verbinding te kunnen maken met verschillende IVERA-APP interfaces (op het zelfde platform, of op andere platforms, is het IVERA-object ITSAPPLOC gedefinieerd voor de IVERA-TLC interface. De IVERA master kan eerst dit object opvragen met behulp van de IVERA-TLC interface, en vervolgens de juiste verbindingseigenschappen voor een ITS Applicatie bepalen.

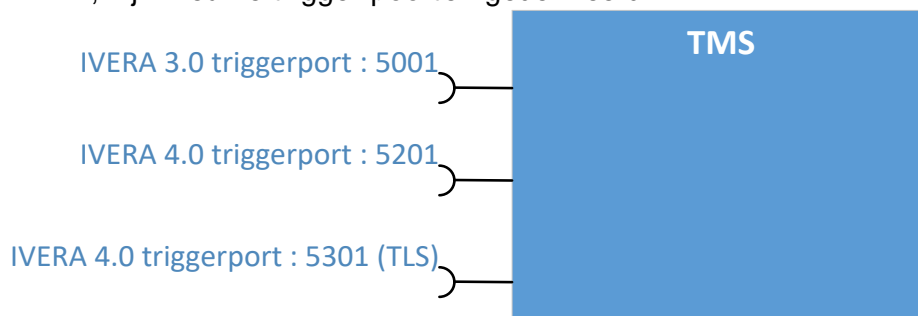
3.1.2 Slave to Master (event- en logboek afhandeling)

De IVERA TMS luistert op een TCP port om berichten te ontvangen die worden verstuurd door IVERA slaves. Deze berichten worden 'trigger events' genoemd.

Hiervoor wordt poort 5201 gebruikt voor unsafe verbindingen (zonder TLS) en 5301 voor safe verbindingen (met TLS).

Een IVERA slave stuurt trigger events om de TMS (IVERA master) te berichten over nieuw opgetreden events. De IVERA master kan dan de bijbehorende actie uitvoeren (zoals het opvragen van logboeken bij de IVERA slave)
Trigger events worden verstuurd naar een trigger poort van de TMS.

Voor IVERA 4, zijn nieuwe trigger-poorten gedefinieerd:



Afbeelding 2 Trigger poorten op een TMS

In IVERA versies vor versie 4, de TMS kon de IVERA slave identificeren door het source IP-adres van waar het trigger event verstuurd te gebruiken. Aangezien meerdere IVERA slaves binnen een systeem kunnen zijn ingezet (waarbij een network stack en IP-adres worden gedeeld), wordt het IVERA trigger uitgebreid met een trigger source identificatie welke door de IVERA master wordt gebruikt om de network location van de bijbehorende IVERA slave te bepalen.

De data welke door de IVERA-slave wordt verstuurd is uitgebreid met de inhoud van het volgende ID object:

1. APPID for triggers from IVERA-APP
2. VRIID for triggers from IVERA-TLC

De trigger events worden verstuurd als een reeks berichten:

```
BerichtSlaveAckHand CarriageReturn  
BerichtSlaveTrigger CarriageReturn  
{ BerichtSlaveTrigger CarriageReturn }
```

Voorbeeld:

```
VRIID= "V10002","KRP55","Dorpstraat/Kerkstraat","FAB X Type Y","iTLC","1997-01-  
17","", "", "", "", ""  
:T= 2001  
:T= 1010
```

De volgorde is:

- Slave verbindt met de trigger poort op de master.
- Slave verstuurt bericht BerichtSlaveAckHand met inhoud van het ID object
- De volgende berichten bevatten een of meerdere BerichtSlaveTrigger
- De slave verbreekt de verbinding.
- Master takes appropriate actions to follow up on the trigger message(s) taking the received ID into account. (e.g. requesting logbook objects)

4. IVERA-protocol

4.1 Inleiding

Het IVERA-protocol is een master-slave protocol op laag 7 (applicatie) van het OSI-model. Via het IVERA-protocol kan een master (beheerscentrale) objecten in een slave (VRI) lezen en schrijven. Een object is gedefinieerd als:

- Een object binnen het IVERA-protocol is iets dat je kunt selecteren en manipuleren als een eenheid.
- Om een object te kunnen selecteren heeft ieder object een unieke naam.
- Alle data van een object is van hetzelfde type.

De volgende tabel geeft een aantal voorbeelden van objecten:

Object	Omschrijving
SG.I	Een object dat de functionele namen van alle signaalgroepen bevat.
VRI.LA	Een object dat alle programma-events bevat die nog niet door de master zijn bevestigd.
VRI.C	Een object waarmee commando's aan de slave gegeven kunnen worden.

Tabel 3.1. Voorbeelden van objecten

De beschrijving van het IVERA-protocol is als volgt opgebouwd:

Paragraaf	Inhoud
Master-Slave	Beschrijving van het gebruikte master-slave principe.
OSI-model	Beschrijving van de plaats van het IVERA-protocol in het 7 lagen OSI-model en de eisen die gesteld worden aan onderliggende netwerklagen.
Toepassing, Applicaties en Automaten	Beschrijving van deze terminologie.
Gebruikers	Een definitie van de begrippen gebruiker en gebruikersgroep
Bericht definitie	Een beschrijving van het protocol tussen de IVERA master en slave
Object definitie	Een algemene definitie van het begrip object.
Object kenmerken	Een aanvulling op de objectdefinitie in de vorm van een aantal objecten met bijzondere kenmerken.

Tabel 3.2. Opbouw beschrijving IVERA-protocol

4.2 Master-Slave

Het IVERA-protocol kent een master (de beheerscentrale) en een slave (de verkeers-regelinstallatie).

Opbouwen van een verbinding

Zowel de master als de slave kunnen het initiatief nemen tot het opbouwen van een verbinding. Nadat de verbinding tot stand is gekomen is er sprake van een master-slave protocol.

Als de master met de slave verbindt is kan de master IVERA objecten opvragen/beïnvloeden. Als de slave met de master verbindt kan de slave alleen maar events versturen.

Om verbinding te maken met de verschillende IVERA-APP interfaces, is het IVERA object ITSAPPLOC gedefinieerd in IVERA-TLC. De master kan dit object eerst opvragen via de IVERA-TLC interface en daaruit de juiste verbinding gegevens halen voor de verbinding met een IVERA-APP interface.

IVERA-APP ondersteunt Transport Layer Security (TLS) zoals beschreven in RFC7525.

Opvragen van informatie

Een master kan informatie opvragen uit de slave door het lezen van objecten.

Schrijven van informatie

Een master kan o.a. een parameterinstelling in de slave wijzigen door het schrijven van de nieuwe waarde naar het bijbehorende object in de slave.

Geven van commando's

Een master kan een commando sturen naar een slave door het schrijven naar een object. De betekenis van het commando is in de definitie van het object opgeslagen.

Een gebeurtenis (event) in de slave

Bij een event in de slave waarover de master geïnformeerd moet worden zal de slave eerst een verbinding opbouwen met de master. Als de verbinding aanwezig is zal de slave een event bericht <BerichtSlaveTrigger> versturen naar de master. Een eventbericht dat autonoom van de slave naar de master wordt verstuurd bevat niet de eventdata, maar is slechts een trigger dat een gebeurtenis is opgetreden. Na het ontvangen van een eventbericht is het de taak van de master om de juiste informatie uit de slave te lezen.

4.3 OSI-model

Het IVERA-protocol bevindt zich op laag 7 “de applicatie laag” in het OSI model. Het IVERA-protocol communiceert met onderliggende netwerklagen via standaard functies (stream of file I/O). Het IVERA-protocol doet de volgende aannames, ten aanzien van onderliggende interfacelagen.

- Onderliggende interfacelagen zorgen voor het opzetten en in stand houden van een “connection oriented” verbinding tussen IVERA-master en IVERA-slave. Een mogelijke invulling van zo'n verbinding is een verbinding op basis van TCP/IP sockets.
- Onderliggende interfacelagen dragen er zorg voor dat de bytes verstuurd door de IVERA-master foutvrij en in dezelfde volgorde aankomen bij de IVERA-slave en visa versa.
- Onderliggende interface lagen dragen zorg voor segmentering en routing.
- Indien datacompressie gewenst dan wel noodzakelijk is, dient dit in lagere interface lagen geïmplementeerd te worden.
- Indien data-encryptie noodzakelijk is, dient dit in lagere interface lagen geïmplementeerd te worden.
- Het IVERA-protocol werkt ten minste over TCP/IP en PPP en eventueel ook rechtstreeks op een fysieke verbinding.

4.4 Toepassingen, Applicaties en Automaten

Het IVERA-protocol is geschikt voor de meerdere toepassingen, zoals o.a.

- Communicatie tussen een beheerscentrale en een verkeersregelininstallatie,
- Communicatie met toeritdoseertoestellen,
- Aansturing van borden in een parkeerverwijzingssysteem.

Voor een toepassing is er een unieke set van objecten. De naamgeving van de objecten binnen een toepassing is afgeleid van de voor die toepassing gebruikte naamconventie.

Een object is uniek gedefinieerd met: <TID, ObjectNaam>
waarbij: TID = toepassing identificatie

Binnen een toepassing kunnen er verschillende hardwareplatforms (automaten) bestaan, ieder met hun eigen specifieke mogelijkheden. De naam van een automaat specifiek object begint altijd met de letter 'X'.

Een automaatspecifiek object is uniek gedefinieerd met: <TID, XID, ObjectNaam>
waarbij: XID = automaatidentificatie.

De achtergrond van deze opzet is dat ontwikkelaars van toepassingen, applicaties en automaten, zelfstandig hun objecten kunnen definiëren, wat het beheer van objecten sterk vereenvoudigt.

NB. Uiteraard is het streven om per toepassing de objecten zoveel mogelijk te standaardiseren.

4.5 IVERA objecten

Het IVERA protocol beschrijft toegang tot IVERA objecten. Sommige objecten hebben betrekking op verkeerskundige controle applicaties en andere objecten hebben betrekking op de TLC: dit wordt per object weergegeven in de onderstaande tabel. Sommige objecten hebben betrekking op zowel de TLC als op ITS Applicaties en dienen voor beide te worden geïmplementeerd.

Ter ondersteuning van de iTLC-architecture zijn nieuwe objecten toegevoegd of objecten gewijzigd. Deze zijn in cursief weergegeven.

X = object kan worden verwacht in deze interface.

P = object is protocol gerelateerd. Object is onderdeel van het IVERA protocol zelf of nodig om IVERA te implementeren en heeft geen directe relatie met de inhoud van de slave.

Name	Description ¹	IVERA-APP	IVERA-TLC	Mandatory
KTIJD	Kalendertijd		X	
<i>TIJD</i>	<i>Actuele systeemtijd</i>	X	X	
<i>DATUM</i>	<i>Actuele systeemdatum</i>	X	X	
JAAR	Actueel jaar		X	
WEEK	Weeknummer		X	
DAG	Nummer van dag van de week		X	
DAG.I	Index dag van de week		X	
BIJZDAG	Bijzondere dag		X	
WKZB	Weeknummer begin zomertijd		X	
WKZE	Weeknummer einde zomertijd		X	
VRIID	Automaatidentificatie		X	
VRIID.I	Index automaatidentificatie		X	
VRIVER	Versienummers		X	
VRIVER.I	Index versienummers		X	
VRISTAT	Automaat toestand		X	
VRISTAT.I	Index statusbronnen		X	
VRIPROG	Automaat programma		X	
VRISUBPROG	Automaat subprogramma		X	
VRIPROG.I	Index programmabronnen		X	
VRIPROGLYST	Programmalijs.		X	
VRIPROGLYSTEXT	Uitgebreide programmalijs		X	
BEDRIJF	Bedrijfstoestand m.b.t. de centrale.		X	
BEDRIJF.I	Index object BEDRIJF		X	
FTPUSER.I	FTP-gebruikersnamen	X	X	
FTPPASS	FTP-passw o rds	X	X	
FTPLOCATION	FTP-locatie	X	X	

¹ The description is in Dutch. It is a copy from the specification which is written in Dutch.

Name	Description ¹	IVERA-APP	IVERA-TLC	Mandatory
VRIFOUT	Actuele foutcode		X	
VRIFOUT.I	Index foutcodes		X	
VRIFSUB	Fout status van subsystemen.		X	
VRIFSUB.I	Index subsystemen.		X	
KLA1	Inschakelen regelen periode 1		X	
KLU1	Uitschakelen regelen periode 1		X	
KLA2	Inschakelen regelen periode 2		X	
KLU2	Uitschakelen regelen periode 2		X	
KLA3	Inschakelen regelen periode 3		X	
KLU3	Uitschakelen regelen periode 3		X	
KLA4	Inschakelen regelen periode 4		X	
KLU4	Uitschakelen regelen periode 4		X	
KLA5	Inschakelen regelen periode 5		X	
KLU5	Uitschakelen regelen periode 5		X	
KLOKPER	Hulpelement klokperiode		X	
KLOKPER.A	Stand klokperiode		X	
KLOKPER.I	Index object KLOKPER		X	
KLOKPROG	Klokperiode programmakeuze		X	
KLOKPROG.A	Stand klokperiode programmakeuze		X	
KLOKPROG.I	Index object KLOKPROG		X	
KLA_AKOEST	Inschakelen akoestische signalen		X	
KLU_AKOEST	Uitschakelen akoestische signalen		X	
KLA_HARD	Inschakelen hoog geluidsvolume		X	
KLU_HARD	Uitschakelen hoog geluidsvolume		X	
CIFGUS	CIF gewenste uitgangssturing	X		
CIFWUS	CIF werkelijk uitgangssturing	X		
CIFIS	CIF ingangsstatus	X		
CIFWPS	CIF werkelijke programmastatus	X		
CIFGPS	CIF gewenste programmastatus	X		
CIFKLOK	CIF kalendertijd	X		
CIFPARM1	CIF parameter tabel 1	X		
CIFPARM2	CIF parameter tabel 2	X		
TGOR	Garantieontruimingstijd (appl)	X		
TGOR1	Garantieontruimingstijd (proces)		X	
TOR	Ontruimingstijd (appl)	X		
TGG	Garantiegroentijd (appl)	X		
TGG1	Garantiegroentijd (proces)		X	
TGGL	Garantiegeeltijd (appl)	X		
TGGL1	Garantiegeeltijd (proces)		X	
TMGL	Maximum geeltijd (appl)	X		
TMGL1	Maximum geeltijd (proces)		X	
TGR	Garantieroodtijd (appl)	X		
TGR1	Garantieroodtijd (proces)		X	
TVG	Vastgroentijd	X		

Name	Description ¹	IVERA-APP	IVERA-TLC	Mandatory
TVAG	Voertuigafhankelijk verlenggroen	X		
TGL	Geeltijd	X		
TMG	Actuele maximumgroentijd	X		
TMG1	Maximumgroentijd 1	X		
TMG2	Maximumgroentijd 2	X		
TMG3	Maximumgroentijd 3	X		
TMG4	Maximumgroentijd 4	X		
TMG5	Maximumgroentijd 5	X		
TMG6	Maximumgroentijd 6	X		
TDH1	Actuele 1e hiaattijd	X		
TDH11	1e hiaattijd periode 1	X		
TDH12	1e hiaattijd periode 2	X		
TDH2	Actuele 2e hiaattijd	X		
TDH21	2e hiaattijd periode 1	X		
TDH22	2e hiaattijd periode 2	X		
TDH	Hiaattijd	X		
TDB	Bezettijd voor aanvraag	X		
TDOG	Bewakingstijd ondergedrag		X	
TDBG	Bewakingstijd bovengedrag		X	
TDFL	Meettijd fluttergedrag		X	
CDFL	Grenswaarde fluttergedrag		X	
TDBP1	Tijd detectiebewaking aan		X	
TDBP2	Tijd detectiebewaking uit		X	
T	Tijdstelling (appl)	X		
T.A	Lopende tijd (appl)	X		
T.I	Index timers (appl)	X		
T.T	Type tijden	X		
C	Counterinstelling (appl)	X		
C.A	Lopende counter (appl)	X		
C.I	Index counters (appl)	X		
C.T	Type counters	X		
P	Parameterinstelling (appl)	X		
P.I	Index parameters (appl)	X		
P.T	Type parameters (appl)	X		
EGGP	EGG parameterinstelling (appl)	X		
EGGP.I	Index EGG parameters (appl)	X		
EGGP.T	Type EGG parameters (appl)	X		
S	Schakelaar (appl)	X		
S.I	Index schakelaars (appl)	X		
S.T	Type schakelaars (appl)	X		
KLB	Klok parameter 1	X		
KLE	Klok parameter 2	X		
KL.I	Index klokparameters	X		
TP	Tijd instelling (proces)		X	
TP.A	Lopende tijd (proces)		X	
TP.I	Index timers (proces)		X	
CP	Counter instelling (proces)	X		
CP.A	Lopende counter (proces)	X		
CP.I	Index counters (proces)	X		

Name	Description ¹	IVERA-APP	IVERA-TLC	Mandatory
PP	Parameter instelling (proces)	X		
PP.I	Index parameters (proces)	X		
SP	Schakelaar (proces)		X	
SP.I	Index schakelaars (proces)		X	
SGE.A	Signaalgroep toestand (ext)		X	
SGI.A	Signaalgroep toestand (int)	X		
TSGE.A	Timer signaalgroep toestand (ext)		X	
TSGI.A	Timer signaalgroep toestand (int)	X		
SGE.LB	Signaalgroep logboek (ext)		X	
SGI.LB	Signaalgroep logboek (int)	X		
SGE.LA	Signaalgroep logboek (onb/ext)		X	
SGI.LA	Signaalgroep logboek (onb/int)	X		
SG.I	Signaalgroep namen	X	X	
LAMP.I	Index lampnamen		X	
LAMP.A	Actuele lampstatus		X	
LAMPINFO	Lamp configuratie		X	
D.A	Detector toestand		X	
TD.A	Timer bezet/onbezet		X	
SWD	Software detector schakelaar		X	
D.LB	Detector logboek		X	
D.LA	Detector logboek (onb)		X	
D.I	Detector namen	X	X	
DC.A	Classificatie detector toestand		X	
DC.I	Index object DC		X	
U.A	Toestand overige uitgangen	X	X	
TU.A	Timer uitgang toestand	X	X	
U.LB	Uitgangen logboek	X	X	
U.LA	Uitgangen logboek (onb)	X	X	
U.I	Index overige uitgangen	X	X	
I.A	Toestand overige ingangen		X	
TI.A	Timer ingang toestand		X	
SWI	Software input schakelaar		X	
I.LB	Ingangen logboek		X	
I.LA	Ingangen logboek (onb)		X	
I.I	Index overige ingangen		X	
LSGE	Lijndump SG-toestand (ext)		X	
LSGI	Lijndump SG-toestand (int)	X		
LD	Lijndump detector toestand		X	
LI	Lijndump ingang toestand		X	
LU	Lijndump uitgang toestand		X	
BL.A	Actueel blok/module/stage	X		
PL.I	Index signaalplannen	X		
PLTXMAX	maximum waarde cyclustijd (* TX_max)	X		
PLTPLON	inschakeltijd signaalplan (* TPL_on)	X		
PLTPLOFF	uitschakeltijd signaalplan (* TPL_off)	X		

Name	Description ¹	IVERA-APP	IVERA-TLC	Mandatory
PLTXA	parameter vooruitschakelen (* TXA[])	X		
PLTXB	parameter SG[] (* TXB[])	X		
PLTXC	parameter EWG[] /SVG[] (* TXC[])	X		
PLTXD	parameter EVG[] /SMG[] (* TXD[])	X		
PLTXE	parameter EMG[] (* TXE[])	X		
VRI.LB	VRI-logboek		X	
VRI.LA	VRI-logboek (onb)		X	
VRI.C	VRI-commando		X	
VRI.A	Actieve storingslijst		X	
PAR.LB	Parameterlogboek	X	X	
PAR.LA	Parameterlogboek (onb).	X	X	
TELINST	Instellingen telprogramma		X	
TELDATA	Data telprogramma		X	
TELMON	Actuele data telprogramma		X	
DATAKOM	Instelling datacommunicatie	X	X	
DATAKOM.I	Index object Datacom	X	X	
AUTHOG	Gebruikersnamen		X	
AUTHOP	Toegangscodes		X	
LOGINNIVEAU	Nummer gebruikersgroep waaronder ingelogd is.	P	P	
OVDEVICE	OV-devices		X	
OVFILTER	OV-filter		X	
OV.LB	OV-logboek		X	
OV.LA	OV-logboek (onb)		X	
DIMINST.I	Index diminstellingen.		X	
DIMINST	diminstellingen.		X	
DIMMEN.I	Index dimstatus.		X	
DIMMEN.A	dimstatus.		X	
AKOESTISCH.I	Index Status akoestische signalen		X	
AKOESTISCH.A	Status akoestische signalen		X	
AKOESTISCH.F	Foutstatus akoestische signalen		X	
PAKOESTISCH	Parameter akoestische signalen		X	
PAKOESTISCH.I	Index Parameter akoestische signalen		X	
EXTRAINFO.A	Informatieve actuele extra info string.	X	X	
EXTRAINFOEXT	Informatieve toelichting extra info string.	X	X	
EXTRAINFO.I	Index extra info	X	X	
EVENTLYST.I	Eventnummers als tekststring	X	X	
EVENTLYST.INFO	Detailinformatie over het event	X	X	
ERROR.CODE	Foutcode	P	P	
ERROR.INFO	Gedetailleerde beschrijving	P	P	
ERROR.CMD	Het commando waar de error bij hoort	P	P	
NOODSTROOM	Instellingen voor de noodstroom voorziening		X	

Name	Description ¹	IVERA-APP	IVERA-TLC	Mandatory
NOODSTROOM.A	Actuele toestand noodstroom voorziening		X	
NOODSTROOM.I	Index object noodstroom voorziening		X	
NOODSTROOM.LA	Logboek (onb.) noodstroom voorziening		X	
NOODSTROOM.LB	Logboek (bev.) noodstroom voorziening		X	
ABON	Abonnementsverzoek	P	P	
BB0	Objectlijst type 0	P	P	
BB1	Objectlijst type 1	P	P	
BBA0	Objectlijst + attributen type 0	P	P	
BBA1	Objectlijst + attributen type 1	P	P	
PING	Ping-commando	P	P	
LOGIN	Login-commando	P	P	
TID	Toepassing identificatienummer	X	X	
XID	Automaat identificatienummer		X	
YID	Applicatie identificatienummer	X		
ZID	Gereserveerd			
TLC.I	TLC namen	X		X
TLC	TLC instellingen	X		X
RIS.I	RIS namen	X		
RIS	RIS instellingen	X		
ITSAPPLOC	Applicatie management referentie		X	X
ITSAPP.I	ITS applicatie gebruikersnamen		X	X
ITSAPP	ITS applicatie instellingen		X	X
ITSSTAT	Huidige status van de ITS applicatie		X	X
USER	IVERA gebruikersinstellingen	P	P	P
APPID.I	Index Identificatie van Applicatie	X		
APPID	Identificatie van Applicatie	X		X
APPIFLOC	Applicatie interface locaties	X		X
APPVER	Versies van de applicatie	X		X
APPVER.I	Index versienummers	X		X
APPFOUT	fouttoestand	X		
APPFOUT.I	Index fouttoestand	X		
APP.LA	Logboek met meldingen van ITS-Applicatie (onb)	X		
APP.LB	Logboek met meldingen van ITS-Applicatie	X		
APP.A	Actieve storingslijst	X		
X* (X-objecten)	TLC Fabrikant specifieke objecten		X	
Y* (Y-objecten)	APP specifieke objecten	X		

Tabel 1 Object allocation

4.6 Objectdefinitie

Een object zoals gedefinieerd in een IVERA-slave bestaat uit:

- unieke objectnaam

- objectattributen
- data-elementen

De objectnaam bevat een unieke naam voor het object. De objectattributen bevatten alle kenmerken van het object. De data-elementen bevatten de in het object opgeslagen data. De data-elementen in een object zijn opgeslagen als een één of meer dimensionale array. Per object is het aantal dimensies en het aantal data-elementen per dimensie instelbaar. Om praktische redenen is het aantal dimensies per object begrensd op 3.

```
Object
= Naam
+ Omschrijving
+ Type
+ UIC { UIC }
+ Logboek
+ Wijzigingsteller
+ Aantal data-elementen { Aantal data-elementen }
+ Index data-elementnaam { Index data-elementnaam }
+ minimum waarde data-element
+ maximum waarde data-element
+ Index data minimum elementwaarde
+ Index data maximum elementwaarde
+ Index data-element type
+ Data-element formaat
+ Data-element stapgrootte
+ Data-element waarde { Data element waarde }
+ Overzicht alle attributen
```

Tabel 3.6. Object definitie, zie ook hoofdstuk 25.

Alle attributen van een object hebben een unieke naam. Deze attribuutnamen zijn weergegeven in de volgende tabel. Het type geeft aan het datatype van het attribuut; getal of tekst:

Attribuut	Type	Omschrijving
N	1	Naam
O	1	Omschrijving
T	0	Type
U	0	User Identificatie Control
L	0	Logboek
W	0	Wijzigingsteller
E	0	aantal data-elementen
I	1	Index verwijzing per dimensie
MIN	0	Minimum data-elementwaarde
MAX	0	Maximum data-elementwaarde
IMIN	1	Index data-element minimumwaarde
IMAX	1	Index data-element maximumwaarde
ITYPE	1	Index data-element type
F	0	Data-element formaat
S	0	Data-element stapgrootte
A	1	Overzicht alle attributen

Tabel 3.3. Object attributen

Indien een attribuut niet is gespecificeerd, staat het de leverancier van een IVERA-automaat vrij zelf de waarde van het attribuut te bepalen. De attributen IMIN en IMAX zijn in het geheel niet gedefinieerd en dus vrij te definiëren.

*NB. Indien het aantal elementen (E) is gespecificeerd als een naam, zie naamconventie (Tabel 7.2 Constanten).
NB. De volgorde van gebruikersgroepen (U) is groep 4,3,2,1.*

In het geval van meerdere dimensies worden het aantal data elementen en de index verwijzingen weergegeven door middel van een volgnummer:

- E1=aantal data elementen dimensie 1, E2=aantal data elementen dimensie 2, etc.
- I1= index dimensie 1, I2=index dimensie 2, etc.

```

N      = ObjectNaam
O      = String
T      = 0 | 1 | 2
U      = Groep4 * 1000 + Groep3 * 100 + Groep2 * 10 + Groep1
L      = 0 | 1
W      = PosIntegerWaarde
E      = PosIntegerWaarde
I      = ObjectNaam
MIN    = IntegerWaarde
MAX    = IntegerWaarde
IMIN   = ObjectNaam
IMAX   = ObjectNaam
ITYPE  = ObjectNaam
F      = PosIntegerWaarde
S      = PosIntegerWaarde
A      = /* Overzicht alle attributen */

```

```

Groep1 = Groep
Groep2 = Groep
Groep3 = Groep
Groep4 = Groep
Groep  = 0 | 4 | 6

```

```

IntegerWaarde = ["-"] PosIntegerWaarde
PosIntegerWaarde = DecWaarde
DecWaarde = Digit { Digit }
Digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

```

```
ObjectNaam = Letter { Eletter } [ "." ] { Eletter }
Letter = "A" .. "Z" | "a" .. "z"
Eletter = Letter | Digit
```

Tabel 3.6. Attributen definitie

NB. Integer waarden zijn getallen in het bereik -2^{31} tot $+2^{31}-1$.

Naam

Ieder object heeft een unieke naam. Een naam bestaat uit een combinatie van letters, cijfers en eventueel één punt. Een naam begint altijd met een letter. Een naam kent geen onderscheid tussen hoofdletters en kleine letters. Een naam bestaat uit maximaal 16 karakters.

Referentie	Omschrijving
KLA1	Inschakeltijden regelen periode 1
D.A	Detectortoestand
D.I	Detectornamen (tekststring)
LSGE	Externe signaalgroep status (tekststring)

Tabel 3.4 Voorbeelden van objectnamen.

Algemene richtlijnen voor de naamgeving van objecten.

- Een naam bestaat uit een unieke combinatie van letters en cijfers.
- Een naam begint met een letter.
- Een punt (.) is bedoeld voor groepsindelingen of categorieën.

Door de indeling van objecten in groepen kan het zoeken van objecten die een relatie met elkaar hebben worden geautomatiseerd. Bijv. "D.*" levert alle aan detectoren gerelateerde objecten.

NB. Voor meer informatie over objectnaamgeving zie "Beheer van objecten", hoofdstuk 5.

Omschrijving

Een omschrijving van het object in leesbare tekst. De omschrijving bevat maximaal 32 karakters.

Type

Het type beschrijft het formaat van de data-elementen in het object.

Type	Omschrijving
0	Getal (32 bits integer)
1	Tekststring
2	Cluster (optie)

Tabel 3.5. Type objecten

Een object met het type 0 (getal) wil zeggen, dat de data-elementen van het object bestaan uit getallen. In het geval dat de data-elementen tekststrings bevatten is het object van het type 1 (tekststring).

Aantal data-elementen

Het actuele aantal data-elementen in een object. In het geval van een object met meerdere dimensies (bijv. een matrix) wordt het aantal data-elementen per dimensie gespecificeerd; het totale aantal data elementen van het object is dan een vermenigvuldiging van het aantal data-elementen van iedere dimensie.

Een object kan 0 tot maximaal 2^{16} (65536) data-elementen bevatten. In het geval dat een object 0 elementen bevat is het object 'leeg'. Indien een 'leeg' object wordt benaderd, zal de slave antwoorden met een foutcode ERR_EMPTY (zie tabel 3.11).

De nummering van de data-elementen begint bij 0 en loopt tot $2^{16}-1$, d.w.z. met 0 wordt het 1st data-element aangeduid en met $2^{16}-1$ het laatste data-element.

Normaliter is het aantal elementen in een object constant; dit is echter geen eis. Een voorbeeld van een object waarvan het aantal elementen varieert, is een eventbuffer. In een eventbuffer correspondeert het aantal data-elementen met het aantal events in het buffer. Het aantal elementen kan in dit geval ook 0 zijn.

Tevens is het mogelijk en in sommige gevallen zelfs wenselijk om een object met 0 elementen te definiëren. Een voorbeeld van een object dat 0 elementen kan bevatten is het object P (parameters). In het geval dat een IVERA-slave geen parameters heeft, zijn er twee keuzen; het object P niet definiëren of een object P met 0 elementen definiëren. In dit geval geniet het de voorkeur een object P met 0 elementen te definiëren, om zo te accentueren dat de IVERA-slave het object P wel ondersteunt, maar dat er geen parameters zijn.

UIC (User Identification Control)

Het UIC attribuut is een masker dat per gebruikersgroep aangeeft welke toegangsrechten een groep van gebruikers heeft op de data-elementen van een object. Hiervoor is per gebruikersgroep een bitmasker gedefinieerd dat aangeeft wat de lees-, schrijf- en executierechten van de gebruikersgroep zijn. De indeling van het UIC bitmasker komt overeen met de gebruikersrechten binnen UNIX.

Waarde	bitmask	Omschrijving
0	000	geen rechten
1	001	alleen executeren
2	010	alleen schrijven
3	011	executeren en schrijven
4	100	alleen lezen
5	101	executeren en lezen
6	110	lezen en schrijven
7	111	executeren, lezen en schrijven

Tabel 3.6. Gebruikersrechten

Daar het IVERA-protocol alleen lezen en schrijven ondersteunt, zijn alleen de volgende opties van belang:

UIC	Omschrijving
0	geen rechten
4	alleen lezen
6	lezen en schrijven

Tabel 3.7. Gebruikersrechten binnen IVERA

Voorbeeld: UIC = 6664

- Groep1 = 4 (alleen lezen)
- Groep2, groep3 en groep4 = 6 (lezen en schrijven)

NB. Hierbij is de aanname gedaan dat iemand die mag schrijven ook mag lezen.

NB. Zie paragraaf 4.8 "Gebruikers" voor een beschrijving van de gebruikersgroepen.

Index data-elementnamen

Een index bevat een Naam (zie BNF ObjectNaam) die verwijst naar een ander object. Een index biedt de mogelijkheid om de elementen van een object van een logische naam te voorzien. De logische namen zijn opgeslagen in de data-elementen van het object waar de index heen verwijst. In een object kan per dimensie een index worden opgegeven.

Een voorbeeld is de index van het object TGL (geeltijden) die verwijst naar het object SG.I (signaalgroepnamen).

Logboek

Voor ieder object is vastgelegd of wijzigingen van de data-elementen moeten worden opgeslagen in het parameterlogboek.

Wijzigingsteller (optioneel)

De wijzigingsteller van een object is bedoeld als een variabele die wordt verhoogd, als een van de data-elementen in een object wijzigt. Deze vlag zou handig kunnen zijn voor objecten met een groot aantal data-elementen die slechts sporadisch van toestand veranderen. De wijzigingsteller voorkomt dat een master die geïnteresseerd is in de wijzigingen, regelmatig alle data elementen moet lezen.

Aangezien de wijzigingsteller in de praktijk niet wordt gebruikt, is het niet voorgeschreven deze met een zinnige waarde te vullen. Wel blijft het attribuut om compatibiliteitsredenen gehandhaafd.

Minimumwaarde en maximumwaarde van data-elementen

Voor ieder object is een algemene minimum- en maximumwaarde instelbaar die geldt voor alle data-elementen. Iedere waarde die naar een data-element van het object wordt geschreven moet aan de volgende voorwaarde voldoen:

$$\text{minimumwaarde} \leq \text{waarde} \leq \text{maximumwaarde}$$

NB. De attributen minimum- en maximumwaarde zijn alleen van toepassing op objecten van het type 0 (getal).

NB. Voor een object van het type 1 (tekststring) corresponderen de minimum- en maximumwaarde met de minimale en maximale lengte van de string.

Indexminimum en -maximum van data-elementen

Voor iedere getalobject is het mogelijk een ander object te gebruiken als minimum of maximum. Dit biedt de mogelijkheid om per data-element een minimum- en maximumwaarde te specificeren. De verwijzing is een naam (zie BNF ObjectNaam) die verwijst naar een object.

Een voorbeeld is het TGGL (garantiegeeltijd) object dat als indexminimum aan het TGL (geeltijd) object is gekoppeld. Op deze manier wordt voorkomen dat een geeltijd onder de garantiegeeltijd ingesteld wordt.

NB. De attributen index minimum- en maximumwaarde zijn alleen van toepassing op objecten van het type 0 (getal).

Indextype van data-elementen

Voor ieder getalobject is het mogelijk een ander object te gebruiken als type. Dit biedt de mogelijkheid om per data-element een type te specificeren. De verwijzing is een naam (zie BNF ObjectNaam) die verwijst naar een object.

Formaat data-element

Per object is er de mogelijkheid één dataformaat te definiëren. Het dataformaat wordt aangeduid door middel van een getal. De betekenis van het dataformaat is afhankelijk van de toepassing.

Stapgrootte data-element

Per object is een stapgrootte gedefinieerd. De stapgrootte geeft aan met welke stapgrootte een waarde in een data-element van het object kan worden ingesteld. Bij het schrijven van data naar een object worden alleen die waardes geaccepteerd die een veelvoud zijn van de stapgrootte; alle tussenliggende waardes worden geweigerd.

NB. Het attribuut stapgrootte is alleen van toepassing op objecten van het type 0 (getal).

NB. Afwijkende reeksen zoals 1, 3, 5, 7 kunnen niet als attributen worden gespecificeerd, wel is het mogelijk deze beperkingen aan te brengen door deze hard in de code van het object te programmeren.

Waarde data-element

Een object kan 0 tot 2^{16} data elementen bevatten.

Overzicht alle attributen

Per object is het mogelijk om via het attribuut 'A' alle attributen in 1 keer te lezen of te wijzigen. Het attribuut 'A' is een tekst string met het volgende formaat:

```
Overzicht alle attributen = AttribuutDef { “,” AttribuutDef }
AttribuutDef = AttribuutNaam “=” AttribuutWaarde
AttribuutNaam = “N” | “O” | “U” | “L” | “W” | “E” | “E1” | “E2” | “I” | “I1” | “I2” | “MIN” | “MAX” | “IMIN” |
“IMAX” | “F” | “S” | “A” | “T”

AttribuutWaarde = IntegerWaarde | AttribuutString1 | AttribuutString2
AttribuutString1= “{ Karakter1 }”
AttribuutString2= Karakter2 { Karakter2 }
Karakter1 = “A”..”Z” | “a”..”z” | “0”..”9” | “ ” | “,” | “.”
Karakter2 = “A”..”Z” | “a”..”z” | “0”..”9” | “.”
```

Tabel 3.6. Overzicht alle attributen

Als alle attributen van een object worden opgevraagd dan dienen minimaal de volgende attributen te worden vermeld:

N	Objectnaam
T	Objecttype
F	Formaat
E	Aantal elementen (of indien meer dimensionaal dan E1 en E2).
U	Toegangsrechten

NB. De waarde van de attributen A en DATA worden uiteraard nooit vermeld.

NB. Voor voorbeelden zie tabel 3.8.

4.7 Objectsoorten

In deze paragraaf worden globaal een aantal soorten objecten en hun kenmerken beschreven. Deze beschrijving is een aanvulling op de formele objectdefinitie (zie paragraaf 4.5 “Objectdefinitie”).

Base-object

Een *base-object* is een object van het type 1. Een *base-object* bevat een lijst met namen van de in de IVERA-slave aanwezige objecten van een bepaald type. Het aantal data-elementen van een *base-object* komt overeen met het aantal objecten van een bepaald type.

De *base-objecten* “BB0” t/m “BB99” leveren de namen van de objecten van een bepaald type. Het volgnummer komt overeen met het objecttype.

De *base-objecten* “BBA0” t/m “BBA99” leveren alle attributen van de objecten van een bepaald type.

Index object

Een *indexobject* is een object van het type 1. In een *indexobject* bevatten de data-elementen logische namen voor de data-elementen in een ander object.

Een voorbeeld is het object “SG.I” (signaalgroepnamen). Dit object kan als index dienen voor andere signaalgroep georiënteerde objecten, zoals het object TGL” (geeltijden).

Commando-object

Vanuit een IVERA-master kunnen via *commando-objecten* commando’s naar een slave worden gestuurd. Het sturen van een commando is geïmplementeerd als het schrijven van een waarde naar een data-element van een object. De combinatie van object, data-element en waarde bepaalt het commando dat wordt gegeven.

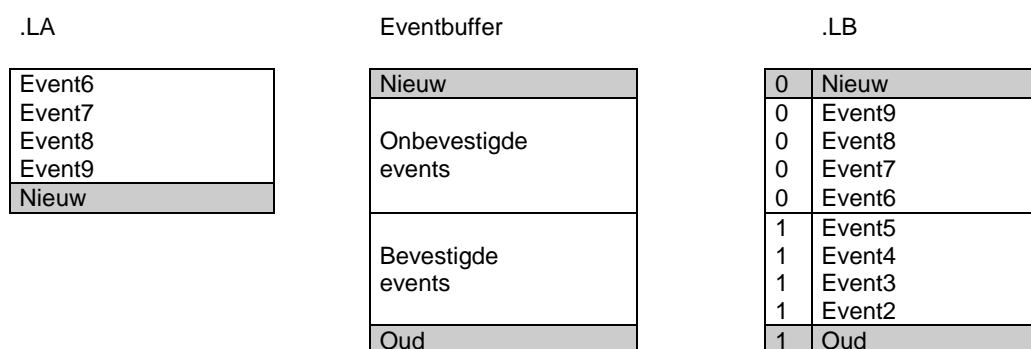
Een voorbeeld is het object “VRI.C”. Door het schrijven van een waarde naar dit object kunnen specifieke alarmen/events in een slave worden gereset.

Event-object

Een event is een gebeurtenis in de slave. Deze events worden door de slave opgeslagen in een zogenaamd *event-object*. De master kan de opgetreden events ophalen door het lezen van het bijbehorende object. Nadat de master een event heeft gelezen, kan de master het gelezen event bevestigen. Een *event-object* wordt gekenmerkt door het feit dat het aantal data-elementen niet constant is. Het aantal data-elementen komt overeen met het aantal events in het *event-object*.

In de slave wordt er onderscheid gemaakt tussen twee *event-objecten*;

- Logboek (.LB); dit object bevat de laatste n events. Waarbij element 0 overeenkomt met het nieuwste event en element $n-1$ met het oudste event. Als het eventbuffer vol is, vervalt het oudste element bij het toevoegen een nieuw event.
- Onbevestigde events (.LA); dit object bevat de nieuwste m onbevestigde events. Waarbij element 0 overeenkomt met het oudste onbevestigde event en element $m-1$ met het nieuwste onbevestigde event. De master kan een event bevestigen door het schrijven van een willekeurige waarde naar een data-element. Na bevestiging wordt het event door de slave uit het object verwijderd.



Figuur 3.1. Event-objectdefinitie

De bedoeling van een event-object (LA) is dat een event ten minste een keer aan de master wordt gemeld. Nadat de master het event heeft gelezen zal de master het event bevestigen. Indien er tussen het lezen en het bevestigen door de master een nieuw event ontstaat, dan gaat dit niet verloren, doordat alleen de gelezen events worden bevestigd.

Tevens kan een event alleen worden bevestigd, als alle voorgaande events zijn bevestigd.

4.8 Gebruikers

Vanaf IVERA versie 4 is de definitie van de gebruikers groepen gewijzigd. Een gebruikers groep Administrator (Admin) voor user account management is nodig. Hiervoor zal groep 4 worden gebruikt.

De gebruikers groepen IVERA versie 4 zijn:

1. De wereld
2. Kantonnier
3. Verkeerskundige en Technisch onderhoud
4. Gebruiker en toegangsbeheer

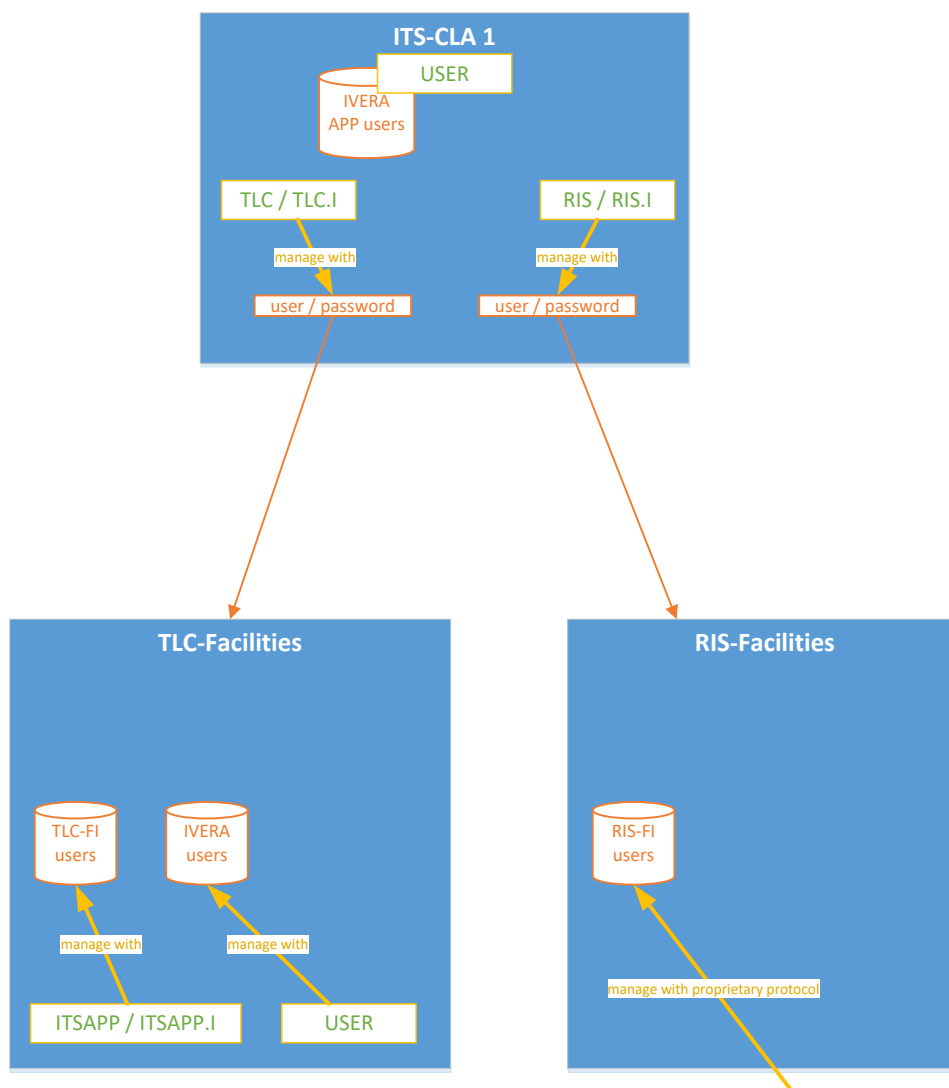
De UIC van het object DATACOM zal hiervoor worden gewijzigd van “6444” naar “6644”

In Afbeelding 3 Object relations and users, wordt de relatie tussen IVERA Objecten en gebruikers management weergegeven

De volgende gebruikers configuraties kunnen worden onderscheiden:

- ‘IVERA users’ zijn gebruikers die IVERA-TLC mogen gebruiken
- ‘TLC-FI users’ zijn gebruikers die mogen registreren bij TLC- Facilities.
- ‘RIS-FI users’ zijn gebruikers die mogen registreren bij RIS-Facilities.
- ‘IVERA APP users’ zijn gebruikers die mogen inloggen bij de IVERA-APP interface.

Alle gebruiker configuraties worden beheerd door IVERA Objecten (in groen weergegeven).



Afbeelding 3 Object relations and users

Inloggen

De gebruikersgroepen hebben een eigen wachtwoord waarmee een gebruiker kan inloggen. Het inloggen gebeurt door het schrijven naar het object "LOGIN". De wachtwoorden worden eenmalig bepaald en zijn niet instelbaar. De wachtwoorden voor het inloggen zijn klantspecifiek.

Alleen de objecten LOGIN en PING kunnen zonder inloggen worden gelezen dan wel geschreven.

4.9 Berichtdefinitie

Voor het lezen of schrijven van een object stuurt een master een bericht in leesbare tekst naar de slave. Het bericht wordt afgesloten met een return (karakter code 13 decimaal). De slave antwoordt met een bericht in leesbare tekst, afgesloten met een return.

```

Bericht = [BerichtID] (BerichtMaster | BerichtSlave) CarriageReturn
BerichtID = "@" PosIntegerWaarde "#"
BerichtMaster = ObjectRef [ "=" ArgumentLijst ]
BerichtSlave = BerichtSlaveErr | BerichtSlaveAck | BerichtSlaveAntw |
  BerichtSlaveAckHand | BerichtSlaveAntwHand | BerichtSlaveTrigger
  
```

```

BerichtSlaveErr = “:E=” SlaveErrCode
BerichtSlaveAck = “:A”
BerichtSlaveAntw = “=” ArgumentLijst
BerichtSlaveAckHand = ObjectRef “=” ArgumentLijst
BerichtSlaveAntwHand = ObjectRef “=” ArgumentLijst
BerichtSlaveTrigger = “:T=” TriggerCode

TriggerCode = PosIntegerWaarde
SlaveErrCode = “0” | “1” | “10” | “11” | “12” | “13” | “14” | “15” | “16” | “17” | “18” | “19”
CarriageReturn = Karakter code 13 decimaal

ObjectRef = ObjectNaam [“.” AttribuutNaam ] | ( [ “/” ElementBereik { “,” ElementBereik } ] )
ElementBereik = “*” | Bereik
Bereik = Element [ “-” [ Element ] ]
Element = (“#” PosIntegerWaarde) | IndexNaam

IndexNaam = Eletter { Eletter | “_” }
ArgumentLijst = Argument { “,” Argument }
Argument = IntegerWaarde | TekstString
TekstString = DubbelQuote { DetailString } DubbelQuote
DubbelQuote = “””
Komma = “,”
AsciiKarakter = Karakter uit de ASCII karakterset met waarde tussen 31 en 127 met uitzondering van
DubbelQuote (ASCII 34) en Komma (ASCII 44)
AsciiString = { AsciiKarakter }
DetailString = { AsciiString | Komma }
leeg = “”

```

Tabel 3.6. Berichtdefinitie.

NB. Het IVERA-protocol maakt geen onderscheid tussen hoofdletters en kleine letters. In zowel objectnamen als attribuutnamen mogen hoofdletters en kleine letters door elkaar gebruikt worden.

NB. Als het bericht vanuit de master een <BerichtId> bevat dient in het antwoord van de slave hetzelfde <BerichtId> te worden teruggestuurd.

NB. Het ondersteunen van een combinatie van indexnamen en indexnummers in één bericht is optioneel (zie definitie Element).

SlaveErrCode	Code	Reden	
ERR_ILLEGAL	0	Bericht voldoet niet aan het IVERA bericht formaat.	R/W
ERR_OVERFLOW	1	De slave heeft onvoldoende geheugen om het bericht te kunnen verwerken.	R/W
ERR_OBJECT	10	Object is niet gedefinieerd in de slave.	R/W
ERR_USER	11	Master heeft geen autorisatie om het object te lezen/schrijven.	R/W
ERR_RANGE	12	Het gespecificeerde element bereik is ongeldig.	R/W
ERR_INDEX	13	De slave kan een of meer opgegeven indices niet vertalen.	R/W
ERR_DIM	14	Indien niet alle dimensies zijn gespecificeerd.	W
ERR_WRANGE	15	Het aantal elementen komt niet overeen met het aantal data argumenten.	W
ERR_DATA	16	De data in de argumentlijst is ongeldig.	W
ERR_EMPTY	17	Het aantal elementen van het object is 0.	R/W
ERR_STEP	18	Waarde is geen veelvoud van de stapgrootte.	W
ERR_ATTRIB	19	Het opgegeven attribuut is ongeldig.	R/W

Tabel 3.11. IVERA-slave foutcodes.

4.10 Object Elementbereik

Een object bestaat uit 1 of meer data elementen. Een object is gedefinieerd als een meerdimensionale array van data-elementen, waarbij het aantal dimensies in theorie onbeperkt is. Voorbeelden van objecten met 2 dimensies zijn o.a. de ontruimingstijdenmatrix en plan- (c.q. programma-)afhankelijke maximumgroentijden.

In het IVERA-protocol is een bereik van data-elementen gedefinieerd als:

```
bereik van data-elementen = [ "/" ElementBereik { "," ElementBereik } ]
ElementBereik = "*" | Bereik
Bereik = Element [ "-" [ Element ] ]
Element = ("#" PosIntegerWaarde) | Naam
```

Bij een object met meer dan 1 dimensie is het bereik gedefinieerd als:

```
ElementBereikdimensie 1, ElementBereikdimensie 2, .. , ElementBereikdimensie n
```

Vertaald in x en y coördinaten komt dit overeen met:

```
y1-y2,x1-x2
```

Het bereik dient te voldoen aan de volgende algemene voorwaarden:

- Geen bereikspecificatie komt overeen met **alle** elementen.
- De nummering van de data-elementen begint bij 0.
- Een bereikdefinitie is ongeldig als:
 - Een van de elementen buiten bereik is.
 - Indien het eerste elementnummer groter is dan het tweede elementnummer.
 - Een naam niet (via een *index-object*) vertaald kan worden in elementnummer.
- Voor schrijfoperaties gelden de volgende aanvullende voorwaarden:
 - Voor iedere dimensie moet het bereik volledig gespecificeerd zijn.
 - Indien de argumentlijst 1 argument bevat, wordt dit argument naar alle elementen in het gespecificeerde bereik geschreven.
 - Indien de argumentlijst meerdere argumenten bevat, moet het aantal argumenten overeenkomen met het aantal elementen in het gespecificeerde bereik.

In de volgende paragrafen zal het specificeren van een element bereik worden toegelicht aan de hand van twee voorbeelden:

1. Geeltijden van een signaalgroep (TGL)
2. Ontruimingstijdenmatrix (TOR)

De VRI in de voorbeelden heeft 4 signaalgroepen. De namen van deze signaalgroepen zijn opgeslagen in het *index object* "SG.I".

Element	Index
0	SG01
1	SG02
2	SG03
3	SG04

Tabel 3.12. Index object SG.I

Het specificeren van het elementbereik voor een matrix wordt verklaard aan de hand van de ontruimingstijdenmatrix van een VRI. De volgende tabel bevat de ontruimingstijdenmatrix voor een VRI met 4 signaalgroepen. De waarden in de tabel corresponderen met denkbeeldige elementnummers.

	SG01	SG02	SG03	SG04
SG01	0	1	2	3
SG02	4	5	6	7
SG03	8	9	10	11
SG04	12	13	14	15

Tabel 3.14. Voorbeeld object met 2 dimensies

4.11 Data context diagrammen

4.11.1 Lezen van objecten

Vanuit een master kunnen objecten worden gelezen door het specificeren van een objectnaam en bereik van elementen. De slave antwoordt met de gewenste data, indien een geldig object wordt gelezen of met een errorbericht in het geval van een fout.

Master	Slave	Omschrijving
<ObjectRef> <ObjectRef> <ObjectRef>	<ObjectRef>=<ArgumentLijst> :E=<SlaveErrCode>	Geldig object gelezen. Fout tijdens lezen object Fout tijdens interpretatie bericht
<BerichtID><ObjectRef> <BerichtID><ObjectRef> <BerichtID><ObjectRef>	<BerichtID>=<ArgumentLijst> <BerichtID>:E=<SlaveErrCode>	Geldig object gelezen. Fout tijdens lezen object Fout tijdens interpretatie bericht

Tabel 3.16. Data-contextdiagram voor het lezen van een object

Een slave antwoordt met een errorbericht “:E=”, als het bericht vanuit de master niet aan de definitie <ObjectRef> voldoet, of als het bericht wel aan de definitie <ObjectRef> voldoet, maar de slave geen antwoord kan geven op de gestelde vraag.

Master	Slave	Omschrijving
SWD SWD/D011-D021 TOR/SG01 TOR/SG02,SG05	SWD=0,0,0,0,0,0 SWD/D011-D021=0,0,0 TOR/SG01=-1,2,3,4 TOR/SG02,SG05=15	Software detectorschakelaar alle detectoren Software detectorschakelaar van D011,D012,D021 Ontruimingstijden van SG01 Ontruimingstijd SG01 -> SG02
@1#SWD @2#SWD/D011-D021 @3#TOR/SG01 @4#TOR/SG02,SG05	@1#=0,0,0,0,0,0 @2#=0,0,0 @3#=-1,2,3,4 @4#=15	Software detectorschakelaar alle detectoren Software detectorschakelaar van D011,D012,D021 Ontruimingstijden van SG01 Ontruimingstijd SG01 -> SG02

Tabel 3.18. Voorbeelden van het lezen van objecten.

4.11.2 Schrijven van objecten

Vanuit een master kan data naar een object worden geschreven door het specificeren van een objectnaam, een bereik van elementen en een lijst met argumenten. Bij een geldig bericht antwoordt de slave met een acknowlegde (ACK) bericht naar de master. In het geval van een fout antwoordt de slave met een errorbericht.

Master	Slave	Omschrijving
<ObjectRef>=<ArgumentLijst> <ObjectRef>=<ArgumentLijst> <ObjectRef>=<ArgumentLijst>	<ObjectRef>=<ArgumentLijst> :E=<SlaveErrCode>	Geldig object geschreven Fout tijdens schrijven object Fout tijdens interpretatie bericht
<BerichtID><ObjectRef>=<ArgumentLijst> <BerichtID><ObjectRef>= <ArgumentLijst> <BerichtID><ObjectRef>= <ArgumentLijst>	<BerichtID>:A <BerichtID>:E=<SlaveErrCode>	Geldig object geschreven Fout tijdens schrijven object Fout tijdens interpretatie bericht

Tabel 3.19. Data context diagram voor het schrijven naar een object.

Een slave antwoordt met een errorbericht “:E=” als:

- het bericht niet voldoet aan de definitie <ObjectRef>=<ArgumentLijst>.
- het bereik niet volledig is gespecificeerd.
- het aantal data argumenten niet overeenkomt met het aantal elementen.
- de data ongeldig is.
- de master geen autorisatie heeft om de data te wijzigen.

Master	Slave	Omschrijving
SWD/#0=2 KLA1/#1=2500 SGE.A/SG02=4 TOR/SG01,SG02=2 PING/#0=5	SWD/#0=2 :E=16 :E=10 TOR/SG01,SG02=2 PING/#0=5	Parameterwijziging geaccepteerd. NAK_DATA, buiten bereik. Gebruiker heeft geen schrijf rechten. Parameterwijziging geaccepteerd. Test verbinding.
@1#SWD/#0=2 @2#KLA1/#1=2500 @3#SGE.A/SG02=4 @4#TOR/SG01,SG02=2 @5#PING/#0=5	@1#:A @2#:E=16 @3#:E=10 @4#:A @5#:A	Parameterwijziging geaccepteerd. NAK_DATA, buiten bereik. Gebruiker heeft geen schrijf rechten. Parameterwijziging geaccepteerd. Test verbinding.

Tabel 3.21. Voorbeelden van het schrijven naar objecten.

4.11.3 Schrijven van meerdere elementen

Door middel van een enkel bericht is het mogelijk om meerdere elementen van een object te schrijven. Dit kan op twee manieren:

- 1 waarde naar meerdere data elementen, of
- een aantal waardes naar een aantal opeenvolgende data elementen.

Master	Slave	Omschrijving
@1#SWD=1 @2#KLA1/*=1000 @3#SWD/D011-D012=1 @4#SWD/D011-D012=3 @5#SWD/D011-D012=1,2 @6#SWD/D011-D021=1,2	@1#:E=14 @2#:A @3#:A @4#:E=16 @5#:A @6#:E=15	Bereik is niet volledig gespecificeerd Alle inschakel tijden regelen periode 1 naar 10:00 SWD/D011=1 en SWD/D012=1 Data niet geldig omdat SWD maximaal 2 is SWD/D011=1, SWD/D012=2 Aantal elementen =3, aantal argumenten = 2

Tabel 3.22. Voorbeelden van het schrijven van meerdere data-elementen naar objecten.

Bij de bovenstaande tabel de volgende opmerking:

In het geval dat met 1 commando meerdere elementen worden gewijzigd, kan het gebeuren dat het schrijven naar één van de elementen wordt geweigerd vanwege ongeldige data. De slave handelt in dit geval als volgt:

- De slave antwoordt met “:E=16”, indien één van de data-elementen ongeldig is.
- Geen van de data-elementen wordt gewijzigd.
- Dit houdt concreet in, dat de slave voor het schrijven van de data eerst alle data-elementen moet controleren.

NB. Bij een schrijfcommando moet altijd het volledige bereik worden gespecificeerd om te voorkomen dat er per ongeluk verkeerde elementen worden gewijzigd. Dit is met name van belang bij het handmatig ingeven van commando's.

4.11.4 Lezen van objectattributen

Vanuit de IVERA-master kunnen de attributen van een object in de slave worden gelezen. De syntax voor het lezen van de attributen is gelijk aan die voor het lezen van het object data-elementen. Het lezen van attributen kan op de volgende manieren:

- Het lezen van 1 attribuut, of
- Het lezen van alle attributen.

Voor een beschrijving van de attributen wordt verwezen naar de paragrafen “Objecten”.

Master	Slave	Omschrijving
SWD:N	SWD:N="swd"	Objectnaam
SWD:T	SWD:T=0	Objecttype
SWD:E	SWD:E=6	4 detectoren
SWD:U	SWD:U=6664	User Identificatie Control
SWD:L	SWD:L=1	Logboek aan
SWD:I	SWD:I="D.I"	Index object(s)
SWD:W	SWD:W=1	Wijzigingsteller
SWD:MIN	SWD:MIN=0	Minimumwaarde
SWD:MAX	SWD:MAX=2	Maximumwaarde
SWD:IMIN	SWD:IMIN=""	Index minimum
SWD:IMAX	SWD:IMAX=""	Index maximum
SWD:S	SWD:S=1	Stapgrootte
SWD:A	SWD:A="/* string met attributen */"	Uitlezen alle attributen
TOR:E	TOR:E=4,4	matrix van 4 bij 4.
TOR:I	TOR:I="SG.I","SG.I"	
TOR:IMIN	TOR:IMIN="TGOR"	

Tabel 3.23. Voorbeelden van het lezen van objectattributen.

NB. Voor de definitie van de string met attributen wordt verwezen naar paragraaf 4.7 "Objectsoorten".

4.11.5 Wijziging van objectattributen (optioneel)

Vanuit de IVERA-master kunnen de attributen van een object in de slave worden gewijzigd. De syntax voor het wijzigen van de attributen is gelijk aan die voor het schrijven van het object data-elementen. Het wijzigen van attributen is alleen mogelijk met gebruikersgroep 4 privileges. Het schrijven van attributen kan op de volgende manieren:

- Het schrijven van 1 attribuut, of
- Het schrijven van meerdere attributen.

Master	Slave	Omschrijving
@1#SWD:L=1	@1#:A	Logboek aan
@2#SWD:A="L=1,MAX=2"	@2#:A	Stapgrootte wordt niet ondersteund door tekstobjecten.
D.I:S=1	:E=19	

Tabel 3.24. Voorbeelden van het schrijven van objectattributen.

NB. Indien een attribuut niet wordt ondersteund, antwoordt de IVERA-slave met `ERR_ATTRIB`.

NB. Voor de definitie van de string met attributen wordt verwezen naar paragraaf 4.7 "Objectsoorten".

4.11.6 Master-slave synchronisatie

Bij een handmatige invoer van IVERA-commando's wordt er geen gebruik gemaakt van berichtnummers.

In het geval van communicatie tussen twee computers, bijvoorbeeld de communicatie tussen een beheerscentrale en een verkeersregelinstallatie, wordt gebruik gemaakt van berichtnummers. De master genereert de berichtnummers, die door slave transparant worden terugzonden. De master vergelijkt het antwoord van de slave met het eerste bericht in het verzendbuffer; indien deze berichten met elkaar overeenkomen, wordt het bericht uit het verzendbuffer verwijderd en wordt het antwoord doorgegeven naar de applicatie. In het geval dat het ontvangen bericht niet overeen komt met het eerste bericht in het verzendbuffer, wordt het binnenkomende bericht genegeerd en zal de master de verbinding opnieuw synchroniseren.

In het geval dat de slave antwoordt met een errorbericht `<ERR_ILLEGAL>`, dat wil zeggen de slave heeft een ongeldig bericht ontvangen, zal de master de verbinding tussen de master en slave opnieuw synchroniseren.

Indien de slave antwoordt met een errorbericht (anders dan `ERR_ILLEGAL`), wordt het bericht uit het verzendbuffer gehaald en de foutcode wordt doorgegeven aan de applicatie.

Time-out

In uitzonderlijke gevallen kan het voorkomen dat de master geen antwoord ontvangt, terwijl de onderliggende netwerklagen aangeven dat de verbinding tussen de master en slave goed is. Voor het onderkennen van deze situatie heeft de IVERA-master een timeout-timer. Na het verstrijken van de time-out zal de master de verbinding tussen de master en slave opnieuw synchroniseren.

Verbinding verbroken

Bij het wegvallen van de verbinding tussen master en slave, zal de onderliggende netwerklaag dit melden aan het IVERA-protocol. Bij het wegvallen van de verbinding zal het IVERA de lopende actie afbreken. Dit houdt in:

- de master verstuurt geen berichten meer totdat de verbinding opnieuw is opgebouwd.
- de slave stopt met versturen van een mogelijk antwoordbericht.

PING

In iedere IVERA-slave is een object "PING" aanwezig. Dit object heeft geen betekenis voor de slave, maar kan door de master gebruikt worden voor het synchroniseren van de master en slave. Tevens biedt PING de mogelijkheid tot het meten van de tijd dat een bericht onderweg is van master naar slave en terug naar de master.

Voor het synchroniseren van de master en slave schrijft de master een getal naar het object.

4.11.7 Gebeurtenis in de slave (trigger)

Een slave kan de master informeren dat een event heeft plaatsgevonden, door het versturen van het bericht <BerichtSlaveTrigger>. De TriggerCode in het bericht is een applicatie specifieke code waarin kan worden gespecificeerd welk type event heeft plaatsgevonden.

De slave zal eerst een verbinding met de master opbouwen en vervolgens het bericht versturen. Het is vervolgens aan de master om de slave te ondervragen en de verbinding te verbreken. In het geval dat de master niet binnen een tijd van 5 minuten inlogt in de slave (via login op IVERA niveau) zal de slave automatisch de verbinding verbreken. Vanaf het moment dat door de master is ingelogd, geldt de normale login time-out.

N.B. Het <BerichtSlaveTrigger> is een uitzondering op het master-slave principe in die zin dat het bericht autonoom door de slave naar de master kan worden gestuurd.

5. Beheer van objecten

De doelstelling van het IVERA-protocol is een fabrikantonafhankelijke oplossing voor de communicatie tussen een beheerscentrale en een VRI. De hiervoor gekozen oplossing is een eenvoudig protocol voor het lezen en schrijven van objecten. Echter het protocol op zich biedt geen enkele functionaliteit. De werkelijke functionaliteit ligt opgeslagen in de objecten.

Bij het definiëren van de objecten van een VRI spelen de volgende aspecten een rol:

- De functionaliteit is nog niet uitgekristalliseerd, waardoor er in de toekomst objecten zullen vervallen en andere zullen worden toegevoegd.
- Naast de standaard (basis) functionaliteit is er de mogelijkheid om per type automatisch specifieke objecten te definiëren.
- Naast de standaardfunctionaliteit bestaat er de mogelijkheid om voor iedere (verkeerskundige-) applicatie specifieke objecten te definiëren.
- Welke objecten zijn in een VRI beschikbaar, dat wil zeggen; welke functionaliteit is er aanwezig in een VRI?
- De doelstelling is een zo eenvoudig, robuust en toekomst vast mogelijk protocol.

Vanwege bovenstaande redenen is er gekozen voor een identificatie van objecten op basis van een unieke naam.

Voor de naamgeving gelden de volgende randvoorwaarden:

- Een eenmaal uitgegeven naam mag worden verwijderd, maar mag **nooit** worden hergebruikt.
- De namen zijn onderverdeeld in 4 categorieën:
 - Algemene objecten per toepassing (A t/m W)
 - Automaat specifieke objecten (X...).
 - Applicatiespecifieke objecten (Y...)
 - Gereserveerd (Z...)
- De algemene objecten per toepassing worden beheerd door een daarvoor aangewezen instantie die regelmatig vergadert en nieuwe namen uitgeeft.
- Bij het vrijgeven van nieuwe namen wordt een document meegeleverd waarin op formele wijze de volledige functionaliteit van het nieuwe object wordt beschreven.
- Het wijzigen van de functionaliteit van een object mag alleen plaatsvinden onder de voorwaarde van “upwards compatible”. Indien dit niet mogelijk is, dient een nieuw object te worden gedefinieerd met de gewenste functionaliteit.

5.1 Homoniemen

Tussen de verschillende toepassingen van het IVERA-protocol kunnen homoniemen ontstaan, d.w.z. een objectnaam komt in meerdere toepassingen voor en de functionaliteit van het object kan per toepassing verschillen. Om objecten uniek te identificeren is er een unieke toepassing identificatie (TID) die via het protocol uit de IVERA-slave kan worden gelezen.

Per toepassing worden de namen van automaat- en applicatiespecifieke objecten (alle objecten die beginnen met X, Y en Z) door de leverancier bepaald, hierdoor kunnen ook per toepassing homoniemen ontstaan. De functionaliteit van de objecten die beginnen met een X, Y of Z wordt bepaald door de naam in combinatie met een unieke identificatie (XID, YID, ZID). Deze identificaties zijn als objecten in de IVERA-slave opgeslagen.

Een voorbeeld van een homoniem is het object STATUS. Per toepassing bevat dit object de status van het aangesloten apparaat. Echter per toepassing kan betekenis van de status verschillen.

5.2 Algemene objecten

Binnen het IVERA-protocol is een aantal objecten gereserveerd. Deze objecten hebben een speciale betekenis binnen het protocol. Zie **Error! Reference source not found.** voor een overzicht van deze objecten.

5.3 Toepassingspecifieke objecten

Alle objecten die beginnen met de letters A t/m W zijn toepassings specifiek. Dwz. de functionaliteit van deze objecten is eenduidig vastgelegd in combinatie met een unieke toepassingidentificatie (TID).

Voor de toepassingidentificatie (TID) zijn de volgende reserveringen gemaakt:

Bereik	Toepassing
1..9999	Verkeersregelininstallaties

Tabel 4.1. TID reserveringen

5.4 Automaatspecifieke objecten

Voor de automaatspecifieke objecten is de beginletter 'X' gereserveerd. De namen van de automaatspecifieke objecten worden door de leverancier van de automaat bepaald.

De functie van een automaatspecifiek object is eenduidig vastgelegd in combinatie met een unieke automaatidentificatie (XID).

6. Beveiliging en gebruikers

Deze paragraaf beschrijft de aanvulling op de IVERA-specificatie voor de beveiliging van een IVERA-toestel tegen inbreuk door derden.

6.1 Gebruikers

Het regeltoestel ondersteunt ten minste vier gebruikers. Hoewel het mogelijk is per regeltoestel meerdere gebruikers te definiëren, is het vanuit beheersoogpunt gewenst dit aantal laag te houden.

De opdrachtgever geeft bij opdracht de gebruikersnamen en passwords door aan de fabrikant. Indien niet opgegeven worden de gebruikersnamen en passwords door de fabrikant bepaald en bij levering schriftelijk aan de opdrachtgever gemeld.

6.2 Lokale wijziging van gebruikersnaam en password

De gebruikersnamen en passwords kunnen lokaal via een instelapparaat van het regeltoestel worden gewijzigd.

6.3 Wijziging gebruikersnaam en password

De gebruikers en bijbehorende passwords zijn gedefinieerd in de IVERA objecten:

- AUTHOG : Gebruikersnamen
- AUTHOP : Passwords

Het aantal elementen van deze objecten bepaalt het aantal gedefinieerde gebruikers.

Een lijst met gebruikersnamen kan worden opgevraagd door het lezen van het object AUTHOG. Door het schrijven naar het object AUTHOG kan een gebruikersnaam worden gewijzigd.

Het object AUTHOP levert bij lezen een lege string, door schrijven kan het password per gebruiker worden gewijzigd. Via het IVERA-protocol kunnen dus op afstand zowel gebruikersnaam als passwords worden gewijzigd.

Let wel: Iedere gebruiker die is ingelogd, kan deze actie uitvoeren. Om te voorkomen dat iemand zomaar een gebruikersnaam of password wijzigt, zal ook het huidige password moeten worden opgegeven.

Het wijzigen van een gebruikersnaam gebeurt door het schrijven naar het bijbehorende element van het object AUTHOG.

```
AUTHOG/<element>="<gebruikersnaam>,<password>,<nieuw1>,<nieuw2>"
```

waarbij:

<element>	: object element (c.q. gebruiker)
<gebruikersnaam>	: huidige gebruikersnaam voor deze gebruiker.
<password>	: huidige password voor deze gebruiker of password van 1 ^{ste} gebruiker.
<nieuw1>	: nieuwe gebruikersnaam voor deze gebruiker.
<nieuw2>	: moet overeenkomen met <nieuw1>.

Het wijzigen van een password gebeurt door het schrijven naar het bijbehorende element van het object AUTHOP.

```
AUTHOP/<element>="<gebruikersnaam>,<password>,<nieuw1>,<nieuw2>"
```

waarbij:

<element>	: object element (c.q. gebruiker)
<gebruikersnaam>	: huidige gebruikersnaam voor deze gebruiker.
<password>	: huidige password voor deze gebruiker of password van 1 ^{ste} gebruiker.
<nieuw1>	: nieuw password voor deze gebruiker.
<nieuw2>	: moet overeenkomen met <nieuw1>.

□ Door deze opzet heeft de 1^{ste} gebruiker meer rechten. Namelijk de 1^{ste} gebruiker kan de gebruikersnamen en passwords van alle gebruikers wijzigen. De overige gebruikers kunnen alleen hun eigen gebruikersnaam en password wijzigen.

□ Als alternatief voor het wijzigen van gebruikersnamen en toegangscode is een fabrikant specifieke methode toegestaan mits deze gebruik maakt van standaard FTP functionaliteit, d.w.z. het laden van een file in de slave. In dit geval dient de slave te beschikken over FTP server functionaliteit.

NB: De gebruikersnaam en password lengte zijn op MAX_FLEN gemaximaliseerd.

6.4 Beheersaspecten

Voor de beheerder heeft de IVERA beveiliging de volgende consequenties.

- Per nieuwe automaat de gebruikersnamen en passwords opgeven aan de fabrikant.
- Het beheren/wijzigen van de gebruikersnamen en passwords.

Gebruikersnamen

Voor de beheerder zijn er meerdere gebruikersnamen en passwords om toegang te verkrijgen tot het regeltoestel. De gebruikersnamen en passwords kunnen zowel lokaal als vanuit de centrale worden gewijzigd. Door de definitie van meerdere gebruikers kunnen verschillende partijen (zonder elkaars password te kennen) toegang verkrijgen tot het regeltoestel.

Voor de keuze van de gebruikersnamen zijn er meerdere mogelijkheden, maar er is een sterke voorkeur de gebruikersnamen eenvoudig te houden; twee mogelijkheden:

1. De gebruikersnaam voor alle regeltoestellen in het beheersgebied is identiek, bijvoorbeeld de naam van de beheersorganisatie.
2. De gebruikersnaam is uniek per regeltoestel, bijvoorbeeld een vast voorvoegsel gevolgd door de identificatie van het regeltoestel.

Andere mogelijkheden zoals volledig willekeurige namen per regeltoestellen zijn uiteraard mogelijk maar worden vanuit beheersoogpunt afgeraden.

Inbellen vanuit de IVERA centrale

De gebruikersnaam en het password van ieder regeltoestel zijn geconfigureerd in een IVERA-centrale. Een gebruiker die is ingelogd in de IVERA-centrale, heeft zodoende direct toegang tot alle regeltoestellen in het beheersgebied zonder gebruikersnaam en password te hoeven opgeven.

□ In principe zijn de gebruikersnamen en passwords van de regeltoestellen alleen bekend bij de 'beheerder'. Een gemiddelde gebruiker van een IVERA-centrale heeft deze kennis niet.

Het wijzigen/beheren van de passwords

De gebruikersnamen en passwords worden per regeltoestel eenmalig geconfigureerd en kunnen tijdens de levensduur worden gewijzigd. Daar de gebruikersnamen relatief eenvoudig zijn, mag worden aangenomen dat de gebruikersnamen na enige tijd bekend zijn. De beveiliging is daarom voor 99% afhankelijk van het password. Het password van alle regeltoestellen zal regelmatig moeten worden gewijzigd. Het password zal voldoende ingewikkeld moeten zijn. De redenen om het password van een gebruiker of alle passwords te wijzigen zijn:

- Periodiek (1x per jaar).
- Na een inbreuk in het systeem van buitenaf.
- Bij een probleem in de beheersorganisatie waardoor de passwords buiten de organisatie bekend zijn geworden.
- Het feit dat iemand van buiten de organisatie “tijdelijk” toegang heeft gekregen tot het regeltoestel via een daarvoor vrijgegeven gebruikersnaam en password en deze toegang niet langer wenselijk is.

Voor de passwords zijn er meerdere mogelijkheden:

1. Één password voor alle regeltoestellen.
2. Een uniek password per regeltoestel.
3. Een uniek password per gebruiker per regeltoestel.

Methode 2 is in principe veiliger, omdat iemand van buitenaf die toegang tot 1 regeltoestel heeft verkregen niet direct toegang heeft tot alle regeltoestellen. Bij methode 1 is het van belang het password periodiek te wijzigen.

N.B. De gebruikersnamen en passwords die door de fabrikant eenmalig worden (voor)geprogrammeerd worden door de fabrikant niet geadministreerd. Als zodanig kan er op de fabrikant geen beroep worden gedaan, indien het password is vergeten/kwijtgeraakt. Wel kan de fabrikant de service verlenen om de gebruikersnamen en passwords (lokaal) opnieuw in te stellen.

De mate van beveiliging

Beveiliging is altijd een samenspel van een technische realisatie en de verantwoordelijke organisatie.

In eerste instantie is de IVERA-beveiliging bedoeld tegen inbreuk in het regeltoestel van buitenaf. Deze beveiliging is noodzakelijk omdat door het gebruik van gestandaardiseerde ‘open’ protocol iedereen met een internet-PC met het regeltoestel kan communiceren. De CHAP implementatie biedt de beveiliging door het gebruik van gebruikersnamen en passwords.

In tweede instantie is de IVERA-beveiliging bedoeld als bescherming binnen de eigen organisatie. Dit is mogelijk door de passwords alleen bekend te maken aan een beperkt aantal mensen binnen de organisatie die met de regeltoestellen moeten werken. Tevens is het mogelijk om aan verschillende partijen verschillende gebruikersnamen en passwords toe te kennen. In principe moet bij een wijziging in de samenstelling van de groep de passwords worden gewijzigd.

Alles tezamen biedt de CHAP implementatie een afdoende beveiliging die past bij de toepassing en de beheersorganisatie. Belangrijk is wel de beveiligingsgegevens binnen een zo select mogelijk gezelschap te houden en regelmatig de passwords te wijzigen.

“Tijdelijke” toegang tot het regeltoestel kan worden verschaft aan derden buiten de eigen organisatie door hiervoor een gebruikersnaam en password vrij te geven. Op het moment dat deze toegang niet meer wenselijk is kan de 1^{ste} gebruiker de gebruikersnaam en/of password

wijzigen en daarmee de toegang blokkeren. Zodoende is het zaak dat het password van de 1^{ste} gebruiker te allen tijde binnen de eigen organisatie blijft.

7. Functionele omschrijving

7.1 Functionele modificatie van parameters (FMP)

Bij FMP wordt bedoeld het vanuit de centrale op een functionele wijze wijzigen van instellingen in de VRI. De gebruiker krijgt op het scherm een overzicht van de beschikbare parameters en hun actuele instelling. Deze parameters zijn overzichtelijk gerangschikt zodat de gebruiker in staat is zonder detailkennis van de installatie een parameter te wijzigen.

Deze eenvoudige aanname heeft echter grote consequenties voor de interactie tussen de centrale en de VRI. Een aantal van de complicaties worden hier kort weergegeven.

- Met de komst van RWSC en CCOL regelingen is het aantal parameters explosief toegenomen.
- Niet iedere installatie heeft dezelfde parameters.
- Per installatie kunnen de minimum- en maximumwaarde van parameters variëren.
- Gekoppelde instellingen. Een voorbeeld hiervan is de geeltijd die niet lager ingesteld mag worden dan de garantie geeltijd van de desbetreffende signaalgroep.
- Voor FMP is een getal als index niet voldoende, maar is een functionele naam per element noodzakelijk.
- Iedere parameter moet uniek zijn.
- Beveiliging, wie mag welke parameters lezen en/of wijzigen.

Al deze aspecten zijn geregeld binnen het IVERA protocol. Waardoor voor de communicatie tussen een centrale en een VRI, FMP is teruggebracht tot het lezen en schrijven van objecten.

Het standaardiseren van namen van verschillende parameters is niet zozeer een IVERA protocol aanpassing, als wel het maken van een extra restrictie op het gebruik van namen in een aantal indexobjecten. Het gaat hierbij over de indexobjecten, die gekoppeld zijn aan de parameters in CCOL of RWS-C, die reeds een naam hebben zoals de volgende indexobjecten:

Index object	Omschrijving
SG.I	Signaalgroep namen
D.I	Detector namen
U.I	Uitgangen namen
I.I	Ingangen namen

Tabel 3.14. Index objecten met omschrijving

De namen in deze objecten moeten exact overeenkomen met de namen die in CCOL of RWS-C zijn geconfigureerd. Zijn bepaalde namen leeg gelaten, dan wordt in IVERA een invulling aangegeven, die overeenkomt met het elementnummer en het type parameter.

De 15^e lege uitgangnaam wordt in IVERA ingevuld met 'U_14'.

De 24^e lege ingangnaam wordt ingevuld met 'I_23'.

7.2 Monitoring

Middels monitoring is het mogelijk op de centrale mee te kijken met de actuele toestand van de VRI. De bekendste vorm van monitoring is het kijken naar de actuele toestand van signaalgroepen en detectoren.

Binnen het IVERA protocol komt monitoring overeen met het lezen van objecten. Een aantal objecten is speciaal gedefinieerd voor monitoring door een zo compact mogelijk dataformaat. Voorbeelden van deze objecten zijn: LSGE, LI en LD.

Door het toevoegen van een timestamp in de objecten is het mogelijk om de informatie op de centrale op te slaan en eventueel later vertraagd of versneld af te spelen. Tevens kunnen hiermee eventuele vertragingen in de communicatie worden gecompenseerd.

Belangrijk bij monitoring is dat de gelezen data consistent is. De consistentie is in de definitie van het IVERA afgedekt. Een voorbeeld van niet consistente data is een bericht waarin twee signaalgroepen tegelijk groen zijn, die in werkelijkheid een conflict met elkaar hebben. Dit zou kunnen ontstaan als de informatie niet op precies hetzelfde moment uit de VRI wordt gelezen.

7.3 Data verzameling

Er is sprake van data verzameling als de VRI informatie verzamelt en tijdelijk opslaat. De beheerscentrale kan op een later tijdstip de informatie uitlezen. Voorbeelden van dataverzameling zijn o.a.

- Telprogramma's
- Logging van events.

Binnen het IVERA protocol kan de centrale de verzamelde data uitlezen door het lezen van objecten. Het wissen van data gebeurt door het geven van opdrachten aan de VRI door het schrijven naar objecten. Het wijzigen van instellingen van een dataverzamelmodule gebeurt op dezelfde manier als het wijzigen van parameters, door het schrijven naar objecten.

7.4 Toevoegen extra elementen

In de VRI zijn er verschillende objecten waarvan een aantal elementen reeds is gedefinieerd. Naast deze elementen is het mogelijk om per installatie extra elementen toe te voegen. Voorbeelden van deze objecten zijn o.a. TP (tijdinstellingen procesbesturing) en SP (schakelaars procesbesturing).

Voor het toevoegen van extra elementen gelden de volgende voorwaarden:

- De functionele namen mogen niet conflicteren met namen van reeds gedefinieerde elementen.
- De toegevoegde namen moeten beginnen met een X.
- Reeds gedefinieerde elementen mogen niet worden gebruikt voor een andere functie.
- De beheerscentrale (IVERA-master) mag deze objecten alleen benaderen via de functionele namen, omdat de elementnummers kunnen wijzigen.

7.5 Naamconventie objecten

Voor de definitie en de naamgeving van objecten voor een verkeersregelininstallatie wordt een eenvoudige conventie gebruikt.

De beginletters X,Y en Z zijn gereserveerd voor specifieke objecten en mogen niet voor andere doeleinden worden gebruikt.

Naam	Omschrijving	Voorbeeld	Omschrijving	Samenvoeging van
KL*	Klok parameter	KLA1	Inschakelen regelen periode 1	L , SG en E
L*	Lijndump	LSGE	Lijndump externe SG toestanden	
VRI*	Algemeen VRI object	VRIID	Automaat identificatie	
WK*	Weeknummer instelling	WKZB	Weeknummer begin zomertijd	
X*	Automaat specifiek	XID	Automaat identificatie nummer	
D	Detector	D.A	Actuele detector toestand	D en .A
SG	Signaalgroep	SGE.A	Actuele SG toestand	SG en .A
SW	Swico input commando	SWD	Detector SWICO	SW en D
U	(Overige) uitgangen	U.A	Actuele uitgangstoestand	U en .A
*.I	Index object	SG.I	Signaalgroepnamen	SG en .I
*.A	Actuele toestand	TP.A	Procesbesturing lopende tijd	T, P en .A
*.LA	Logboek (onbevestigd)	VRI.LA	VRI logboek (onbevestigd)	VRI en .LA
*.LB	Logboek (alle)	VRI.LB	VRI logboek (alle)	VRI en .LB
*P	Procesbesturing	SP	Schakelaar in procesbesturing	S en P

Tabel 7.1 Naamconventie

In de tabel met objecten worden voor het opgeven van het aantal data elementen de volgende constanten gebruikt:

Constante	Omschrijving
NUMSG	Aantal signaalgroepen
NUMD	Aantal detectie-ingangen
NUMSP	Aantal schakelaars (procesbesturing)
NUMTP	Aantal timers (procesbesturing)
NUMU	Aantal uitgangen (proces + applicatie)
NUMI	Aantal ingangen (proces + applicatie)
NUMDC	Aantal classificatiedetectors
NUMPROG	Aantal in het toestel aanwezige programma's.
NUMPROGEXT	Lengte van de uitgebreide programmalijs
NUMKLOKPER	Aantal klokperiodes
NUMKLOKPROG	Aantal klokgestuurde programmakeuzes
NUMDAG	Aantal dagen van de week (==14)
NUMBIJZDAG	Aantal bijzondere dagen
NUMOVINST	Aantal instellingen openbaar vervoer
NUMOVFILTER	Aantal filterregels t.b.v. openbaar vervoer
NUMLAMP	Aantal lampen
NUMTELDATA	Aantal telcircuits
NUMEVENT	Aantal events
NUMINFO	Aantal automaat info regels
NUMERR	Aantal fouten
NUMABON	Maximum aantal abonnementen
MAX_FLEN	Maximale lengte van functionele elementnamen.

Tabel 7.2 Constanten

De constante MAX_FLEN (=32) bepaalt de maximale lengte van functionele namen. Langere strings worden afgekapt op 32, er volgt geen foutmelding.

8. Dataformaten

8.1 Object type 0

Een object van type 0 is toepasbaar voor alle informatie die middels een getal kan worden weergegeven. Om de informatie toegankelijk te maken zijn er een aantal data formaten voor getal objecten gedefinieerd. Het data formaat geldt voor alle data elementen van een object.

Formaat	Omschrijving	Weergave	Waarde
1	Getal met eenheid 1		
2	Getal met eenheid 0.1		
3	Getal met eenheid 0.01		
10	Schakelaar (uit/aan)		0=uit, 1=aan
11	Software Input Commando		0=neutraal, 1=uit, 2=aan
20	Klok (uur/minuut)	HH:MM	uur * 100 + minuut
21	Klok (uur/minuut/seconde)	HH:MM:SS	uur * 10000 + minuut * 100 + seconde
22	Datum (jaar/maand/dag)	JJJJ:MM:DD	jaartal * 10000 + maand * 100 + dag
23	Datum (maand/dag)	MM:DD	maand * 100 + dag
24	Dag van de week		
25	Jaartal		0=elk jaar, >999 = jaartal
26	Weeknummer		0=elke week, 1-53=weeknummer
30	Signaalgroep toestand extern		
40	Detector toestand		
50	VRI status		
60	Fout status		
100	Element type		

Tabel 8.1 Data formaten object type 0
NB. Jaar is 4 cijfers.

Dag van de week (24)

Waarde	Omschrijving
1	maandag
2	dinsdag
3	woensdag
4	donderdag
5	vrijdag
6	zaterdag
7	zondag
8	elke dag
9	werkdagen
10	weekeinde
11	feestdag
12	koopavond
13	koopzondag
14	speciaal

Tabel 8.2 Dag van de week

Signaalgroep toestand extern (30)

Waarde	Karakter	Signaalgroep toestand
0	R	Rood
1	G	Groen
2	A	Geel
3	W	Wit knipperen (OV)
4	O	Gedoofd
5	F	Geel knipperen
6	6	gereserveerd
7	7	gereserveerd

Tabel 8.3 Signaalgroep toestand extern

8.2 Object type 1

Een object van type 1 is toepasbaar voor alle informatie die door middel van een ASCII-string kan worden weergegeven. Om de informatie toegankelijk te maken zijn er een aantal dataformaten voor objecten gedefinieerd. Alle objecten van een bepaald type hebben dezelfde kenmerken.

Formaat	Omschrijving
1	Ruwe tekst.
2	Indexnaam
10	Instellen gebruikersnaam/toegangscode
20	Kalendertijd
21	Klokperiode
22	Klokperiode programmakeuze
30	Toestand van alle externe signaalgroep toestanden.
40	Toestand van alle detectoren.
41	Toestand van een classificatiedetector
50	Toestand van overige ingangen of uitgangen.
60	Programmalijs uitbreid
70	Lijst met bijzondere dagen
80	Openbaar Vervoer devices
81	Openbaar Vervoer detectorfilter.
82	OV event
100	Event
101	Parameterevent
110	Lampconfiguratie
111	Lampstatus
200	Instellingen telprogramma
201	Data telprogramma
202	Actuele data telprogramma
300	Abonnement

Tabel 8.4 Data formaten objecttype

9. Management Interface TLC Faciliteiten

De afbeelding hieronder beschrijft de manier waarop een IVERA master het accesspoint van de management interface van een ITS_CLA kan bepalen.

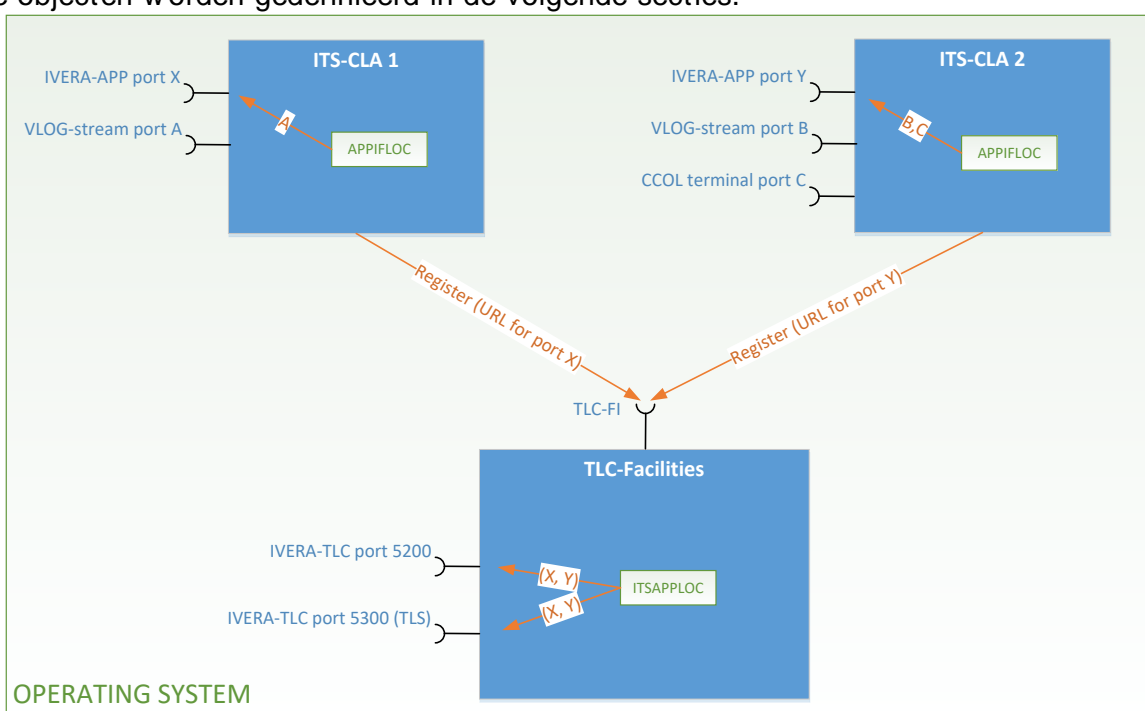
Aangezien de IVERA-TLC van de TLC-Faciliteiten beschikbaar is op een bekend address/port kan dit door een IVERA master gebruikt worden om de inhoud op te vragen van het IVERA Object 'ITSAPPLOC'.

Dit object bevat per ITS-CLA een URL, welke verwijst naar de management interface van de ITS-CLA (de URL wordt verzonden bij registratie van ITS-CLA naar TLC-Faciliteiten).

Een IVERA-master kan deze URL gebruiken om verbinding te maken tot de management interface.

Als deze management interface IVERA-APP is, dan kan de master het IVERA-object 'APPIFLOC' opvragen. Dit object bevat een lijst URL welke verwijst naar alle andere interfaces van de specifieke ITS-CLA.

De objecten worden gedefinieerd in de volgende secties.



Afbeelding 4 resolving connection points of ITS-A

9.1 Object ITSAPPLOC

Het object ITSAPPLOC bevat de URI welke definieert waar de ITS Application management interface te bereiken is.

Attribuut	Type	Object	Omschrijving
N	1	ITSAPPLOC	Naam
O	1	URI van een ITS-Application zoals gegeven bij ITS-A Registratie met de TLC-Facilities	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
L	0		Logboek
W	0		Wijzigingsteller
E	0	NUMITSAPP	aantal data-elementen
I	1	ITSAPP.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
IMIN	1		Index data-element minimumwaarde
IMAX	1		Index data-element maximumwaarde
ITYPE	1		Index data-element type
F	0	401	Data-element formaat
S	0		Data-element stapgrootte
A	1		Overzicht alle attributen

Tabel 5 Object attributen ITSAPPLOC

Dit object bevat toegangs data voor de ITS applicaties welke actieve toegang hebben tot deze TLC.

Een ITS Applicatie is gedefinieerd per “programmanummer” en wordt eenmaal getoond.

Formaat: Resource identifier (401), type 1

ResourceIdentifier = uri + “,” + [omschrijving]

uri = AsciiString conform **Error! Reference source not found.**

omschrijving = AsciiString

Usage: Een element lezen

ITSAPPLOC/< element> = ”< uri> ,[omschrijving]”

< element> : objectelement (index nummer)

< uri> : uniform resource identification (het is aanbevolen het poortnummer in de uri op te nemen)

< omschrijving> : bevat additionele informatie

Als toevoeging op standaard schema's (zoals as http, https), zijn de volgende URI schemes gedefinieerd voor IVERA:

URI scheme	Description
ivera-app	Identifies ivera-app protocol
ivera-apps	Identifies ivera-app protocol using TLS
ivera-tlc	Identifies ivera-tlc protocol
ivera-tlcs	Identifies ivera-tlc protocol using TLS
tlc-fi	Identifies tlc-fi protocol
tlc-fis	Identifies tlc-fis protocol using TLS
ris-fi	Identifies ris-fi protocol
ris-fis	Identifies ris-fis protocol using TLS
Vlog	Identifies v-log protocol
Vlogs	Identifies v-log protocol using TLS
Ccol	Identifies ccol parser protocol
Ccols	Identifies ccol parser protocol using TLS

Voorbeeld resource identifier elements:

ivera-apps://10.10.39.40:5100,Regelapplicatie spits

http://10.10.40.10,OV prioriteitsappl

https://10.10.41.19,Snelheidsadviesapplicatie

Voorbeelden:

Een element lezen:

ITSAPPLOC/#1

ITSAPPLOC/#1= "ivera-apps://10.10.39.40:5100,Regelapplicatie spits"

Meerdere elementen lezen:

ITSAPPLOC

ITSAPPLOC= "ivera-apps://10.10.39.40:5100,Regelapplicatie spits", " http://10.10.40.10,OV prioriteitsappl"

Een leeg element lezen:

ITSAPPLOC/#3

ITSAPPLOC/#3= ""

9.1.1 Object ITSAPP.I

Het IVERA Object ITSAPP.I geeft de gebruikersnamen voor de geconfigureerde ITS applicaties. De gebruikersnamen kunnen worden gewijzigd door het schrijven van ITSAPP.

Attribuut	Type	Object	Omschrijving
N	1	ITSAPP.I	Naam
O	1	ITS applicatie user management	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMITSAPP	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 6 Object attributen ITSAPP

9.1.2 Object ITSAPP

Het IVERA Object ITSAPP kan worden gebruikt om de ITS applications te beheren.

Attribuut	Type	Object	Omschrijving
N	1	ITSAPP	Naam
O	1	ITS applicatie user management	Omschrijving
T	1	1	Type
U	0	6644	User Identificatie Control
E	0	NUMITSAPP	aantal data-elementen
I	1	ITSAPP.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	402	Data-element formaat
S	0		Data-element stapgrootte

Tabel 7 Object attributen ITSAPP

Formaat: ITSAPP Gebruiker (402), type 1

ITSAPPGebruiker = [GebruikersnaamITSA + "," [+ Type] + "," [+ Programma] + "," [+ Wachtwoord] + "," [+ Kruispunt]]

Letter = "A" .. "Z" | "a" .. "z"

Digit = "0" .. "9"

Digit1to9 = "1" .. "9"

UnderScore = "_"

Hyphen = "-"

GebruikersnaamITSA = Letter { Letter | Digit | UnderScore | Hyphen }

Type = "CONTROL" | "PROVIDER" | "CONSUMER"

Programma = Digit1to9 [Digit]

Wachtwoord = AsciiString
Kruispunt = AsciiString

Alleen een IVERA administrator kan de instellingen voor gebruikersbeheer wijzigen (username, password en type). Om deze instellingen voor een gebruiker te wijzigen dient het volgende formaat te worden gebruikt:

Gebruik: een element schrijven

ITSAPP/< element> = "[< gebruikersnaam> ,< type>],< programma>],< wachtwoord>],< kruispunt>]]"

waarbij:

< element> : objectelement (index nummer of indexnaam)
 < gebruikersnaam> : gebruikersnaam van de ITS applicatie (Alleen door ADMIN te wijzigen)
 < type> : het type ITS applicatie (CONTROL| PROVIDER| CONSUMER) (Alleen door ADMIN te wijzigen)
 < programma> : het programma nummer (1..99). (Vanaf gebruikers niveau 3 te wijzigen.) Alleen van toepassing voor een ITS-CLA application (CONTROL)
 < wachtwoord> : het wachtwoord voor de ITS applicatie. (Alleen door ADMIN te wijzigen)
 < kruispunt> : Identificatie (naam) van het kruispunt dat de ITS-CLA mag regelen Alleen van toepassing voor een ITS-CLA application (CONTROL)

Een gebruiker kan worden verwijderd door het schrijven van een lege string door de administrator.

Het lezen van het object ITSAPP zal een string geven in het volledige formaat, maar zonder wachtwoord.

Gebruik: een element lezen

ITSAPP/< element> = "< gebruikersnaam> ,< type> ,< programma> ,< kruispuntId> "

Voorbeelden:

Een element schrijven:

ITSAPP/#1= "Control1,CONTROL,96,Password,KRP55"

Een element lezen:

ITSAPP/#1

ITSAPP/#1= "Control1,CONTROL,96,KRP55"

Een gebruiker verwijderen:

ITSAPP/#1= ""

9.1.3 Object ITSSTAT

Het object ITSSTAT bevat de ITS application status:

Attribuut	Type	Object	Omschrijving
N	1	ITSSTAT	Naam
O	1	ITS applicatie status	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMITSAPP	aantal data-elementen
I	1	ITSAPP.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1 ruwe tekst	Data-element formaat
S	0		Data-element staggrootte

Tabel 8 Object attributen ITSSTAT

Dit object geeft de huidige status van de ITS applicaties weer. De mogelijke waarden hangen af van het type van de ITS applicatie

- Disconnected
- Connected
- NotConfigured
- Offline
- ReadyToControl
- StartControl
- InControl
- EndControl
- Error

Voorbeeld:

Een element lezen:

ITSSTAT/#2

ITSSTAT/#3= "InControl"

9.2 IVERA Gebruiker Beheer

9.2.1 Object USER

Het IVERA Object USER zal in IVERA 4 worden gebruikt om de gebruikers te definiëren die de IVERA interface kunnen gebruiken. (Beide interfaces IVERA-TLC and IVERA-APP)

Een administrator heeft speciale rechten in het IVERA protocol. Alleen een administrator kan de gebruikersnaam en gebruikers groep wijzigen. Het wachtwoord kan worden gewijzigd door een administrator of door de gebruiker zelf. Er kan meer dan een administrator zijn in de lijst van gebruikers.

Belangrijke opmerking over de eerste gebruiker, USER/#0:

- USER/#0 is altijd een administrator.
- De gebruikers groep van USER/#0 kan niet worden gewijzigd
- USER/#0 kan nooit worden verwijderd uit de lijst met gebruikers
- Elke admin gebruiker kan de gebruikersnaam en wachtwoord van USER/#0 wijzigen.

Het lezen van dit object zal de gebruikersnaam en de gebruikers groep geven.

Attribuut	Type	Object	Omschrijving
N	1	USER	Naam
O	1	IVERA gebruiker management	Omschrijving
T	1	1	Type
U	0	6666	User Identificatie Control
E	0	NUMUSERS	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	404	Data-element formaat
S	0		Data-element stapgrootte

Tabel 9 Object attributen USER

Format: IveraGebruiker (404), type 1

IveraGebruiker =

Gebruikersnaam + “,” + Gebruikersgroep [+ “,” + Wachtwoord + “,” + Nieuw 1 + “,” + Nieuw 2]

Gebruikersnaam = AsciiString

Gebruikersgroep = “1” | “2” | “3” | “4”

Wachtwoord = AsciiString

Nieuw 1 = AsciiString
Nieuw 2 = AsciiString

Gebruik: een element schrijven

USER/< element> = "< gebruikersnaam> ,< gebruikersgroep> [< wachtwoord> ,< nieuw 1> ,< nieuw 2>]"

waarbij:

< element> : objectelement (index nummer)
< gebruikersnaam> : gebruikersnaam van de gebruiker (alleen te wijzigen door Admin)
< gebruikersgroep> : de IVERA gebruikersgroep (1..4). Admin= 4. (alleen te wijzigen door Admin)
< wachtwoord> : het wachtwoord van de gebruiker of de huidige ingelogde Admin gebruiker
< nieuw 1> : nieuw wachtwoord voor deze gebruiker
< nieuw 2> : nieuw wachtwoord voor deze gebruiker (moet overeenkomen met nieuw 1)

Voorbeelden:

Een element schrijven:

USER/#2= " admin2,4,secret1,new Secret,new Secret "

Een element lezen:

USER/#2

USER/#2= " admin2,4 "

10. Object definitie

De objecten met attributen zijn gedefinieerd in de volgende hoofdstukken en paragrafen en worden daar nader toegelicht.

10.1 Gebruikersgroepen

Het IVERA protocol definieert 4 gebruikersgroepen. In de toepassing voor verkeersregelininstallaties worden de gebruikersgroepen als volgt gedefinieerd:

1. De wereld
2. Kantonnier
3. Verkeerskundige
4. Technisch onderhoud (fabrikant)

10.2 Overzicht alle objecten

De volgende tabel bevat een overzicht van de objectattributen zoals gedefinieerd in het IVERA protocol.

Attribuut	Type	Omschrijving
N	1	Naam
O	1	Omschrijving
T	0	Type
U	0	User Identificatie Control
L	0	Logboek
W	0	Wijzigingsteller
E	0	aantal data-elementen
I	1	Index verwijzing per dimensie
MIN	0	Minimum data-elementwaarde
MAX	0	Maximum data-elementwaarde
IMIN	1	Index data-element minimumwaarde
IMAX	1	Index data-element maximumwaarde
ITYPE	1	Index data-element type
F	0	Data-element formaat
S	0	Data-element stapgrootte
A	1	Overzicht alle attributen

Tabel 10.1 Object attributen

De tabel op de volgende bladzijde bevat een overzicht van alle objecten, met uitzondering van de automaat (X..).

Indien een attribuut niet is gespecificeerd, staat het de leverancier van een IVERA-automaat vrij zelf de waarde van het attribuut te bepalen. De attributen IMIN en IMAX zijn in het geheel niet gedefinieerd en dus vrij te definiëren.

NB. Indien het aantal elementen (E) is gespecificeerd als een naam, zie naamconventie (tabel 5.2).

NB. De volgorde van gebruikersgroepen (U) is groep 4,3,2,1.

11. Datum en tijd

11.1 Algemeen datum en tijd

Voor de datum en tijd zijn de volgende objecten gedefinieerd:

Object	Omschrijving
KTIJD	Kalendertijd
TIJD	Actuele systeemtijd
DATUM	Actuele systeemdatum
JAAR	Actueel jaar
WEEK	Weeknummer
DAG.I	Index dag van de week
DAG	Nummer van dag van de week
BIJZDAG	Bijzondere dag
WKZB	Weeknummer begin zomertijd
WKZE	Weeknummer einde zomertijd

Tabel 3.14. Objecten datum en tijd

Een verkeersregeltoestel schakelt functionaliteit op basis van de tijd. Het gaat om bijvoorbeeld de maximum groentijden voor ochtend- en de avondspits, maar ook om bijvoorbeeld het toestaan van de aansturing van de akoestische signalering en de wens tot regelen en knipperen van de installatie.

Het object KTIJD is het object om de actuele kalendertijd uit te lezen. Het instellen van de tijd vindt plaats met de objecten DATUM en TIJD.

Deze laatste 2 objecten worden ook toegepast om het regeltoestel aan een beheercentrale te synchroniseren.

De objecten DAG en WEEK bevat het actuele dag van de week respectievelijk het weeknummer. In de regeltoestellen worden deze automatisch bepaald aan de hand van de datum en is daarmee niet wijzigbaar.

Met het object BIJZDAG zijn bijzondere dagen in te stellen, zoals de kerstdagen.

Weeknummer voor het begin en einde van de zomertijd voor het huidige jaar worden met de objecten WKZB en WKZE ingesteld. Wordt een 0 ingevuld, dan wordt de zomer/wintertijd-omschakeling gebaseerd op de Midden-Europese tijdzone, waaronder ook Nederland valt.

11.2 Objecten datum en tijd

11.2.1 KTIJD

Het object KTIJD geeft de kalendertijd weer:

Attribuut	Type	Object	Omschrijving
N	1	KTIJD	Naam
O	1	Kalendertijd	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	20	Data-element formaat
S	0		Data-element stapgrootte

Tabel 11.1 Object attributen

Formaat: Kalendertijd (20), type 1

De kalendertijd is een string met datum, tijd en dag van de week in het volgende formaat:

DDD JJJJ-MM-DD UU:MM:SS

Element	Omschrijving
DDD	Weekdag (MAA,DIN,WOE,DON,VRY,ZAT,ZON).
JJJJ	Jaar (1900 .. 2099)
MM	Maand (01 .. 12)
DD	Dag van de maand (01 .. 31)
UU	Uur (00 .. 23)
MM	Minuut (00 .. 59)
SS	Seconde (00 .. 59)

Tabel 11.2 Definitie kalendertijd

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
KTIJD/#0	KTIJD/#0=<DDD JJJJ-MM-DD UU:MM:SS>	Opvragen actuele kalendertijd
KTIJD/#0="MAA 2007-11-09 23:43:12"	:E11	Aanpassen kalendertijd niet mogelijk en resulteert in een foutmelding

Tabel 11.3 Voorbeelden KTIJD

11.2.2 TIJD

In een iTLC wordt elk platform gesynchroniseerd via een NTP server. Vanaf IVERA versie 4 zijn de DATUM en TIJD objecten daarom read-only.

Het object TIJD geeft de actuele systeemtijd weer:

Attribuut	Type	Object	Omschrijving
N	1	TIJD	Naam
O	1	Actuele systeemtijd	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	235959	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	21	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 11.4 Object attributen

Formaat: klok (uur/minuut/seconde) (21), type 0

Weergave: HH:MM:SS

Waarde: uur * 10000 + minuut * 100 + seconde

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TIJD/#0	TIJD/#0=<HHMMSS>	Opvragen actuele systeemtijd

Tabel 11.5 Voorbeelden TIJD

11.2.3 DATUM

In een iTLC wordt elk platform gesynchroniseerd via een NTP server. Vanaf IVERA versie 4 zijn de DATUM en TIJD objecten daarom read-only.

Het object DATUM geeft de actuele systeemdatum weer:

Attribuut	Type	Object	Omschrijving
N	1	DATUM	Naam
O	1	Actuele systeemdatum	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	22	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 11.6 Object attributen

Formaat: datum (jaar/maand/dag) (22), type 0

Weergave: JJJ:MM:DD

Waarde: jaartal * 10000 + maand * 100 + dag

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
DATUM/#0	DATUM/#0=<JJJMMDD>	Opvragen actuele systeemdatum

Tabel 11.7 Voorbeelden DATUM

11.2.4 JAAR

Het object JAAR geeft het actuele systeemjaar weer:

Attribuut	Type	Object	Omschrijving
N	1	JAAR	Naam
O	1	Actueel jaar	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	25	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 11.8 Object attributen

Formaat: jaartal (25), type 0

Weergave:

Waarde: jaartal (0=elk jaar, >999 = jaartal)

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
JAAR	JAAR=<JJJJ>	Opvragen actuele systeemjaar
JAAR/#0=<jjjj_nie uw>	:E11	Aanpassen jaar is niet mogelijk.

Tabel 11.9 Voorbeelden JAAR

11.2.5 WEEK

Het object WEEK geeft het actuele weeknummer weer:

Attribuut	Type	Object	Omschrijving
N	1	WEEK	Naam
O	1	weeknummer	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0	1	Minimum data-elementwaarde
MAX	0	53	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	26	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 11.10 Object attributen

Formaat: weeknummer (26), type 0

Weergave:

Waarde: 0=elke week, 1-53=weeknummer

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
WEEK	WEEK =<WW>	Opvragen actuele weeknummer
WEEK//#0=1	:E11	Aanpassen weeknummer is niet mogelijk.

Tabel 11.11 Voorbeelden WEEK

11.2.6 DAG.I

Het object DAG.I geeft de naam van de dag van de week weer:

Attribuut	Type	Object	Omschrijving
N	1	DAG.I	Naam
O	1	index dag van de week	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMDAG	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 11.12 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
DAG.I	DAG.I="maandag","dinsdag","woensdag","donderdag","vrijdag","zaterdag","zondag","elke dag","werkdagen","weekeinde","feestdag","koopavond","koopzondag","speciaal"	Opvragen index object
DAG.I/#0	DAG.I/#0="maandag"	Opvragen index 0
DAG.I/#14	:E=12	Opvragen niet bestaande index

Tabel 11.13 Voorbeelden DAG.I

11.2.7 DAG

Het object DAG geeft het actuele nummer van de dag van de week weer:

Attribuut	Type	Object	Omschrijving
N	1	DAG	Naam
O	1	Nummer van dag van de week	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0	1	Minimum data-elementwaarde
MAX	0	7	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	24	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 11.14 Object attributen

Formaat: dag van de week (24), type 0

Waarde	Omschrijving
1	maandag
2	dinsdag
3	woensdag
4	donderdag
5	vrijdag
6	zaterdag
7	zondag
8	elke dag
9	werkdagen
10	weekeinde
11	feestdag
12	koopavond
13	koopzondag
14	speciaal

Tabel 11.15 Dag van de week

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
DAG	DAG=<nummer van de actuele dag>	Opvragen dagnummer: 1=maandag, 2=dinsdag, 3=woensdag, 4=donderdag, 5=vrijdag, 6=zaterdag, 7=zondag

Tabel 11.16 Voorbeelden DAG

11.2.8 BIJZDAG

Het object BIJZDAG geeft de bijzondere dag weer:

Attribuut	Type	Object	Omschrijving
N	1	BIJZDAG	Naam
O	1	Bijzondere dag	Omschrijving
T	1	1	Type
U	0	6644	User Identificatie Control
E	0	NUMBIJZDAG	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	70	Data-element formaat
S	0		Data-element stapgrootte

Tabel 11.17 Object attributen

Formaat: Lijst met bijzondere dagen (70), type 1

Voor dagen in de lijst met bijzondere dagen wordt de volgende syntax gehanteerd:

Bijzonderedaglystentry = Dag + “,” Dagtype + [“,” Toelichting]

Dag = Datum(23)|Datum(22)| Weekdag|0

Weekdag = Dag van de week (1,2,3,4,5,6 of 7)

Dagtype = Dag van de week (11,12,13 of 14)

Toelichting = AsciiString

NB:

Wanneer Dag de waarde 0 heeft is deze entry voor BIJZDAG buiten gebruik. De waarden in de andere velden worden dan genegeerd. Ze moeten echter wel aan het bovenstaande formaat voldoen.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
BIJZDAG/*		De lijst met bijzondere dagen wordt in het juiste formaat getoond en komt overeen met die in de automaat.
BIJZDAG/#2		De tweede entry in de lijst met bijzondere dagen wordt in het juiste formaat getoond.
BIJZDAG/#4="20080430,12,koningin nedag"	BIJZDAG/#4=20080430,12,koninginnedag	De vierde entry in de lijst met bijzondere dagen wordt in het juiste formaat getoond met aanpassing.

Tabel 11.18 Voorbeelden BIJZDAG

11.2.9 WKZB

Het object WKZB geeft het weeknummer begin zomertijd weer:

Attribuut	Type	Object	Omschrijving
N	1	WKZB	Naam
O	1	Weeknummer begin zomertijd	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0	1	Minimum data-elementwaarde
MAX	0	53	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 11.19 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
WKZB/#0	WKZB/#0=13	Opvragen weeknummer begin zomertijd.
WKZB/#0=<weeknummer b_nieuw>	WKZB/#0=<weeknummer b_nieuw>	Weeknummer begin zomertijd overgenomen.
WKZB/#0=54	:E11	Foutief weeknummer

Tabel 11.20 Voorbeelden WKZB

11.2.10 WKZE

Het object WKZE geeft het weeknummer einde zomertijd weer:

Attribuut	Type	Object	Omschrijving
N	1	WKZE	Naam
O	1	Weeknummer einde zomertijd	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0	1	Minimum data-elementwaarde
MAX	0	53	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 11.21 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
WKZE/#0	WKZE/#0=43	Opvragen weeknummer einde zomertijd.
WKZE/#0=<weeknummer e_nieuw>	WKZE/#0=<weeknummer e_nieuw>	Weeknummer einde zomertijd overgenomen.
WKZE/#0=54	:E11	Foutief weeknummer

Tabel 11.22 Voorbeelden WKZE

12. Identificatie

12.1 Algemeen identificatie

Voor de identificatie zijn de volgende objecten gedefinieerd:

Object	Omschrijving
VRIID.I	Index automaatidentificatie
VRIID	Automaatidentificatie
VRIVER.I	Index versienummers
VRIVER	Versienummers
TID	Toepassing identificatienummer
XID	Automaat identificatienummer

Tabel 3.14. Objecten identificatie

Elk regeltoestel is identificeerbaar. Dit is tenminste noodzakelijk om deze aan een beheerscentrale te kunnen koppelen.

12.2 Objecten identificatie

12.2.1 VRIID.I

Het object VRIID.I geeft de index automaat identificatie weer:

Attribuut	Type	Object	Omschrijving
N	1	VRIID.I	Naam
O	1	Index automaat identificatie	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 12.1 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Het object VRIID.I bevat een aantal elementen waarmee een VRI gekarakteriseerd wordt:

VRIID.I		VRI identificatie
0	INST_NR	Installatienummer
1	KRP_NR	Kruispuntnummer
2	KRP_NAAM	Kruispuntnaam
3	AUT_TYPE	Automaattype
4	PAK_TYPE	Pakkettype
5	INST_DATUM	Datum van installatie
6	RESERVE6	Gereserveerd
7	RESERVE7	Gereserveerd
8	RESERVE8	Gereserveerd
9	RESERVE9	Gereserveerd

Tabel 12.2 Automaatidentificatie

INST_NR

Een uniek nummer of naam van de installatie, waaronder de installatie bij de leverancier is geregistreerd.

Voorbeeld: "V10002".

KRP_NR

Unieke identificatie van het kruispunt. Voorbeeld: "KRP55".

KRP_NAAM

De naam van het kruispunt. Meestal twee straatnamen gescheiden door een "/".

Voorbeeld: "Dorpstraat/Kerkstraat"

AUT_TYPE

Een typeaanduiding van het regeltoestel.

Voorbeeld: "FAB X Type Y"

PAK_TYPE

Omschrijving van het applicatieprogramma.

Voorbeeld: "CCOL".

INST_DATUM

Datum waarop de installatie in bedrijf is gesteld.

Voorbeeld: "1997-01-17".

RESERVE6 t/m RESERVE9

De RESERVEx elementen zijn gereserveerd voor specificatie in een toekomstige IVERA versie. Het is niet toegestaan hier een fabrikantspecifieke invulling aan te geven.

NB. De VRIID elementen zijn niet aan een bepaald formaat gebonden.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRIID.I	VRIID.I="INST_NR","KRP_N R","KRP_NAAM","AUT_TYPE ","PAK_TYPE","INST_DATU M","RESERVE6","RESERVE 7","RESERVE8","RESERVE9 "	Het resultaat van het indexobject en mag in geen geval lege velden weergeven
VRIID.I/#9	VRIID.I/#9="RESERVE9"	Automaat retourneert huidige inhoud van het 10 ^e element.
VRIID.I/#10	:E=12	Respons op het opvragen van een index buiten de vastgestelde grens.

Tabel 12.3 Voorbeelden VRIID.I

12.2.2 VRIID

Het object VRIID geeft de automaat identificatie weer:

Attribuut	Type	Object	Omschrijving
N	1	VRIID	Naam
O	1	Automaat identificatie	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1	VRIID.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, ruwe tekst	Data-element formaat
S	0		Data-element stapgrootte

Tabel 12.4 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRIID	VRIID="<installatienummer>"," <kruispuntnummer>","<kruis puntnaam>","<automaattype> ","<pakkettype>","<datum van installatie>"," " " " " " " "	Automaat retourneert huidige inhoud van object vrijd volgens specificaties fabrikant
VRIID/#0	VRIID/#0="<installatienumme r>"	Automaat retourneert huidige inhoud van het installatienummer.
VRIID/#10	:E=12	Respons op het opvragen van een index buiten de range.

Tabel 12.5 Voorbeelden VRIID

12.2.3 VRIVER.I

Het object VRIVER.I geeft de index automatisch identificatie weer:

Attribuut	Type	Object	Omschrijving
N	1	VRIVER.I	Naam
O	1	Index versienummers	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 12.6 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Het object VRIVER.I bevat een aantal versienummers van de in de VRI aanwezige software:

VRIVER.I		Versie nummers
0	APPL	Kruispuntapplicatie
1	PAKKET	Pakket
2	INTERFACE	Interface
3	PROCES	Procesbesturing
4	BEWAKER	Autonome bewaker
5	FAB1	Fabrikantspecifiek (indexnaam en waarde)
6	FAB2	Fabrikantspecifiek (indexnaam en waarde)
7	FAB3	Fabrikantspecifiek (indexnaam en waarde)
8	FAB4	Fabrikantspecifiek (indexnaam en waarde)
9	FAB5	Fabrikantspecifiek (indexnaam en waarde)

Tabel 12.7 Versie nummers

APPL

Versienummer van het kruispunt specifieke programma.

Pakket

Versienummer van het applicatie pakket. Voorbeelden van applicaties zijn o.a. CCOL en RWS-C.

Interface

Versienummer van de interface tussen de applicatie en de procesbesturing. Voorbeeld van zo'n interface is de CVN C-interface.

Proces

Versienummer van de procesbesturing. De procesbesturing bevat de basissoftware van het regeltoestel.

Bewaker

Versienummer van de autonome bewakingsprocessor.

FAB1 t/m FAB5

Afhankelijk van het type regeltoestel kan de fabrikant nog enkele regeltoestel specifieke versienummers definiëren. Fabrikanten zijn vrij om een fabrikantspecifieke invulling te geven aan de FABx elementen, voor zowel indexnaam als waarde.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRIVER.I	VRIVER.I="APPL","PAKKET", "INTERFACE","PROCES","B EWAKER","FAB1","FAB2","F AB3","FAB4","FAB5"	Het resultaat van het indexobject
VRIVER.I/#4	VRIVER.I/#4="BEWAKER"	Automaat retourneert huidige versie van de bewaker.
VRIVER.I/#10	:E=12	Respons op het opvragen van een index buiten de range.

Tabel 12.8 Voorbeelden VRIVER.I

12.2.4 VRIVER

Het object VRIVER geeft de versienummers weer:

Attribuut	Type	Object	Omschrijving
N	1	VRIVER	Naam
O	1	Versienummers	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1	VRIVER.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, ruwe tekst	Data-element formaat
S	0		Data-element stapgrootte

Tabel 12.9 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRIID/inst_nr	VRIID/inst_nr="<installatienu mmer>"	Automaat retourneert versie installatienummer
VRIVER/#1	VRIVER/#1="<versienummer _pakket>"	Automaat retourneert versie met index 1
VRIVER/#0="1,2,3, 4,5"	:E=11	Schrijven naar de versie is niet mogelijk

Tabel 12.10 Voorbeelden VRIVER

12.2.5 TID

Het object TID geeft het toepassing identificatienummer weer:

Attribuut	Type	Object	Omschrijving
N	1	TID	Naam
O	1	Toepassing identificatienummer	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 12.11 Object attributen

Voor dit toepassingsgebied is het bereik 1..9999 gereserveerd. De TID bevat het versienummer van dit document "IVERA Objectdefinitie Verkeersregelininstallaties". Bijv. 301 is versie 3.01.

12.2.6 XID

Het object XID geeft het automatisch identificatienummer weer:

Attribuut	Type	Object	Omschrijving
N	1	XID	Naam
O	1	automaat identificatienummer	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 12.12 Object attributen

Binnen dit toepassingsgebied is het XID als volgt gedefinieerd:

Bereik	Gereserveerd voor
1.000.000 .. 1.999.999	TPA
2.000.000 .. 2.999.999	Vialis
3.000.000 .. 3.999.999	Peek (of Imtech)
4.000.000 .. 4.999.999	Siemens
5.000.000 .. 5.999.999	Ko Hartog Verkeerstechiek
6.000.000 .. 6.999.999	SWARCO

Tabel 12.13 XID

12.2.7 Automaat specifieke events

Afhankelijk van het type regeltoestel is het mogelijk dat er events zijn die niet in een van de gedefinieerde categorieën vallen. De betekenis van automaatspecifieke events (100.000..199.999) is eenduidig vastgelegd in combinatie met het unieke automaatidentificatienummer (XID).

13. Toegang

13.1 Algemeen toegang

Voor de toegang zijn de volgende objecten gedefinieerd:

Object	Omschrijving
FTPUSER.I	FTP-gebruikersnamen
FTPPASS	FTP-passwords
FTPLOCATION	FTP-locatie
DATAKOM.I	Index object Datacom
DATAKOM	Instelling datacommunicatie
AUTHOG	Gebruikersnamen
AUTHOP	Toegangscodes
LOGINNIVEAU	Nummer gebruikersgroep waaronder ingelogd is.
LOGIN	Login-commando

Tabel 3.14. Objecten toegang

Objecten FTPUSER.I, FTPPASS en FTPLOCATION

Genoemde objecten bevatten de configuratie van de gebruikersnamen en passwords van de FTP-server. Via het IVERA-protocol kunnen op afstand alleen de passwords van de FTP-server worden gewijzigd. De gebruikersnamen en de gebruikersrechten zijn vast gedefinieerd.

Het aantal elementen van deze objecten bepaalt het aantal gedefinieerde gebruikers.

Iedere gebruiker die in IVERA is ingelogd kan wijzigingen uitvoeren. Om te voorkomen dat iemand zomaar een password wijzigt, zal ook het huidige password moeten worden opgegeven.

Een lijst met gebruikersnamen kan worden opgevraagd door het lezen van het object FTPUSER.I. Het schrijven naar het object FTPUSER.I is niet mogelijk. Het object FTPUSER.I kan als index gebruikt worden voor FTPPASS en FTPLOCATION.

Normaliter is de FTP-gebruikersnaam gelijk aan de gebruikersnaam uit FTPUSER.I. Als een VRI dit niet kan ondersteunen dan is het toegestaan om een wel ondersteunde gebruikersnaam te retourneren. Deze moet dan gebruikt worden in de FTP-sessie, en bij het wijzigen van het password. De gebruikersnamen mogen niet gewijzigd worden.

Deze objecten hoeven slechts door de VRI ondersteund te worden voor zover de implementatie van FTP in de VRI dat toelaat. Centrales zullen ze geheel moeten ondersteunen, aangezien er VRI's kunnen zijn die deze functionaliteit volledig toepassen.

In het FTPLOCATION object is per FTP-gebruiker aangegeven in welke directory de gewenste gegevens staan ten opzichte van de root van deze gebruiker (dit is dus niet noodzakelijkerwijs de root van het file systeem, en kan per gebruiker anders zijn). Indien voor alle FTP-gebruikers de directory in FTPLOCATION root ("/") is mag het FTPLOCATION object weggelaten worden. Een centrale mag er dan van uit gaan dat na het inloggen met FTP de juiste directory gekozen is.

Het is toegestaan dat de locatie zich wijzigt bij het wisselen van programma. Daarom is het, indien de automaat het FTPLOCATION object implementeert, noodzakelijk om de locatie op te vragen voordat een FTP-sessie gestart wordt.

Na aanspreken van fasebewaking is de procesbesturing verantwoordelijk voor het geven van het DUMP-commando('s) en het opslaan van de gegenereerde dump. De dump wordt in een file opgeslagen welke middels FTP opgehaald kan worden. De naam en de plaats van het bestand is terug te vinden in het FTPLOCATION-object (gebruiker dumpinfo). De VRI is verantwoordelijk voor het beheer van de dumpfiles.

Doel is om op gestandaardiseerde manier programmadumps uit de VRI op te halen. Voor het ophalen van bestanden wordt FTP gebruikt (RFC959).

Voor het ophalen van de dump wordt gebruik gemaakt van dezelfde FTP-objecten als bij het laden. Bij het maken van de verbinding wordt dan de FTP-gebruiker *dumpinfo* gebruikt.

Objecten DATACOM.I en DATACOM

Deze objecten hebben betrekking op de communicatie tussen het regeltoestel en de beheerscentrale. Dit object bevat onder andere het telefoonnummer en het IP-adres van de beheerscentrale.

Het object bevat ook enkele verbindingseigenschappen.

Objecten AUTHOG en AUTHOP

Deze objecten geven de gebruikersnamen en toegangscode weer.

Objecten LOGINNIVEAU en LOGIN

Deze objecten hebben betrekking op het inloggen op het regeltoestel. Het object LOGINNIVEAU geeft nummer van de gebruikersgroep waaronder is ingelogd weer. Het object LOGIN is bedoeld om het login-commando te versturen.

13.2 Objecten toegang

13.2.1 FTPUSER.I

Het object FTPUSER.I geeft de FTP-gebruikersnamen weer:

Attribuut	Type	Object	Omschrijving
N	1	FTPUSER.I	Naam
O	1	FTP-gebruikersnamen	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 13.1 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

De gebruikersnamen en gebruikersrechten zijn vastgelegd in de volgende tabel:

Index	Gebruikersnaam (FTPUSER.I)	Directory (FTPLOCATION)	Rechten
0	Admin	De root van het file systeem.	De gebruiker heeft lees- en schrijfrechten op het gehele file systeem. De gebruiker kan de FTP passwords van alle andere FTP gebruikers wijzigen.
1	Upload	De directory waar de applicatie dient te worden geschreven.	Het lezen/schrijven van een nieuwe applicatie.
2	Service	De root van het file systeem.	De gebruiker heeft leesrechten op het gehele file systeem.
3	Trafficinfo	De directory waar de analyse files zijn opgeslagen.	Lezen van de analyse informatie (MV files, V-log, etc).
4	loginfo	De directory waar de log informatie van de actieve applicatie staat.	Lezen van de log informatie van de actieve applicatie.
5	dumpinfo	De directory waar de verschillende soorten dumps staan.	Het lezen en verwijderen van dumpbestanden.
6	FAB1		
7	FAB2		
8	FAB3		
9	anonymous	Een directory waarin informatie over de regelaar staat (bijvoorbeeld een HTML file).	Alleen lezen. Voor deze gebruiker wordt bij het inloggen elk password geaccepteerd.

Tabel 13.2 gebruikersnamen en gebruikersrechten

Voor het laden van programma's moet dus een FTP-sessie als gebruiker *upload* gemaakt worden met de automaat.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
FTPUSER.I	FTPUSER.I="admin", "upload","service", "trafficinfo", "loginfo", "RESERVE6", "RESERVE7", "RESERVE8", "RESERVE9", "anonymous"	De gebruikersnamen RESERVE6..9 kunnen automatisch specifieke namen bevatten. Het resultaat mag in geen geval lege indexvelden tonen
FTPUSER.I/#5	FTPUSER.I/#5="DUMPINFO"	Respons in het opvragen van de index 5.

Tabel 13.3 Voorbeelden FTPUSER.I

13.2.2 FTTPASS

Het object FTTPASS geeft de FTP-passwords weer:

Attribuut	Type	Object	Omschrijving
N	1	FTTPASS	Naam
O	1	FTP-passwords	Omschrijving
T	1	1	Type
U	0	6666	User Identificatie Control
E	0	10	aantal data-elementen
I	1	FTPUSER.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	10	Data-element formaat
S	0		Data-element stapgrootte

Tabel 13.4 Object attributen

Formaat: Instellen gebruikersnaam/toegangscode (10), type 1

Voor het wijzigen van gebruikersnamen, toegangscode en rechten wordt de volgende syntax gehanteerd:

```

Commando = Gebruikersnaam " " Toegangscode " " Nieuw1 " " Nieuw2
Gebruikersnaam = AsciiString
Toegangscode = AsciiString
Nieuw1 = AsciiString
Nieuw2 = AsciiString
  
```

Het object FTTPASS levert bij lezen een lege string indien de FTP gebruikersnaam gelijk is aan de in het object FTP.I gespecificeerde naam. Wanneer een afwijkende FTP gebruikersnaam dient te worden gebruikt, dan zal bij het lezen van het object FTTPASS deze afwijkende FTP gebruikersnaam worden getourneerd.

Het wijzigen van een password gebeurt door het schrijven naar het bijbehorende element van het object FTTPASS.

```
FTTPASS/<element>="<FTP gebruikersnaam>,<password>,<nieuw1>,<nieuw2>"
```

waarbij:

```

<element>      : objectelement (index nummer of een gebruikersnaam uit FTPUSER.I)
<gebruikersnaam> : de FTP gebruikersnaam
<password>     : huidige password voor deze gebruiker of password van 1ste gebruiker.
<nieuw1>       : nieuw password voor deze gebruiker.
<nieuw2>       : moet overeenkomen met <nieuw1>.
  
```

Door deze opzet heeft de 1^{ste} gebruiker meer rechten. Namelijk de 1^{ste} gebruiker kan de passwords van alle gebruikers wijzigen. De overige gebruikers kunnen alleen hun eigen password wijzigen.

Een eenvoudige VRI met maar één FTP-gebruiker mag voor alle elementen van FTPPASS dezelfde FTP-gebruikersnaam teruggeven. De directory in FTPLOCATION geeft vervolgens aan waar de verlangde gegevens zich bevinden.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
FTPPASS	FTPPASS="Fadmin","Fupload","Fservice","Ftrafficinfo","Floginfo","Ftdc","FRESERVE7","FRESERVE8","FRESERVE9","Fanonymous"	De inlognamen kunnen verschillen van de logische gebruikersnamen. Resultaten mogen ook leeg zijn
FTPPASS/#1="Fupload,<upwd>,ivtest,ivtest"	FTPPASS/#1="Fupload,<upwd>,ivtest,ivtest"	Verander het wachtwoord voor de gebruiker 'upload' (Vervang <upwd> door het huidige paswoord voor de gebruiker 'upload')
FTPPASS/#1="Fupload,ivtest,<upwd>,<upwd>"	FTPPASS/#1="Fupload,ivtest,<upwd>,<upwd>"	Verander het wachtwoord voor de gebruiker 'upload' (Vervang <upwd> door het originele paswoord voor de gebruiker 'upload')

Tabel 13.5 Voorbeelden FTPPASS

13.2.3 FTPLOCATION

Het object FTPLOCATION geeft de FTP-locatie weer:

Attribuut	Type	Object	Omschrijving
N	1	FTPLOCATION	Naam
O	1	FTP-locatie	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1	FTPUSER.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, ruwe tekst	Data-element formaat
S	0		Data-element staggrootte

Tabel 13.6 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
FTPLOCATION	FTPLOCATION="/" "/D2/appl1" "/D2/appl1" "/D2/appl1" "/D2/tdc" "RESERVE7" "RESERVE8" "RESERVE9" "/D2/WebServer"	Vraag object FTPLOCATION op. FTPLOCATION is een optioneel object, wanneer deze niet is geïmplementeerd mag de centrale er vanuit gaan dat na inloggen met FTP de juiste directory wordt gekozen

Tabel 13.7 Voorbeelden FTPLOCATION

13.2.4 DATACOM.I

Het object DATACOM.I geeft de index object datacommunicatie weer:

Attribuut	Type	Object	Omschrijving
N	1	DATACOM.I	Naam
O	1	index object datacom	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	16	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 13.8 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Het object DATACOM.I bevat een aantal elementen waarmee een communicatieverbinding gekarakteriseerd wordt:

DATACOM.I		Beschrijving
0	Telefoon_centrale	Telefoonnummer centrale
1	IP_adres_centrale	IP adres centrale
2	Poortnummer	Poortnummer
3	Triggerevents	Triggerevents
4	Terugbeltijd	Terugbeltijd
5	Log_datacomevents	Log datacomevents
6	IP_adres_VRI	IP-adres regeltoestel
7	TO_communicatie	Time-out communicatie
8	TO_modem	Time-out opbouw modemverbinding
9	TO_PPP	Time-out opbouw PPP-connectie
10	TO_triggerpoort	Time-out openen triggerpoort centrale
11	TO_respons	Time-out respons van centrale
12	Retrytijd	Retry-tijd
13	Retrymaximum	Maximum aantal retries
14	TO_IVERA_sessie	Time-out IVERA sessie
15	Reserve15	gereserveerd

Tabel 13.9 Datacommunicatie-instellingen

Telefoonnummer centrale

Het telefoonnummer bestaat uit maximaal 16 digits. Aanvullend zijn de speciale karakters "*", "#", en "," toegestaan.

Bijvoorbeeld: "0334541777", "0,0334541777" of "*99#".

IP-adres centrale

Het IP-adres van de centrale wordt gespecificeerd in het formaat: 255.255.255.255

Poortnummer

Het poortnummer is een integer getal. Default poortnummer is 5001.

Trigger-events

Het element triggerevents bevat een lijst van events gescheiden door komma's, die aangeeft voor welke events er een trigger melding naar de centrale moet worden verstuurd. Bij gebruik van kieslijnen dient de VRI, indien er geen actieve lijnverbinding is, eerst een uitbelverbinding naar de centrale tot stand te brengen conform de DATACOM instellingen. Bij

gebruik van een always-online verbinding met de centrale wordt de trigger onmiddellijk verstuurd

Bijvoorbeeld: "1010,1020,2500,100123,200456"

Terugbeltijd

Door middel van het gebruik van deze terugbeltijd kan worden gecontroleerd of het opbouwen van een verbinding vanuit de automaat goed functioneert.

Er kan een tijd in minuten worden ingesteld, waarna de automaat de centrale belt en een Testtriggerevent (6000) verstuurt. Daarna zet de automaat de terugbeltijd op 0.

Als de terugbeltijd 0 is, betekent dit, dat de terugbelfunctie uitgeschakeld is.

Log datacomevents

Default worden datacommunicatie-events (events in het bereik 6000-6999) gelogd.

Door de waardes 0 en 1 in dit element kan expliciet aangegeven worden, of datacommunicatie-events moeten worden gelogd.

IP-adres regeltoestel

Het IP-adres van het regeltoestel wordt gespecificeerd in het formaat: 255.255.255.255

Time-out communicatie

Time-out in seconden voor het in stand houden van de verbinding, terwijl er geen IP-communicatie plaatsvindt. De default waarde is 300 seconden.

Time-out opbouw modemverbinding

Time-out in seconden voor het tot stand komen van een modemverbinding met de centrale, wanneer het regeltoestel de verbindingsofbouw initieert. De default waarde is 120 seconden.

Time-out opbouw PPP-connectie

Time-out in seconden voor het tot stand komen van een PPP-connectie met de centrale. De default waarde is 60 seconden.

Time-out openen triggerpoort centrale

Time-out in seconden voor het openen van de triggerpoort in de centrale.

De default waarde is 30 seconden.

Time-out respons van centrale

Time-out in seconden voor het wachten op een respons op een trigger (het openen van poort 5000) vanuit de centrale.

De default waarde is 300 seconden.

Retry-tijd

Wachttijd in seconden door het regeltoestel gehanteerd tussen de retries. De default waarde is 180 seconden.

Maximum aantal retries

Het maximale aantal retries wanneer er geen trigger verstuurd kan worden naar de centrale. De default waarde is 5.

Time-out IVERA sessie

Time-out in seconden voor het wachten op een commando van de centrale of sturen van een bericht naar de centrale op IVERA protocol niveau. De default waarde is 60 minuten. Na het verstrijken van deze time-out wordt de verbinding op IVERA niveau verbroken.

Afhankelijk van de mogelijkheden van de VRI, kan het zijn dat één of meer van de in DATACOM opgenomen elementen niet instelbaar zijn. In dat geval kan het betreffende element van DATACOM niet gewijzigd worden en is de inhoud ervan slechts informatief.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
DATACOM.I	DATACOM.I="TELEFOON_CENTRALE","IP_ADRES_CENTRALE","POORTNUMMER","TRIGGEREVENTS","TERUGBELTIJD","LOG_DATACOMEVENTS","IP_ADRES_VRI","TO_COMMUNICATIE","TO_MODEM","TO_PPP","TO_TRIGGERPOORT","TO_RESPONSE","RETRYTIJD","RETRYMAXIMUM","RESERVE14","RESERVE15"	Vraag de index van DATACOM op
DATACOM.I/#14	DATACOM.I/#14="TO_IVERA_sessie"	Vraag het object DATACOM.I op
DATACOM/TO_IVERA_sessie=60	DATACOM/TO_IVERA_sessie=60	Stel de timeout in in het object DATACOM in seconden

Tabel 13.10 Voorbeelden DATACOM.I

13.2.5 DATACOM

Het object DATACOM geeft de instelling datacommunicatie weer:

Attribuut	Type	Object	Omschrijving
N	1	DATACOM	Naam
O	1	Instelling datacommunicatie	Omschrijving
T	1	1	Type
U	0	6644	User Identificatie Control
E	0	16	aantal data-elementen
I	1	DATACOM.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, ruwe tekst	Data-element formaat
S	0		Data-element stapgrootte

Tabel 13.11 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
DATACOM/#0="1234"	DATACOM/#0="1234"	Wijzig de parameter 'telefoonnummer centrale'
DATACOM/#1="192.168.42.1"	DATACOM/#1="192.168.42.1"	Wijzig de parameter 'IP_adres_centrale'
DATACOM/#1	DATACOM/#1="192.168.42.1"	Verificatie dat waarde wordt overgenomen
DATACOM/#0="+311234567890123"	:E=16	Wijzig de waarde van het eerste element uit object DATACOM met een waarde > 16 digits, en teken(s)

Tabel 13.12 Voorbeelden DATACOM

13.2.6 AUTHOG

Het object AUTHOG geeft de gebruikersnamen weer:

Attribuut	Type	Object	Omschrijving
N	1	AUTHOG	Naam
O	1	gebruikersnamen	Omschrijving
T	1	1	Type
U	0	6666	User Identificatie Control
E	0	10	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	10	Data-element formaat
S	0		Data-element stapgrootte

Tabel 13.13 Object attributen

Formaat: Instellen gebruikersnaam/toegangscode (10), type 1

Voor het wijzigen van gebruikersnamen, toegangscode en rechten wordt de volgende syntax gehanteerd:

```

Commando = Gebruikersnaam “,” Toegangscode “,” Nieuw1 “,” Nieuw2”
Gebruikersnaam = AsciiString
Toegangscode = AsciiString
Nieuw1 = AsciiString
Nieuw2 = AsciiString
  
```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
AUTHOG/#0	AUTHOG/#0="<usr1>"	Lees de 1e gebruikersnaam
AUTHOG/#0="<usr1>,<pwd1>,USERT,USERT"	AUTHOG/#0="<usr1>,<pwd1>,>,USERT,USERT"	Verander de gebruikersnaam met aangepaste huidige gebruikersnaam en wachtwoord
AUTHOG/#0	AUTHOG/#0="USERT"	Lees de 1e gebruikersnaam
AUTHOG/#0="USERT,<pwd1>,<usr1>,<usr1>"	AUTHOG/#0="USERT,<pwd1>,>,<usr1>,<usr1>"	Schrijf een gebruikersnaam terug in authog

Tabel 13.14 Voorbeelden AUTHOG

13.2.7 AUTHOP

Het object AUTHOP geeft de toegangscode weer:

Attribuut	Type	Object	Omschrijving
N	1	AUTHOG	Naam
O	1	toegangscode	Omschrijving
T	1	1	Type
U	0	6666	User Identificatie Control
E	0	10	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	10	Data-element formaat
S	0		Data-element stapgrootte

Tabel 13.15 Object attributen

Formaat: Instellen gebruikersnaam/toegangscode (10), type 1

Voor het wijzigen van gebruikersnamen, toegangscode en rechten wordt de volgende syntax gehanteerd:

```

Commando = Gebruikersnaam “,” Toegangscode “,” Nieuw1 “,” Nieuw2”
Gebruikersnaam = AsciiString
Toegangscode = AsciiString
Nieuw1 = AsciiString
Nieuw2 = AsciiString

```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
AUTHOP/#0	AUTHOP/#0=	Er wordt geen wachtwoord getoond deze is onzichtbaar uit veiligheidsredenen
AUTHOP/#0=<usr1>,<pwd1>,!#%&'()*+-./:;<=>@[^_`{}~,!#%&'()*+-./:;<=>@[^_`{}~,!#%&'()*+-./:;<=>@[^_`{}~	AUTHOP/#0=<usr1>,<pwd1>,!#%&'()*+-./:;<=>@[^_`{}~,!#%&'()*+-./:;<=>@[^_`{}~	Verander de gebruikersnaam met aangepaste gebruikersnaam en wachtwoord

Tabel 13.16 Voorbeelden AUTHOP

13.2.8 LOGINNIVEAU

Het object LOGINNIVEAU geeft nummer van de gebruikersgroep waaronder is ingelogd weer:

Attribuut	Type	Object	Omschrijving
N	1	LOGINNIVEAU	Naam
O	1	Nummer gebruikersgroep waaronder is ingelogd	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0	1	Minimum data-elementwaarde
MAX	0	4	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 13.17 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
Loginniveau	loginniveau=1	Lees het huidige inlogniveau uit

Tabel 13.18 Voorbeelden LOGINNIVEAU

13.2.9 LOGIN

Het LOGIN object is gewijzigd vanaf IVERA versie 4. Vanaf IVERA versie 4 dient de gebruiker in te loggen met een gebruikersnaam en wachtwoord. De login methode met behulp van een 4 cijferige pincode komt te vervallen.

Het object LOGIN is bedoeld om het login-commando te versturen:

Attribuut	Type	Object	Omschrijving
N	1	LOGIN	Naam
O	1	Login-commando	Omschrijving
T	1	1	Type
U	0	6666	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	405	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 13.19 Object attributen

Voorbeeld

Inloggen vindt plaats door de juiste gebruikersnaam en wachtwoord op te geven:
 LOGIN/#0 = "admin,secret"

Formaat: Login (405), type 1

LoginCommand = Gebruikersnaam + "," + Wachtwoord

Gebruikersnaam = AsciiString

Wachtwoord = AsciiString

Gebruik: login

LOGIN/#0= "< gebruikersnaam> ,< wachtwoord> "

waarbij:

< gebruikersnaam> : gebruikersnaam van de IVERA gebruiker.

< wachtwoord> : het wachtwoord voor de IVERA gebruiker.

Een ongeldige login resulteert in een foutmelding :E16.

Het lezen van het LOGIN object is niet gedefinieerd.

Wanneer een lege string naar het LOGIN object wordt geschreven, wordt de gebruiker uitgelogd.

Examples:

Logging in:
LOGIN/#0= "admin,secret"

Logging out:
LOGIN/#0= ""

Met het object LOGINNIVEAU wordt het huidige inlogniveau weergegeven, of te wel het hoogste niveau van inloggen van een gebruikersgroep.

Indien de ingevoerde login onjuist is wordt door de IVERA-slave de foutcode :E=16 geretourneerd. Na 3x foutief inloggen wordt de communicatieverbinding volledig - dat betekent bij kiesverbindingen met inbegrip van de fysieke verbinding - door de VRI verbroken. Voordat de verbinding wordt verbroken is het mogelijk, maar niet verplicht, dat foutcode :E=16 wordt verstuurd.

Indien een inlogpoging wordt gedaan voor een ander niveau dan waarop reeds is ingelogd en deze poging faalt, dan blijft de gebruiker op het huidige inlogniveau. Mocht dit drie keer achter elkaar mislukken dan zal de verbinding worden verbroken.

13.2.10 Datacommunicatie events

Datacommunicatie events hebben betrekking op gebeurtenissen in de datacommunicatie, zoals het tot stand komen van een verbinding, etc.

Code	Omschrijving	Detailinfo	VRI.A
6000	Testtrigger		
6001	Begin fysieke verbinding		
6002	Einde fysieke verbinding		
6003	Poging tot inbreuk		
6004	Uitbellen naar centrale		
6005	Login	<Inlogniveau>	
6006	Logout		

Tabel 13.20 Data communicatie events

NB: De kolom 'VRI.A' geeft aan per event of deze bij actief of inactief zijn in de automaat, zal worden getoond in respectievelijk zal worden verwijderd uit de 'Actieve storingenlijst'. Indien dit event inactief wordt gemaakt door een ander event, is dit expliciet aangegeven met 'eventcode'.

14. VRI algemeen

14.1 Algemeen VRI algemeen (procesbesturing)

Voor de VRI algemeen zijn de volgende objecten gedefinieerd:

Object	Omschrijving
TP.I	Index timers (proces)
TP	Tijd instelling (proces)
TP.A	Lopende tijd (proces)
SP.I	Index schakelaars (proces)
SP	Schakelaar (proces)
EXTRAINFO.I	Index extra info
EXTRAINFO.A	Informatieve actuele extra info string.
EXTRAINFOEXT	Informatieve toelichting extra info string.
VRI.C	VRI-commando
VRI.A	Actieve storingslijst
PAR.LB	Parameterlogboek
PAR.LA	Parameterlogboek (onb).

Tabel 3.14. Objecten VRI algemeen

Deze objecten hebben betrekking op de procesbesturing. In de volgende paragrafen wordt hier aan gerefereerd.

Objecten TP.I, TP en TP.A

De tijdinstellingen in de procesbesturing bestaan uit: TP (tijdinstelling), TP.I (Index) en TP.A (lopende tijd). Naast de in de index-tabel gedefinieerde elementen kunnen er per installatie specifieke elementen worden toegevoegd.

Objecten SP.I en SP

De schakelaars in de procesbesturing bestaan uit: SP (schakelaar instellingen) en SP.I (Index). Naast de in de index-tabel gedefinieerde elementen kunnen er per installatie specifieke elementen worden toegevoegd.

NB. Voor de voorwaarden die gelden voor het toevoegen van extra elementen zie "Toevoegen extra elementen".

Objecten EXTRAINFO.I, EXTRAINFO.A en EXTRAINFOEXT

Met behulp van een aantal objecten kan informatie worden ingewonnen over VRI specifieke deelsystemen. Men moet hierbij denken aan systemen die af en toe voorkomen, niet generiek zijn of niet voorzien zijn. Denk aan een weerstation, sluis of een verkeersmeetpunt.

Hiervoor zijn 3 objecten:

EXTRAINFO.A
EXTRAINFO.I
EXTRAINFOEXT

Welke informatie wordt gegeven is vrij, zodat de mogelijkheid er is om in een centrale informatie over een VRI weer te geven zonder dat de software (ontwikkelaar) van de centrale iets hoeft te weten over de op te vragen gegevens.

De gebruiker moet natuurlijk wel weten welke informatie hij ziet. In het object EXTRAINFOEXT staat daarvoor een beschrijving.

De dynamische informatie zelf staat in EXTRAINFO.A. EXTRAINFO.I is het bijbehorende index object.

VRI-commando's

Via het object VRI.C kunnen commando's worden gegeven aan de VRI. De codes vormen een subgroep (5000..5999) in de eventcodes zoals die beschreven worden in de onderstaande paragraaf 14.2.10 "Commando-events".

Ook kunnen resetcommando's gegevens worden via het object VRI.C met een deel van de subgroep (4000..4999) in de eventcodes zoals die in dezelfde paragraaf benoemd worden. Dit laatste is ook omschreven in de paragraaf 16.2.11.

Met behulp van het VRI.A object kan de lijst met actieve storingen worden opgevraagd. De automaat zorgt voor het beheer op deze lijst. Storingen die niet meer actief zijn in de automaat worden dus 'automatisch' verwijderd uit deze lijst, waardoor het aantal indexen in de lijst kan verschillen per uitgevoerde VRI.A commando.

Commando events

Er zijn een aantal commando events om de werking van de VRI te beïnvloeden. Het gaat dan om:

- Testen puts;
- Gewenste VRI-status
- Gewenst programmanummer
- Gewenst subprogrammanummer
- Herstart VRI

Parameterwijziging

Het formaat van een parameterwijziging is dataformaat 101. Parameterwijzigingen worden vastgelegd in PAR.LA en PAR.LB.

Afhankelijk van de versie van de implementatie van IVERA in de VRI en de mogelijkheden van de VRI bevat het event naast de nieuwe waarden van de gewijzigde parameters eventueel ook de oude waarden van die parameters (zie beschrijving formaat 101 in de paragraaf 14.2.12).

Programma events

Vanuit het regelprogramma worden foutmeldingen als programma-event meegegeven. Het gaat dan onder andere om meldingen als:

- VRI status wijziging;
- Programmaomschakeling;
- Fasebewaking;
- Garantietijdonderschrijding;
- Grens van 90% van een of meerdere logboeken bereikt;
- Fout in de seriele koppeling;
- Onder- of bovenspanningsmelding.

14.2 Objecten VRI algemeen (procesbesturing)

14.2.1 TP.I

Het object TP.I geeft de index voor alle timers weer:

Attribuut	Type	Object	Omschrijving
N	1	TP.I	Naam
O	1	Index timers	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMTP	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 14.1 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

TP.I	
GEELKNIP	Tijd geelknippen
VASTGEEL	Tijd vastgeel
ALLESROOD	Alles roodtijd

Tabel 14.2 Tijdinstellingen procesbesturing

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TP.I	TP.I="GEELKNIP","VASTGEEL","ALLESROOD"	Vraag object TP.I op
TP.I/#0	TP.I/#0="GEELKNIP"	Vraag element uit object TP.I weer op ter controle
TP.I/#0="Test"	:E=11	Wijziging van een element is niet toegestaan.

Tabel 14.3 Voorbeelden TP.I

14.2.2 TP

Het object TP geeft de tijd instelling voor alle timers weer:

Attribuut	Type	Object	Omschrijving
N	1	TP	Naam
O	1	Tijd instelling	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMTP	aantal data-elementen
I	1	TP.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2, getal met eenheid 0.1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 14.4 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TP	TP=<waarde 1>, <waarde 2>,...<waarde n>.	Vraag de tijdsinstellingen van object TP op.
TP/#0=243342	:E=16	ERR_DATA het bericht voldoet niet aan het IVERA bericht formaat.
TP/#0=100	:E=11	ERR_USER de master heeft geen autorisatie om het object te schrijven.

Tabel 14.5 Voorbeelden TP

14.2.3 TP.A

Het object TP.A geeft de lopende tijd voor alle timers weer:

Attribuut	Type	Object	Omschrijving
N	1	TP.A	Naam
O	1	Lopende tijd	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMTP	aantal data-elementen
I	1	TP.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2, getal met eenheid 0.1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 14.6 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TP.A:A	TP.A:A="N=TP.A,T=0,E=<NUMTP>,U=4444,I=TP.I,MIN=0,S=1,F=2,O='Lopende tijd (proces)'"	Vraag de attributen van het object TP.A op
TP.A/#0	TP.A/#0=<waarde_1>	Vraag een data-element van het object TP.A op
TP.A/#0=<nieuwe_waarde>	:E=11	Wijzig de waarde met een andere geldige waarde. Dit is niet toegestaan.

Tabel 14.7 Voorbeelden TP.A

14.2.4 SP.I

Het object SP.I geeft de index voor alle schakelaars weer:

Attribuut	Type	Object	Omschrijving
N	1	SP.I	Naam
O	1	Index schakelaars	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMSP	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 14.8 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

SP.I	
KLCON	Klok continue (0=via klok, 1=continue)
KLEEN	Klok overbrugging eenmalig
DPBRT	Detectiebewaking in procesbesturing

Tabel 14.9 Schakelaars procesbesturing

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
SP.I	SP.I="KLCON","KLEEN","DPBRT"	Vraag de elementen van object SP.I op
SP.I/#0	SP.I/#0="KLCON"	Vraag het eerste SP.I-element op.
SP.I/#<aantal	:E=12	Vraag ongeldig element bereik op

elementen +1>		
---------------	--	--

Tabel 14.10 Voorbeelden SP.I

14.2.5 SP

Het object SP geeft de schakelaar toestand voor alle schakelaars weer:

Attribuut	Type	Object	Omschrijving
N	1	SP	Naam
O	1	schakelaar	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMSP	aantal data-elementen
I	1	SP.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	1	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	10, schakelaar (uit/aan)	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 14.11 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
SP/#0	SP/#0=<waarde>	Vraag het object SP op
SP:E	SP:E=<aantal sp>	Vraag aantal elementen op, het object bevat minimaal drie data-elementen
SP/<naam>=<waarde2>	:E=11	Voer een waarde in binnen de gespecificeerde range voor het object SP, bij een te laag inlogniveau

Tabel 14.12 Voorbeelden SP

14.2.6 EXTRAINFO.I

Het object EXTRAINFO.I geeft de index extra info weer:

Attribuut	Type	Object	Omschrijving
N	1	EXTRAINFO.I	Naam
O	1	index extra info	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMINFO	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 14.13 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
EXTRAINFO.I	Aantal waarden is afhankelijk van configuratie. EXTRAINFO.I="<indexnaam1>","<indexnaam2>", ...	Vraag EXTRAINFO.I op

Tabel 14.14 Voorbeelden EXTRAINFO.I

14.2.7 EXTRAINFO.A

Het object EXTRAINFO.A geeft de informatieve actuele extra info string weer:

Attribuut	Type	Object	Omschrijving
N	1	EXTRAINFO.A	Naam
O	1	informatieve actuele extra info string	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMINFO	aantal data-elementen
I	1	EXTRAINFO.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	60	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, ruwe tekst	Data-element formaat
S	0		Data-element stapgrootte

Tabel 14.15 Object attributen

De objecten EXTRAINFO.A en EXTRAINFOEXT hebben formaat 1. Dit formaat ondersteunt niet het gebruik van DubbelQuote (ASCII 34).

Om dit soort tekens toch te kunnen gebruiken, gebruiken we een codering overgenomen uit HTML. (zie: http://www.w3.org/TR/html4/charset.html#idx-character_encoding-4 §5.3.1)

```
Extendedstring={AsciiString | ExtendedKar }
ExtendedKar = &#+extendedAscii+;
ExtendedAscii = 32..255 /* code van een bijzonder teken */
```

Het resultaat van een Extendedstring blijft een geldige string volgens formaat 1.

Voorbeeld:

Omschrijving:

Een VRI heeft een weerstation en houdt de capaciteit gegevens van de hoofdrijbaan bij.

De inhoud van de objecten kan er dan als volgt uitzien:

```
EXTRAINFO.I/#0 = "OMGTEMP"
EXTRAINFO.I/#1 = "WINDSTERKTE"
EXTRAINFO.I/#2 = "INTENSITEIT_FC02"
EXTRAINFO.I/#3 = "INTENSITEIT_FC08"
```

```
EXTRAINFOEXT/#0="Omgevingstemperatuur gemeten bij VRI in &#248C"
EXTRAINFOEXT/#1="Windsterkte en richting gemeten op 3m hoogte"
EXTRAINFOEXT/#2="Aantal voertuigen per uur&#44; rijsnelheid N299 N-richting"
EXTRAINFOEXT/#3="Aantal voertuigen per uur&#44; rijsnelheid N299 Z-richting"
```

```
EXTRAINFO.A/#0="22&#248;C"
EXTRAINFO.A/#1="ZW3"
EXTRAINFO.A/#2="2412 VTG/UUR gem. 61 km/h"
EXTRAINFO.A/#3="812 VTG/UUR gem. 93 km/h"
```

44 is de ASCII code voor het komma teken 248 is de (extended) ASCII code voor het graden teken. Het komma teken mag ook gewoon in de string staan.

De master kan deze objecten uitlezen en in zijn user interface tonen aan de gebruiker. De master dient uiteraard de extended string te vertalen naar de gewenste weergave.

In dit geval kan dit er als volgt uitzien:

22°C	Omgevingstemperatuur gemeten bij VRI in °C
ZW3	Windsterkte en richting gemeten op 3m hoogte
2412 VTG/UUR gem. 61 km/h	Aantal voertuigen per uur, rijsnelheid N299 N-richting
812 VTG/UUR gem. 93 km/h	Aantal voertuigen per uur, rijsnelheid N299 Z-richting

14.2.8 EXTRAINFOEXT

Het object EXTRAINFOEXT geeft de informatieve toelichting extra info string weer:

Attribuut	Type	Object	Omschrijving
N	1	EXTRAINFOEXT	Naam
O	1	informatieve toelichting extra info string	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMINFO	aantal data-elementen
I	1	EXTRAINFO.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	60	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, ruwe tekst	Data-element formaat
S	0		Data-element stapgrootte

Tabel 14.16 Object attributen

Zie ook het object EXTRAINFO.A in de voorgaande paragraaf.

14.2.9 VRI.C

Het object VRI.C geeft het VRI-commando weer:

Attribuut	Type	Object	Omschrijving
N	1	VRI.C	Naam
O	1	VRI-commando	Omschrijving
T	1	0	Type
U	0	6664	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 14.17 Object attributen

Voor de command's wordt verwezen naar paragraaf 14.1, VRI-command's.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRI.C/#0=4001	VRI.C/#0=4001	Reset alle storingen
VRI.C:A	VRI.C:A="T=0,E=1,U=6664, MIN=0,S=1,F=1,N=VRI.C,T=0 ,F=1,L=0,U=6664,E=1,I=<Ind ex verwijzing per dimensie>,IMIN=<Index data- element minimumwaarde>,IMAX=<Ind ex data-element maximumwaarde>,S=1,MIN= 0,MAX=999999,O='VRI- commando'"	Opvragen van VRI.C object attributen
VRI.C:E	VRI.C:E=1	Opvragen van elementen in VRI.C object
VRI.C/#1	:E=12	Opvragen van elementen in VRI.C object
VRI.C:MIN	VRI.C:MIN=0	Opvragen van attribuut in VRI.C object
VRI.C:MIN=10	VRI.C:MIN=10	Wijzigen van attribuut uit VRI.C object
VRI.C:MINA	:E=19	Opvragen van ongeldig attribuut
VRI.C:MIN="Hallo"	:E=16	Wijzigen van attribuut uit VRI.C object naar een ongeldige waarde
VRI.C:MAX	VRI.C:MAX=<max_waarde>	Opvragen van attribuut in VRI.C object
VRI.C:MIN=<1000 000>	:E=16	Wijzigen van attribuut naar een ongeldige waarde

Tabel 14.18 Voorbeelden VRI.C

14.2.10 Commando-events

Commando-events hebben betrekking op commando's in de VRI. Zo wordt bijvoorbeeld het commando test puts gemeld als eventcode 5001. Daarin tegen is het ook mogelijk om de puts te testen door het schrijven van de waarde 5001 naar het object "VRI.C".

NB. Indien een VRI een commando niet ondersteunt, antwoordt de VRI met de foutcode ERR_DATA.

Code	Omschrijving
5001	Test putsarmatuur
5022	Geeft de VRI toestemming om naar lokaal bedrijf te gaan. Er wordt niet meer geluisterd naar de programmawens van de centrale maar van de lokale bediening / weekautomaat.
5023	Opheffen toestemming lokaal bedrijf. De VRI luistert alleen naar de wens van de programmawens van de centrale. Deze wens is vastgelegd in elementnr. 2 van resp. VRISTAT en VRIPROG. Afhankelijk van de implementatie in de VRI wordt hier al of niet gehoor aan gegeven.
5100..5199	Gewenste VRI-status vanuit centrale
5200..5299	Gewenste programmanummer vanuit centrale
5300..5399	Gewenste subprogrammanummer vanuit centrale
5990	Geeft VRI opdracht een warme herstart uit te voeren
9990	Geeft VRI opdracht een warme herstart uit te voeren (verouderd)

Tabel 14.19 Commando events

Gewenste VRI-status

De 5100 events corresponderen met de nummers in tabel. 3.4. (status) en tabel. 6.5. (bron). Een 5100 event kan met en zonder detailinformatie voorkomen.

Een 5100 event is gespecificeerd als: 5100 + <status>, <bron> waarbij de bronvermelding optioneel is.

De VRI-programmastatus vanuit de centrale kan worden geselecteerd door het commando: *5100 + gewenste status.*

Bijvoorbeeld:

- 5100; Automatisch bedrijf.
- 5110; Regelen,

Gewenst programmanummer

De 5200 events corresponderen met de nummers in tabel 6.6. (bron). Een 5200 event kan met en zonder detailinformatie voorkomen.

Een 5200 event is gespecificeerd als: 5200 + <programmanummer>, <bron> waarbij de bronvermelding optioneel is.

Het gewenste programmanummer vanuit de centrale kan worden gezet door het commando: *5200 + gewenste programma.*

Bijvoorbeeld:

- 5200; geen programmakeuze.
- 5205; programma 5.

NB. Zie functionele beschrijving.

Gewenst subprogrammanummer

De 5300 events corresponderen met de nummers in tabel 6.6. (bron). Een 5300 event kan met en zonder detailinformatie voorkomen.

Een 5300 event is gespecificeerd als: 5300 + <programmanummer>, [<bron>] waarbij de bronvermelding optioneel is.

Het gewenste programmanummer vanuit de centrale kan worden gezet door het commando: *5300 + gewenste programma.*

Bijvoorbeeld:

- 5300; geen subprogrammakeuze.
- 5305; subprogramma 5.

Herstart VRI

Het 5990 event heeft een onmiddellijke warme herstart van de VRI tot gevolg, mits de toestand van de VRI dit toestaat.

De herstart wordt alleen uitgevoerd als de waarde van VRISTAT zodanig is, dat een herstart zonder risico voor de verkeersafwikkeling kan plaatsvinden. De toegestane waarden zijn 1 (Buiten bedrijf), 2 (Gedoofd), 3 (Knipperen extern), 4 (Knipperen) en 5 (Knipperen tijdens inschakelen).

Wanneer de VRI zich in een andere toestand dan de toegestane bevindt, kan een herstart slechts bereikt worden, door de VRI door een commando in de een geschikte toestand te brengen en vervolgens de opdracht tot herstarten te geven.

Een 5990 event kan alleen zonder detailinformatie voorkomen.

NB: Er bestaan implementaties van IVERA versie 1.30 die dit event ondersteunen als commando event 9990. Die variant moet als verouderd worden beschouwd, aangezien ze buiten de 5000 range valt die voor commando events is gereserveerd. Een implementatie van een centrale kan er echter voor kiezen om beide varianten te ondersteunen voor maximale compatibiliteit.

NB: Bij het uitlezen van het object "VRI.C" is het resultaat ongedefinieerd.

14.2.11 VRI.A

Het object VRI.A geeft de actieve storingslijst weer:

Attribuut	Type	Object	Omschrijving
N	1	VRI.A	Naam
O	1	Actieve storingslijst	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	0 .. 150	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	100	Data-element formaat
S	0		Data-element stapgrootte

Tabel 14.20 Object attributen

Formaat: Event (100), type 1

Een eventobject bevat een event in de slave in een leesbare ASCII string. Een event bestaat uit een timestamp, een eventcode en een nadere detaillering van het event in tekst. De detailinformatie is afhankelijk van de eventcode.

```
EventBericht = TimeStamp + "," + Bevestigd + "," + EventCode + "," + DetailInfo
TimeStamp = "jjjjmmdd:uummss"
Bevestigd = 0 | 1
EventCode = IntegerWaarde
DetailInfo = Detailstring
```

Met behulp van het IVERA VRI.A object kan de lijst met actieve storingen worden opgevraagd. De automaat zorgt voor het beheer op deze lijst. Storingen die niet meer actief zijn in de automaat worden dus 'automatisch' verwijderd uit deze lijst, waardoor het aantal indexen in de lijst kan verschillen per uitgevoerd VRI.A commando.

De response lijst is altijd chronologisch gesorteerd op Timestamp (recentste eerst). Timestamp is het moment van optreden van de storing.

NB. Een EventCode kan meerdere keren in de lijst voorkomen, in de DetailInfo is terug te vinden over welke storing het gaat. Bijvoorbeeld een detector of signaalgroep in storing.

NB. Herstelmeldingen worden per definitie niet in deze lijst gezet, omdat dit type aangeeft dat de storing is verholpen.

Eventbuffer

Iedere event heeft een unieke code. De volgende tabel geeft een overzicht van verschillende eventcategorieën en de daarvoor gereserveerde codes.

Eventcode	Omschrijving
1000..1999	I/O-events
2000..2999	Programma-events/fouten
3000..3999	Bewakerevents/fouten
4000..4999	Resetevents
5000..5999	Commando-events
6000..6099	Datacommunicatie-events
100000..199999	Automaatspecifieke events.
200000..299999	Applicatiespecifieke events.

Tabel 14.21 Eventcategorieën

Per event is er de mogelijkheid tot detailinformatie. Per eventcode is het formaat van de detailinformatie weergegeven. In het geval dat de detailinformatie niet beschikbaar is, hoeven de argumenten niet te worden ingevuld. Zo kan bijvoorbeeld een lampfout worden gemeld als:

Voorbeeld	Omschrijving
19970117:150023,0,1010,SG02,R,1,1,2,0	SG02, rood, lantaarn=1, laag, aantal defecte rode lampen=2, niet laatste
19970117:141854,0,1010,1	Lampfout nummer 1
19970117:170127,0,1010,SG02,,,1	SG02, laag

Tabel 14.22 Voorbeeld van lampfoutevents

I/O-events

In de categorie I/O-events valt de informatie die inzicht verschaft in de aard van een opgetreden storing in I/O van een VRI. De storingen zijn verdeeld in de volgende groepen:

- Lampfouten
- Detectiefouten

Lampfouten

De rapportering van lampfouten is sterk afhankelijk van de mogelijkheden van de VRI. De ideale VRI zou de eerste en laatste lampfout per kleur/categorie/signaalgroep moeten kunnen detecteren. In het geval dat een lampfout is hersteld, moet automatisch een herstelmelding volgen.

Code	Omschrijving	Detailinfo	VRI.A
1010	Lampfout	<SgRef>,<Kleur>,<Lref>,<Cat>,<ALF>,<LRF> [,<STEND>]	Ja

Tabel 14.23 Lampfoutevent

SgRef	Signaalgroepnaam of -nummer	
Kleur	Rood/geel/groen	R A G
Lref	Lantaarnnummer	0..9
Cat	Categorie	0..3
ALF	Aantal lampfouten voor de SgRef/Kleur combinatie.	0..99
LRF	Laatste rode-lampfout	0 1
STEND	Onbekend, start of einde	0 1 2

Categorie	Omschrijving
0	Onbekend
1	Laag
2	Hoog
3	Onderlicht

Tabel 14.24 Lamp categorieën

NB. Op het moment dat een lamp wordt vervangen, terwijl de automaat in bedrijf is, wordt een herstel melding verstuurd op het moment dat alle lampen van de signaalgroep zijn hersteld.

NB. Een bericht met ALF=0 is een herstel melding.

NB. Een lampfout kan in de meeste automaten alleen gedetecteerd worden in de aan toestand.

NB: Het invullen van het veld <STEND> is optioneel. Als bekend is, of het om het begin of het einde van een fouttoestand gaat, kan dat in <STEND> aangegeven worden.

NB: De kolom 'VRI.A' geeft aan per event of deze bij actief of inactief zijn in de automaat, zal worden getoond in respectievelijk zal worden verwijderd uit de 'Actieve storingenlijst'. Indien dit event inactief wordt gemaakt door een ander event, is dit expliciet aangegeven met 'eventcode'.

Voorbeeld 1:

Verstuurd commando:

VRI.A

Response bericht:

VRI.A="bericht 5","bericht 4","bericht 3","bericht 2","bericht 1"

NB. Waarbij bericht 5 verondersteld wordt het jongste bericht te zijn en bericht 1 het oudste bericht.

Voorbeeld 2:

Verstuurd commando:

VRI.A

Response bericht:

VRI.A="bericht 5","bericht 4","bericht 2","bericht 1"

NB. Waarbij bericht 3 niet meer actief is en uit de lijst is verwijderd.

Voorbeeld 3 :

In onderstaande voorbeeld staan een aantal events zoals deze in een automaat kunnen voorkomen in de actuele stringenlijst.

Verstuurd commando:

VRI.A

Response bericht:

VRI.A="20121212:121212,0,1010,SG02,R,1,1,2,0","20121212:110518,0,4016,1",20121212:101601,0,1010,1

Vervolgens wordt het event 4016 'Noodstroomvoedingbericht' opgeheven en daarom uit de actuele stringenlijst gehaald.

Verstuurd commando:

VRI.A

Response bericht:

VRI.A="20121212:121212,0,1010,SG02,R,1,1,2,0","20121212:101601,0,1010,1"

14.2.12 PAR.LB

Het object PAR.LB geeft het parameterlogboek weer:

Attribuut	Type	Object	Omschrijving
N	1	PAR.LB	Naam
O	1	Parameterlogboek	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	0 .. 1000	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	101	Data-element formaat
S	0		Data-element stapgrootte

Tabel 14.25 Object attributen

Formaat: Parameterevent (101), type 1

Een parametereventobject bevat een parameterwijziging in de slave in een leesbare ASCII string. Wijzigingen van een parameter, door welke partij dan ook, leiden tot een parameterevent. Parameters welke als read-only (zie ITYPE) zijn gedefinieerd worden niet als event gemeld. De uitvoering van opdrachten tot parameterwijziging waarbij de nieuwe waarde gelijk is aan de oude waarde, worden niet als event gemeld.

Wanneer meer dan één parameter door een dergelijke opdracht wordt gewijzigd, leidt elk van de wijzigingen van een parameter tot een apart event. Een event bestaat uit een timestamp, een parameternaam (objectnaam), een element en de nieuwe waarde. De timestamp heeft een jaaraanduiding met 4 cijfers. Optioneel kan aan het event de oude waarde toegevoegd worden.

```
ParameterEventBericht = TimeStamp + "," + Bevestigd + "," + ObjectNaam + "/" + Element + [" + Element] + "=" + NieuweWaarde + [" + OudeWaarde ]
TimeStamp = "jjjjmdd:uummss"
Bevestigd = 0 | 1
ObjectNaam = /* Zie BNF definitie in IVERA functionele specificatie */
NieuweWaarde = AsciiString
OudeWaarde = AsciiString
AsciiString = /* Zie BNF definitie in IVERA functionele specificatie */
```

NB: Als de waarde van de parameter een string is, komt deze dus zonder omringende dubbelquotes in het parametereventbericht; dit ter voorkoming van geneste dubbelquotes in dat bericht.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
PAR.LB:E	PAR.LB:E=<aantal elementen>	Vraag totale logboek op
PAR.LB/#<aantal-1>	PAR.LB/#<aantal-1>= TimeStamp + "," + bevestigd + "," + ObjectNaam + [":" + AttribuutNaam] + "/" + Element + "=" + NieuweWaarde + ["," + OudeWaarde]	Vraag laatste entry op
PAR.LB:O	PAR.LB:O='Parameterlogboek'	Vraag een attribuut op van het object PAR.LB

Tabel 14.26 Voorbeelden PAR.LB

14.2.13 PAR.LA

Het object PAR.LA geeft het parameterlogboek (onbevestigd) weer:

Attribuut	Type	Object	Omschrijving
N	1	PAR.LA	Naam
O	1	Parameterlogboek (onbevestigd)	Omschrijving
T	1	1	Type
U	0	6666	User Identificatie Control
E	0	0 .. 1000	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	101	Data-element formaat
S	0		Data-element staggrootte

Tabel 14.27 Object attributen

Zie de paragraaf 14.2.12 voor de formaat omschrijving.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
PAR.LA/#0	De VRI geeft een <ParameterEventBericht> terug. Zie referentie.	Vraag de waarde van het eerste onbevestigde logboekobject op.
PAR.LA/*=""		Maak het object waar waarde <ParameterEventBericht> uit gehaald is leeg.
PAR.LA/#0	:E=17	Vraag nogmaals de waarde van het eerste onbevestigde logboekobject op, na de wijziging.

Tabel 14.28 Voorbeelden PAR.LA

14.2.14 Programma-events

De programma-events omvatten algemene events en storings in de VRI.

Code	Omschrijving	Detailinfo	VRI.A
2000	Programma event	<IdNummer>	
2001	VRI status wijziging	<IdNummer>	
2002	Programmaomschakeling	<IdNummer>	
2003	Brugingreep	<IdNummer>,<BeginEinde>	
2004	Brandweeringreep	<IdNummer >,<BeginEinde>	
2005	AHOB melding	<IdNummer>,<BeginEinde>	
2500	Fasebewaking		
2501	GUS-WUS fouten CVN C-interface	<SgRef>	
2502	Rekentijdproblemen		
2503	Garantietijdonderschrijding	<SgRef>,<ExternSGToestand>,<Tijd>	
2504	Maximumtijdoverschrijding	<SgRef>,<ExternSGToestand>,<Tijd>	
2505	Start niet kunnen regelen door storing		Ja
2506	Einde niet kunnen regelen door storing		
2510	Overig Logboek 90% vol grens bereikt.	<Obj>	
2511	VRI.LA Logboek 90% vol grens bereikt.		
2512	PAR.LA Logboek 90% vol grens bereikt.		
2513	OV.LA Logboek 90% vol grens bereikt.		
2600	Seriële koppeling - ontbreken levenssignaal.	<IdNummer>	Ja
2601	Seriële koppeling - geen communicatie.	<IdNummer>	Ja
2700	Onderspanningsmelding		Ja
2701	Bovenspanningsmelding		Ja
2702	Telefoonnummer centrale kwijt		Ja

Tabel 14.29 Programma events

SgRef	Signaalgroepnaam of -nummer
IdNummer	Uniek identificatienummer van het element in het regeltoestel.
BeginEinde	Start/Einde indicatie (start =0, einde=1)
ExternSGToestand	Zie tabel 4.1.4.
Tijd	Tijd in 1/10 seconde eenheden.
Obj	objectnaam

NB. De automaat levert een foutcode en mogelijk een verklarende tekst. De te nemen actie is vastgelegd in de centrale.

NB: De kolom 'VRI.A' geeft aan per event of deze bij actief of inactief zijn in de automaat, zal worden getoond in respectievelijk zal worden verwijderd uit de 'Actieve storingslijst'. Indien dit event inactief wordt gemaakt door een ander event, is dit expliciet aangegeven met 'eventcode'.

Toelichting bij event 2510: Overig Logboek 90% vol grens bereikt:

Dit event is voornamelijk bedoeld om het verlies van events te voorkomen bij het vollopen van een logboek. Indien event 2510 wordt opgenomen als triggerevent in het object datacom zal bij het bereiken van de 90% grens een triggerevent 2510 worden verstuurd naar de centrale. De centrale dient er vervolgens voor te zorgen dat de logboeken worden gelezen. Hierbij gelden de volgende uitgangspunten:

- De trigger treed alleen op indien de 90% vanaf een lagere waarde wordt bereikt. (Dus bij de overgang van 89% naar 90%)
- Is de trigger eenmaal opgetreden voor een object zal deze pas nogmaals optreden als het betreffende object (logboek) eerst voldoende leeg geweest is. Richtwaarde hiervoor is 50%.
- Als detailinfo wordt de objectnaam van het betreffende logboek meegegeven. Dit kan bijvoorbeeld D.LA zijn maar ook een XLOG.LA is mogelijk.

Toelichting bij event 2511, 2512 en 2513. Events zijn gelijk aan event 2510 maar dan voor een specifiek logboek. Voor een logboek zal echter maar 1 event optreden. Als event 2511 optreed omdat VRI.LA voor 90% vol is zal niet ook event 2510 daardoor optreden. Events

2511, 2512 en 2513 hebben geen detailinfo omdat het object bekend is vanwege de koppeling met het eventnummer.

Autonome bewaker

Events van de autonome bewaker worden gemeld, indien de installatie daartoe instaat is.

Code	Omschrijving	Detailinfo	VRI.A
3000	Algemeen bewakerevent	Nummer	Ja
3001	Conflict	<SgRef>,<SgRef>,<ConflictType>,<Tijd>	Ja
3002	Lampfout	<IdNummer>	Ja
3003	Meer dan 1 kleur	<SgRef>	Ja
3004	Geelknipperfout	<SgRef>	Ja
3005	Garantietijdonderschrijding	<SgRef>,<ExternSGToestand>,<Tijd>	Ja
3006	Maximumtijdoverschrijding	<SgRef>,<ExternSGToestand>,<Tijd>	Ja
3007	Fout in eindschakelaar	<IdNummer>	Ja
3008	Witknipperfout	<SgRef>	Ja
3009	Halfconflict OV	<SgRef>,<SgRef>	Ja
3010	Volgordebewaking	<SgRef>,<ExternSGToestand>,<ExternSGToestand>	Ja

Tabel 14.30 Bewaker events

SgRef	Signaalgroepnaam of -nummer
IdNummer	Uniek identificatienummer van het element in het regeltoestel.
ConflictType	0 = ontruimingstijdonderschrijding 1 = conflict algemeen 2 = groen/groen-conflict 3 = groen/geel-conflict 4 = geel/geel-conflict
ExternSGToestand	Zie tabel 4.1.4.
Tijd	Werkelijk gerealiseerde tijd in 1/10 seconde eenheden.

NB. Bij melding 3001 is <Tijd> alleen van belang bij conflicttype 0. Conflict type 1 kan worden gebruikt, indien er geen onderscheid gemaakt kan worden tussen conflicttypes 2, 3 en 4.

NB: De kolom 'VRI.A' geeft aan per event of deze bij actief of inactief zijn in de automaat, zal worden getoond in respectievelijk zal worden verwijderd uit de 'Actieve storingslijst'. Indien dit event inactief wordt gemaakt door een ander event, is dit expliciet aangegeven met 'eventcode'.

14.2.15 Commando's en events

Parameterevent (101)

Een parametereventobject bevat een parameterwijziging in de slave in een leesbare ASCII string.

Wijzigingen van een parameter, door welke partij dan ook, leiden tot een parameterevent. Parameters welke als read-only (zie ITYPE) zijn gedefinieerd worden niet als event gemeld. De uitvoering van opdrachten tot parameterwijziging waarbij de nieuwe waarde gelijk is aan de oude waarde, worden niet als event gemeld.

Wanneer meer dan één parameter door een dergelijke opdracht wordt gewijzigd, leidt elk van de wijzigingen van een parameter tot een apart event.

Een event bestaat uit een timestamp, een parameternaam (objectnaam), een element en de nieuwe waarde. De timestamp heeft een jaaraanduiding met 4 cijfers.

Optioneel kan aan het event de oude waarde toegevoegd worden.

```
ParameterEventBericht = TimeStamp + "," + Bevestigd + "," + ObjectNaam + "/" + Element + ["," + Element] + "=" + NieuweWaarde + ["," + OudeWaarde ]
TimeStamp = "jjjjmmdd:uummss"
Bevestigd = 0 | 1
ObjectNaam = /* Zie BNF definitie in IVERA functionele specificatie */
NieuweWaarde = AsciiString
OudeWaarde = AsciiString
AsciiString = /* Zie BNF definitie in IVERA functionele specificatie */
```

NB: Als de waarde van de parameter een string is, komt deze dus zonder omringende dubbelquotes in het parametereventbericht; dit ter voorkoming van geneste dubbelquotes in dat bericht.

15. Protocol

15.1 Algemeen protocol

Voor de protocollen zijn de volgende objecten gedefinieerd:

Object	Omschrijving
ERROR.CODE	Foutcode
ERROR.INFO	Gedetailleerde beschrijving
ERROR.CMD	Het commando waar de error bij hoort
ABON	Abonnementsverzoek
BB0	Objectlijst type 0
BB1	Objectlijst type 1
BBA0	Objectlijst + attributen type 0
BBA1	Objectlijst + attributen type 1
PING	Ping-commando

Tabel 15.1 Objecten protocol

De objecten ERROR.CODE, ERROR.INFO en ERROR.CMD

Het ERROR object kan worden gebruikt om van de laatste NUMERR fouten aanvullende informatie op te vragen. Op index 0 bevindt zich de meest recente fout, op NUMERR-1 bevindt zich de oudste fout. Een typische waarde voor NUMERR ligt tussen 5 en 10. Per sessie wordt een error object bijgehouden. Bij het uitloggen worden de errorobjecten geleege. Bij het wijzigen van het login-niveau blijven de errorobjecten gehandhaafd.

Het object ABON

Met behulp van het IVERA push mechanisme kan een IVERA cliënt zich abonneren op wijzigingen in een object of element bereik van een object. Voor het abonneren kan gebruik worden gemaakt van het ABON bericht. Als data wordt de functionele naam van het object of het bereik van elementen (hierbij worden alle IVERA notatiewijzen om een bereik aan te duiden geaccepteerd) waarop geabonneerd dient te worden verstuurd.

Indien er geen data met het ABON bericht wordt meegestuurd (dus puur het commando 'ABON' wordt uitgevoerd) dan worden de actuele abonnementen opgevraagd.

Abonnementen zijn alleen geldig voor de huidige IVERA sessie. Bij het uitloggen worden alle abonnementen opgezegd. Bij het wijzigen van het login-niveau blijven de abonnementen gehandhaafd.

Er kan alleen een abonnement genomen worden op alle in de automaat ondersteunde objecten en element bereiken.

De objecten BB0, BB1, BBA0 en BBA1

Met de objecten BBx is inzicht te krijgen van welk type het formaat is. Met de objecten BBAX ontstaat aanvullend op de objecten BBx de weergave van de toegepaste attributen.

Het object PING

Dit object is bedoeld voor het testen van de communicatie tussen de beheerscentrale en het regeltoestel.

15.2 Objecten protocol

15.2.1 ERROR.CODE

Het object ERROR.CODE geeft de foutcode weer:

Attribuut	Type	Object	Omschrijving
N	1	ERROR.CODE	Naam
O	1	Foutcode	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMMERR	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0		Data-element stapgrootte

Tabel 15.2 Object attributen

De ERROR.CODE is initieel gelijk aan: -1.

Een voorbeeld is opgenomen in de paragraaf 15.2.3.

15.2.2 ERROR.INFO

Het object ERROR.INFO geeft de gedetailleerde beschrijving weer:

Attribuut	Type	Object	Omschrijving
N	1	ERROR.INFO	Naam
O	1	gedetailleerde beschrijving	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMMERR	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, ruwe tekst	Data-element formaat
S	0		Data-element stapgrootte

Tabel 15.3 Object attributen

De ERROR.INFO is initieel gelijk aan: "". Het " karakter (dubbele quote) wordt in de terugmelding vervangen door twee keer een ' karakter (enkele quote).

Een voorbeeld is opgenomen in de paragraaf 15.2.3.

15.2.3 ERROR.CMD

Het object ERROR.CMD geeft het commando waar de error bij hoort weer:

Attribuut	Type	Object	Omschrijving
N	1	ERROR.CMD	Naam
O	1	het commando waar de error bij hoort	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMMERR	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, ruwe tekst	Data-element formaat
S	0		Data-element stapgrootte

Tabel 15.4 Object attributen

Het commando waar de error bij hoort, op te vragen via ERROR.CMD, is initieel gelijk aan: “”. Het “ karakter (dubbele quote) wordt in de terugmelding vervangen door twee keer een ‘ karakter (enkele quote).

Voorbeeld:

```
TGL/SG02=9
:E=16
ERROR.CODE/#0
ERROR.CODE/#0=16
ERROR.INFO/#0
ERROR.INFO/#0="Waarde buiten bereik. Verwacht [3, 6]; Ontvangen:9"
ERROR.CMD/#0
ERROR.CMD/#0="TGL/SG02=9"
ERROR.CMD
ERROR.CMD="TGL/SG02=9" ,"DATACOM/#0-#1=' '112A' ' ,''10.1.1.' ' ,...
```

Richtlijn voor de foutmelding.

Het verdient aanbeveling om bij de controle op de juistheid van een commando een duidelijke formulering in de foutmelding te zetten en waar mogelijk een hint te geven voor een juist commando. Als bijvoorbeeld op boven- en ondergrens wordt getest dan niet alleen melden dat de waarde buiten het bereik ligt maar ook de grenzen aangeven waarop getest werd.

15.2.4 ABON

Het object ABON geeft het abonnementsverzoek weer:

Attribuut	Type	Object	Omschrijving
N	1	ABON	Naam
O	1	abonnementsverzoek	Omschrijving
T	1	1	Type
U	0	6666	User Identificatie Control
E	0	NUMABON	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	300	Data-element formaat
S	0		Data-element stapgrootte

Tabel 15.5 Object attributen

Formaat: Abonnement (300), type 1

Dataformaat=ObjectRef

ObjectRef = zie functionele IVERA specificatie, zie hoofdstuk 3.8

Het schrijven naar objecten binnen een abonnement (<ObjectRef>=<ArgumentLijst>) is niet toegestaan.

Wanneer een niet geldig object/element bereik wordt meegegeven wordt :E=16 als resultaat gegeven. Het object wordt niet opgenomen in de abonnementslijst. Als het een abonnement op een leeg object betreft wordt het abonnement geaccepteerd (zie voorbeeld VRI.LA). Het meegeven van een toekenning (abon/#0="TGL/#0=20") is niet toegestaan en geeft :E=16 als resultaat.

Voorbeeld 1:

Omschrijving:

Middels het onderstaande bericht abonneert de cliënt zich op alle geeltijd wijzigingen.

Verstuurd bericht:

ABON/#0="TGL"

Direct verkregen antwoord :

ABON/#0="TGL"

TGL=30,30,30,20,20,20,20,30

Verkregen antwoord na wijziging geeltijd:

TGL=40,30,30,20,20,20,20,30

Voorbeeld 2:

Omschrijving:

Middels het onderstaande bericht abonneert de cliënt zich op wijzigingen van signaalgroep statussen van signaalgroepen 02 t/m 08 en van de detectie statussen van detectoren DF0201 t/m DF0211.

Verstuurd bericht:

ABON/#1-#2="SGE.A/02-08","DA./DF0201-DF0211"

Direct verkregen antwoord :

ABON/#1-#2="SGE.A/02-08","D.A/DF0201-DF0211"

SGE.A/02-08=1,0

D.A/DF0201-DF0211=1,1

Verkregen antwoord na wijziging van signaalgroep statussen 02 t/m 08:
SGE.A/02-08=0,0

Verkregen antwoord na wijziging detector statussen DF0201 t/m DF0211:
D.A/DF0201-DF0211=1,0

Voorbeeld 3:

Omschrijving:

Indien na uitvoeren van voorbeeld 1 en voorbeeld 2 het commando ABON wordt gegeven, worden de huidige abonnementen opgevraagd.

Verstuurd bericht:

ABON

Verkregen antwoord :

ABON="TGL","SGE.A/02-08","D.A/DF0201-DF0211"," "," "," "," "," "," "," "," "," "," "," "," "," "," "," "," "," "," "," "
NB. Het totaal aantal geretourneerde elementen is gelijk aan NUMABON.

Voorbeeld 4:

Omschrijving:

Middels het onderstaande bericht wordt het 2e abonnement verwijderd.

Verstuurd bericht:

ABON/#1=""

Verkregen antwoord :

ABON/#1=""

Voorbeeld 5:

Omschrijving:

Abonnement op een leeg object zoals een leeg logboek.

Verstuurd bericht:

ABON/#1="VRI.LA"

Verkregen antwoord :

ABON/#1="VRI.LA"

:E=17

15.2.5 BB0

Het object BB0 geeft de objectlijst van type 0 weer:

Attribuut	Type	Object	Omschrijving
N	1	BB0	Naam
O	1	objectlijst type 0	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	1 ..	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, ruwe tekst	Data-element formaat
S	0		Data-element stapgrootte

Tabel 15.6 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
BB0	BB0="<objectnaam1>,<objectnaam2>,...,etc"	De automaat retourneert een lijst van alle objecten van het type 0 (==waarde)
BB0/#0	BB0="<objectnaam1>"	Vraag een element van object BB0 op
BB0:A	BB0:A="N=BB0,O='Objectlijst type 0',T=1,U=4444,F=1"	Vraag attributen van object BB0 op

Tabel 15.7 Voorbeelden BB0

15.2.6 BB1

Het object BB1 geeft de objectlijst van type 1 weer:

Attribuut	Type	Object	Omschrijving
N	1	BB0	Naam
O	1	objectlijst type 1	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	1 ..	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, ruwe tekst	Data-element formaat
S	0		Data-element stapgrootte

Tabel 15.8 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
BB1	BB1="<objectnaam1>,<objectnaam2>,...,etc"	De automaat retourneert een lijst van alle objecten met attributen van het type 1 (==waarde)
BB1/#0	BB1="<objectnaam1>"	Vraag een element van object BB1 op
BB1:A	BB1:A="N=BB1,O='Objectlijst type 1',T=1,U=4444,F=1"	Vraag attributen van object BB1 op

Tabel 15.9 Voorbeelden BB1

15.2.7 BBA0

Het object BBA0 geeft de objectlijst en de attributen van type 0 weer:

Attribuut	Type	Object	Omschrijving
N	1	BBA0	Naam
O	1	objectlijst + attributen type 0	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	1 ..	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, ruwe tekst	Data-element formaat
S	0		Data-element stapgrootte

Tabel 15.10 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
BBA0	BBA0=/"*string met objecten en hun attributen*/"	De automaat retourneert een lijst van alle objecten met attributen van het type 0 (==waarde)
BBA0/#0	BBA0/#0="<objectnaam1 met attributen ervan>"	Vraag een element van object BBA0 op
BBA0:A	BBA0:A="N=BBA0,O='Objectlijst + attributen type 0',T=1,U=4444,F=1"	Vraag attributen van object BBA0 op

Tabel 15.11 Voorbeelden BBA0

15.2.8 BBA1

Het object BBA1 geeft de objectlijst en de attributen van type 1 weer:

Attribuut	Type	Object	Omschrijving
N	1	BBA1	Naam
O	1	objectlijst + attributen type 1	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	1 ..	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, ruwe tekst	Data-element formaat
S	0		Data-element stapgrootte

Tabel 15.12 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
BBA1	BBA1=/"*string met objecten en bijbehorende attributen*/"	De automaat retourneert een lijst van alle objecten met attributen van het type 1 (==tekst)
BBA1:A	BBA1:A="N=BBA1,O='Objectlijst + attributen type 1',T=1,U=4444,F=1"	Vraag attributen van object BBA1 op
BBA1/#0	BBA1/#0="<objectnaam1 met attributen ervan>"	Vraag een element van object BBA1 op

Tabel 15.13 Voorbeelden BBA1

Uitgebreid voorbeeld

In een IVERA-slave met 4 objecten van het type getal (TGL, TGGL, TOR, TGOR) bevatten de objecten BB0 en BBA0 beide 4 data-elementen.

Master	Slave antwoord
BBA0	BBA0="N=TGL,T=0, E=4,L=1,U=6664,I=SG.I,S=1,MIN=2,MAX=10,IMIN=1,OMAX=10,O='Geeltijd'", "N=TGGL,T=0,E=4,L=0,U=4444,I=SG.I,S=1,MIN=0,MAX=10,O='Garantiegeeltijd'", "N=TOR,T=0,E1=4,E2=4,L=1,U=6664,I1=SG.I,I2=SG.I,S=1,MIN=-1,MAX=10,IMIN=1,OMAX=10,O='Ontruimingstijd'", "N=TGOR,T=0,E1=4,E2=4,L=0,U=4444,I1=SG.I,I2=SG.I,S=1,MIN=-1,MAX=10,IMIN=1,OMAX=10,O='Garantieontruimingstijd'"
BB0	BB0="TGL","TGGL","TOR","TGOR"

Tabel 3.8. Voorbeeld van objecten BB0 en BBA0.

15.2.9 PING

Het object PING geeft het ping-commando weer:

Attribuut	Type	Object	Omschrijving
N	1	PING	Naam
O	1	Ping-commando	Omschrijving
T	1	0	Type
U	0	6666	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 15.14 Object attributen

Voorbeeld

Master	Slave
@1#PING/#0=5	@1#:A

Tabel 3.25. Voorbeeld PING

16. VRI status

16.1 Algemeen VRI status

Voor de VRI status zijn de volgende objecten gedefinieerd:

Object	Omschrijving
VRISTAT.I	Index statusbronnen
VRISTAT	Automaat toestand
VRIPROG	Automaat programma
VRISUBPROG	Automaat subprogramma
VRIPROG.I	Index programmabronnen
VRIPROGLYST	Programmalijs
VRIPROGLYSTEXT	Uitgebreide programmalijs
BEDRIJF.I	Index object BEDRIJF
BEDRIJF	Bedrijfs

Tabel 16.1 Objecten VRI status

Waarde VRI-status

De waarde van de VRI-status geeft de werkelijke toestand van de verkeersregelininstallatie weer. De VRI-status is ook bedoeld voor de gewenste toestand van bedieningsorganen, zoals het politiepaneel. De informatie kan worden uitgelezen via het object "VRISTAT". De volgende toestanden zijn beschikbaar:

Buiten bedrijf

De installatie is gedoofd en er is geen spanning op de uitgangen, dwz. het hoofdrelais is afgeschakeld

Gedoofd

De installatie is "softwarematig" gedoofd. Dit betekent dat de lampen gedoofd zijn. Het hoofdrelais kan afhankelijk van het type installatie in deze situatie nog bekrachtigd zijn.

Knippen (extern)

De installatie knippert buiten op straat, echter intern wordt er geregeld. Deze situatie komt alleen voor tijdens het uitvoeren van tests waarbij de afloop van het regelprogramma wordt gecontroleerd. Indien deze toestand wordt gecommandeerd, gaat het regeltoestel over naar de toestand geel knippen. Of het regeltoestel automatisch over gaat naar extern geel knippen en intern regelen, is afhankelijk van het type regeltoestel. In sommige gevallen is nog een 'handmatige' actie nodig om dit te bewerkstelligen

Knippen

De installatie bevindt zich in de toestand knippen. Indien deze toestand wordt gecommandeerd, gaat het regeltoestel over naar de toestand geel knippen.

Inschakelknippen

Tijdens de inschakelprocedure dient de installatie een gedefinieerde tijd te knippen (waarbij de voorwaarschuwing is aangeschakeld). Deze toestand wordt aangeduid met inschakel knippen.

Vastgeel

Vast (of statisch) geel tijdens inschakelprocedure.

Alles rood tijdens inschakelen

Overgangstoestand van vastgeel naar regelen, waarbij alle signaalgroepen rood tonen.

Alles rood

Alle signaalgroepen tonen rood.

Alles rood tijdens uitschakelen

Overgangstoestand van regelen naar knipperen, waarbij alle signaalgroepen rood tonen.

Regelen

Toestand waarin het regelprogramma actief is.

Automatisch bedrijf

Automatisch bedrijf is geen toestand van de installatie. Automatisch bedrijf is de default voor de gewenste VRI status. Een bron die geen invloed wil hebben op de werkelijk VRI status geeft als gewenste VRI status automatisch bedrijf.

Aanwezige programma's

In een VRI kunnen meerdere programma's geladen zijn. Het object VRIPROGLYST bevat een lijst met de aanwezige programma's, waarbij het aantal elementen van het object overeenkomt met het aantal aanwezige programma's (NUMPROG). Ieder element bevat een programmanummer.

Het object VRIPROGLYSTEXT bevat een lijst met de aanwezige (sub)programma's, waarbij het aantal elementen van het object overeenkomt met het aantal aanwezige (sub)programma's (NUMPROGEXT). Ieder element bevat het programma nummer, het subprogrammanummer (indien van toepassing), het type van het programma, de regelstructuur (indien van toepassing), het applicatiepakket en een omschrijving.

Specificaties voor het laden

- Bij het laden wordt er altijd 1 bestand naar de VRI verstuurd. Dit ene bestand bevat alle gegevens welke nodig zijn om een nieuw programma in de VRI te laden (het is dus mogelijk om bijv. een ingepakt bestand te verzenden naar de VRI, alwaar het bestand uitgepakt wordt tot meerdere bestanden).
- Het te laden bestand bevat tevens gegevens (als bijvoorbeeld Type, Regelstructuur en Omschrijving) over het nieuwe programma (zoals genoemd in IVERA Objectdefinitie, Objecttype 1, nr. 60: Programmalijsjt).
- Het uploaden van een programma houdt in, dat slechts en alleen de software van de procesbesturing (incl. OS) en applicatie vervangen mogen worden. De software van de bewakingsunit blijft te allen tijde ongewijzigd.
- Programma's worden geladen in de directory zoals aangegeven in het object FTPLOCATION met index "UPLOAD".
- Tijdens het laden mag de cliënt besluiten het laden te stoppen.
- wanneer tijdens het laden van een programma de TCP-verbinding wegvalt (of wanneer het laden afgebroken wordt), dan wordt dit programma niet opgenomen in het object "VRIPROGLYST". M.a.w. alleen programma's die succesvol door een integriteitstest komen worden zichtbaar in "VRIPROGLYST".
- Naamgeving van te laden programma's voldoet aan de volgende opmaak:
PRxxxxyy.ext
Hierbij is
 - 'xxxx' het kruispuntnummer
 - 'yy' het programmanummer
 - 'ext' de extensie van het bestand (= optioneel); lengte van de extensie is niet gelimiteerd.

Wanneer bij het laden ook de procesbesturing vernieuwd wordt, dan hoeft niet gegarandeerd te worden dat instellingen en logs (als bijvoorbeeld telresultaten) bewaard blijven.

Centraal bedrijf versus lokaal bedrijf

In elke VRI kan worden bepaald of een centrale de regeltoestand (en regelprogramma) van de VRI mag veranderen, of lokaal het bedieningspaneel. Deze keuze, lokaal of centraal, kan tijdens bedrijf worden veranderd. Het veranderen van deze keuze wordt uiteindelijk door de centrale bepaald, maar kan lokaal worden aangevraagd. Er zijn een aantal mogelijkheden:

- Aanvragen lokaal bedrijf aan de centrale;
- Teruggeven van de lokale bediening aan de centrale;
- Gecombineerd bedrijf;

Aanvragen lokaal bedrijf aan de centrale

Als men lokaal de VRI wil bedienen wordt een 'aanvraag lokaal' naar de centrale verstuurd. De centrale kan vervolgens 'toestemming lokaal' geven. Dan is lokale programmakeuze via het bedieningspaneel mogelijk. De aanvraag 'toestemming lokaal' automatisch wordt opgeheven na een time-out van 2 uur geen interactie van de lokale bediener met het bedieningspaneel. Wanneer de automaat in de toestand lokaal bedrijf staat, mag de time-out niet meer optreden. Zodra de centrale 'toestemming lokaal' intrekt gaat de VRI direct naar 'centraal bedrijf' en is lokale bediening niet meer mogelijk. Met het commando `VRI.C/#0 = 5023` overruled de centrale te allen tijde de lokale bediening.

Teruggeven van de lokale bediening aan de centrale

Het is ook mogelijk om lokaal de bediening weer terug te geven aan de centrale, terwijl er nog 'toestemming lokaal' is. Dan komt de VRI opnieuw in 'centraal bedrijf'. Een centrale kan ook vooraf 'toestemming lokaal' geven. Dan is lokale bediening direct mogelijk. Wanneer 'toestemming lokaal' door de centrale vroegtijdig wordt gegeven, blijft de VRI regelen volgens wens van de centrale, totdat de gebruiker bij de VRI de 'aanvraag lokaal' op het bedieningspaneel inschakelt. Omdat de centrale reeds toestemming gegeven heeft, krijgt de gebruiker direct lokaal bedrijf.

Gecombineerd bedrijf

Het is mogelijk om de VRI te configureren in 'gecombineerd bedrijf' door de fabrikant. Is dit het geval, dan kunnen centrale en bedieningspaneel, tegelijkertijd de regeltoestand (en regelprogramma) van de VRI veranderen. De laatste wens geldt en wordt onderling 'gekopieerd'. In gecombineerd bedrijf hebben de commando's 5022 en 5023 geen invloed. De toestand gecombineerd bedrijf blijft actief dus lokale en centrale bediening blijft mogelijk.

De VRI dient ervoor te zorgen dat bij het overschakelen van centraal naar lokaal bedrijf (en andersom), de regeltoestand (en regelprogramma) niet veranderd. Daartoe zal de VRI bij het omschakelen van centraal naar lokaal de wens van het bedieningspaneel eerst gelijk maken aan die van de centrale, en bij het omschakelen van lokaal naar centraal, de centrale wens eerst gelijk maken aan die van het bedieningspaneel.

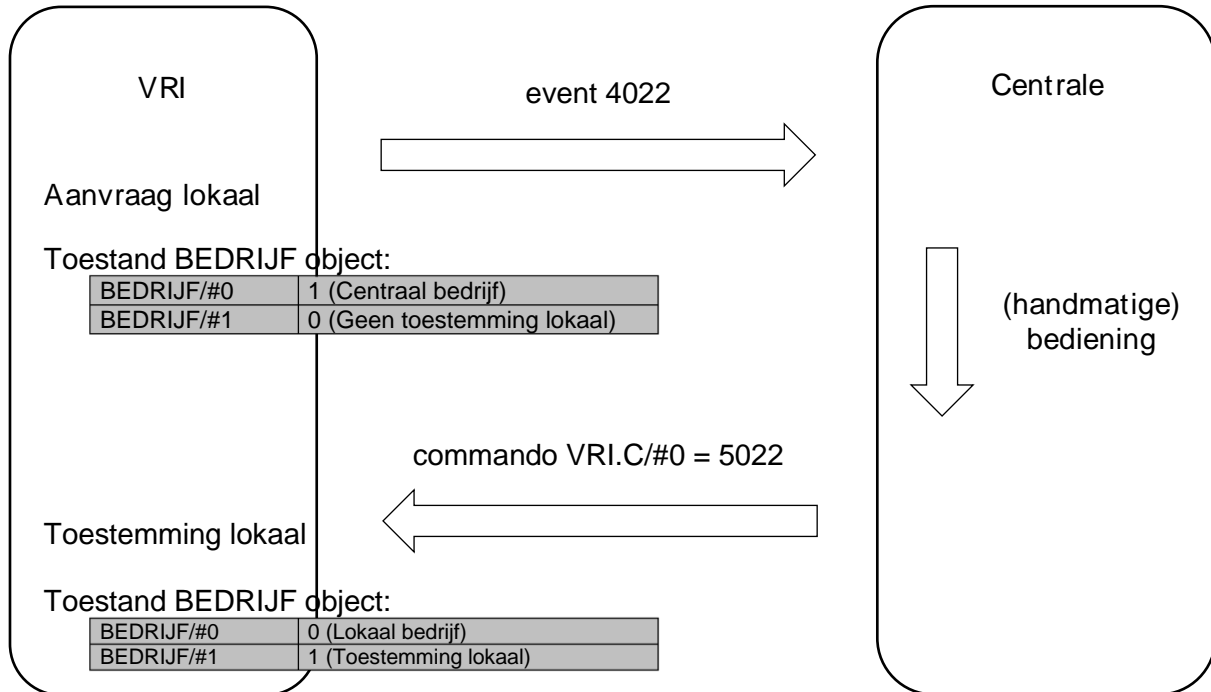
Om te voorkomen dat een VRI niet meer lokaal te bedienen is, wanneer geen communicatie met de centrale meer mogelijk is, wordt automatisch 'toestemming lokaal' gegeven indien er gedurende een time-out (default 5 minuten) geen IVERA-verbinding meer is.

De methode waarop de 'aanvraag lokaal' kan worden gedaan is regeltoestel specifiek. Dit kan via een drukknop of schakelaar van het bedieningspaneel.

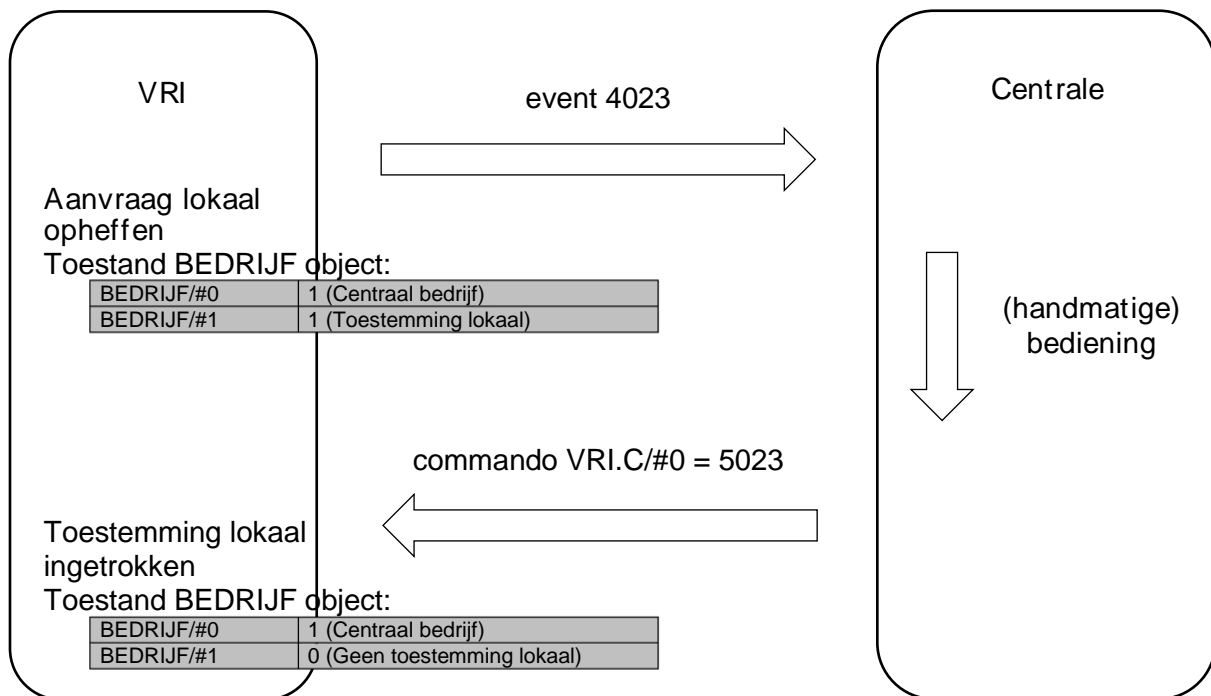
Ten behoeve van de hierboven beschreven functionaliteit is het object `BEDRIJF` gedefinieerd. Daarbij tevens het index object `BEDRIJF.I`. Na herstart of spanningsuitval onthoudt de VRI de toestand van object `BEDRIJF`.

In de onderstaande figuur is de werking van de omschakeling van centraal naar lokaal bedrijf en terug weergegeven:

Aanvragen “aanvraag lokaal bedrijf”:

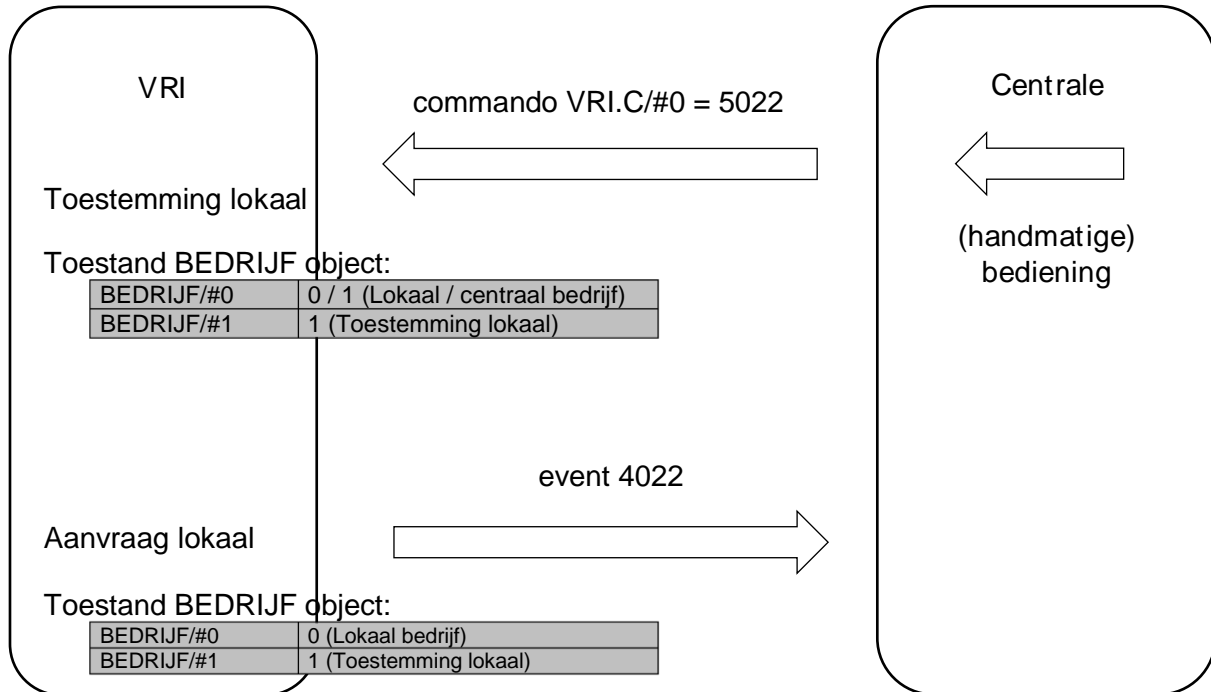


Opheffen “aanvraag lokaal bedrijf”:

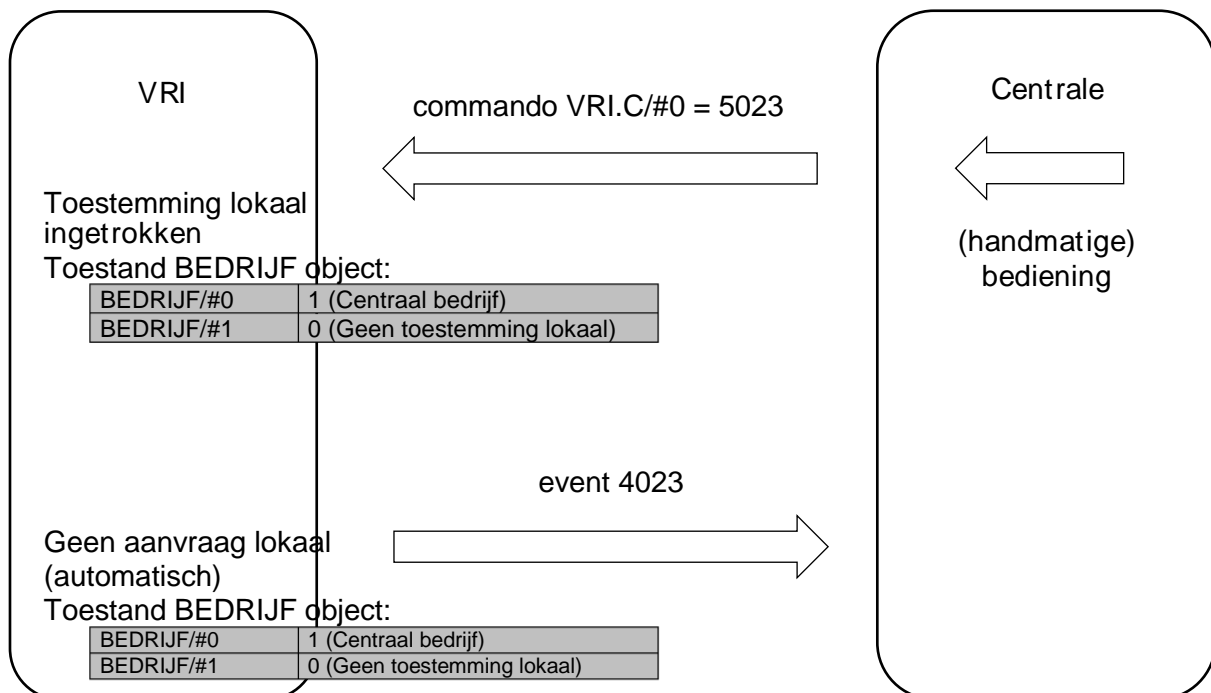


De events 4022 en 4023 worden verstuurd naar de centrale, ongeacht of dit event in de triggerlijst is opgenomen.

Geven "toestemming lokaal bedrijf":



Intrekken "toestemming lokaal bedrijf":



De events 4022 en 4023 worden verstuurd naar de centrale, ongeacht of dit event in de triggerlijst is opgenomen.

16.2 Objecten VRI status

16.2.1 VRISTAT.I

Het object VRISTAT.I geeft de index statusbronnen weer:

Attribuut	Type	Object	Omschrijving
N	1	VRISTAT.I	Naam
O	1	Index statusbronnen	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 16.2 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Het object VRISTAT.I bevat de actuele VRI status indexen van diverse bronnen in het regeltoestel:

VRISTAT.I		Automaat toestand
0	AKT	Actuele toestand
1	BD	Gewenste toestand bedieningspaneel
2	CEN	Gewenste toestand centrale
3	APPL	Gewenste toestand applicatie
4	KLOK	Gewenste toestand klok
5	PROCES	Gewenste toestand procesbesturing
6	BEWAKER	Gewenste toestand autonome bewaker
7	RESERVE7	Gereserveerd
8	RESERVE8	Gereserveerd
9	RESERVE9	Gereserveerd

Tabel 16.3 VRI status

De volgende index-namen zijn van toepassing:

AKT

Actuele of werkelijke toestand van de VRI.

BD

Gewenste toestand zoals aangevraagd via het bedieningspaneel. Het bedieningspaneel bevat de schakelaars gedoofd, geel knipperen, alles rood en automatisch bedrijf.

CEN

Gewenste toestand zoals aangevraagd vanuit de beheerscentrale.

APPL

Gewenste toestand zoals aangevraagd vanuit de applicatie. Bij een CCOL of RWSC regeling is de GPS in de CVN C-interface bepaald.

KLOK

Gewenste toestand zoals aangevraagd door de kloktijden tabel.

PROCES

Gewenste toestand zoals aangevraagd vanuit de procesbesturing. Bijvoorbeeld bij het herhaaldelijk optreden van fouten in het applicatieprogramma zal de procesbesturing de VRI naar knippen sturen.

BEWAKER

Bij een verkeersonveilige situatie zal de autonome bewaker de toestand knippen of gedoofd aanvragen afhankelijk van de aard van de situatie.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRISTAT.I	VRISTAT.I="AKT","BD","CEN","APPL","KLOK","PROCES","BEWAKER","RESERVE7","RESERVE8","RESERVE9"	Vraag object VRISTAT.I op
VRISTAT.I:A	VRISTAT.I:A="N=VRISTAT.I,T=1,E=10,U=4444,MAX=32,F=2,O='Index statusbronnen'"	Vraag attributen op van het object VRISTAT.I
VRISTAT.I/#0	VRISTAT.I/#0="AKT"	Vraag 1 ^e element op van het object VRISTAT.I

Tabel 16.4 Voorbeelden VRISTAT

16.2.2 VRISTAT

Het object VRISTAT geeft de automaat toestand weer:

Attribuut	Type	Object	Omschrijving
N	1	VRISTAT	Naam
O	1	Automaat toestand	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1	VRISTAT.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	10	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 16.5 Object attributen

Het object VRISTAT bevat de actuele VRI status en de gewenste VRI toestand van diverse bronnen in het regeltoestel:

Waarde	Omschrijving	Werkelijk	Gewenst
0	Automatisch bedrijf	Nee	Ja
1	Buiten bedrijf	Ja	Ja
2	Gedoofd	Ja	Ja
3	Knippen (extern)	Ja	Ja
4	Knippen	Ja	Ja
5	Knippen tijdens inschakelen	Ja	Nee
6	Vastgeel	Ja	Nee
7	Alles rood tijdens inschakelen (*1)	Ja	Nee
8	Alles rood	Ja	Ja
9	Alles rood tijdens uitschakelen (*1)	Ja	Nee
10	Regelen	Ja	Ja

Tabel 16.6 VRI status

*1 Indien de installatie deze toestand niet kan onderscheiden kan als alternatief de toestand "Alles rood" worden gemeld.

De keuze van de werkelijke VRI status geschiedt op basis van de gewenste VRI-status van diverse bronnen (functionele eenheden) in de verkeersregelinstallatie. De statussen die als gewenste VRI status mogen worden aangevraagd zijn in de tabel weergegeven met “Ja” in de kolom gewenst.

Eenheid	Omschrijving
Bedieningspaneel Centrale	Via het bedieningspaneel kan een wegbeheerder de installatie bedienen. Op afstand is het mogelijk de installatie in een bepaalde toestand te forceren, bijvoorbeeld bij een speciale gebeurtenis.
Applicatie	Vanuit de applicatie is het mogelijk om in specifieke situaties de installatie te laten knippen of te doven. Bijvoorbeeld in het geval van een brug- of spoorweg ingreep.
Klok	In de kloktijden tabel is vastgelegd op welke tijdstippen van de dag de installatie regelt.
Procesbesturing	Vanuit de procesbesturing is het mogelijk iedere gewenste toestand aan te vragen.
Autonome bewaker	Bij een verkeersonveilige situatie of een storing kan de bewaker de installatie laten knippen of doven.

Tabel 16.7 Bronnen gewenste VRI status

In de verkeersregelinstallatie is de prioriteit van de diverse bronnen vastgelegd. Waarbij geldt dat de autonome bewaker altijd de hoogste prioriteit heeft.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRISTAT/#0	VRISTAT.I/#0="AKT"	Vraag een element op van het object VRISTAT
VRISTAT/#0="HAL LO"	:E=11	Wijzig opgevraagde element
VRISTAT/#0	VRISTAT.I/#0="AKT"	Vraag ter controle het element weer op

Tabel 16.8 Voorbeelden VRISTAT

16.2.3 VRIPROG.I

Het object VRIPROG.I geeft de index programmabronnen weer:

Attribuut	Type	Object	Omschrijving
N	1	VRIPROG.I	Naam
O	1	Index programmabronnen	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 16.9 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Het object VRIPROG.I bevat een aantal elementen waarmee een programma's gekarakteriseerd wordt:

VRIPROG.I		Automaat programma
0	AKT	Actueel programma
1	BD	Gewenst programma bedieningspaneel
2	CEN	Gewenst programma centrale
3	APPL	Gewenst programma applicatie
4	KLOK	Gewenst programma klok
5	PROCES	Gewenst programma procesbesturing
6	RESERVE6	Gereserveerd
7	RESERVE7	Gereserveerd
8	RESERVE8	Gereserveerd
9	RESERVE9	Gereserveerd

Tabel 16.10 Programmanummer

AKT

Actuele of werkelijk programmanummer van de VRI.

BD

Gewenst programma zoals aangevraagd via het bedieningspaneel.

CEN

Gewenst programma zoals aangevraagd vanuit de beheerscentrale.

APPL

Gewenst programma zoals aangevraagd vanuit de applicatie. Bij een CCOL- of RWSC-regeling is dit niet geïmplementeerd.

KLOK

Gewenst programma zoals aangevraagd door de kloktijden tabel.

PROCES

Gewenst programma zoals aangevraagd door de procesbesturing.

NB: Het wijzigen van het programmanummer via het object VRIPROG kan wijzigingen in IVERA objecten met regeling specifieke parameters tot gevolg hebben. Denk hierbij o.a. aan toestanden van schakelaars en bedrijfstijden. Het verdient daarom aanbeveling om bij de bediening van de automaat en bij het ontwerp van meerdere programma's voor hetzelfde kruispunt met dergelijke wisselingen rekening te houden. Als de namen als index worden gebruikt door de centrale dan bieden unieke namen een mogelijke oplossing bij de programmering.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRIPROG.I	VRIPROG.I="AKT","BD","CEN", "APPL","KLOK","PROCES", "RESERVE6","RESERVE7", "RESERVE8","RESERVE9"	Opvragen van elementen in VRIPROG.I object
VRIPROG.I:A	VRIPROG.I:A="N=VRIPROG.I, T=1,E=10,U=4444,MAX=32, F=2,O='Index programmabronnen'"	Opvragen van VRIPROG.I object attributen
VRIPROG.I?	:E=10	Foutieve syntax invoer VRIPROG.I object
VRIPROG.I/#10	:E=12	Opvragen ongeldig element bereik

Tabel 16.11 Voorbeelden VRIPROG.I

16.2.4 VRIPROG

Het object VRIPROG geeft de automaat programma weer:

Attribuut	Type	Object	Omschrijving
N	1	VRIPROG	Naam
O	1	Automaat programma	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1	VRIPROG.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	99	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 16.12 Object attributen

In een VRI kunnen meerdere programma's geladen zijn. Het object VRIPROG bevat het actuele programma en het gewenste programma van diverse bronnen. De prioriteit van de diverse bronnen is vastgelegd in de VRI. Het programma is een nummer vanaf 1. Een waarde van 0 als gewenst programma geeft aan dat een bron geen programma heeft geselecteerd.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRIPROG/#0	VRIPROG/#0=<actueel programma>	Vraag een element uit object VRIPROG op
VRIPROG:A	VRIPROG:A="N=VRIPROG,T=0,E=10,U=4444,I=VRIPROG.I,MIN=0,MAX=99,S=1,F=1,O='Automaat programma'"	Vraag attributen op van het object VRIPROG
VRIPROG/#0=1	:E=11	Voer een schrijffactie uit op het object VRIPROG

Tabel 16.13 Voorbeelden VRIPROG

16.2.5 VRISUBPROG

Het object VRISUBPROG geeft de automaat programma weer:

Attribuut	Type	Object	Omschrijving
N	1	VRISUBPROG	Naam
O	1	Automaat subprogramma	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1	VRIPROG.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	99	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 16.14 Object attributen

Een programma kan eventueel subprogramma's bevatten. Het object VRISUBPROG bevat het actuele subprogramma en het gewenste subprogramma van diverse bronnen. De prioriteit van de diverse bronnen is vastgelegd in de VRI. Het subprogramma is een nummer vanaf 1. Een waarde van 0 als gewenst programma geeft aan dat een bron geen subprogramma heeft geselecteerd.

VRISUBPROG	VRIPROG.I	Automaat programma
0	AKT	Actueel subprogramma
1	BD	Gewenst subprogramma bedieningspaneel
2	CEN	Gewenst subprogramma centrale
3	APPL	Gewenst subprogramma applicatie
4	KLOK	Gewenst subprogramma klok
5	PROCES	Gewenst subprogramma procesbesturing
6	RESERVE6	Gereserveerd
7	RESERVE7	Gereserveerd
8	RESERVE8	Gereserveerd
9	RESERVE9	Gereserveerd

Tabel 16.15 Subprogrammanummer

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRISUBPROG/#2	VRISUBPROG/#2=1	Opvragen van een element in VRISUBPROG object
VRISUBPROG:A	VRISUBPROG:A= "N=VRISUBPROG,T=0,E=10, U=4444,I=VRIPROG.I,MIN=0, MAX=99,S=1,F=1, O='Automaat subprogramma'"	Opvragen van VRISUBPROG object attributen
VRISUBPROG:O	VRISUBPROG:O = "Automaat subprogramma"	Opvragen van een attribuut in VRISUBPROG object

Tabel 16.16 Voorbeelden VRISUBPROG

16.2.6 VRIPROGLYST

Het object VRIPROGLYST geeft de programmalijs weer:

Attribuut	Type	Object	Omschrijving
N	1	VRIPROGLYST	Naam
O	1	programmalijs	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMPROG	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	99	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 16.17 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRIPROGLYST/#0	VRIPROGLYST/#0=<waarde >	Opvragen van een element in VRIPROGLYST object
VRIPROGLYST:A	VRIPROGLYST:A="N=VRIPROGLYST,T=0,E=<NUMPROG>,U=4444,MIN=0,MAX=99,S=1,F=1,O='Programmalijs'"	Opvragen van VRIPROGLYST object attributen
VRIPROGLYST		Opvragen van de lijsten met aanwezige programma's

Tabel 16.18 Voorbeelden VRIPROGLYST

16.2.7 VRIPROGLYSTEXT

Het object VRIPROGLYSTEXT geeft de uitgebreide programmalijst weer:

Attribuut	Type	Object	Omschrijving
N	1	VRIPROGLYSTEXT	Naam
O	1	Uitgebreide programmalijst	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMPROGEXT	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	99	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	60	Data-element formaat
S	0		Data-element stapgrootte

Tabel 16.19 Object attributen

Formaat: Programmalijst uitgebreid(60), type 1

Voor (sub)programma's in de uitgebreide programmalijst wordt de volgende syntax gehanteerd:

Programmalijstentry = Programmanummer + "," + Subprogrammanummer + "," + Type + "," + Regelstructuur + "," + Pakketype + "," + Omschrijving

Programmanummer = 0..99

Subprogrammanummer = 0..99 (0 als programma geen subprogramma's kent of er nog geen informatie over subprogramma's beschikbaar is)

Type = AsciiString

Regelstructuur = AsciiString

Pakketype = AsciiString

Omschrijving = AsciiString

Een programma is een zelfstandige executable in het verkeersregeltoestel. Een subprogramma is een deel van een executable dat onder bepaalde omstandigheden de door een verkeersregeltoestel gerealiseerde verkeersregeling bepaalt. Een verkeersregeltoestel kan één of meer programma's bevatten; een programma kan één of meer subprogramma's bevatten.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRIPROGLYSTEXT/#0	VRIPROGLYSTEXT/#0=Programmanummer + "," + Subprogrammanummer + "," + Type + "," + Regelstructuur + "," + Pakketype + "," + Omschrijving	Opvragen van een element in VRIPROGLYSTEXT object
VRIPROGLYSTEXT:A	VRIPROGLYSTEXT:A="N=VRIPROGLYSTEXT,T=1,E=<NUMPROGEXT>,U=4444,F=60,O='Uitgebreide programmalijst'"	Opvragen van VRIPROGLYSTEXT object attributen
VRIPROGLYST/#0=1	:E=11	Een schrijffactie is niet toegestaan.

Tabel 16.20 Voorbeelden VRIPROGLYSTEXT

16.2.8 BEDRIJF.I

Het object BEDRIJF.I geeft de index object BEDRIJF weer:

Attribuut	Type	Object	Omschrijving
N	1	BEDRIJF.I	Naam
O	1	Index object bedrijf	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	2	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 16.21 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Het object BEDRIJF.I bevat een aantal elementen waarmee BEDRIJF gekarakteriseerd wordt:

BEDRIJF	BEDRIJF.I	Omschrijving
0	CENTRAAL	Geeft aan of VRI in lokaal (0) , centraal (1) of gecombineerd (2) bedrijf staat (kan runtime veranderen tussen 0 en 1. De waarde 2 is vast).
1	TOESTEMMING	Geeft aan of er door de centrale niet (0) of wel (1) toestemming lokaal gegeven is (kan runtime veranderen). In de toestand gecombineerd bedrijf staat hier altijd de waarde 1.

Een voorbeeld is opgenomen in de paragraaf 16.1.

16.2.9 BEDRIJF

Het object BEDRIJF geeft de bedrijfstoestand m.b.t. de centrale weer:

Attribuut	Type	Object	Omschrijving
N	1	BEDRIJF	Naam
O	1	bedrijfstoestand m.b.t. de centrale	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	2	aantal data-elementen
I	1	BEDRIJF.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	2	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 16.22 Object attributen

Een voorbeeld is opgenomen in de paragraaf 16.1.

16.2.10 BEDRIJF events

Daarnaast zijn er twee VRI.C commando's gedefinieerd, namelijk 5022 voor 'toestemming lokaal' en 5023 voor opheffen 'toestemming lokaal'.

Commando's:

Commando code	Omschrijving
5022	Geeft de VRI toestemming om naar lokaal bedrijf te gaan. Er wordt niet meer geluisterd naar de programmawens van de centrale maar van de lokale statuskeuze
5023	Opheffen toestemming lokaal bedrijf. De VRI luistert alleen naar de wens van de programmawens van de centrale. Deze wens is vastgelegd in elementnr. 2 van resp. VRISTAT en VRIPROG. Afhankelijk van de implementatie in de VRI wordt hier al of niet gehoor aan gegeven.

Code	Omschrijving	Detailinfo
4022	'Aanvraag toestemming lokaal' is gedaan door gebruiker bij VRI.	
4023	'Aanvraag toestemming lokaal' is ingetrokken door gebruiker bij VRI.	

16.2.11 Resetevents

Reset events hebben betrekking op gebeurtenissen in de VRI en commando's waarmee storingen gereset worden.

Zo wordt bijvoorbeeld het commando om alle storingen te resetten gemeld als eventcode 4001. Daarin tegen is het ook mogelijk om alle storingen te resetten door het schrijven van de waarde 4001 naar het object "VRI.C".

Code	Omschrijving	Detailinfo	VRI.A
4000	Algemeen resetevent	<IdNummer>	
4001	Reset van alle storingen		
4002	Reset van detectiealarmen		
4003	Reset van lampfouten		
4004	Reset van applicatiefouten		
4005	Reset van tellers		
4006	Reset teller applicatiefouten		
4007	Reset teller aantal GUS-WUS fouten		
4008	Reset teller fasebewakingsfouten		
4009	Reset teller executietijdoverschrijdingen		
4010	Netspanning uitsterfbericht		
4011	Opstartbericht	<Soort start>	
4012	Deur open politie paneel	0 = gesloten, 1 = open	
4013	Deur open wegbeheerder	0 = gesloten, 1 = open	
4014	Deur open energie compartiment	0 = gesloten, 1 = open	
4015	Testbericht noodkreetmelder		
4016	Noodstroomvoedingbericht	0 = gedeactiveerd, 1 = geactiveerd	Ja

Tabel 16.23 Reset events

IdNummer Uniek identificatie nummer van het element in het regeltoestel.
 Soort start Aanduiding van het soort start (koud = 1, warm = 0, warm+ = 2)

De verschillende soorten van een start van een VRI (event 4011) hebben het volgende effect:

Koud Koude start, waarbij alle dynamische gegevens, parameterwijzigingen, logboeken etc. gewist worden.
 Warm Warme start waarbij alle gegevens, parameterwijzigingen, logboeken etc. behouden blijven.
 Warm+ De betekenis van "Warm+" is gelijk aan de betekenis van "Warm". Het wordt echter aangeraden om voor VRI implementaties gebruik te maken van "Warm" aangezien "Warm+" in een toekomstige IVERA specificatie zal komen te vervallen.

NB: De kolom 'VRI.A' geeft aan per event of deze bij actief of inactief zijn in de automaat, zal worden getoond in respectievelijk zal worden verwijderd uit de 'Actieve storingenlijst'. Indien dit event inactief wordt gemaakt door een ander event, is dit expliciet aangegeven met 'eventcode'.

Toelichting bij event 4001: reset van alle storingen

De volgende storingen worden bij het geven van het commando 4001 minimaal gereset:

- detectiestoring
- lampfouten
- GUS/WUS-fouten
- fasebewakingsfouten
- executietijdfouten

Van iedere storing die wordt gereset, wordt in betreffende logboeken (xxx.LA en xxx.LB) hiervan een herstelmelding opgeslagen.

Indien van toepassing, worden de tellers in het object CP tevens gereset.

In principe wordt voor iedere storing die wordt opgeheven in betreffende logboeken (xxx.LA en xxx.LB) een herstelmelding opgeslagen. Er geldt een uitzondering voor fouten waarvan het herstel alleen na een herstart kan worden vastgesteld, en fouten die worden hersteld gedurende een tijdsperiode waarin de automaat is uitgeschakeld. Een centrale dient daarom

een herstartmelding 4011 als een impliciete reset van alle fouten te interpreteren. Daarnaast kan een resetmelding 4001 ook een impliciete opheffing van alle voorafgaande fouten tot gevolg hebben. Wanneer een foutmelding impliciet is gereset maar toch nog steeds actief blijkt te zijn, dan wordt daar opnieuw in het logboek melding van gemaakt.

Toelichting bij events 4012, 4013 en 4014;

Een event met de waarde 0 betekent "deur is dichtgegaan.

Een event met de waarde 1 betekent "deur is opengegaan.

17. Klokperioden

17.1 Algemeen klokperioden

Voor de VRI status zijn de volgende objecten gedefinieerd:

Object	Omschrijving
KLA1	Inschakelen regelen periode 1
KLU1	Uitschakelen regelen periode 1
KLA2	Inschakelen regelen periode 2
KLU2	Uitschakelen regelen periode 2
KLA3	Inschakelen regelen periode 3
KLU3	Uitschakelen regelen periode 3
KLA4	Inschakelen regelen periode 4
KLU4	Uitschakelen regelen periode 4
KLA5	Inschakelen regelen periode 5
KLU5	Uitschakelen regelen periode 5
KLOKPER.I	Index object KLOKPER
KLOKPER	Hulpelement klokperiode
KLOKPER.A	Stand klokperiode
KLOKPROG.I	Index object KLOKPROG
KLOKPROG	Klokperiode programmakeuze
KLOKPROG.A	Stand klokperiode programmakeuze

Per VRI kunnen per dag van de week 5 periodes (KLA1 t/m KLU5) worden gespecificeerd. Deze periodes geven aan de momenten van de dag waarop de VRI regelt. Een regelperiode is de tijd tussen het in- en uitschakelmoment van een VRI. De tijden worden in minuten nauwkeurig opgegeven waarbij de seconde stand impliciet nul is.

Een periode van 00:00 tot 24:00 geeft aan dat de VRI continue in bedrijf is.

Een periode van 00:00 tot 00:00 geeft aan dat er geen tijdsperiode is gespecificeerd. In dat geval wordt er voor de VRI op de betreffende dag geen wens om te regelen opgegeven.

De begintijd mag niet groter zijn dan de eindtijd. Daarmee wordt voorkomen, dat een gekozen periode met verschillende dag-types voor begin en einde van de periode langer blijkt te duren dan bij het instellen van de parameters werd beoogd. Indien hieraan niet voldaan wordt kan de VRI reageren door het retourneren van een foutmelding, of bij KLA groter dan KLU wordt de periode als uitgeschakeld beschouwd.

Het index-object DAG.I kan naast de namen van (groepen van) dagen ook aanduidingen als "feestdag", "koopzondag" en "koopavond" bevatten.

Als in één van de objecten voor regelperiodes zo'n term (dag van de week met waarde 11 t/m 14) voorkomt, wordt voor het antwoord op de vraag of het een bijzondere dag is de lijst met bijzondere dagen (BIJZDAG) gebruikt.

Het bepalen van de relevante instelling van de klokmomenten verloopt dan als volgt:

1. Kijk in de lijst met bijzondere dagen, of de huidige dag er in voorkomt;
2. Controleer, of er een klokmoment voor het bijbehorende dag-type gedefinieerd is;
3. Herhaal dit tot dit het geval is dan wel tot de lijst volledig verwerkt is.

Gebruik de gegevens voor de gevonden bijzondere dag. Als er geen bijzondere dag gevonden is:

1. Kijk of er een klokmoment gedefinieerd is voor elke dag.
2. Gebruik de gevonden gegevens voor elke dag
3. Als er geen schakelmoment is op grond van de 'elke dag' instellingen:

- a. Doe hetzelfde voor werkdagen en weekeinde
- b. Als er geen gegevens voor werkdagen of weekeinde gevonden zijn:
 - Gebruik de gegevens voor de 'gewone' dag van de week.

NB. In het algemeen zal het bij een combinatie van een object voor Inschakelen regelen en voor Uitschakelen regelen zó zijn, dat, als de aanduiding van een "bijzondere dag" gebruikt wordt, deze in beide voorkomt. Daarmee wordt voorkomen, dat de lengte van de regelperiode afhangt van de inhoud van de lijst met bijzondere dagen.

In een VRI kunnen klokperiodes programmakeuze (KLOKPROG) worden gespecificeerd. Deze periodes geven aan welk (sub)programma actief moet zijn, wanneer de desbetreffende klokperiode actief is.

Óf er geregeld wordt, wordt bepaald door de elementen KLA1 t/m KLA5 en KLU1 t/m KLU5. De klok gestuurde programmakeuze bepaalt wanneer welk programma actief is, in de periode dat er geregeld wordt.

De wijze van bepaling van het actief zijn van een programmakeuze op een bepaalde dag, wordt net zo uitgevoerd als bij klok gestuurde regelperiodes.

Als er geen KLOKPROG aanwezig is, er geen KLOKPROG ingevuld is of er geen van de objecten KLOKPROG actief is, wordt de programmakeuze niet door de klok beïnvloed.

17.2 Objecten klokperioden

17.2.1 KLA1

Het object KLA1 geeft het inschakelen regelen periode 1 weer:

Attribuut	Type	Object	Omschrijving
N	1	KLA1	Naam
O	1	Inschakelen regelen periode 1	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMDAG	aantal data-elementen
I	1	DAG.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	2400	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	20	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 17.1 Object attributen

Formaat: Klok (uur/minuut) (20), type 0

De kloktijd is een getal met de tijd in het volgende formaat:

Waargave: UUMM

Waarde: uur * 100 + minuut

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
KLA1/*=0	KLA1/*=0	Begintijdstip gelijk aan 00:00 uur
KLA1/*=2400	KLA1/*=2400	Begintijdstip gelijk aan 24:00 uur

Tabel 17.2 Voorbeelden KLA1

17.2.2 KLU1

Het object KLU1 geeft het uitschakelen regelen periode 1 weer. Verwezen wordt naar paragraaf 17.2.1 voor de inhoudelijke omschrijving.

17.2.3 KLA2

Het object KLA2 geeft het inschakelen regelen periode 2 weer. Verwezen wordt naar paragraaf 17.2.1 voor de inhoudelijke omschrijving.

17.2.4 KLU2

Het object KLU2 geeft het uitschakelen regelen periode 2 weer. Verwezen wordt naar paragraaf 17.2.1 voor de inhoudelijke omschrijving.

17.2.5 KLA3

Het object KLA3 geeft het inschakelen regelen periode 3 weer. Verwezen wordt naar paragraaf 17.2.1 voor de inhoudelijke omschrijving.

17.2.6 KLU3

Het object KLU3 geeft het uitschakelen regelen periode 3 weer. Verwezen wordt naar paragraaf 17.2.1 voor de inhoudelijke omschrijving.

17.2.7 KLA4

Het object KLA4 geeft het inschakelen regelen periode 4 weer. Verwezen wordt naar paragraaf 17.2.1 voor de inhoudelijke omschrijving.

17.2.8 KLU4

Het object KLU4 geeft het uitschakelen regelen periode 4 weer. Verwezen wordt naar paragraaf 17.2.1 voor de inhoudelijke omschrijving.

17.2.9 KLA5

Het object KLA5 geeft het inschakelen regelen periode 5 weer. Verwezen wordt naar paragraaf 17.2.1 voor de inhoudelijke omschrijving.

17.2.10 KLU5

Het object KLU5 geeft het uitschakelen regelen periode 5 weer. Verwezen wordt naar paragraaf 17.2.1 voor de inhoudelijke omschrijving.

17.2.11 KLOKPER.I

Het object KLOKPER.I geeft de index object KLOKPER weer:

Attribuut	Type	Object	Omschrijving
N	1	KLOKPER.I	Naam
O	1	Index object klokper	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMKLOKPER	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 17.3 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
KLOKPER.I	KLOKPER.I="klok1","klok2","klok3",...	Vraag statische lijst met klokperiodes op.

Tabel 17.4 Voorbeelden KLOKPER.I

17.2.12 KLOKPER

Het object KLOKPER geeft het hulpelement klokperiode weer:

Attribuut	Type	Object	Omschrijving
N	1	KLOKPER	Naam
O	1	Hulpelement klokperiode	Omschrijving
T	1	1	Type
U	0	6644	User Identificatie Control
E	0	NUMKLOKPER	aantal data-elementen
I	1	KLOKPER.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	21	Data-element formaat
S	0		Data-element stapgrootte

Tabel 17.5 Object attributen

Formaat: Klokperiode (21), type 1

Voor de klokperiode wordt de volgende syntax gehanteerd:

```
Klokper=  
[StartJaar] + “,” + [StartWeek] + “,” + StartDag + “,” + StartTijd + “,” +  
[EindJaar] + “,” + [EIndWeek] + “,” + EindDag + “,” + EindTijd [ + “,” + Toelichting ]  
  
StartJaar = Jaartal  
StartWeek = Weeknummer  
StartDag = Dag van de week  
StartTijd = uur*100 + minuut  
EindJaar = Jaartal  
EIndWeek = Weeknummer  
EindDag = Dag van de week  
EindTijd = uur*100 + minuut  
Toelichting = AsciiString
```

Bij de bepaling van StartWeek en Eindweek geldt het volgende:

Voor het JWD formaat wordt de norm ISO 8601 aangehouden:

- De eerste week van elk jaar is de week waarin 4 Januari valt;
- Elke jaar heeft minimaal 52 weken;
- Indien nieuwjaarsdag op een Donderdag valt heeft het jaar 53 weken;
- Indien de nieuwjaarsdag van een schrikkeljaar op Woensdag valt heeft het jaar ook 53 weken;

Bij de bepaling van startmoment en eindmoment (op basis van Jaar, Week, Dag en Tijd) geldt het volgende:

- Indien de StartDag van het type feestdag(11), koopavond(12), koopzondag(13) of speciaal(14) is, dan dient de EindDag van hetzelfde type te zijn.
- Indien de Start- en Eind Dag van het type feestdag(11), koopavond(12), koopzondag(13) of speciaal(14) is, zal dit automatisch leiden tot een klokperiode van een enkele dag.
- Indien de StartDag van het type feestdag(11), koopavond(12), koopzondag(13) of speciaal(14) is, is het gebruik van de velden StartJaar en StartWeek niet toegestaan.
- Indien de StartDag van het type weekeinde(10), is zal het werkelijke startmoment plaatsvinden op een zaterdag(6). Als de startgegevens worden ingevuld tijdens een weekeinde tussen start- en eindtijd, wordt er onmiddellijk gestart.
- Indien de StartDag van het type werkdagen(9), is zal het werkelijke startmoment plaatsvinden op een maandag(1). Als de startgegevens worden ingevuld op een werkdag tussen start- en eindtijd, wordt er onmiddellijk gestart.
- Indien de EindDag van het type feestdag(11), koopavond(12), koopzondag(13) of speciaal(14) is, is het gebruik van de velden EindJaar en EindWeek niet toegestaan.
- Indien de EindDag van het type weekeinde(10) of ‘elke dag’(8), is zal het werkelijke einde moment plaatsvinden op een zondag(7). Als de eindgegevens worden ingevuld tijdens een weekeinde tussen start- en eindtijd, wordt er onmiddellijk gestopt.
- Indien de EindDag van het type werkdagen(9), is zal het werkelijke einde moment plaatsvinden op een vrijdag(5). Als de eindgegevens worden ingevuld op een werkdag tussen start- en eindtijd, wordt er onmiddellijk gestopt.
- Indien de StartDag niet in gebruik is (0) dan moeten alle andere velden ook als niet in gebruik zijn gespecificeerd (dus ook 0, of leeg indien het optionele velden betreft).
- Indien de EindDag niet in gebruik is (0) dan moeten alle andere velden ook als niet in gebruik zijn gespecificeerd (dus ook 0, of leeg indien het optionele velden betreft).
- Het start moment (StartJaar+StartWeek+StartDag+StartTijd) mag niet groter zijn dan het eind moment (EindJaar+EindWeek+EindDag+EindTijd). Daarmee wordt voorkomen, dat een gekozen periode met verschillende dagtypes voor begin en

einde van de periode langer blijkt te duren dan bij het instellen van de parameters werd beoogd.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
KLOKPER	KLOKPER=",,0,0,,,0,0",,,0,0,, ,0,0", "[,Toelichting]"	Vraag de aanwezige klokperiodes op.
KLOKPER/#0="Y, W,D,T1,Y,W,D,T2, [Toelichting]"	KLOKPER/#0="Y,W,D,T1,Y, W,D,T2[,Toelichting]"	Stel een klokperiode in waar de huidige tijd in valt.

Tabel 17.6 Voorbeelden KLOKPER

17.2.13 KLOKPER.A

Het object KLOKPER.A geeft stand klokperiode weer:

Attribuut	Type	Object	Omschrijving
N	1	KLOKPER.A	Naam
O	1	Stand klokperiode	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMKLOKPER	aantal data-elementen
I	1	KLOKPER.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	1	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	10, schakelaar (uit/aan)	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 17.7 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
KLOKPER.A	KLOKPER.A=1,0,0,...	Vraag de actieve klokperiodes op.

Tabel 17.8 Voorbeelden KLOKPER.A

17.2.14 KLOKPROG.I

Het object KLOKPROG.I geeft de index object KLOKPROG weer:

Attribuut	Type	Object	Omschrijving
N	1	KLOKPROG.I	Naam
O	1	Index object klokprog	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMKLOKPROG	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 17.9 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
KLOKPROG.I	KLOKPROG.I="prog1","prog2","prog3",...	Vraag de index klokgestuurde programma's op.

Tabel 17.10 Voorbeelden KLOKPROG.I

17.2.15 KLOKPROG

Het object KLOKPROG geeft de klokperiode programmakeuze weer:

Attribuut	Type	Object	Omschrijving
N	1	KLOKPROG	Naam
O	1	Klokperiode programmakeuze	Omschrijving
T	1	1	Type
U	0	6644	User Identificatie Control
E	0	NUMKLOKPROG	aantal data-elementen
I	1	KLOKPROG.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	22	Data-element formaat
S	0		Data-element stapgrootte

Tabel 17.11 Object attributen

Formaat: Klokperiode programmakeuze (22), type 1

Voor de klokperiode programmakeuze wordt de volgende syntax gehanteerd:

```
Klokprog=
KlokPeriode + "," +
[Programmanummer] + "," + [Subprogrammanummer] [ + "," + Toelichting ]
```

```
KlokPeriode = Naam klokperiode
Programmanummer = 0..99 (0 is geen keuze)
Subprogrammanummer = 0..99 (0 is geen keuze)
Toelichting = AsciiString
```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
KLOKPROG/#0="klok1,0,[Toelichting]"	KLOKPROG/#0="klok1,X,[Toelichting]",...	Stel een klokprogramma in
KLOKPROG.A	KLOKPROG.A=1,0,0,...	Vraag actieve klokprogramma op
KLOKPROG/#0="klok1,-1,"	:E=16	Stel een klokprogramma in met een buiten de range programma nummer

Tabel 17.12 Voorbeelden KLOKPROG

17.2.16 KLOKPROG.A

Het object KLOKPROG.A geeft stand klokperiode programmakeuze weer:

Attribuut	Type	Object	Omschrijving
N	1	KLOKPROG.A	Naam
O	1	Stand klokperiode programmakeuze	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMKLOKPROG	aantal data-elementen
I	1	KLOKPROG.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	1	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	10, schakelaar (uit/aan)	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 17.13 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
KLOKPROG.A	KLOKPROG.A=1,0,0,...	Roep actuele status op van KLOKPROG.A . Periode klok1 is actief, dus programma X ook.

Tabel 17.14 Voorbeelden KLOKPROG.A

18. Detectie

18.1 Algemeen detectie

De detectie afhandeling binnen IVERA is onder te verdelen in de fysieke detectie afhandeling, de bijbehorende tijdsinstellingen en is inclusief het opvragen en/of loggen van het detectie gedrag.

Dit hoofdstuk heeft betrekking op alle objecten die onder de index D.I vallen. Een mogelijke uitzondering hierin zijn de hiaatmetingen binnen RWS-C. Deze vallen onder de index SG.I, vanaf RWS-C versie komende na 1.11.21 valt dit onder D.I. Het hoofdstuk is opgedeeld in de volgende onderdelen:

- De detectiealgoritme objecten;
- De monitorings- / loggings objecten;
- De regeltechnische objecten;
- De attributen.

Voor detectie zijn de volgende objecten gedefinieerd:

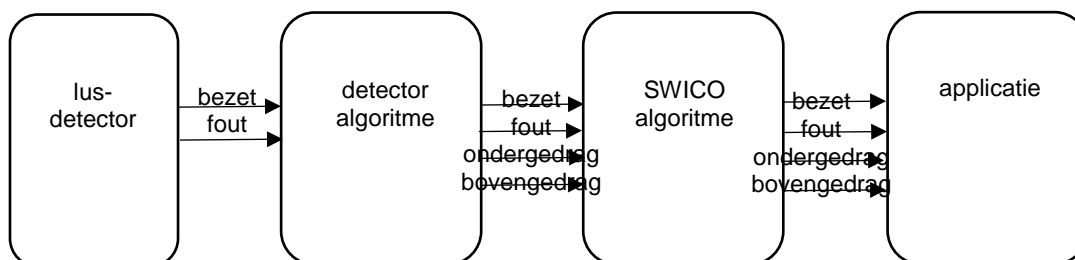
Object	Omschrijving
D.I	Index
D.A	Actuele toestand.
TD.A	Lopende timer bezet/onbezet.
SWD	SWICO detectie.
D.LB	Detector logboek (laatste xx veranderingen).
D.LA	Detector logboek (onbevestigd).
TDBG	Bewakingstijd bovengedrag
TDOG	Bewakingstijd ondergedrag
TDFL	Meettijd voor fluttergedrabewaking
CDFL	Grenswaarde voor Fluttergedragbewaking
TDBP1	Tijd detectiebewaking aan
TDBP2	Tijd detectiebewaking uit
DC.I	Index object DC
DC.A	classificatiedetectortoestand
LD	Lijndump detectortoestand

Tabel 18.1 Detectie objecten

Detectiealgoritme

De detector toestand zoals die kan worden uitgelezen via het IVERA protocol is gebaseerd op een standaard verwerking van detectie informatie binnen de verkeersregelininstallatie. Voordat de detectie binnenkomt in de applicatie vinden er een paar tussenstappen plaats.

Deze standaard verwerking is weergegeven in het volgende figuur.



Note: Swico op/af komt wordt niet op de CVN-C interface aangeboden en daarmee niet aan de applicatie. Daarnaast is het zo dat de applicatie zelf ook op detectiefouten kan toetsen.

Het onderdeel detectie begint bij de passage van de detectielus of bediening van een drukknop. De aanwezigheid van een voertuig wordt gedetecteerd door een lusdetector, die in het verkeersregeltoestel of sectiekast is opgenomen. Deze lusdetector bewaakt de detectielus op onderbreking van de lusdraad. In geval van een drukknop vindt geen bewaking plaats op de onderbreking van een ader.

De actuele bezet- en foutstatus vanuit de lusdetector wordt continu door het detectiealgoritme gelezen. Het detectiealgoritme geeft deze informatie transparant door en controleert op basis van vier tijdstellingen met hun lopende timers (TDOG, TDBG, TDFL en CDFL) alle detectoren op onder-, boven- en fluttergedrag. Een onder- en boven- en fluttergedrag wordt doorgegeven aan het volgende proces.

Bewaking fluttergedrag is het detecteren van fluttergedrag van detectoren zodat de regeling hierop kan acteren. Voor het detecteren van fluttergedrag zijn een meettijd (TDFL) en een grenswaarde (CDFL) nodig. Tijdens de meettijd wordt het aantal bezetmeldingen van een detector geteld. Er is sprake van fluttergedrag indien het aantal getelde bezetmeldingen gelijk of groter is dan de ingestelde grenswaarde.

Het SWICO-algoritme stelt de beheerder in staat om de actuele detectorstatus door middel van softwareschakelaars (SWICO's) te overschrijven. In het geval dat er geen SWICO is gezet, wordt de actuele detectortoestand transparant doorgegeven aan de applicatie. In het geval dat een SWICO is gezet, wordt de actuele toestand niet doorgegeven aan de applicatie, dit om eventuele maatregelen in de applicatie ten aanzien van lusfouten ongedaan te maken.

De detectortoestand bevat naast de SWICO-bits de actuele informatie van de lus (dus niet de informatie die aan de applicatie wordt doorgegeven), dit om op afstand te kunnen controleren of eventuele storingen reeds zijn verholpen. In de beheerscentrale kan de informatie, die wordt doorgegeven aan de applicatie, worden afgeleid op basis van het SWICO-algoritme.

Afhankelijk van de implementatie kan het hier boven omschreven principe afwijken.

De detector toestand bestaat uit:

Bit	Naam	Omschrijving
0	Bezet	Actuele bezet/onbezet status van de lus.
1	Fout	Actuele foutstatus van de lus
2	onder/bovengedrag	Actuele onder/bovengedrag status zoals bepaald door detectiealgoritme. (Ondergedrag als bezet = 0, bovengedrag als bezet = 1).
3	SWICO op	Lus permanent bezet.
4	SWICO af	Lus permanent onbezet.
5	Fluttergedrag	Op de lus is fluttergedrag geconstateerd.

Tabel 18.2 Detector toestand

Speciaal voor de bewaking op ondergedrag is er per dag van de week een instelbare periode (TDBP1 en TDBP2) gedefinieerd waarop de detectie bewaking actief is. De timers die gebruikt worden voor de bewaking halteren bij het einde bewakingsperiode en lopen verder bij de aanvang van een nieuwe periode.

De monitoring

Alle detectie gebeurtenissen worden opgeslagen in een detector logboek. Hiermee is naderhand de geschiedenis, in beperkte mate van passages en/of meldingen, te herleiden. Elke verandering wordt met een tijdstempel gelogd in het betreffende logboek.

Foutmeldingen resulteren in event 1020. Deze zijn terug te vinden in het VRI logboek en op te vragen door middel van VRI.LA. Verwezen wordt naar het object VRI.C.

De regeltechnische objecten

Afhankelijk van het applicatie of het type regeltoestel is het mogelijk om een hiaat- en/of bezettijd per detector te specificeren. De toepassingswijze is verder applicatieafhankelijk.

De bezettijd (TDB) bepaalt de vertraging tussen het bezet worden van de detector en het zetten van een aanvraag in de applicatie. Indien de detector voor het verstrijken van deze tijd afvalt wordt er geen aanvraag gezet.

De hiaattijd (TDH) bepaalt de vertraging tussen het afvallen van de detector en het afvallen van het meetkriterium in de applicatie. Indien de detector voor het verstrijken van deze tijd weer opkomt blijft het meetkriterium actief.

Per detector kunnen er afhankelijk van de applicatie 1 of meerdere hiaattijden worden gedefinieerd, indien er sprake is van een RWS-C applicatie. Voor de hiaattijden zijn de objecten TDH1 tot en met TDH22 gereserveerd.

Voor de omschrijving van deze objecten wordt verwezen naar de CCOL en RWS-C documentatie.

Classificatie

De objecten DC.I en DC.A zijn bedoeld om voertuigen te classificeren. Er worden voertuig relevante gegevens in het object opgenomen en op de interface aangeboden.

Lijndump

Het object LD geeft een string weer met daarin een timestamp en voor iedere detector een karakter die de actuele toestand weergeeft. De timestamp geeft aan het moment van de laatste wijziging.

18.2 Objecten detectie

18.2.1 D.I

Het object D.I geeft de index weer van alle detectoren:

Attribuut	Type	Object	Omschrijving
N	1	D.I	Naam
O	1	Detectornamen	Omschrijving
T	0	1	Type
U	0	4444	User Identificatie Control
E	0	NUMD	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0		Data-element formaat
S	0	2	Data-element stapgrootte

Tabel 18.3 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Het standaardiseren van namen van verschillende parameters is niet zozeer een IVERA protocol aanpassing, als wel het maken van een extra restrictie op het gebruik van namen in een aantal indexobjecten. Het gaat hierbij over de indexobjecten, die gekoppeld zijn aan de parameters in CCOL of RWS-C, die reeds een naam hebben zoals de volgende indexobjecten:

De namen in deze objecten moeten exact overeenkomen met de namen die in CCOL of RWS-C zijn geconfigureerd. Zij bepaalde namen leeg gelaten, dan wordt in IVERA een invulling aangegeven, die overeenkomt met het elementnummer en het type parameter.

De n^e lege indexnaam wordt in IVERA ingevuld met 'D_(n-1)', waarbij (n-1) een getal weergeeft.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
D.I	D.I="021","022","D_2"	Weergave index alle detectoren
D.I/#0	D.I/#0="021"	Opvragen toestand detector met index 0

Tabel 18.4 Voorbeelden index detectoren

18.2.2 D.A

Het object D.A geeft de toestand weer van alle detectoren:

Attribuut	Type	Object	Omschrijving
N	1	D.A	Naam
O	1	Detectortoestand	Omschrijving
T	0	0	Type
U	0	4444	User Identificatie Control
E	0	NUMD	aantal data-elementen
I	1	D.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	63	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	40, detectortoestand	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.5 Object attributen

Formaat: Toestand van de detector (40), type 0

Het object geeft voor iedere detector een karakter die de actuele toestand weergeeft. In totaal kent een detector 64 toestanden (6 bits); de codering in de vorm van karakters (voor een eventuele lijndump) is "0" .. "9" , "A" .."V" en "a".."x" De toestanden 24 t/m 31 en 56 t/m 63 zijn illegaal en kunnen onder normale omstandigheden niet voorkomen. Het veld bitmask geeft de bitcodering weer zoals vastgelegd in Tabel 18.6.

Waarde	Karakter	Bitmask	Toestand Detector toestand	Applicatie				
				Bezet	Fout	Onder	Boven	Flutter
0	0	000000	Onbezet	0	0	0	0	0
1	1	000001	Bezet	1	0	0	0	0
2	2	000010	Onbezet en fout	0	1	0	0	0
3	3	000011	Bezet en fout	1	1	0	0	0
4	4	000100	Onbezet en ondergedrag	0	0	1	0	0
5	5	000101	Bezet en bovengedrag	1	0	0	1	0
6	6	000110	Onbezet, fout en ondergedrag	0	1	1	0	0
7	7	000111	Bezet, fout en bovengedrag	1	1	0	1	0
8	8	001000	SWICO op, Onbezet	1	0	0	0	0
9	9	001001	SWICO op, Bezet	1	0	0	0	0
10	A	001010	SWICO op, Onbezet en fout	1	0	0	0	0
11	B	001011	SWICO op, Bezet en fout	1	0	0	0	0
12	C	001100	SWICO op, Onbezet en ondergedrag	1	0	0	0	0
13	D	001101	SWICO op, Bezet en bovengedrag	1	0	0	0	0
14	E	001110	SWICO op, Onbezet, fout en ondergedrag	1	0	0	0	0
15	F	001111	SWICO op, Bezet, fout en bovengedrag	1	0	0	0	0
16	G	010000	SWICO af, Onbezet	0	0	0	0	0
17	H	010001	SWICO af, Bezet	0	0	0	0	0
18	I	010010	SWICO af, Onbezet en fout	0	0	0	0	0
19	J	010011	SWICO af, Bezet en fout	0	0	0	0	0
20	K	010100	SWICO af, Onbezet en ondergedrag	0	0	0	0	0
21	L	010101	SWICO af, Bezet en bovengedrag	0	0	0	0	0
22	M	010110	SWICO af, Onbezet, fout en ondergedrag	0	0	0	0	0
23	N	010111	SWICO af, Bezet, fout en bovengedrag	0	0	0	0	0
24	X	011000	illegale toestand	0	0	0	0	0
25	X	011001	illegale toestand	0	0	0	0	0
26	X	011010	illegale toestand	0	0	0	0	0
27	X	011011	illegale toestand	0	0	0	0	0
28	X	011100	illegale toestand	0	0	0	0	0
29	X	011101	illegale toestand	0	0	0	0	0
30	X	011110	illegale toestand	0	0	0	0	0
31	X	011111	illegale toestand	0	0	0	0	0
32	a	100000	Onbezet en fluttergedrag	0	0	0	0	1
33	b	100001	Bezet en fluttergedrag	1	0	0	0	1
34	c	100010	Onbezet, fout en fluttergedrag	0	1	0	0	1
35	d	100011	Bezet, fout en fluttergedrag	1	1	0	0	1
36	e	100100	Onbezet, ondergedrag en fluttergedrag	0	0	1	0	1
37	f	100101	Bezet, bovengedrag en fluttergedrag	1	0	0	1	1
38	g	100110	Onbezet, fout, ondergedrag en fluttergedrag	0	1	1	0	1
39	h	100111	Bezet, fout, bovengedrag en fluttergedrag	1	1	0	1	1
40	i	101000	SWICO op, Onbezet en fluttergedrag	1	0	0	0	0
41	j	101001	SWICO op, Bezet en fluttergedrag	1	0	0	0	0
42	k	101010	SWICO op, Onbezet, fout en fluttergedrag	1	0	0	0	0
43	l	101011	SWICO op, Bezet, fout en fluttergedrag	1	0	0	0	0
44	m	101100	SWICO op, Onbezet, ondergedrag en fluttergedrag	1	0	0	0	0
45	n	101101	SWICO op, Bezet, bovengedrag en fluttergedrag	1	0	0	0	0
46	o	101110	SWICO op, Onbezet, fout, ondergedrag en fluttergedrag	1	0	0	0	0
47	p	101111	SWICO op, Bezet, fout, bovengedrag en fluttergedrag	1	0	0	0	0
48	q	110000	SWICO af, Onbezet en fluttergedrag	0	0	0	0	0
49	r	110001	SWICO af, Bezet en fluttergedrag	0	0	0	0	0
50	s	110010	SWICO af, Onbezet, fout en fluttergedrag	0	0	0	0	0
51	t	110011	SWICO af, Bezet, fout en fluttergedrag	0	0	0	0	0
52	u	110100	SWICO af, Onbezet, ondergedrag en fluttergedrag	0	0	0	0	0
53	v	110101	SWICO af, Bezet, bovengedrag en fluttergedrag	0	0	0	0	0
54	w	110110	SWICO af, Onbezet, fout, ondergedrag en fluttergedrag	0	0	0	0	0
55	x	110111	SWICO af, Bezet, fout, bovengedrag en fluttergedrag	0	0	0	0	0
56	X	111000	illegale toestand	0	0	0	0	0
57	X	111001	illegale toestand	0	0	0	0	0
58	X	111010	illegale toestand	0	0	0	0	0
59	X	111011	illegale toestand	0	0	0	0	0
60	X	111100	illegale toestand	0	0	0	0	0
61	X	111101	illegale toestand	0	0	0	0	0
62	X	111110	illegale toestand	0	0	0	0	0
63	X	111111	illegale toestand	0	0	0	0	0

Tabel 18.6 Detector toestand (alle combinaties)

Ter verduidelijking een aantal voorbeelden met detector 021 als toegepast element:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
D.A/021	D.A/021=1	Opvragen toestand detector 021, de respons is gelijk aan de omschrijving "detector toestand" uit de tabel x.y. 1 = bezet detector
D.A/021	D.A/021=51	Opvragen toestand detector 021, de respons is gelijk aan: SWICO af, bezet, fout en fluttergedrag

Tabel 18.7 Voorbeelden status detectoren

18.2.3 TD.A

Het object T.DA geeft de lopende timer bezet/onbezet weer van alle detectoren:

Attribuut	Type	Object	Omschrijving
N	1	TD.A	Naam
O	1	Timer bezet/onbezet	Omschrijving
T	0	0	Type
U	0	4444	User Identificatie Control
E	0	NUMD	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2, getal met de eenheid 0.1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.8 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
T.DA/#0	T.DA/021=1234	Actuele bezet/onbezetduur.

Tabel 18.9 Voorbeelden timers detectoren

18.2.4 SWD

Het object SWD geeft de SWICO toestand weer van alle detectoren:

Attribuut	Type	Object	Omschrijving
N	1	SWD	Naam
O	1	Software detectorschakelaar	Omschrijving
T	0	0	Type
U	0	6664	User Identificatie Control
E	0	NUMD	aantal data-elementen
I	1	D.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	2	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	11	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.10 Object attributen

Formaat: Software Input Commando (11), type 0

0=neutraal, 1=uit, 2=aan

De volgende tabel bevat enkele voorbeelden met element 021.

Voorbeeld	Respons vanuit de applicatie	Omschrijving
SWD/021	SWD/021=0	Opvragen SWICO toestand detector 021, de respons is gelijk aan : 0 = verkeersafhankelijk 1 = UIT 2 = AAN

Tabel 18.11 Voorbeelden SWICO status detectoren

18.2.5 D.LB

Het object D.LB geeft het detectorlogboek weer:

Attribuut	Type	Object	Omschrijving
N	1	D.LB	Naam
O	1	Detector logboek	Omschrijving
T	0	1	Type
U	0	4444	User Identificatie Control
E	0	0 .. 1000	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	40, toestand van alle detectoren	Data-element formaat
S	0		Data-element stapgrootte

Tabel 18.12 Object attributen

Verwezen wordt naar VRI.LA en VRI.LB voor de detailbeschrijving.

De volgende tabel bevat enkele voorbeelden met element 021 (1^e element in het logboek):

Voorbeeld	Respons vanuit de applicatie	Omschrijving
D.LB/#0	D.LB/#0=202300:1	Opvragen toestand detector 021 uit het logboek, de respons is gelijk aan de omschrijving "detector toestand" uit de tabel x.y. 202300 = tijdstempel 1 = bezet detector
D.LB/#0	D.LB/#0=202301:t	Opvragen toestand detector 021 uit het logboek, de respons is gelijk aan: 202301 = tijdstempel t = SWICO af, bezet, fout en fluttergedrag

Tabel 18.13 Voorbeelden status detectoren in het logboek

18.2.6 D.LA

Het object D.LA geeft het detectorlogboek weer (onbevestigd):

Attribuut	Type	Object	Omschrijving
N	1	D.LA	Naam
O	1	Detector logboek (onbevestigd)	Omschrijving
T	0	1	Type
U	0	6666	User Identificatie Control
E	0	0 .. 1000	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	40, toestand van alle detectoren	Data-element formaat
S	0		Data-element stapgrootte

Tabel 18.14 Object attributen

Verwezen wordt naar VRI.LA en VRI.LB voor de detailbeschrijving.

De volgende tabel bevat enkele voorbeelden met element 021 (1^e element in het logboek):

Voorbeeld	Respons vanuit de applicatie	Omschrijving
D.LA/#0	D.LA/#0=202300:1	Opvragen toestand detector 021 uit het logboek, de respons is gelijk aan de omschrijving "detector toestand" uit de tabel x.y. 202300 = tijdstempel 1 = bezet detector
D.LA/#0	D.LA/#0=202301:t	Opvragen toestand detector 021 uit het logboek, de respons is gelijk aan: 202301 = tijdstempel t = SWICO af, bezet, fout en fluttergedrag

Tabel 18.15 Voorbeelden status detectoren in het logboek

18.2.7 TDBG

Het object TDBG geeft de bewakingstijd bovengedrag weer van alle detectoren:

Attribuut	Type	Object	Omschrijving
N	1	TDBG	Naam
O	1	Bewakingstijd bovengedrag	Omschrijving
T	0	0	Type
U	0	6644	User Identificatie Control
E	0	NUMD	aantal data-elementen
I	1	D.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	20	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.16 Object attributen

Formaat: Klok (uur/minuut) (20), type 0

De kloktijd is een getal met de tijd in het volgende formaat:

Waargave: UUMM

Waarde: uur * 100 + minuut

De volgende tabel bevat enkele voorbeelden met element 021.

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TDBG/021=0	TDBG/021=0	Detector wordt niet bewaakt op bovengedrag.
TDBG/021	TDBG/021=0	Opvragen huidige instelling
TDBG/021=100	TDBG/021=100	Detector wordt niet bewaakt op bovengedrag gedurende 1 uur.
TDBG/021	TDBG/021=100	Opvragen huidige instelling

Tabel 18.17 Voorbeelden detectiebewaking

18.2.8 TDOG

Het object TDOG geeft de bewakingstijd ondergedrag weer van alle detectoren:

Attribuut	Type	Object	Omschrijving
N	1	TDOG	Naam
O	1	Bewakingstijd ondergedrag	Omschrijving
T	0	0	Type
U	0	6644	User Identificatie Control
E	0	NUMD	aantal data-elementen
I	1	D.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	20	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.18 Object attributen

Formaat: Klok (uur/ minuut), type 0

De kloktijd is een getal met de tijd in het volgende formaat:

Waargave: UUMM

Waarde: uur * 100 + minuut

De volgende tabel bevat enkele voorbeelden met element 021:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TDOG/021=0	TDOG/021=0	Detector wordt niet bewaakt op ondergedrag.
TDOG/021	TDOG/021=0	Opvragen huidige instelling
TDOG/021=100	TDOG/021=100	Detector wordt niet bewaakt op ondergedrag gedurende 1 uur.
TDOG/021	TDOG/021=100	Opvragen huidige instelling

Tabel 18.19 Voorbeelden detectiebewaking

18.2.9 TDFL

Het object TDFL geeft de meettijd fluttergedrag weer van alle detectoren:

Attribuut	Type	Object	Omschrijving
N	1	TDFL	Naam
O	1	Meettijd fluttergedrag	Omschrijving
T	0	0	Type
U	0	6644	User Identificatie Control
E	0	NUMD	aantal data-elementen
I	1	D.I	Index verwijzing per dimensie
MIN	0	-1	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met de eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.20 Object attributen

De volgende tabel bevat enkele voorbeelden met element 021.

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TDFL/021=60	TDFL/021=60	Meettijd fluttergedrag ingesteld op 60 seconden.
TDFL/#0	TDFL/021=60	Opvragen huidige instelling van het 1 ^e element.

Tabel 18.21 Voorbeelden detectiebewaking

18.2.10 CDFL

Het object CDFL geeft de meettijd fluttergedrag weer van alle detectoren:

Attribuut	Type	Object	Omschrijving
N	1	CDFL	Naam
O	1	Grenswaarde fluttergedrag	Omschrijving
T	0	0	Type
U	0	6644	User Identificatie Control
E	0	NUMD	aantal data-elementen
I	1	D.I	Index verwijzing per dimensie
MIN	0	-1	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met de eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.22 Object attributen

De volgende tabel bevat enkele voorbeelden met element 021:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
CDFL/021=40	CDFL/021=40	Grenswaarde fluttergedrag ingesteld op 40.
CDFL/#0	CDFL/021=40	Opvragen huidige instelling van het 1 ^e element.

Tabel 18.23 Voorbeelden detectiebewaking

18.2.11 TDBP1

Het object TDBP1 geeft de tijd detectiebewaking AAN weer van alle detectoren:

Attribuut	Type	Object	Omschrijving
N	1	TDBP1	Naam
O	1	Tijd detectiebewaking aan	Omschrijving
T	0	0	Type
U	0	6644	User Identificatie Control
E	0	7	aantal data-elementen
I	1	DAG.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	2400	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	20	Data-element formaat

S	0	1	Data-element stapgrootte
---	---	---	--------------------------

Tabel 18.24 Object attributen

Formaat: Klok (uur/minuut), type 0

De kloktijd is een getal met de tijd in het volgende formaat:

Waargave: UUMM

Waarde: uur * 100 + minuut

Voorbeelden zijn opgenomen in de paragraaf 18.2.12.

18.2.12 TDBP2

Het object TDBP2 geeft de tijd detectiebewaking UIT weer van alle detectoren:

Attribuut	Type	Object	Omschrijving
N	1	TDBP2	Naam
O	1	Tijd detectiebewaking uit	Omschrijving
T	0	0	Type
U	0	6644	User Identificatie Control
E	0	7	aantal data-elementen
I	1	DAG.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	2400	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	20	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.25 Object attributen

Formaat: Klok (uur/minuut), type 0

De kloktijd is een getal met de tijd in het volgende formaat:

Waargave: UUMM

Waarde: uur * 100 + minuut

De volgende tabel bevat enkele voorbeelden met element 021:

Voorbeeld	Omschrijving
TDOG/021=800 TDBP1/021=600 TDBP2/021=2200	Als een detector gedurende 8 uur onbezet is tussen 06:00 en 22:00 wordt het alarm ondergedrag gezet.
TDOG/021=4800 TDBP1/021=600 TDBP2/021=2200	De detectiebewaking is gedurende 16 uur per dag actief. Een bewakingstijd op ondergedrag van 48 uur komt dan overeen met een tijd van 48/16=3 dagen. Gedurende de periode dat er niet wordt bewaakt, wordt de toestand van de detector niet getoetst. Het opkomen van de detector in deze periode wordt genegeerd.

Tabel 18.26 Voorbeelden detectiebewaking in combinatie met TDBP1 en TDBP2.

18.2.13 DC.I

Het object DC.I geeft de index weer van het object DC van alle snelheidsdetectoren:

Attribuut	Type	Object	Omschrijving
N	1	DC.I	Naam
O	1	Index object classificatietoestand	Omschrijving
T	0	1	Type
U	0	4444	User Identificatie Control
E	0	NUMDC	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 18.27 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
DC.I	DC.I= <classificatiedetector_1>, <classificatiedetector_2>,..., <classificatiedetector_n>	Het antwoord bevat de namen van de classificatiedetectors. Het resultaat mag in geen geval lege velden tonen

Tabel 18.28 Voorbeelden DC.I

18.2.14 DC.A

Het object DC.A geeft de classificatietoestand weer van alle snelheidsdetectoren:

Attribuut	Type	Object	Omschrijving
N	1	DC.A	Naam
O	1	classificatietoestand	Omschrijving
T	0	1	Type
U	0	4444	User Identificatie Control
E	0	NUMDC	aantal data-elementen
I	1	DC.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	41	Data-element formaat
S	0		Data-element stapgrootte

Tabel 18.29 Object attributen

Formaat: Toestand van een classificatiedetector(41), type 1

Voor classificatiedetectors wordt de volgende syntax gehanteerd:

[Snelheid] + “,” + [Lengte] + “,” + [Hoogte] + “,” + [Breedte] + “,” + [Klasse-indeling] + “,” + [Klasse] + “,” + [Volume] + “,” + [Gewicht] + “,” + [Aantal personen] + “,” + [Betrouwbaar] + “,” + [Richting] + “,” + [Tijd onbezet]

Snelheid = = snelheid in hectometers/uur

Lengte = lengte in decimeters

Hoogte = hoogte in decimeters

Breedte = breedte in decimeters

Klasse-indeling = 0 | 1 | 2

Klasse = 0..12

Volume = volume in kubieke meters

Gewicht = gewicht in decakilo's

Aantal personen = 0..99

Betrouwbaar (nee | ja) = 0 | 1

Richting (normaal | tegengesteld) = 0 | 1

Tijd onbezet = tijd sinds laatste meting in tienden van seconden

Wanneer de detector Klasse-indeling 0 gebruikt, is de indeling van de voertuigen in klassen –conform de definitie van de CVN-interface versie 3.0- als volgt:

Waarde	Omschrijving
0	geen voertuigpassage/informatie
1	personenauto
2	vrachtwagen
3	bus
4	personenauto + aanhanger
5	vrachtwagen + aanhanger
6	niet gebruikt
7	ongeldig voertuig

Tabel 18.30 Voertuigklassen volgens indelingstype 0

Wanneer de detector Klasse-indeling 1 gebruikt, is -de door de FHWA opgestelde- indeling van de voertuigen in klassen als volgt:

Element	Omschrijving
0	motorfietsen
1	personenauto's
2	andere ongelede voertuigen met 2 assen en 4 wielen
3	autobussen
4	ongelede vrachtwagens met 2 assen en 6 wielen
5	ongelede vrachtwagens met 3 assen
6	ongelede vrachtwagens met 4 of meer assen
7	vrachtwagens met 1 aanhanger en max. 4 assen
8	vrachtwagens met 1 aanhanger en 5 assen
9	vrachtwagens met 1 aanhanger en 6 of meer assen
10	vrachtwagens met meer dan 1 aanhanger en max 5 assen
11	vrachtwagens met meer dan 1 aanhanger en 6 assen
12	vrachtwagens met meer dan 1 aanhanger en 7 of meer assen

Tabel 18.31 Voertuigklassen volgens indelingstype 1

Wanneer de detector Klasse-indeling 2 gebruikt, is de indeling van de voertuigen in klassen als volgt:

Waarde	Omschrijving
0	voetgangers
1	wielrijders/snorfietsen/bromfietsen
2	motorfietsen
3	personenauto's
4	personenauto's met aanhanger
5	bestelwagens
6	ongelede vrachtwagens
7	trucks met oplegger
8	vrachtwagens met aanhangwagen
9	autobussen

Tabel 18.32 Voertuigklassen volgens indelingstype 2

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
DC.A	DC.A="<[Snelheid] + "," + [Lengte] + "," + [Hoogte] + "," + [Breedte] + "," + [Klasse-indeling] + "," + [Klasse] + "," + [Volume] + "," + [Gewicht] + "," + [Aantal personen] + "," + [Betrouwbaar] + "," + [Richting] + "," + [Tijd onbezet]>"	Vraag de toestand op
DC.A/#0="10, 10, 10, 10, 1, 6, 10, 10, 10, 1, 1, 0,3"	:E=11	Schrijven is niet toegestaan.

Tabel 18.33 Voorbeelden DC.A

18.2.15 LD

Het object LD geeft de lijndump detectortoestand weer van alle detectoren:

Attribuut	Type	Object	Omschrijving
N	1	LD	Naam
O	1	Lijndump detectortoestand	Omschrijving
T	0	1	Type
U	0	4444	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	40	Data-element formaat
S	0		Data-element stapgrootte

Tabel 18.34 Object attributen

Formaat: Toestand van alle detectoren (40), type 1

Het object geeft een string met daarin een timestamp en voor iedere detector een karakter die de actuele toestand weergeeft. De timestamp geeft aan het moment van de laatste wijziging.

```
DetToestand = HMS_timestamp ":" { DetKarakter }
HMS_timestamp = "hhmmss"
DetKarakter = /* zie tabel detectortoestand */
```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
LD/#0	LD/#0=<waarde>	waarbij <waarde> de volgende syntax heeft: HMS_timestamp ":" { DetKarakter } HMS_timestamp = "hhmmss" DetKarakter = /* zie tabel detectortoestand */
LD:A	LD:A="N=LD,T=1,E=1,U=4444,F=40,O='Lijndump detector toestand'"	

Tabel 18.35 Voorbeelden LD

18.3 Detectie fout event

Met de detectiefout events worden wijzigingen in de foutstatus (bovengedrag, ondergedrag, lusfout) en wijzigingen van de software schakelaar gemeld aan de beheerscentrale. Ieder event bericht bevat de volledige foutstatus per detector inclusief de actuele bezet/onbezet toestand.

Code	Omschrijving	Detailinfo
1020	Detectiefout	<DetRef>,<Bezet>,<BOV>,<OND>,<LF>,<SW> [,<STBOV>,<STOND>,<STLF>[,<FLT>,<STFLT>]]

Tabel 18.36 Detectiefout event

DetRef	Detectornaam of -nummer	
Bezet	Bezet/onbezet	0 1
BOV	Bovengedrag	0 1
OND	Ondergedrag	0 1
LF	Lusfout	0 1
SW	Softwareschakelaar	0 1 2
STBOV(*)	Bovengedrag (geen melding, start, einde)	0 1 2
STOND(*)	Ondergedrag (geen melding, start, einde)	0 1 2
STLF(*)	Lusfout (geen melding, start, einde)	0 1 2
FLT(*)	Flutter	0 1
STFLT(*)	Flutter (geen melding, start, einde)	0 1 2

NB: Het invullen van de velden <STBOV>, <STOND>, <STLF> en <FLT>, <STFLT> is optioneel. Als bekend is, of het om het begin of het einde van een van de drie fouttoestanden gaat, kan dat in <STBOV>, <STOND>, <STLF> of <STFLT> aangegeven worden.

NB: De velden aangemerkt met () zijn optioneel om in te vullen. Als bekend is, of het om het begin of het einde van een van de drie fouttoestanden gaat, kan dat in <STBOV>, <STOND>, <STLF> of <STFLT> aangegeven worden.*

Ter verduidelijking een voorbeeld met detector 021 als toegepast element:

Code	Voorbeeld
1020	"021",1,0,0,0,0,0,0,0,0

Reset- en Commando events

Via het object VRI.C kunnen (reset)commando's worden gegeven aan de VRI. De codes vormen de subgroepen 4000 – 4999 en 5000..5999) in de eventcodes.

Resetcommando's gegevens worden via het object VRI.C met een deel van de subgroep (4000..4999) in de eventcodes zoals die in dezelfde paragraaf benoemd worden.

Code	Omschrijving	Detailinfo
4001	Reset van alle storingen	
4002	Reset van detectiealarmen	

Tabel 18.37 Reset events

Commando-events hebben betrekking op commando's in de VRI. Zo wordt bijvoorbeeld het commando test puts gemeld als eventcode 5001. Daarin tegen is het ook mogelijk om de puts te testen door het schrijven van de waarde 5001 naar het object "VRI.C".

NB. Indien een VRI een commando niet ondersteunt, antwoordt de VRI met de foutcode ERR_DATA.

Code	Omschrijving
5001	Test putsarmatuur

Tabel 18.38 Commando events

19. In- en uitgangen

19.1 Algemeen in- en uitgangen

Voor de in- en uitgangen zijn de volgende objecten gedefinieerd:

Object	Omschrijving
U.I	Index overige uitgangen
U.A	Toestand overige uitgangen
TU.A	Timer uitgangstoestand
U.LB	Uitgangenlogboek
U.LA	Uitgangenlogboek (onb)
I.I	Index overige ingangen
I.A	Toestand overige ingangen
TI.A	Timer ingangstoestand
SWI	Software inputschakelaar
I.LB	Ingangenlogboek
I.LA	Ingangenlogboek (onb)
LI	Lijndump ingangtoestand
LU	Lijndump uitgangtoestand

Tabel 19.1 In- en uitgangen objecten

(overige) uitgangen

Voor de overige uitgangen zijn de volgende objecten gedefinieerd: U.A (actuele toestand), U.I (Index).

(overige) ingangen

Voor de overige ingangen zijn de volgende objecten gedefinieerd: I.A (actuele toestand), I.I (Index) en SWI (SWICO).

Voor zowel de in- als de uitgangen zijn de toestanden, timer toestanden, lijndump en logboeken beschikbaar.

19.2 Objecten in- en uitgangen

19.2.1 U.I

Het object U.I geeft index overige uitgangen weer van alle uitgangen:

Attribuut	Type	Object	Omschrijving
N	1	U.I	Naam
O	1	Index overige uitgangen	Omschrijving
T	0	1	Type
U	0	4444	User Identificatie Control
E	0	NUMU	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 19.2 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Het standaardiseren van namen van verschillende parameters is niet zozeer een IVERA protocol aanpassing, als wel het maken van een extra restrictie op het gebruik van namen in een aantal indexobjecten. Het gaat hierbij over de indexobjecten, die gekoppeld zijn aan de parameters in CCOL of RWS-C, die reeds een naam hebben zoals de volgende indexobjecten:

De namen in deze objecten moeten exact overeenkomen met de namen die in CCOL of RWS-C zijn geconfigureerd. Zij bepaalde namen leeg gelaten, dan wordt in IVERA een invulling aangegeven, die overeenkomt met het elementnummer en het type parameter.

De n^e lege indexnaam wordt in IVERA ingevuld met 'U_(n-1)', waarbij (n-1) een getal weergeeft.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
U.I	U.I=<Waarde_1>,<Waarde_2>,<Waarde_n>	Opvragen index overige uitgangen. Het resultaat mag in geen geval lege velden tonen.
U.I/#0="Test"	:E=11	Schrijven is niet toegestaan.

Tabel 19.3 Voorbeelden U.I

19.2.2 U.A

Het object U.A geeft de toestand overige uitgangen weer van alle uitgangen:

Attribuut	Type	Object	Omschrijving
N	1	U.A	Naam
O	1	Toestand overige uitgangen	Omschrijving
T	0	0	Type
U	0	4444	User Identificatie Control
E	0	NUMU	aantal data-elementen
I	1	U.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 19.4 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
U.A	U.A="<toestandswaarde_1>,<toestandswaarde_1>,...,..."	Lees het object U.A. VRI automaat weergeeft waarden in de vorm van type 0 (== getal)
U.A:A	U.A:A="N=U.A,T=0,E=<NUMU>,U=4444,I=U.I,MIN=0,S=1,F=1,O='Toestand overige uitgangen'"	Vraag attributen op van object U.A
U.A:E	U.A:E=<NUMU>	Vraag aantal elementen op van object U.A

Tabel 19.5 Voorbeelden U.A

19.2.3 TU.A

Het object TU.A geeft de timer overige uitgangen weer van alle uitgangen:

Attribuut	Type	Object	Omschrijving
N	1	TU.A	Naam
O	1	timer uitgangstoestand	Omschrijving
T	0	0	Type
U	0	4444	User Identificatie Control
E	0	NUMU	aantal data-elementen
I	1	U.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2, getal met eenheid 0.1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 19.6 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TU.A	TU.A="<waarde_1>,<waarde_2>,...,etc"	Lees het object TU.A. VRI automaat weergeeft waarden van type 0 (== getal)
TU.A:A	TU.A:A="N=TU.A,T=0,E=<NUMU>,U=4444,I=U.I,MIN=0,S=1,F=2,O='Timer bezet/onbezet'"	Vraag attributen op van object TU.A
TU.A/#0	TU.A/#0=<waarde_1>	Vraag een element op van het object TU.A

Tabel 19.7 Voorbeelden TU.A

19.2.4 U.LB

Het object U.LB geeft uitgangenlogboek weer van alle uitgangen:

Attribuut	Type	Object	Omschrijving
N	1	U.LB	Naam
O	1	uitgangenlogboek	Omschrijving
T	0	1	Type
U	0	4444	User Identificatie Control
E	0	0 .. 1000	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	50	Data-element formaat
S	0		Data-element stapgrootte

Tabel 19.8 Object attributen

Formaat: Toestand van overige in- en uitgangen (50), type 1

Het object geeft een string met daarin een timestamp en voor iedere in/uitgang een karakter die de actuele toestand weergeeft. De timestamp geeft aan het moment van de laatste wijziging.

```

IOTOestand = HMS_timestamp + "." + { IOKarakter }
HMS_timestamp = "hhmmss"
IOKarakter = "0" | "1"

```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
U.LB		De automaat retourneert de huidige inhoud van het object U.LB

Tabel 19.9 Voorbeelden U.LB

19.2.5 U.LA

Het object U.LA geeft uitgangenlogboek (onbevestigd) weer van alle uitgangen:

Attribuut	Type	Object	Omschrijving
N	1	U.LA	Naam
O	1	Uitgangenlogboek (onbevestigd)	Omschrijving
T	0	1	Type
U	0	6666	User Identificatie Control
E	0	0 .. 1000	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	50	Data-element formaat
S	0		Data-element stapgrootte

Tabel 19.10 Object attributen

Formaat: Toestand van overige in- en uitgangen (50), type 1

Het object geeft een string met daarin een timestamp en voor iedere in/uitgang een karakter die de actuele toestand weergeeft. De timestamp geeft aan het moment van de laatste wijziging.

```

IOToestand = HMS_timestamp + ":" + { IOKarakter }
HMS_timestamp = "hhmmss"
IOKarakter = "0" | "1"

```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
U.LA	Retourneert inhoud U.LA	Lees de huidige inhoud van het object U.LA uit
U.LA/#0=""	U.LA/#0=""	Bevestig het eerste element in het object U.LA
U.LA	:E=17	Nadat alle elementen zijn bevestigd retourneert de automaat de code :E=17 (ERR_EMPTY) indien de inhoud van U.LA wordt opgevraagd

Tabel 19.11 Voorbeelden U.LA

19.2.6 I.I

Het object I.I geeft index overige ingangen weer van alle ingangen:

Attribuut	Type	Object	Omschrijving
N	1	I.I	Naam
O	1	Index overige ingangen	Omschrijving
T	0	1	Type
U	0	4444	User Identificatie Control
E	0	NUMI	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 19.12 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Het standaardiseren van namen van verschillende parameters is niet zozeer een IVERA protocol aanpassing, als wel het maken van een extra restrictie op het gebruik van namen in een aantal indexobjecten. Het gaat hierbij over de indexobjecten, die gekoppeld zijn aan de parameters in CCOL of RWS-C, die reeds een naam hebben zoals de volgende indexobjecten:

De namen in deze objecten moeten exact overeenkomen met de namen die in CCOL of RWS-C zijn geconfigureerd. Zijn bepaalde namen leeg gelaten, dan wordt in IVERA een invulling aangegeven, die overeenkomt met het elementnummer en het type parameter.

De n^e lege indexnaam wordt in IVERA ingevuld met 'I_(n-1)', waarbij (n-1) een getal weergeeft.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
I.I		Vraag de waarde van I. Op. De gemelde namen van de ingangen hebben het juiste formaat;het resultaat mag in geen geval lege velden tonen

Tabel 19.13 Voorbeelden I.I

19.2.7 I.A

Het object I.A geeft de toestand overige ingangen weer van alle ingangen:

Attribuut	Type	Object	Omschrijving
N	1	I.A	Naam
O	1	Toestand overige ingangen	Omschrijving
T	0	0	Type
U	0	4444	User Identificatie Control
E	0	NUMI	aantal data-elementen
I	1	I.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 19.14 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
I.A/#0	I.A/#0=<waarde_ingangstoestand_1>	Vraag een data-element op van object I.A
I.A/#0=<nieuwe_waarde>	:E=11	Wijzig de waarde ervan met een andere geldige waarde

Tabel 19.15 Voorbeelden I.A

19.2.8 TI.A

Het object TI.A geeft de timer overige ingangen weer van alle ingangen:

Attribuut	Type	Object	Omschrijving
N	1	TI.A	Naam
O	1	timer ingangstoestanden	Omschrijving
T	0	0	Type
U	0	4444	User Identificatie Control
E	0	NUMI	aantal data-elementen
I	1	I.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2, getal met eenheid 0.1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 19.16 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TI.A		De gemelde duur van de ingangstoestand komt overeen met die in de automaat.

Tabel 19.17 Voorbeelden TI.A

19.2.9 SWI

Het object SWI geeft de software inpuitschakelaar weer van alle ingangen:

Attribuut	Type	Object	Omschrijving
N	1	SWI	Naam
O	1	Software inpuitschakelaar	Omschrijving
T	0	0	Type
U	0	6644	User Identificatie Control
E	0	NUMI	aantal data-elementen
I	1	I.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	2	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	11	Data-element formaat
S	0		Data-element stapgrootte

Tabel 19.18 Object attributen

Formaat: Software Input Commando (11), type 0

0=neutraal, 1=uit, 2=aan

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
SWI/#0	SWI/#0=0	Vraag een data element (ingang) op
SWI/#0=1	SWI/#0=1	Wijzig de waarde van dit element met een geldige waarde (0, 1 of 2)
SWI/#0=1	:E=11	Wijzig de waarde van dit element met een geldige waarde (0, 1 of 2), inlogniveau gelijk aan 1 of 2.

Tabel 19.19 Voorbeelden SWI

19.2.10 I.LB

Het object I.LB geeft ingangenlogboek weer van alle ingangen:

Attribuut	Type	Object	Omschrijving
N	1	I.LB	Naam
O	1	ingangenlogboek	Omschrijving
T	0	1	Type
U	0	4444	User Identificatie Control
E	0	0 .. 1000	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	50	Data-element formaat
S	0		Data-element stapgrootte

Tabel 19.20 Object attributen

Formaat: Toestand van overige in- en uitgangen (50), type 1

Het object geeft een string met daarin een timestamp en voor iedere in/uitgang een karakter die de actuele toestand weergeeft. De timestamp geeft aan het moment van de laatste wijziging.

```

IOToestand = HMS_timestamp + ":" + { IOKarakter }
HMS_timestamp = "hhmmss"
IOKarakter = "0" | "1"

```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
I.LB "d.lb"	De automaat retourneert een lijst met de huidige toestand van de ingangen in het logboek I.LB, Bijv: I.LB="115537:000000000000 000000000"	Lees de huidige inhoud van het ingangen logboek bevestigd

Tabel 19.21 Voorbeelden I.LB

19.2.11 I.LA

Het object I.LA geeft ingangenlogboek (onbevestigd) weer van alle ingangen:

Attribuut	Type	Object	Omschrijving
N	1	I.LA	Naam
O	1	Ingangenlogboek (onbevestigd)	Omschrijving
T	0	1	Type
U	0	6666	User Identificatie Control
E	0	0 .. 1000	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	50	Data-element formaat
S	0		Data-element stapgrootte

Tabel 19.22 Object attributen

Formaat: Toestand van overige in- en uitgangen (50), type 1

Het object geeft een string met daarin een timestamp en voor iedere in/uitgang een karakter die de actuele toestand weergeeft. De timestamp geeft aan het moment van de laatste wijziging.

```

IOToestand = HMS_timestamp + ":" + { IOKarakter }
HMS_timestamp = "hhmmss"
IOKarakter = "0" | "1"

```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
I.LA	De automaat toont de huidige inhoud van het object I.LA, waarvan het bevestigde element is verwijderd.	Lees de huidige inhoud van het object I.LA uit
I.LA/#0=""	I.LA/#0=""	Bevestig het eerste element in het object I.LA
I.LA	:E=17	Nadat alle elementen zijn bevestigd retourneert de automaat de code :E=17 (ERR_EMPTY) indien de inhoud van I.LA wordt opgevraagd

Tabel 19.23 Voorbeelden I.LA

19.2.12 LI

Het object LI geeft lijndump ingangtoestand weer van alle ingangen:

Attribuut	Type	Object	Omschrijving
N	1	LI	Naam
O	1	Lijndump ingangtoestand	Omschrijving
T	0	1	Type
U	0	4444	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	50	Data-element formaat
S	0		Data-element stapgrootte

Tabel 19.24 Object attributen

Formaat: Toestand van overige in- en uitgangen (50), type 1

Het object geeft een string met daarin een timestamp en voor iedere in/uitgang een karakter die de actuele toestand weergeeft. De timestamp geeft aan het moment van de laatste wijziging.

```

IOToestand = HMS_timestamp + ":" + { IOKarakter }
HMS_timestamp = "hhmmss"
IOKarakter = "0" | "1"

```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
LI	LI=HMS_timestamp + ":" + { IOKarakter }	De lijndump geeft de actuele status van de overige ingangen: HMS_timestamp = "hhmmss" IOKarakter = "0" of "1" De gemelde ingangstoestand komt overeen met die in de automaat.
LI/#0="123"	:E=11	Schrijven niet toegestaan.

Tabel 19.25 Voorbeelden LI

19.2.13 LU

Het object LI geeft lijndump uitgangstoestand weer van alle uitgangen:

Attribuut	Type	Object	Omschrijving
N	1	LU	Naam
O	1	Lijndump uitgangstoestand	Omschrijving
T	0	1	Type
U	0	4444	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	50	Data-element formaat
S	0		Data-element stapgrootte

Tabel 19.26 Object attributen

Formaat: Toestand van overige in- en uitgangen (50), type 1

Het object geeft een string met daarin een timestamp en voor iedere in/uitgang een karakter die de actuele toestand weergeeft. De timestamp geeft aan het moment van de laatste wijziging.

```

IOToestand = HMS_timestamp + ":" + { IOKarakter }
HMS_timestamp = "hhmmss"
IOKarakter = "0" | "1"

```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
LU	LU=HMS_timestamp + ":" + { IOKarakter }	De lijndump geeft de actuele status van de overige uitgangen: HMS_timestamp = "hhmmss" IOKarakter = "0" of "1" De gemelde uitgangstoestand komt overeen met die in de automaat.
LU/#0="123"	:E=11	Schrijven niet toegestaan.

Tabel 19.27 Voorbeelden LU

20. Signaalgroepen

20.1 Algemeen signaalgroepen

De regeltechnische communicatie naar de weggebruiker vindt plaats met behulp van signaalgroepen en uitgangsignalen. Dit hoofdstuk omschrijft alle objecten met betrekking tot de signaalgroepen.

Binnen IVERA zijn de signaalgroepen onder te verdelen in de fysieke signaalgroep afhandeling, de bijbehorende tijdsinstellingen, dimmen van de signaalgroepen, de akoestische signaalgevers en het opvragen en/of loggen van de signaalgroep toestanden.

Dit hoofdstuk heeft betrekking op alle objecten die onder de index SG.I vallen.

Het hoofdstuk / deze paragraaf is opgedeeld meerdere onderdelen:

- De signaalgroep toestand objecten;
- De garantietijden objecten
- De lampinformatie;
- Het dimmen van de signaalgroepen;
- Het aansturen van de akoestische signalen.

Voor signaalgroepen zijn de volgende objecten gedefinieerd:

Object	Omschrijving
SG.I	Index
SGE.A	Actuele toestand extern (rood, groen, geel, etc).
TSGE.A	Lopende timer externe signaalgroep toestand.
SGE.LB	Logboek externe signaalgroep toestand (laatste xx veranderingen).
SGE.LA	Logboek externe signaalgroep toestand (onbevestigd).
TGOR1	Garantie ontruimingstijden zoals gedefinieerd in de procesbesturing.
TGG1	Garantiegroentijd (proces)
TGGL1	Garantiegeeltijd (proces)
TMGL1	Maximum geeltijd (proces)
TGR1	Garantieroodtijd (proces)
LAMP.I	Index lampnamen
LAMP.A	Actuele lampstatus
LAMPINFO	Lampconfiguratie
DIMINST.I	Index diminstellingen
DIMINST	Diminstellingen
DIMMEN.I	Index dimstatus
DIMMEN.A	Dimstatus
AKOESTISCH.I	Index status akoestische signalen
AKOESTISCH.A	Status akoestische signalen
AKOESTISCH.F	Foutstatus akoestische signalen
KLA_AKOEST	Inschakelen akoestische signalen
KLU_AKOEST	Uitschakelen akoestische signalen
KLA_HARD	Inschakelen hoog geluidsvolume
KLU_HARD	Uitschakelen hoog geluidsvolume
PAKOESTISCH.I	Index parameter akoestische signalen
PAKOESTISCH	Parameter akoestische signalen
LSGE	Lijndump SG-toestand (ext)

Tabel 20.1 Signaalgroep objecten

Signaalgroep toestanden

Signaalgroepen kennen de toestanden groen, geel en rood als extern zichtbare toestand. Intern kent elke signaalgroep meer toestanden, waaronder vast- verleng- en meeverlengroen. De interne toestanden zijn applicatie afhankelijk en er zijn verschillen tussen de applicatiepakketten Ccol en RWS-C.

Logboeken

Alle signaalgroep gebeurtenissen worden opgeslagen in een logboek. Hiermee is naderhand de geschiedenis, in beperkte mate van signaalgroepovergangen, te herleiden. Elke verandering wordt met een tijdstempel gelogd in het betreffende logboek.

Garantietijden

In de regelprocessor van het verkeersregeltoestel zijn een aantal ontruimingstijden gedefinieerd. Normaliter worden de signaalgroepen vanuit het applicatiepakket conflictvrij aangestuurd en door de procesbesturing doorgegeven aan de lampschakelaars. Het applicatiepakket gebruikt hiervoor de ontruimingstijden. Waarbij geldt dat ontruimingstijden nooit lager ingesteld mogen worden dan de garantie ontruimingstijden (TGOR1 in de procesbesturing)

De objecten TGG1, TGGL1 en TGR1 bevatten de garantietijden zoals gedefinieerd in de procesbesturing.

De lampinformatie

Per signaalgroep kunnen meerdere lampen zijn aangesloten. De configuratie over deze lampen wordt binnen IVERA vastgelegd. Per lamp is informatie beschikbaar als gemeten stroom, aantal lampen (bij signaalgroep georiënteerde meting) en defecte lampen.

Voor lampen worden binnen IVERA, conform de Regeling Verkeerslichten (RVV), de volgende definitie gehanteerd.

Verkeerslantaarn	Toestel voor het tonen van verkeerslichten.
Verkeerslicht	Driekleurige verkeerslichten, tweekleurige verkeerslichten en tram/bus-lichten.
Licht	Een verkeerslantaarn is samengesteld uit lichten.
Signaalgroep	Een verzameling van verkeerslichten die hetzelfde lichtbeeld tonen.

Tabel 20.2 Lampdefinitie

Een lamp binnen IVERA is synoniem voor een licht dan wel een verzameling van lichten. Een lamp heeft te allen tijde slechts betrekking op één signaalgroep.

De doelstelling van de objectmodellering van lampen binnen IVERA is:

- Het kunnen bepalen van de actuele status van één licht (individuele lampbewaking).
- Het kunnen bepalen van de actuele status van een verzameling van lichten (lampbewaking per signaalgroep).
- Het kunnen uitlezen en eventueel kunnen wijzigen van de instellingen ten aanzien van de lampbewaking.

Naast de standaard IVERA parameters is het mogelijk om fabrikantspecifieke parameters (X objecten) te definiëren waarmee de instellingen van de lampbewaking bekeken dan wel gewijzigd kunnen worden. Aanvullende fabrikantspecifieke statusinformatie over de lampen is mogelijk via XLAMP.A

Het dimmen van de signaalgevers

Signaalgevers worden gedimd om de lichtopbrengst aan te passen aan de omgevingsvariabelen. Dimmen kan op meerdere wijzen plaatsvinden. Van een lichtcel en een externe bron tot en met de binnen IVERA beschikbare astronomische schakeltabel.

Het aansturen van akoestische signalen

Er zijn diverse signaalgevers in de markt met diverse mogelijkheden. Daarnaast is er ook de mogelijkheid de aansturing vanuit de regelapplicatie te verzorgen. Verzorging vanuit de regelapplicatie maakt het mogelijk om geluid te regelen op basis van aanvragen en signaalgroepafwikkeling. Echter de parameters binnen de regelapplicaties zijn niet universeel waardoor de instellingen per automaat bekeken moeten worden door iemand met kennis van de regelapplicatie.

IVERA biedt met deze objecten de mogelijkheid de instellingen universeel te maken echter IVERA kan niet de functionaliteit bieden zoals binnen de regelapplicaties mogelijk is.

Er zijn twee wegen om tot deze functionaliteit te komen.

1) Men kan de uitgangssignalen van de regelapplicatie afhankelijk van de instellingen en tijd wel of niet doorsturen naar de rateltickers.

2) Men kan de akoestische wens aanbieden op een overige ingang van de regelapplicatie. De regelapplicatie verzorgt dan de aansturing van de signaalgevers. Op deze wijze kunnen overgangen netter gerealiseerd worden door een aangevangen cyclus geheel of te maken. De interface met de regelapplicatie valt echter buiten de scope van IVERA.

Met behulp van een aantal objecten kan het geluidsniveau van de akoestische signaalgevers worden beïnvloed. Dit kunnen rateltickers / akoestische signalering zijn ten behoeve van visueel gehandicapten, waarschuwingbellen voor trams e.d. .

Hiervoor zijn 9 objecten beschikbaar:

- AKOESTISCH.I
- AKOESTISCH.A
- AKOESTISCH.F
- KLA_AKOEST
- KLU_AKOEST
- KLA_HARD
- KLU_HARD
- PAKOESTISCH
- PAKOESTISCH.I

20.2 Objecten signaalgroepen

20.2.1 SG.I

Het object SG.I geeft de index weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	SG.I	Naam
O	1	Signaalgroepnamen	Omschrijving
T	0	1	Type
U	0	4444	User Identificatie Control
E	0	NUMSG	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 20.3 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Het standaardiseren van namen van verschillende parameters is niet zozeer een IVERA protocol aanpassing, als wel het maken van een extra restrictie op het gebruik van namen in een aantal indexobjecten. Het gaat hierbij over de indexobjecten, die gekoppeld zijn aan de parameters in CCOL of RWS-C, die reeds een naam hebben zoals de volgende indexobjecten:

De namen in deze objecten moeten exact overeenkomen met de namen die in CCOL of RWS-C zijn geconfigureerd. Zij bepaalde namen leeg gelaten, dan wordt in IVERA een invulling aangegeven, die overeenkomt met het elementnummer en het type parameter.

De n^e lege indexnaam wordt in IVERA ingevuld met 'FC_(n-1)', waarbij (n-1) een getal weergeeft.

De volgende tabel bevat enkele voorbeelden met elementen 02, 05 en een leeg element:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
SG.I	SG.I="02","05","FC_2"	Weergave index alle signaalgroepen
SG.I/#0	SG.I/#0="02"	Opvragen index signaalgroep met index 0

Tabel 20.4 Voorbeelden index signaalgroepen

20.2.2 SGE.A

Het object SGE.A geeft de externe signaalgroepstatus weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	SGE.A	Naam
O	1	Signaalgroepstatus (extern)	Omschrijving
T	0	0	Type
U	0	4444	User Identificatie Control
E	0	NUMSG	aantal data-elementen
I	1	SG.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	7	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	30	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 20.5 Object attributen

Formaat: Signaalgroep status extern (30), type 0

Waarde	Karakter	Signaalgroep status
0	R	Rood
1	G	Groen
2	A	Geel
3	W	Wit knipperen (OV)
4	O	Gedoofd
5	F	Geel knipperen
6	6	gereserveerd
7	7	gereserveerd

Tabel 20.6 Signaalgroepstatus extern

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
SGE.A	SGE.A=0, 0, 1, ...	Vraag de actuele externe signaalgroepstatus op. De signaalgroepstatus komen overeen met het uitgestuurde beeld.
SGE.A/#0	SGE.A/#0=<waarde>	Vraag een element op uit object SGE.A. De signaalgroepstatus komt overeen met het uitgestuurde beeld.
SGE.A/#0=0	:E=11	Schrijven is niet toegestaan.

Tabel 20.7 Voorbeelden externe signaalgroepstatus

20.2.3 TSGE.A

Het object TSGE.A geeft de timer van de externe signaalgroepstatus weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TSGE.A	Naam
O	1	Timer signaalgroepstatus (extern)	Omschrijving
T	0	0	Type
U	0	4444	User Identificatie Control
E	0	NUMSG	aantal data-elementen
I	1	SG.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2, getal met eenheid 0.1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 20.8 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TSGE.A/#0	TSGE.A/#0 = <waarde>	Opvragen van een element in TSGE.A object
TSGE.A:O	TSGE.A:O = "Timer signaalgroepstatus (ext)"	Opvragen van een attribuut in TSGE.A object

TSGE.A:O="Hallo" :E=11	Wijzigen attribuut niet toegestaan.
------------------------	-------------------------------------

Tabel 20.9 Voorbeelden timer externe signaalgroepstoestand

20.2.4 SGE.LB

Het object SGE.LB geeft het externe signaalgroeplogboek weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	SGE.LB	Naam
O	1	Signaalgroeplogboek (extern)	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	0 .. 1000	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	30	Data-element formaat
S	0		Data-element stapgrootte

Tabel 20.10 Object attributen

Formaat: Toestand van alle externe signaalgroepstoestanden (30), type 1

Het object geeft een string met daarin een timestamp en voor iedere signaalgroep een karakter die de actuele toestand weergeeft. De timestamp geeft aan het moment van de laatste wijziging.

```
SGtoestand = HMS_timestamp ":" { SGKarakter }
HMS_timestamp = "hhmmss"
SGKarakter = /* zie tabel signaalgroepstoestand (extern) */
```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
SGE.LB	De VRI geeft de waarde <sge.lb waarde> terug. Deze is het formaat hhmmss:SGKarakter	Vraag de waarde van logboekobject SGE.LB op.

Tabel 20.11 Voorbeelden object SGE.LB

20.2.5 SGE.LA

Het object SGE.LA geeft het externe (onbevestigde) signaalgroeplogboek weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	SGE.LA	Naam
O	1	Signaalgroeplogboek (onbevestigd/extern)	Omschrijving
T	1	1	Type
U	0	6666	User Identificatie Control
E	0	0 .. 1000	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	30	Data-element formaat
S	0		Data-element stapgrootte

Tabel 20.12 Object attributen

Formaat: Toestand van alle externe signaalgroepstoestanden (30), type 1

Het object geeft een string met daarin een timestamp en voor iedere signaalgroep een karakter die de actuele toestand weergeeft. De timestamp geeft aan het moment van de laatste wijziging.

```
SGtoestand = HMS_timestamp ":" { SGKarakter }
HMS_timestamp = "hhmmss"
SGKarakter = /* zie tabel signaalgroepstoestand (extern) */
```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
SGE.LA	De VRI geeft de waarde <sge.la waarde> terug. Deze is het formaat hhmmss:SGKarakter.	Vraag de waarde van logboekobject SGE.LA op.
SGE.LA/#0=""		Maak het eerste object leeg.
SGE.LA	De VRI geeft de huidige inhoud van het object SGE.LA, waarvan het bevestigde element is verwijderd. Bijv: SGE.LA="085012:FFFFFFO OFFFF","085028:AAAAARRR AAAA","085032:RRRRRRRR RRRR","085046:GGRRRR RRRRG","093226:OOOOOO OOOOOO","093836:FFFFFFO OFFFF","093852:AAAAARR RAAAA","093856:RRRRRRR RRRRR","094130:FFFFFFO OFFFF".	Meermaals het object SGE.LA/#0="" opgeven resulteert in :E17, buffer is leeg.

Tabel 20.13 Voorbeelden object SGE.LA

20.2.6 TGOR1

Het object TGOR1 geeft de garantie ontruimingstijd (procesbesturing) weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TGOR1	Naam
O	1	Garantie ontruimingstijd (procesbesturing)	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMSG , NUMSG	aantal data-elementen
I	1	SG.I , SG.I	Index verwijzing per dimensie
MIN	0	-4	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2, getal met eenheid 0.1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 20.14 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TGOR1:E	TGOR1:E=<aantal sg>, <aantal sg>	Aantal elementen opvragen van object TGOR1
TGOR1/#0	TGOR1/#0=<waarde>,<waar de>,...	Een element opvragen van object TGOR1: waarde in automatisch: alle TGOR1 met 1e SG
TGOR1:I	TGOR1:I=SG.I,SG.I,	Indexverwijzing opvragen van object TGOR1

Tabel 20.15 Voorbeelden object TGOR1

20.2.7 TGG1

Het object TGG1 geeft de garantie groentijd (procesbesturing) weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TGG1	Naam
O	1	Garantie groentijd (procesbesturing)	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMSG	aantal data-elementen
I	1	SG.I	Index verwijzing per dimensie
MIN	0	-1	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2, getal met eenheid 0.1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 20.16 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TGG1:E	TGG1:E=<aantal sg>	Aantal elementen opvragen
TGG1/#0	TGG1/#0=<waarde>	Een element opvragen
TGG1:I	TGG1:I=SG.I,	Indexverwijzing opvragen

Tabel 20.17 Voorbeelden object TGG1

20.2.8 TGGL1

Het object TGGL1 geeft de garantie geeltijd (procesbesturing) weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TGGL1	Naam
O	1	Garantie geeltijd (procesbesturing)	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMSG	aantal data-elementen
I	1	SG.I	Index verwijzing per dimensie
MIN	0	-1	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2, getal met eenheid 0.1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 20.18 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TGGL1:E	TGGL1:E=<aantal sg>	Aantal elementen opvragen
TGGL1/#0	TGGL1/#0=<waarde>	Een element opvragen
TGGL1:I	TGGL1:I=SG.I	Indexverwijzing opvragen

Tabel 20.19 Voorbeelden object TGGL1

20.2.9 TMGL1

Het object TMGL1 geeft de maximum geeltijd (procesbesturing) weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TMGL1	Naam
O	1	Maximum geeltijd (procesbesturing)	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMSG	aantal data-elementen
I	1	SG.I	Index verwijzing per dimensie
MIN	0	-1	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde

ITYPE	1		Index data-element type
F	0	2, getal met eenheid 0.1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 20.20 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TMGL1:E	De VRI geeft <aantalElementen> als waarde terug. Dit betreft het aantal data-elementen in object TMGL1.	Vraag aantal data-elementen in object TMGL1 op.
TMGL1/#0	De VRI geeft als waarde: <waarde> in een eenheid van 0.1 seconden.	Vraag de waarde van het object op.

Tabel 20.21 Voorbeelden object TMGL1

20.2.10 TGR1

Het object TGR1 geeft de garantie roodtijd (procesbesturing) weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TGR1	Naam
O	1	Garantie roodtijd (procesbesturing)	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMSG	aantal data-elementen
I	1	SG.I	Index verwijzing per dimensie
MIN	0	-1	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2, getal met eenheid 0.1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 20.22 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TGR1:E	TGR1:E=<aantalElementen>	Aantal elementen opvragen
TGR1/#0	TGR1/#0=<waarde>	Een element opvragen
TGR1:I	TGR1:I=SG.I,	Indexverwijzing opvragen

Tabel 20.23 Voorbeelden object TGR1

20.2.11 LAMP.I

Het object LAMP.I geeft de index van alle lampnamen:

Attribuut	Type	Object	Omschrijving
N	1	LAMP.I	Naam
O	1	Index lampnamen	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMLAMP	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 20.24 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
LAMP.I	<p>De VRI retourneert een lijst van alle bewaakte lampen aanwezig in het systeem.</p> <p>Bijv: LAMP.I="02R1","08R1","24R1","25R1","26R1","34R1","35R1","36R1","43R1","44R1","104R1","108R1","02R2","08R2","34R2","35R2","36R2","108R2","02A1","08A1","24A1","25A1","26A1","43A1","44A1","104A1","108A1","02A2","08A2","108A2","02G1","08G1","24G1","25G1","26G1","34G1","35G1","36G1","43G1","44G1","104G1","108G1","02G2","08G2","34G2","35G2","36G2","108G2"</p> <p>;Het resultaat mag in geen geval lege velden tonen</p>	Vraag de aanwezige lampen op uit het object LAMP.I.

Tabel 20.25 Voorbeelden object LAMP.I

20.2.13 LAMPINFO

Het object LAMPINFO geeft de lampconfiguratie weer van alle lampen:

Attribuut	Type	Object	Omschrijving
N	1	LAMPINFO	Naam
O	1	lampconfiguratie	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMLAMP	aantal data-elementen
I	1	LAMP.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	110	Data-element formaat
S	0		Data-element stapgrootte

Tabel 20.28 Object attributen

Formaat: Lampconfiguratie (110), type 1

Voor de informatie over aspecten/lampen wordt de volgende syntax gehanteerd. De lampconfiguratieparameters **SgRef**, **Lref**, **Kleur** en **Cat**, komen overeen met de informatie die wordt meegegeven met het lampfoutevent (1010).

LampConfiguratie = SgRef+ “,” + Lref + “,” + Kleur + “,” + Cat + “,” + [Aantal] + “,” [Actie] + “,” + [LampType]

SgRef =AsciiString De functionele naam van de bijbehorende signaalgroep.

Lref = 0..9999 Het volgnummer van het armatuur.

Kleur = De kleur of functie van het aspect.

R	Rood aspect
DR	Dubbel rood aspect
A	Geel aspect
G	Groen aspect
W	Wachtlamp
O	Overige

Cat = 0..3 Categorie Zie tabel 6.19

Aantal = 1..9 Het aantal parallel geschakelde aspecten.

Actie = Actie in het geval van een lampfout.

0	Geen actie.
1	Zacht uitschakelen (via alles rood)
2	Hard uitschakelen (direct naar knipperen)

LampType =

G	Gloeilamp
H	Halogeenlamp
K	Kryptonlamp
L1	Led1-module
L2	Led2-module
O	Overige

In het geval dat er signaalgroepgeoriënteerd wordt gemeten en er meerdere armaturen parallel worden aangesloten op een meetcircuit kan dit worden weergegeven met de argumenten Aantal en Lref.

Een aantal voorbeelden:

SgRef	Lref	Aantal	Kleur	Opmerkingen
FC02	1	1	R	Het rode aspect van armatuur 2.1
FC02	2	1	DR	Het rode aspect van armatuur 2.2. Het aspect is uitgevoerd met dubbelrood.
FC02	124	3	G	De groene aspecten van armatuur 2.1, 2.2. en 2.4 zijn parallel aangesloten op 1 meetcircuit.

In het geval er sprake is van combinatorische logica waarbij het regeltoestel uitschakelt, als er combinatie van rode lampen defect is, kan dit worden weergegeven door meerdere lampen te definiëren.

Een voorbeeld waarbij de VRI uitschakelt, als beide rode lampen van signaalgroep defect zijn.

SgRef	Lref	Aantal	Kleur	Actie	Opmerkingen
FC02	1	1	R	0	Het rode aspect van armatuur 2.1
FC02	2	1	R	0	Het rode aspect van armatuur 2.2.
FC02	12	2	R	1	Het resultaat van de combinatorische functie.

De volgende tabel bevat een aanvullend voorbeeld:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
LAMPINFO	De VRI levert een lijst van lamp informatie volgens formaat type 110 (lampconfiguratie). Bijv: LAMPINFO="FC02,1,R,0,1,0","FC08,1,R,0,1,0","FC24,1,R,0,1,1","FC25,1,R,0,1,1","FC26,1,R,0,1,1","FC34,1,R,0,1,0","FC35,1,R,0,1,0","FC36,1,R,0,1,0","FC43,1,R,0,1,1","FC44,1,R,0,1,1","FC104,1,R,0,1,1","FC108,1,R,0,1,0","FC02,2,R,0,1,0","FC08,2,R,0,1,0","FC34,2,R,0,1,0","FC35,2,R,0,1,0","FC36,2,R,0,1,0","FC108,2,R,0,1,0","FC02,1,A,0,1,0","FC08,1,A,0,1,0","FC24,1,A,0,1,0","FC25,1,A,0,1,0","FC26,1,A,0,1,0","FC43,1,A,0,1,0","FC44,1,A,0,1,0","FC104,1,A,0,1,0","FC108,1,A,0,1,0","FC02,2,A,0,1,0","FC08,2,A,0,1,0","FC108,2,A,0,1,0","FC02,1,G,0,1,0","FC08,1,G,0,1,0","FC24,1,G,0,1,0","FC25,1,G,0,1,0","FC26,1,G,0,1,0","FC34,1,G,0,1,0","FC35,1,G,0,1,0","FC36,1,G,0,1,0","FC43,1,G,0,1,0","FC44,1,G,0,1,0","FC104,1,G,0,1,0","FC108,1,G,0,1,0","FC02,2,G,0,1,0","FC08,2,G,0,1,0","FC34,2,G,0,1,0","FC35,2,G,0,1,0","FC36,2,G,0,1,0","FC108,2,G,0,1,0"	Vraag de extra lamp informatie op uit het object LAMPINFO.

Tabel 20.29 Voorbeeld object LAMPINFO

20.2.14 DIMINST.I

Het object DIMINST.I geeft de index van de diminstellingen weer:

Attribuut	Type	Object	Omschrijving
N	1	DIMINST.I	Naam
O	1	Index diminstellingen	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 20.30 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Het dimmen van de lantaarns kan gestuurd worden door verschillende bronnen. Voor de centrale zijn de objecten DIMINST en DIMINST.I om het dimgedrag in te stellen:

DIMINST.I		Omschrijving
0	DIREKT	Algemene instelling om andere instellingen te passeren. 0=geen invloed, 1 altijd dimmen, 2 nooit dimmen.
1	LOKAAL1	Lokaal ingangssignaal1
2	LOKAAL2	Lokaal ingangssignaal2
3	APPLICATIE	Wens applicatie
4	NOODSTROOM	Wens noodstroomvoorziening.
5	DIMZON	Schakelaar of op basis van berekende zon tijden gedimd moet worden, volgens de Tabel 20.32 Zonsondergang en Zonsopgangtijden. (0= Niet dimmen op basis van element 6 en 7 / 1= Ook dimmen op basis van element 6 en 7)
6	ZONSONDERGANG	Starttijd dimmen t.o.v. zonsondergang (Tijd in minuten, negatief getal is voor zonsondergang, positief getal is na zonsondergang)
7	ZONSOPGANG	Eindtijd dimmen t.o.v. zonsopgang (Tijd in minuten, negatief getal is voor zonsopgang, positief getal is na zonsopgang)
8	FAB2	Vrij te gebruiken (indexnaam en waarde)
9	FAB1	Vrij te gebruiken (indexnaam en waarde)

Tabel 20.31 Object DIMINST

Indien een waarde 0 is wordt op het betreffende item niet gedimd. Dit geldt niet voor Zonsondergang en Zonsopgang, hiervoor is element 5 opgenomen.

Voor het dimmen op basis van zonsondergang en zonsopgang is Tabel 20.32 opgenomen met per week de toe te passen tijden. Deze tijden worden gecombineerd met de in te stellen tijden.

Week	Zonsondergang	Zonsopgang	Week	Zonsondergang	Zonsopgang
1	16.41	8.48	27	21.02	4.26
2	16.50	8.45	28	20.58	4.33
3	17.00	8.40	29	20.51	4.41
4	17.12	8.32	30	20.42	4.50
5	17.25	8.23	31	20.31	5.01
6	17.38	8.11	32	20.19	5.12
7	17.51	7.58	33	20.05	5.23
8	18.04	7.44	34	19.51	5.34
9	18.17	7.30	35	19.35	5.46
10	18.29	7.14	36	19.20	5.57
11	18.42	6.58	37	19.03	6.08
12	18.54	6.42	38	18.47	6.20
13	19.06	6.26	39	18.31	6.31
14	19.18	6.10	40	18.14	6.43
15	19.30	5.54	41	17.59	6.55
16	19.42	5.38	42	17.43	7.07
17	19.54	5.24	43	17.28	7.19
18	20.05	5.10	44	17.15	7.32
19	20.17	4.57	45	17.02	7.44
20	20.28	4.46	46	16.51	7.57
21	20.38	4.36	47	16.42	8.09
22	20.47	4.28	48	16.35	8.20
23	20.55	4.23	49	16.30	8.30
24	21.00	4.20	50	16.28	8.38
25	21.03	4.19	51	16.29	8.44
26	21.04	4.22	52	16.33	8.48
			53	16.41	8.48

Tabel 20.32 Zonsondergang en Zonsopgangtijden

NB: Deze tijden zijn gebaseerd op de donderdagen in de betreffende week in 2008. Midden Nederland GMT+1 excl. aanpassing t.b.v. zomertijd (52°00 NB 5°00 OL).

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
DIMINST.I	DIMINST.I="DIREKT", "LOKAL1", "LOKAAL2", "APPLICATIE", "NOODSTROOM", "DIMZON", "ZONSONDERGANG", "ZONSOPGANG", "FAB2", "FAB1	Vraag het object DIMINST.I op.

Tabel 20.33 Voorbeelden object DIMINST.I

20.2.15 DIMINST

Het object DIMINST geeft de diminstellingen weer:

Attribuut	Type	Object	Omschrijving
N	1	DIMINST	Naam
O	1	Diminstellingen	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	10	aantal data-elementen
I	1	DIMINST.I	Index verwijzing per dimensie
MIN	0	-90	Minimum data-elementwaarde
MAX	0	90	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 20.34 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
DIMINST/DIRECT =1	Er wordt gedimd	Zet dimmen direct aan
DIMMEN.A/AKTU EEL	DIMMEN.A/AKTUEEL=1	Vraag DimStatus op
DIMINST/DIRECT =2	Er wordt niet gedimd	Zet dimmen direct uit

Tabel 20.35 Voorbeelden object DIMINST

20.2.16 DIMMEN.I

Het object DIMMEN.I geeft de index van de dimstatus weer:

Attribuut	Type	Object	Omschrijving
N	1	DIMMEN.I	Naam
O	1	Index dimstatus	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 20.36 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

De dimstatus van de lantaarns kan worden uitgelezen van de verschillende bronnen. Voor de centrale zijn de objecten DIMMEN, DIMMEN.I en DIMMEN.A om de status van dimgedrag te monitoren:

DIMMEN.I		Omschrijving
0	AKTUEEL	Actuele dimstatus
1	LOKAAL1	Lokaal ingangssignaal1
2	LOKAAL2	Lokaal ingangssignaal2
3	APPLICATIE	Wens applicatie
4	NOODSTROOM	Wens noodstroomvoorziening.
5	ASTROKLOK	Wens astroklok
6	TIJDAAN	Actuele tijd start dimmen indien gedimd wordt op basis van berekende zontijden.
7	TIJDUIT	Actuele tijd einde dimmen indien gedimd wordt op basis van berekende zontijden.
8	FAB2	Vrij te gebruiken (indexnaam en waarde)
9	FAB1	Vrij te gebruiken (indexnaam en waarde)

Tabel 20.37 Object DIMMEN

In het object DIMMEN.A wordt per signaal de status aangegeven. In AKTUEEL wordt het gezamenlijk resultaat weergegeven. Er wordt gedimd als element 0 (AKTUEEL) 1 is.

AKTUEEL is 1 als DIMINST/#0 gelijk aan 1 is.

AKTUEEL is 0 als DIMINST/#0 gelijk aan 2 is.

AKTUEEL is 1 als DIMINST/#0 gelijk aan 0 is en 1 ander element van DIMMEN.A is ook 1 is en het bijbehorende element in DIMINST ook 1 is.

NB: Binnen IVERA begint de week op maandag.

Voorbeeld 1

Dimmen op basis van zonstand: Als het 16.20 uur is (15 minuten voor zonsondergang). Het is week 53. Object DIMMEN.A ziet er dan als volgt uit:

- Toestand DIMMEN.A object:

DIMMEN.A /#0	1 (Aktueel)
DIMMEN.A /#1	0 (Lokaal 1)
DIMMEN.A /#2	0 (Lokaal 2)
DIMMEN.A /#3	0 (Applicatie)
DIMMEN.A /#4	0 (Noodstroom)
DIMMEN.A /#5	1 (Astroklok)
DIMMEN.A /#6	1605 (tijdaan)
DIMMEN.A /#7	918 (tijduit)
DIMMEN.A /#8	0 (Fab 2)
DIMMEN.A /#9	0 (Fab 1)

Er wordt gedimd op basis van de berekende zontijd, want element 5 is gelijk aan 1. Ook zijn de tijden waarop het dimmen wordt ingeschakeld weergegeven in de elementen 6 en 7. Deze tijden worden altijd weergegeven ook als dimmen op basis van zontijden is uitgeschakeld.

Voorbeeld 2

Dimmen op basis van Noodstroom: Als het 31 minuten voor zonsondergang is. Object DIMMEN.A ziet er dan als volgt uit:

- Toestand DIMMEN.A object:

DIMMEN.A /#0	1 (Aktueel)
DIMMEN.A /#1	0 (Lokaal 1)
DIMMEN.A /#2	0 (Lokaal 2)
DIMMEN.A /#3	0 (Applicatie)
DIMMEN.A /#4	0 (Noodstroom)
DIMMEN.A /#5	0 (Astroklok)
DIMMEN.A /#6	1605 (tijdaan)
DIMMEN.A /#7	918 (tijduit)
DIMMEN.A /#8	0 (Fab 2)
DIMMEN.A /#9	0 (Fab 1)

Voorbeeld 3

Maskeren van een dimingang: Als 31 minuten voor zonsondergang is en lokaal ingangssignaal 1 is actief. Object DIMMEN.A ziet er dan als volgt uit:

- Toestand DIMMEN.A object:

DIMMEN.A /#0	0 (Actueel)
DIMMEN.A /#1	1 (Lokaal 1)
DIMMEN.A /#2	0 (Lokaal 2)
DIMMEN.A /#3	0 (Applicatie)
DIMMEN.A /#4	0 (Noodstroom)
DIMMEN.A /#5	0 (Astroklok)
DIMMEN.A /#6	1605 (tijdaan)
DIMMEN.A /#7	918 (tijduit)
DIMMEN.A /#8	0 (Fab 2)
DIMMEN.A /#9	0 (Fab 1)

Toelichting: Het lokale signaal is actief maar in object DIMINST is het overeenkomstige element 0. Er is dan wel te zien dat dit signaal actief is maar dit heeft geen invloed op de actuele DIMMEN.A.

20.2.17 DIMMEN.A

Het object DIMMEN.A geeft de dimstatus weer:

Attribuut	Type	Object	Omschrijving
N	1	DIMMEN.A	Naam
O	1	Dimstatus	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1	DIMINST.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	2359	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 20.38 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
DIMMEN.A/TIJDUI T	DIMMEN.A/TIJDUIT=848	Vraag tijd zonsopgang op
DIMMEN.A/TIJD AN	DIMMEN.A/TIJDAN=1641	Vraag tijd zonsondergang op
DATUM/#0=20130 325		Pas de datum aan
DIMMEN.A/TIJDUI T	DIMMEN.A/TIJDUIT=626	Vraag tijd zonsopgang op
DIMMEN.A/TIJD AN	DIMMEN.A/TIJDAN=1906	Vraag tijd zonsondergang op

Tabel 20.39 Voorbeelden object DIMINST.I

20.2.18 AKOESTISCH.I

Het object AKOESTISCH.I geeft de index van de status akoestische signalen weer:

Attribuut	Type	Object	Omschrijving
N	1	AKOESTISCH .I	Naam
O	1	Index status akoestische signalen	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	5	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 20.40 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Werking

De objecten AKOESTISCH.I en AKOESTISCH.A zijn voor het weergeven van de huidige status. In object AKOESTISCH.F zijn de fout statussen van de akoestische signaalgevers per signaalgroep weergegeven:

AKOESTISCH.A	AKOESTISCH.I	Omschrijving
0	Signaal	Status signaal (0 = uit, 1= aan)
1	Volume	Status volume (0 = zacht, 1= hard)
2	HardwareFout	Status hardware (0 = oke, 1= fout)
3	Fab2	Reserve
4	Fab1	Reserve

Tabel 20.41 Objecten akoestische signalen

In AKOESTISCH.A is de huidige status te lezen. Hierin kan direct gezien worden of de akoestische signalen aan staan en of deze hard staan.

Indien de signaalgevers een fout aangeven wordt dit zichtbaar in element 2 (HardwareFout). De waarde van hardwarefout kan 0 of 1 zijn. Tevens treed hierdoor event 1030 op.

In AKOESTISCH.F komt de foutstatus van de signaalgevers per signaalgroep te staan. Indien één element in AKOESTISCH.F een fout aangeeft wordt element 2 (HardwareFout) van AKOESTISCH.A de waarde 1.

NB: Wanneer de VRI niet de mogelijkheid heeft aan te geven welke signaalgever defect is kan object AKOESTISCH.F weggelaten worden. De algemene foutmelding in object AKOESTISCH.A kan dan als verzamelmelding gebruikt worden.

Bij een storing of herstel zal, indien het event 1030 is opgegeven, het event 1030 worden verstuurd naar de beheerscentrale. Zie hiervoor paragraaf 20.2.28.

Indien van toepassing wordt op de overige ingangen van de regelapplicatie het volgende aangeboden:

Waarde	Naam ingang	Betekenis
0	Akoestisch	Applicatie bepaald zelf of signaalgevers aan of uit zijn
1	Akoestisch	Applicatie zet signaalgevers aan
2	Akoestisch	Applicatie zet signaalgevers uit. (aangevangen cyclus mag worden afgemaakt zodat bijvoorbeeld een ratel altijd gevolgd wordt door een onderbroken ratel en een naloop tik.)

In AKOESTISCH.F wordt per signaalgroep de foutstatus van de geluidssignaalgever gegeven. Uiteraard zal bijna nooit elke signaalgroep van geluidssignaalgever zijn voorzien. Voor signaalgroepen zonder geluidssignaalgever staat in dit object een 0.

AKOESTISCH.F	SG.I	Omschrijving
<0=geen fout, andere waarde is foutcode>	<Naam van signaalgroep>	Foutindicatie, van geluidssignaal per signaalgroep.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
AKOESTISCH.I	AKOESTISCH.I="Signaal","Volume","HardwareFout","Fab2","Fab1"	Opvragen object AKOESTISCH.I

Tabel 20.42 Voorbeelden object AKOESTISCH.I

20.2.19 AKOESTISCH.A

Het object AKOESTISCH.A geeft de status akoestische signalen weer:

Attribuut	Type	Object	Omschrijving
N	1	AKOESTISCH.A	Naam
O	1	Status akoestische signalen	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	5	aantal data-elementen
I	1	AKOESTISCH.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0		Data-element stapgrootte

Tabel 20.43 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
AKOESTISCH.A/#2	AKOESTISCH.A/#2=1	Er is sprake van een hardwarefout.
AKOESTISCH.F	AKOESTISCH.F=?,?,?,...	Per signaalgroep een getal. Het getal is 0 voor geen fout, een andere waarde voor een fout. Indien er geen signaalgever aanwezig is moet de waarde ook 0 zijn.
AKOESTISCH.A/#2	AKOESTISCH.A/#2=0	De fout is hersteld.
AKOESTISCH.F	AKOESTISCH.F=0,0,0,...	

Tabel 20.44 Voorbeelden object AKOESTISCH.A

20.2.20 AKOESTISCH.F

Het object AKOESTISCH.F geeft de foutstatus akoestische signalen weer:

Attribuut	Type	Object	Omschrijving
N	1	AKOESTISCH.F	Naam
O	1	Foutstatus akoestische signalen	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMSG	aantal data-elementen
I	1	SG.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0		Data-element stapgrootte

Tabel 20.45 Object attributen

Zie de paragraaf 20.2.19 voor een voorbeeld.

20.2.21 KLA_AKOEST

Het object KLA_AKOEST geeft het inschakeltijdstip van de akoestische signalen:

Attribuut	Type	Object	Omschrijving
N	1	KLA_AKOEST	Naam
O	1	Inschakelen akoestische signalen	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMDAG	aantal data-elementen
I	1	DAG.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	2400	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	20	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 20.46 Object attributen

Formaat: Klok (uur/ minuut) (20), type 0

De kloktijd is een getal met de tijd in het volgende formaat:

Waargave: UUMM

Waarde: uur * 100 + minuut

Werking

Voor elke dag zijn twee perioden gedefinieerd met behulp van KLA_AKOEST, KLU_AKOEST, KLA_HARD en KLU_HARD. Binnen deze perioden staat het geluid aan respectievelijk het volume hard. Als het geluid uit staat en het volume hard zal het resultaat zijn dat het geluid uit staat.

In de objecten KLA_AKOEST en KLU_AKOEST staan de tijdstellingen per dag waarbinnen de akoestische signalering actief is.

In de objecten KLA_HARD en KLU_HARD staan de tijdstellingen per dag waarbinnen de akoestische signalering met een hoog geluidsniveau werkt. Buiten deze perioden is het geluidsniveau gedempt.

De werking van de objecten KLA_AKOEST, KLU_AKOEST, KLA_HARD en KLU_HARD komt overeen met KLA1 en KLU1 zoals omschreven in paragraaf 17.2.1.

Voorbeeld

De waarden zijn als volgt ingesteld

ELM	KLA_AKOEST	KLU_AKOEST	KLA_HARD	KLU_HARD
0 (ma)	700	1800	900	1800
1 (di)	700	1800	900	2000
2 (wo)	700	1800	900	1800
3 (do)	700	1800	900	1800
4 (vr)	700	2000	900	2000
5 (za)	800	2000	1000	2000
6 (zo)	800	1900	1000	1800
7 (elke)	0	0	0	0
8 (werk)	0	0	0	0
9 (w'kend)	0	0	0	0
10 (feest)	0	0	0	0
11 (koop)	0	0	0	0
12 (koopz)	0	0	0	0
13 (spec)	0	0	0	0

20.2.22 KLU_AKOEST

Het object KLU_AKOEST geeft het uitschakeltijdstip van de akoestische signalen:

Attribuut	Type	Object	Omschrijving
N	1	KLU_AKOEST	Naam
O	1	Uitschakelen akoestische signalen	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMDAG	aantal data-elementen
I	1	DAG.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	2400	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	20	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 20.47 Object attributen

Formaat: Klok (uur/ minuut) (20), type 0

De kloktijd is een getal met de tijd in het volgende formaat:

Waargave: UUMM

Waarde: uur * 100 + minuut

Voor de voorbeelden wordt verwezen naar paragraaf 20.2.21.

20.2.23 KLA_HARD

Het object KLA_HARD geeft het inschakeltijdstip voor het hoog volume niveau van de akoestische signalen:

Attribuut	Type	Object	Omschrijving
N	1	KLA_HARD	Naam
O	1	Inschakelen hoog geluidsvolume	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMDAG	aantal data-elementen
I	1	DAG.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	2400	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	20	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 20.48 Object attributen

Formaat: Klok (uur/ minuut) (20), type 0

De kloktijd is een getal met de tijd in het volgende formaat:

Waargave: UUMM

Waarde: uur * 100 + minuut

Voor de voorbeelden wordt verwezen naar paragraaf 20.2.21.

20.2.24 KLU_HARD

Het object KLU_HARD geeft het uitschakeltijdstip voor het hoog volume niveau van de akoestische signalen:

Attribuut	Type	Object	Omschrijving
N	1	KLU_HARD	Naam
O	1	Uitschakelen hoog geluidsvolume	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMDAG	aantal data-elementen
I	1	DAG.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	2400	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	20	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 20.49 Object attributen

Formaat: Klok (uur/minuut) (20), type 0

De kloktijd is een getal met de tijd in het volgende formaat:

Waargave: UUMM

Waarde: uur * 100 + minuut

Voor de voorbeelden wordt verwezen naar paragraaf 20.2.21.

20.2.25 PAKOESTISCH.I

Het object PAKOESTISCH.I geeft de index van de parameter akoestische signalen weer:

Attribuut	Type	Object	Omschrijving
N	1	PAKOESTISCH	Naam
O	1	Index parameter akoestische signalen	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	2	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 20.50 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Werking

In het object PAKOESTISCH wordt de werking bepaald. Dit om diverse scenario's te kunnen ondersteunen. Deze beschrijving gaat uit van het aan en uit zetten van het geluid. Dit zal in de praktijk ook het meest variëren qua gebruik. Echter voor hard/zacht is hetzelfde mogelijk.

PAKOESTISCH kan als read only uitgevoerd zijn, in dat geval zijn de waarden informatief.

PAKOESTISCH	PAKOESTISCH.I	Omschrijving
0	Signaal	Werking signaal (zie Tabel 20.53)
1	Volume	Werking volume (zie Tabel 20.53)

Tabel 20.51 Object attributen

Voor de schakeltijden uit dit voorbeeld wordt verwezen naar paragraaf 20.2.21.

Volgens deze instellingen zal op zondag het geluid om 8.00 uur aan gaan, om 10.00 uur zal het geluid op volume hard gaan. Om 18.00 uur gaat het volume weer gedempt en om 19.00 uur gaat het geluid weer uit.

Op dinsdag zal het geluid om 18.00 uur uit gaan. Dit ondanks het feit dat de klok voor hard nog later ingesteld staat.

20.2.26 PAKOESTISCH

Het object PAKOESTISCH geeft de parameter akoestische signalen weer:

Attribuut	Type	Object	Omschrijving
N	1	PAKOESTISCH	Naam
O	1	Parameter akoestische signalen	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	2	aantal data-elementen
I	1	PAKOESTISCH.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	3	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0		Data-element stapgrootte

Tabel 20.52 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
PAKOESTISCH/*= 0	PAKOESTISCH/*=0	Signaal/volume staat volgt AKOESTISCH.A. (Dus indirect naar de betreffende klokken.)
PAKOESTISCH/*= 1	PAKOESTISCH/*=1	Signaal/volume staat aan/hard indien AKOESTISCH.A dit aangeeft en de regelapplicatie dit aangeeft. In de praktijk zal dat betekenen dat in een bepaalde periode het geluid aangaat bij een aanvraag.
PAKOESTISCH/*= 3	PAKOESTISCH/*=3	Signaal volume staat aan/hard indien AKOESTISCH.A dit aangeeft of de regelapplicatie dit aangeeft. In de praktijk zal dat betekenen dat in een bepaalde periode het geluid continue aangaat en buiten die periode alleen bij een aanvraag.
PAKOESTISCH/*= 4	PAKOESTISCH/*=4	Signaal volume staat aan/hard indien de regelapplicatie dit aangeeft. De aansturing is dan geheel afhankelijk van de regelapplicatie.

Tabel 20.53 Voorbeelden object PAKOESTISCH

20.2.27 LSGE

Het object LSGE geeft de lijndump SG-toestanden (extern) voor signaalgroepen weer:

Attribuut	Type	Object	Omschrijving
N	1	LSGE	Naam
O	1	Lijndump SG-toestand (extern)	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	30	Data-element formaat
S	0		Data-element stapgrootte

Tabel 20.54 Object attributen

Formaat: Toestand van alle externe signaalgroepstanden (30), type 1

Het object geeft een string met daarin een timestamp en voor iedere signaalgroep een karakter die de actuele toestand weergeeft. De timestamp geeft aan het moment van de laatste wijziging.

```
SGtoestand = HMS_timestamp ":" { SGKarakter }
HMS_timestamp = "hhmmss"
SGKarakter = /* zie tabel signaalgroepstand (extern) */
```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
LSGE	LSGE="HHMMSS:RRGR..."	Vraag de linedump van de externe signaalgroepstanden op. Het tijdstempel is actueel en de dump komt overeen met het uitgestuurde beeld. De VRI geeft als waarde <hhmmss:xxx>,<hhmmss:xxx>... etc., waarbij x de karakterwaarde is van de toestand van de externe signaalgroep.

Tabel 20.55 Voorbeelden object LSGE

20.2.28 Events

Akoestische storing

Met akoestische storing worden wijzigingen in de foutstatus van de akoestische signalering gemeld aan de beheerscentrale.

Code	Omschrijving	Detailinfo	VRI.A
1030	Akoestischefout	<BeginEinde>[,<AkRef>]	Ja

BeginEinde Start/Einde indicatie (start =0, einde=1)

AkRef Nummer of naam van signaalgever

21. Openbaar vervoer

21.1 Algemeen Openbaar vervoer

Voor het openbaar vervoer zijn de volgende objecten gedefinieerd:

Object	Omschrijving
OVDEVICE	OV-devices
OVFILTER	OV-filter
OV.LB	OV-logboek
OV.LA	OV-logboek (onb)

Tabel 21.1 Openbaar Vervoer objecten

Om Openbaar Vervoer meldingen door de procesbesturing te kunnen laten filteren, zijn filterregels te definiëren.

Elke binnengekomen melding wordt per filterregel geëvalueerd. Filtering vindt plaats van de eerste filterregel (index 0 in OVFILTER) naar de laatste.

Een melding voldoet aan een filterregel, wanneer alle in de filterregel gespecificeerde (dus niet lege) attributen overeenkomen met de melding.

Voldoet de melding *niet* aan de filterregel, dan dient de volgende filterregel geëvalueerd te worden.

Wanneer een melding *wel* voldoet aan een filterregel, dan wordt de bijbehorende actie door de procesbesturing uitgevoerd. Er is keuze uit 2 acties:

- de melding wordt doorgegeven aan de CVN-interface door het opzetten van de in de filterregel opgegeven detector.;
- de melding wordt verder genegeerd.

Voldoet géén enkele filterregel, dan wordt de melding niet doorgegeven aan de CVN-interface.

De melding wordt doorgegeven aan de CVN-interface door de in de filterregel genoemde detector op de interface gedurende 1 seconde op te zetten.

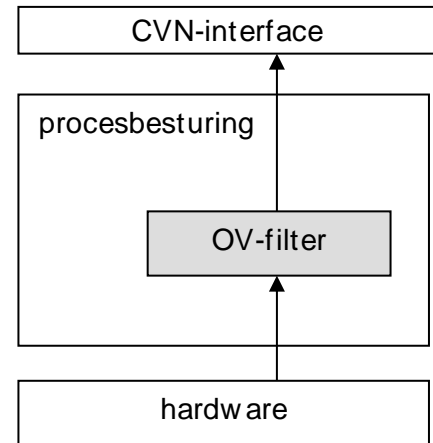
Zijn er geen filterregels aanwezig (OVFILTER-object ontbreekt in dat geval), dan worden de meldingen ongefilterd doorgegeven aan de CVN-interface via het DSI-buffer. Hierdoor is het nog steeds mogelijk het filteren in de verkeersregelapplicatie te laten plaatsvinden. Echter, beïnvloeding op de standaardwijze via IVERA is dan niet mogelijk.

Voor iedere melding wordt een OpenbaarVervoerEvent gegenereerd (OV.LA/LB). Per event wordt aangegeven of de melding al dan niet gefilterd is (dus: of één van de filterregels invloed heeft gehad op het verwerken van de melding). Door het analyseren van het Openbaar Vervoer logboek kan de werking van het filter geëvalueerd worden.

Openbaar vervoer gegevens (seriële selectieve detectie) welke door de VRI ingelezen worden, kunnen via de CVN-interface aan de regelapplicatie aangeboden worden. Hierbij leest de procesbesturing de gegevens van de wegkantsystemen in en biedt deze aan. De gegevens kunnen op 2 plaatsen gefilterd worden: in de procesbesturing of in de verkeersregelapplicatie.

Om tot een uniforme manier van instellen van filtering te komen, is hier een IVERA-object gedefinieerd. Hier wordt er vanuit gegaan dat de filtering in de procesbesturing plaatsvindt. Met het IVERA-object kan het filter worden ingesteld of worden opgevraagd.

Via de objecten OVDEVICE en OVFILTER kan een filter gedefinieerd worden, aan de hand waarvan de procesbesturing seriële selectieve detectie-meldingen al dan niet aan de regelapplicatie doorgeeft. Tevens is een voorziening opgenomen in het openbaar vervoer logboek (OV.LA/LB) waardoor de werking van het filter geëvalueerd kan worden.



Via het object OVDEVICE kan van maximaal 10 apparaten (SICS, VECOM, KAR enz.) de gegevens opgevraagd worden. Dit object kan niet door een gebruiker gewijzigd worden, omdat het de hardware-matige configuratie betreft.

Het object OVFILTER kan worden gebruikt om het openbaar vervoer filter op te vragen of te wijzigen. Het filter bepaalt aan de hand van criteria welke meldingen die ingelezen zijn, doorgegeven mogen worden aan de CVN-interface. Iedere melding wordt bijgeschreven in het OV.LA/LB met daarbij aangegeven of de betreffende melding wel of niet door het filter doorgelaten is. Hiermee kan de werking van het filter geëvalueerd worden.

Het doorgeven aan de CVN-interface kan gerealiseerd worden door gebruik te maken van detectie-ingangen. Per filterregel moet een detectie-ingang aangegeven worden, welke actief gemaakt moet worden, wanneer de OV-melding door de betreffende regel doorgelaten zou worden. De procesbesturing is tevens verantwoordelijk voor het resetten van de detectie-ingang.

21.2 Objecten openbaar vervoer

21.2.1 OVDEVICE

Het object OVDEVICE geeft de OV-devices weer:

Attribuut	Type	Object	Omschrijving
N	1	OVDEVICE	Naam
O	1	OV-devices	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	80	Data-element formaat
S	0		Data-element stapgrootte

Tabel 21.2 Object attributen

Formaat: Openbaar Vervoer devices (80), type 1

Dit objecttype geeft aan van welke Openbaar Vervoer devices de VRI gebruik maakt:

OvDeviceEntry = [OvDeviceNr + "," + OvDeviceType]

OvDevicentr = Nummer van het device (0..9)

OvDeviceType = 0 | 1 | 2 | 3

Waarde	Omschrijving
0.	SICS
1.	VECOM
2.	KAR
3.	VETAG

Tabel 21.3 Openbaar vervoer devices

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
OVDEVICE/#0	OVDEVICE="0,0"	Vraag een element van het object OVDEVICE op.
OVDEVICE:A	OVDEVICE:A="N=OVDEVIC E, T=1,E=10,U=4444,F=80, O='OV-devices'"	Vraag de attributen op van object OVDEVICE
OVDEVICE:E	OVDEVICE:E=10	Opvragen van aantal elementen van object OVDEVICE
OVDEVICE/#10	:E=12	Vraag data-element op hoger dan MAX element bereik

Tabel 21.4 Voorbeelden object OVDEVICE

21.2.2 OVFILTER

Het object OVFILTER geeft het OV-filter weer:

Attribuut	Type	Object	Omschrijving
N	1	OVFILTER	Naam
O	1	OV-filter	Omschrijving
T	1	1	Type
U	0	6644	User Identificatie Control
E	0	NUMOVFILTER	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	81	Data-element formaat
S	0		Data-element stapgrootte

Tabel 21.5 Object attributen

Formaat: Openbaar Vervoer Detectorfilter (81), type 1

Voorbeeld:

Een filterregel welke alle meldingen met lijnnummer 41 doorgeeft aan de CVN-interface:

OvFilterEntry = 2,,,,41,,,,,,,,,,,,,DE0201

Voor de instellingen van het Openbaar Vervoer filter wordt de volgende syntax gehanteerd:

OvFilterEntry =

OvActie + "," +
 [OvDeviceNr] + "," +
 [OvDeviceType] + "," +
 [OvLusNummer] + "," +
 [OvLijnNummer] + "," +
 [OvVoertuigCategorie] + "," +
 [Richtingsaanduiding] + "," +
 [Voertuigstatus] + "," +
 [Prioriteitsklasse] + "," +
 [Type melding] + "," +
 [OvStiptheidsklasse] + "," +
 [OvStiptheidMin] + "," +
 [OvStiptheidMax] + "," +
 [OvRitnummer] + "," +
 [OvRitCategorie] + "," +
 [OvRoute] + "," +
 [OvWagennummer] + "," +
 [Detector]

OvActie = 0 | 1 | 2
 '0' = filterregel niet actief
 als regel voldoet:
 '1' = melding negeren,
 '2' = detector opzetten in CVN-interface

OvDeviceNr = 0..9 (uit OVDEVICE)

OvDeviceType = 0 | 1 | 2 | 3 (uit OVDEVICE)

OvLusNummer = 0..127 (element 0 van de selectieve detectie volgens CVN 3.0)

OvLijnNummer = 0..9999 (element 2 van de selectieve detectie volgens CVN 3.0)

OvVoertuigCategorie = 0..99 (element 1 van de selectieve detectie volgens CVN 3.0)

Richtingsaanduiding = 0..255 (element 6 van de selectieve detectie volgens CVN 3.0)

Voertuigstatus = 0..99 (element 7 van de selectieve detectie volgens CVN 3.0)

Prioriteitsklasse =	0..99 (element 8 van de selectieve detectie volgens CVN 3.0)
Type melding =	0..99 (element 18 van de selectieve detectie volgens CVN 3.0)
OvStiptheidsklasse =	0..99 (element 9 van de selectieve detectie volgens CVN 3.0)
OvStiptheidMin =	-3600..3600 (ondergrens, element 10 van de selectieve detectie volgens CVN 3.0)
OvStiptheidMax =	-3600..3600 (bovengrens, element 10 van de selectieve detectie volgens CVN 3.0)
OvRitnummer =	0..9999 (element 15 van de selectieve detectie volgens CVN 3.0)
OvRitCategorie =	0..99 (element 16 van de selectieve detectie volgens CVN 3.0)
OvRoute =	0..99 (element 17 van de selectieve detectie volgens CVN 3.0)
OvWagennummer =	0..32767 (element 5 van de selectieve detectie volgens CVN 3.0)
Detector =	Naam ; functionele detectornaam zoals getoond in D.I welke éénmalig gezet kan worden n.a.v. een gefilterde melding

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
OVFILTER	OVFILTER="0,,,,,,,,,,,,,""0,, ,,,,,,,,,,,,,"" ...	Uitgangssituatie is een lege OV filter. De waarden dienen binnen aangegeven bereik te zijn volgens obj.def. OvFilterEntry= OvActie + "," + [OvDeviceNr] + "," + [OvDeviceType] + "," + [OvLusNummer] + "," + [OvLijnNummer] + "," + [OvVoertuigCategorie] + "," + [Richtingsaanduiding] + "," + [Voertuigstatus] + "," + [Prioriteitsklasse] + "," + [Type melding] + "," + [OvStiptheidsklasse] + "," + [OvStiptheidMin] + "," + [OvStiptheidMax] + "," + [OvRitnummer] + "," + [OvRitCategorie] + "," + [OvRoute] + "," + [OvWagennummer] + "," + [Detector]
OVFILTER/#0="2, 0,0,X,,,,,,,,,,,,,""Y"	OVFILTER/#0="2,0,0,,X,,,,,,,,,,,,, ,,,,,,,,Y"	Stel een filter regel in waarbij inmeldingen van lijn X doorgelaten worden, en detectielus Y activeert.

Tabel 21.6 Voorbeelden object OVFILTER

21.2.3 OV.LB

Het object OV.LB geeft het OV-logboek weer:

Attribuut	Type	Object	Omschrijving
N	1	OV.LB	Naam
O	1	OV-logboek	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	0 .. 1000	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	82	Data-element formaat
S	0		Data-element stapgrootte

Tabel 21.7 Object attributen

Formaat: OV-event (82), type 1

Een openbaarvervoerevent object bevat een openbaarvervoermelding in de slave in een leesbare ascii-string. Een event bestaat uit een timestamp en een OV-melding

OvEventBericht = TimeStamp + "," + Bevestigd + "," + OV-melding

TimeStamp = "jjjjmdd:uummss"

Bevestigd = 0 | 1

OV-melding =

OvFilterGepasseerd + "," +

OvDeviceNr + "," +
OvDeviceType + "," +
[OvLusNummer] + "," +
[OvLijnNummer] + "," +
[OvVoertuigCategorie] + "," +
[Richtingsaanduiding] + "," +
[Voertuigstatus] + "," +
[Prioriteitsklasse] + "," +
[Type melding] + "," +
[OvStiptheidsklasse] + "," +
[OvStiptheid] + "," +
[OvSnelheid] + "," +
[OvStopstreepAfstand] + ',' +
[OvStopstreepTijd] + "," +
[OvRitnummer] + "," +
[OvRitCategorie] + "," +
[OvRoute] + "," +
[OvWagennummer] + "," +
[Detector]

OVFilterGepasseerd = 0 | 1 | 2 | 3

de waarde '0' geeft aan dat er geen enkele filterregel van toepassing is op de melding.
de waarde '1' geeft aan dat de melding onder invloed van een filterregel is doorgegeven.
de waarde '2' geeft aan dat de melding onder invloed van een filterregel is genegeerd.
de waarde '3' geeft aan dat de melding bij ontbreken van filterregels is doorgegeven aan het DSI-buffer van de CVN-interface.

OvDeviceNr = 0..9 (uit OVDEVICE)

OvDeviceType = 0 | 1 | 2 | 3 (uit OVDEVICE)

OvLusNummer = 0..127 (element 0 van de selectieve detectie volgens CVN 3.0)

OvLijnNummer = 0..9999 (element 2 van de selectieve detectie volgens CVN 3.0)

OvVoertuigCategorie = 0..99 (element 1 van de selectieve detectie volgens CVN 3.0)

Richtingsaanduiding = 0..255 (element 6 van de selectieve detectie volgens CVN 3.0)

Voertuigstatus = 0..99 (element 7 van de selectieve detectie volgens CVN 3.0)

Prioriteitsklasse = 0..99 (element 8 van de selectieve detectie volgens CVN 3.0)

Type melding = 0..99 (element 18 van de selectieve detectie volgens CVN 3.0)

OvStiptheidsklasse = 0..99 (element 9 van de selectieve detectie volgens CVN 3.0)

OvStiptheid = -3600..3600 (element 10 van de selectieve detectie volgens CVN 3.0)

OvSnelheid = 0..99 (element 12 van de selectieve detectie volgens CVN 3.0)

OvStopstreepAfstand = -99..9999 (element 13 van de selectieve detectie volgens CVN 3.0)

OvStopstreepTijd = 0..255 (element 14 van de selectieve detectie volgens CVN 3.0)

OvRitnummer = 0..9999 (element 15 van de selectieve detectie volgens CVN 3.0)

OvRitCategorie = 0..99 (element 16 van de selectieve detectie volgens CVN 3.0)

OvRoute = 0..99 (element 17 van de selectieve detectie volgens CVN 3.0)

OvWagennummer = 0..32767 (element 5 van de selectieve detectie volgens CVN 3.0)

Detector = Naam ; functionele detectornaam zoals getoond in D.I welke éénmalig gezet kan worden n.a.v. een gefilterde melding

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
OV.LB	OV.LB= "JJJMMDD:UUMMSS,1,0,0,0,77,1,0,0,0, 0,0,0,0,0,0,0,0,0,0,Y", "JJJMMDD:UUMMSS,1,2,0,0,77,1,0,0,0, 0,0,0,0,0,0,0,0,0,0,Y", "JJJMMDD:UUMMSS,1,0,0,0,77,1,0,0,0, 0,0,0,0,0,0,0,0,0,0,Z", "JJJMMDD:UUMMSS,1,1,0,0,77,1,0,0,0, 0,0,0,0,0,0,0,0,0,0,Y"	Opvragen waarden van OV-meldingen van de VRI op. Meldingen zijn als bevestigde meldingen terug te vinden in het algemene logboek.

Tabel 21.8 Voorbeelden object OV.LB

21.2.4 OV.LA

Het object OV.LB geeft het OV-logboek weer:

Attribuut	Type	Object	Omschrijving
N	1	OV.LA	Naam
O	1	OV-logboek (onbevestigd)	Omschrijving
T	1	1	Type
U	0	6666	User Identificatie Control
E	0	0 .. 1000	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	82	Data-element formaat
S	0		Data-element stapgrootte

Tabel 21.9 Object attributen

Formaat: OV-event (82), type 1

Voor het formaat, zie paragraaf: 21.2.3.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
OV.LA	OV.LA= "JJJMMDD:UUMMSS,0,1,0,0,77,1,0,0,0, 0,0,0,0,0,0,0,0,0,0,Y", "JJJMMDD:UUMMSS,0,0,0,0,77,1,0,0,0, 0,0,0,0,0,0,0,0,0,0,Z", "JJJMMDD:UUMMSS,0,2,0,0,77,1,0,0,0, 0,0,0,0,0,0,0,0,0,0,Y", "JJJMMDD:UUMMSS,0,0,0,0,77,1,0,0,0, 0,0,0,0,0,0,0,0,0,0,Y"	Opvragen waarden van OV-meldingen van de VRI op. Een element zonder waarde moet leeg zijn. In het gewenst resultaat wordt gesuggereerd dat hier 0 moet staan. De response kan dus zijn: "JJJMMDD:UUMMSS,0,2,0,,77,1,,,,,,,,,,,,,Y" De selectieve detectie gebeurtenissen van de OVFILTER test zijn geregistreerd in het logboek.

Tabel 21.10 Voorbeelden object OV.LA

22. Noodstroom

22.1 Algemeen noodstroom

Voor de noodstroom voorzieningen zijn de volgende objecten gedefinieerd:

Object	Omschrijving
NOODSTROOM.I	Index object noodstroom voorziening
NOODSTROOM	Instellingen voor de noodstroom voorziening
NOODSTROOM.A	Actuele toestand noodstroom voorziening
NOODSTROOM.LA	Logboek (onb.) noodstroom voorziening
NOODSTROOM.LB	Logboek (bev.) noodstroom voorziening

Tabel 22.1 noodstroom objecten

Met enige regelmaat wordt in een verkeersregelininstallatie een UPS of noodstroomvoorziening toegepast. Een noodstroomvoorziening dient bij netspanning uitval de VRI te kunnen voorzien van het benodigde vermogen, voor kortere of langere tijd. Om de werking van deze voorziening te kunnen garanderen, is het zaak om de (laad)toestand van dit apparaat te kunnen opvragen en bewaken.

Met behulp van de betreffende noodstroom objecten kan de actuele toestand en de gebeurtenissen m.b.t. een noodstroomvoorziening worden opgevraagd, en kunnen parameters worden ingesteld.

22.2 Objecten noodstroom

22.2.1 NOODSTROOM.I

Het object NOODSTROOM.I geeft de index van het object noodstroom voorziening weer:

Attribuut	Type	Object	Omschrijving
N	1	NOODSTROOM.I	Naam
O	1	index object noodstroom voorziening	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	25	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 22.2 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Het standaardiseren van namen van verschillende parameters is niet zozeer een IVERA protocol aanpassing, als wel het maken van een extra restrictie op het gebruik van namen in een aantal indexobjecten. Het gaat hierbij over de indexobjecten, die gekoppeld zijn aan de parameters in CCOL of RWS-C, die reeds een naam hebben zoals de volgende indexobjecten:

De namen in deze objecten moeten exact overeenkomen met de namen die in CCOL of RWS-C zijn geconfigureerd. Zij bepaalde namen leeg gelaten, dan wordt in IVERA een invulling aangegeven, die overeenkomt met het elementnummer en het type parameter.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
NOODSTROOM.I	NOODSTROOM.I=UPS_actief, UPS_comfout, UPS_defect, UPS_tijd_act, UPS_tijd_over, ACCU_laag, ACCU_vol, ACCU_Q, ACCU_V, ACCU_OA, ACCU_LA, ACCU_C, ACCU_T, Min_ACCU_V, Max_ACCU_V, Max_ACCU_OA, Max_ACCU_LA, Min_ACCU_T, Max_ACCU_T, Fab1, Fab2, Fab3, Fab4, Fab5, Fab6	Opvragen van de indexnamen van het object NOODSTROOM.I

Tabel 22.3 Voorbeelden object NOODSTROOM.I

In het object NOODSTROOM.A is de actuele toestand van de noodstroomvoorziening uit te lezen. In het object NOODSTROOM.I is voor elk element uit NOODSTROOM.A de naam vastgelegd. De indeling van het object is als volgt:

Element	Waarde (NOODSTROOM.A)	Indexnaam (NOODSTROOM.I)	Omschrijving
0	-1 .. 2	UPS_actief	-1 = Geen UPS aanwezig 0 = UPS niet actief 1 = UPS is actief 2 = UPS is actief in testbedrijf
1	0 .. 1	UPS_comfout	0 = UPS communicatie OK 1 = UPS communicatie FOUT
2	0 .. 1	UPS_defect	0 = UPS is OK 1 = UPS is defect
3	-1 ... MAXINT	UPS_tijd_act	-1 = niet gebruikt Tijd UPS actief (in minuten)
4	-1 ... MAXINT	UPS_tijd_over	-1 = niet gebruikt Verwachte tijd UPS nog actief (minuten)
5	-1 ... 1	ACCU_laag	-1 = niet gebruikt 0 = ACCU lading boven minimum grenswaarde 1 = ACCU lading minimum onderschreden
6	-1 ... 1	ACCU_vol	-1 = niet gebruikt 0 = ACCU lading onder maximum grenswaarde 1 = ACCU lading boven maximum grenswaarde
7	-1 ... 100	ACCU_Q	-1 = niet gebruikt Accu kwaliteit in procenten van de nieuwwaarde
8	-1 ... MAXINT	ACCU_V	-1 = niet gebruikt Accu spanning in eenheden van 0.1V
9	-1 ... MAXINT	ACCU_OA	-1 = niet gebruikt Accu ontladstroom in eenheden van 1mA
10	-1 ... MAXINT	ACCU_LA	-1 = niet gebruikt Accu laadstroom in eenheden van 1mA
11	-1 ... 100	ACCU_C	-1 = niet gebruikt Accu lading in procenten van de nieuwwaarde
12	-99 ... 100	ACCU_T	-99 = niet gebruikt Accu temperatuur in °C
13	-1 ... MAXINT	Min_ACCU_V	-1 = niet gebruikt Min. Accu spanning in eenheden van 0.1V
14	-1 ... MAXINT	Max_ACCU_V	-1 = niet gebruikt Max. Accu spanning in eenheden van 0.1V
15	-1 ... MAXINT	Max_ACCU_OA	-1 = niet gebruikt Max. Accu ontladstroom mA
16	-1 ... MAXINT	Max_ACCU_LA	-1 = niet gebruikt Max. Accu laadstroom in mA
17	-99 ... 100	Min_ACCU_T	-99 = niet gebruikt Min. Accu temperatuur in °C
18	-99 ... 100	Max_ACCU_T	-99 = niet gebruikt Max. Accu temperatuur in °C
19		Fab1	Fabrikant specifiek (indexnaam en waarde)
20		Fab2	Fabrikant specifiek (indexnaam en waarde)
21		Fab3	Fabrikant specifiek (indexnaam en waarde)
22		Fab4	Fabrikant specifiek (indexnaam en waarde)
23		Fab5	Fabrikant specifiek (indexnaam en waarde)
24		Fab6	Fabrikant specifiek (indexnaam en waarde)

Tabel 22.4 Weergave object NOODSTROOM.I

22.2.2 NOODSTROOM

Het object NOODSTROOM geeft de instellingen voor de noodstroom voorziening weer:

Attribuut	Type	Object	Omschrijving
N	1	NOODSTROOM	Naam
O	1	Instellingen voor de noodstroom voorziening	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	25	aantal data-elementen
I	1	NOODSTROOM.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 22.5 Object attributen

In het object NOODSTROOM worden de relevante instellingen ten behoeve van de noodstroom vastgelegd. De indeling van dit object komt overeen met NOODSTROOM.A en NOODSTROOM.I.

Element	Waarde	Omschrijving
0	0-1	UPS aanwezig (0 = geen UPS aanwezig, 1 = UPS aanwezig)
1	0	gereserveerd
2	0	gereserveerd
3	0	gereserveerd
4	0	gereserveerd
5	0-100	Ondergrens Accu lading (%). Komt de accu lading hieronder dan wordt de waarde Accu laag '1'.
6	0-100	Bovengrens Accu lading (%). Komt de accu lading boven deze grens dan wordt de waarde Accu vol '1'.
7	0	gereserveerd
8	0	gereserveerd
9	0	gereserveerd
10	0	gereserveerd
11	0	gereserveerd
12	0	gereserveerd
13	0-32767	Ondergrens Accu spanning (0.1V). Komt de accuspanning hieronder dan wordt het overeenkomstige event in het logboek geschreven.
14	0-32767	Bovengrens Accuspanning (0.1V). Komt de accuspanning boven deze grens, dan wordt een event in het logboek geschreven.
15	0-32767	Bovengrens Accu ontladstroom (mA). Komt de ontladstroom boven deze grens, dan wordt een event in het logboek geschreven.
16	0-32767	Bovengrens Accu laadstroom (mA). Komt de laadstroom boven deze grens, dan wordt een event in het logboek geschreven.
17	-50 - +100	Ondergrens Accu temperatuur (°C). Komt de temperatuur onder deze grens, dan wordt een event in het logboek geschreven.
18	-50 - +100	Bovengrens Accu temperatuur (°C). Komt de temperatuur boven deze grens, dan wordt een event in het logboek geschreven.
19		Fabrikant specifiek
20		Fabrikant specifiek
21		Fabrikant specifiek
22		Fabrikant specifiek
23		Fabrikant specifiek
24		Fabrikant specifiek

Tabel 22.6 Instellingen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
NOODSTROOM	NOODSTROOM=1,0,0,0,0,20,90,0,0,0,0,0,0,0,11,13,5000,5000,0,40,0,0,0,0,0,0	Vraag de gemaakte instellingen op van het object NOODSTROOM .
NOODSTROOM/Max_ACCU_T=40	NOODSTROOM/Max_ACCU_T=40	Stel de accu temp. bovengrens in op 40 graden Celsius

Tabel 22.7 Voorbeelden object NOODSTROOM

22.2.3 NOODSTROOM.A

Het object NOODSTROOM.A geeft de actuele toestand van de noodstroom voorziening weer:

Attribuut	Type	Object	Omschrijving
N	1	NOODSTROOM.A	Naam
O	1	Actuele toestand noodstroom voorziening	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	25	aantal data-elementen
I	1	NOODSTROOM.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 22.8 Object attributen

Voorbeelden

NOODSTROOM.A	;vraag alle actuele waarden op van het object noodstroom.a
NOODSTROOM.A=0,0,0,-1,-1,0,0,90,123,0,150,71,25,113,134,1045,988,1,32,0,0,0,0,0,0	
NOODSTROOM/#11	;vraag de acculading op
NOODSTROOM/#11=71	
NOODSTROOM/Max_ACCU_T	;vraag de maximale gemeten accu temperatuur op
NOODSTROOM/Max_ACCU_T=32	

LET OP: De gemeten minimum en maximum waarden (elementen 13 t/m 18) worden bij spanningsuitval van de VRI bewaard. De waarden kunnen worden gereset met het commando VRI.C/#0=4032.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
NOODSTROOM.A	NOODSTROOM.A=0,0,0,-1,-1,0,0,90,123,0,150,71,25,113,134,1045,988,1,32,0,0,0,0,0,0	Opvragen actuele waarden van het object NOODSTROOM.A.
NOODSTROOM.A/ACCU_T	NOODSTROOM.A/ACCU_T=25	Opvragen van de actuele accu temperatuur.
VRI.C/#0=4032	VRI.C/#0=4032	Reset de minima en maxima van het object NOODSTROOM.A
NOODSTROOM.A	NOODSTROOM.A=0,0,0,-1,-1,0,0,90,123,0,150,71,25,0,0,0,0,0,0,0,0,0,0,0,0	Opvragen van de actuele waarden van het object NOODSTROOM.A ter controle van de reset.

Tabel 22.9 Voorbeelden object NOODSTROOM.A

22.2.4 NOODSTROOM.LA

Het object NOODSTROOM.LA geeft het logboek (onbevestigd) van de noodstroom voorziening weer:

Attribuut	Type	Object	Omschrijving
N	1	NOODSTROOM.LA	Naam
O	1	Logboek (onbevestigd) noodstroom voorziening	Omschrijving
T	1	1	Type
U	0	6666	User Identificatie Control
E	0	0 .. 100	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	100	Data-element formaat
S	0		Data-element stapgrootte

Tabel 22.10 Object attributen

Formaat: Event (100), type 1

Een eventobject bevat een event in de slave in een leesbare ASCII string. Een event bestaat uit een timestamp, een eventcode en een nadere detaillering van het event in tekst. De detailinformatie is afhankelijk van de eventcode.

```
EventBericht = TimeStamp + "," + Bevestigd + "," + EventCode + "," + DetailInfo
TimeStamp = "jjjjmdd:uummss"
Bevestigd = 0 | 1
EventCode = IntegerWaarde
DetailInfo = Detailstring
```

In de onderstaande tabel zijn de mogelijke events opgenomen:

Code	Omschrijving	Detailinfo
3500	UPS actief	<Oorzaak>
3501	UPS communicatie fout	<Status>
3502	UPS defect	<Status>
3503	Accu status	<Laag>, <Vol>, <Kwaliteit>, <Spanning>, <Ontlaadstroom>, <Laadstroom>, <Lading>, <Temperatuur>, <Tijdsduur1>, <Tijdsduur2>
3504	Accu min. spanning onderschreden	<Spanning>, <min. Spanning>
3505	Accu max. spanning overschreden	<Spanning>, <max. Spanning>
3506	Accu max. ontladstroom overschreden	<Ontlaadstroom>, <max. Ontlaadstroom>
3507	Accu max. laadstroom overschreden	<Laadstroom>, <max. Laadstroom>
3508	Accu min. temperatuur onderschreden	<Temperatuur>, <min. Temperatuur>
3509	Accu max. temperatuur overschreden	<Temperatuur>, <max. Temperatuur>

Tabel 22.11 Noodstroom events

Oorzaak	0 = netspanning aanwezig 1 = netspanning afwezig 2 = testbedrijf
Status	0= nee, 1=ja
Tijdsduur1	Aantal minuten dat de UPS actief is
Tijdsduur2	Aantal minuten dat de UPS naar verwachting nog actief kan zijn
Laag	1 = Accu lading is beneden de ingesteld grenswaarde uit het object NOODSTROOM (element 0)
Hoog	1 = Accu lading is boven de ingestelde grenswaarde uit het object NOODSTROOM (element 1)
Kwaliteit	Accu kwaliteit in procenten van de nieuwwaarde
Spanning	Accu spanning in eenheden van 0.1V
Ontlaadstroom	Accu ontladstroom in eenheden van 1mA

Laadstroom	Accu laadstroom in eenheden van 1mA
Lading	Accu lading in procenten van de nieuwwaarde
Temperatuur	Accu temperatuur in °C
Min. Spanning	Min. Accu spanning in eenheden van 0.1V
Max. Spanning	Max. Accu spanning in eenheden van 0.1V
Max. Ontlaadstroom	Max. Accu ontlaadstroom mA
Max. Laadstroom	Max. Accu laadstroom in mA
Min. Temperatuur	Min. Accu temperatuur in °C
Max. Temperatuur	Max. Accu temperatuur in °C

Indien een waarde onbekend is mag een veld leeg worden gelaten.

In het object NOODSTROOM.LA worden alle onbevestigde events m.b.t. de noodstroom vastgelegd, voorzien van datum tijd stempel. Zodra ze door de centrale zijn bevestigd worden ze uit het object NOODSTROOM.LA verwijderd.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
NOODSTROOM.L A	NOODSTROOM.LA="20131023:101010,0 ,3500,1", "20131023:101110,0,3500,0"	Vraag het logboek op.
NOODSTROOM.L A/#0=""	NOODSTROOM.LA/#=0	Bevestig alle meldingen. Herhaal laatste commando zo vaak als nodig is.

Tabel 22.12 Voorbeelden object NOODSTROOM.LA

22.2.5 NOODSTROOM.LB

Het object NOODSTROOM.LB geeft het logboek (bevestigd) van de noodstroom voorziening weer:

Attribuut	Type	Object	Omschrijving
N	1	NOODSTROOM.LB	Naam
O	1	Logboek (bevestigd) noodstroom voorziening	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	0 .. 100	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	100	Data-element formaat
S	0		Data-element stapgrootte

Tabel 22.13 Object attributen

Formaat: Event (100), type 1

Een eventobject bevat een event in de slave in een leesbare ASCII string. Een event bestaat uit een timestamp, een eventcode en een nadere detaillering van het event in tekst. De detailinformatie is afhankelijk van de eventcode.

```
EventBericht = TimeStamp + "," + Bevestigd + "," + EventCode + "," + DetailInfo
TimeStamp = "jjjjmdd:uummss"
Bevestigd = 0 | 1
EventCode = IntegerWaarde
DetailInfo = Detailstring
```

In het logboek NOODSTROOM.LB staan alle noodstroom gerelateerde events (bevestigd en onbevestigd). Het formaat van dit logboek is gelijk aan die van het NOODSTROOM.LA, zoals opgenomen in paragraaf 22.2.4.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
NOODSTROOM.L B	NOODSTROOM.LB="20131023:101010,1 ,3503,0,0,75,12,500,0,75,32,0,3400"	Vraag het logboek op met bevestigde events.

Tabel 22.14 Voorbeelden object NOODSTROOM.LB

23. Telprogramma

23.1 Algemeen telprogramma

Voor het telprogramma zijn de volgende objecten gedefinieerd:

Object	Omschrijving
TELINST	Instellingen telprogramma
TELDATA	Data telprogramma
TELMON	Actuele data telprogramma

Tabel 23.1 telprogramma objecten

Verkeersregeltoestellen hebben de beschikking over dataverzameling in de vorm van teldata. Passage van voertuigen en de registratie van roodlichtrijders, in het algemeen bepaald op de koplussen. Roodlichtrijders worden geteld in combinatie met de roodstand van de signaalgroep.

Het standaardtelprogramma biedt de mogelijkheid om tot maximaal 48 telcircuits te definiëren.

De instellingen per telcircuit kunnen worden uitgelezen door het lezen van het object "TELINST". Het wijzigen van de instellingen kan door het schrijven naar het object "TELINST".

De door de VRI verzamelde telgegevens kunnen worden uitgelezen door het lezen van het object "TELDATA".

De actuele waarden van tellingen kunnen opgevraagd worden door het lezen van het object TELMON.

23.2 Objecten telprogramma

23.2.1 TELINST

Het object TELINST geeft de instellingen telprogramma voor alle telcircuits weer:

Attribuut	Type	Object	Omschrijving
N	1	TELINST	Naam
O	1	Instellingen telprogramma	Omschrijving
T	1	1	Type
U	0	6664	User Identificatie Control
E	0	NUMTELDATA	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	200	Data-element formaat
S	0		Data-element stapgrootte

Tabel 23.2 Object attributen

Formaat: Instellingen telprogramma (200), type 1

TELINST

= datum + “,”
 + tijd + “,”
 + herhalingstijd + “,”
 + duur + “,”
 + mode + “,”
 + functie + “,”
 + roodstart + “,”
 + stop + “,”
 + detector + “,”
 + signaalgroep

datum = jaar * 10000 + maand * 100 + dag van de maand

tijd = uur * 100 + minuut

herhalingstijd = uur * 100 + minuut

duur = uur * 100 + minuut

mode

= “S” ; stop teller; op dit circuit wordt niet (meer) geteld
 | “P” ; 12-168 perioden tellen
 | “C” ; continu tellen

functie

= “D” ; doorroodrijders tellen
 | “I” ; intensiteit tellen
 | “R” ; roodovergangen tellen

roodstart

= IntegerWaarde ; tijd in 1/10 seconde.

Stop

= IntegerWaarde ; tijd in 1/10 seconde

Detector

= Naam ; functionele detectornaam zoals getoond in D.I

Signaalgroep

= Naam ; functionele signaalgroepnaam zoals getoond in SG.I

De volgende dimensies m.b.t. het telprogramma gelden:

	min	max
aantal telcircuits	24	48
aantal telperioden	12	168

De VRI geeft zelf in de eigenschappen van het betreffende object (TELINST) het werkelijk aantal aanwezige telcircuits aan.

Het aantal aanwezige telperioden per circuit is herleidbaar uit de opmaak van de response van het object TELDATA, zie paragraaf 0.

Hieronder wordt in de specificatie uitgegaan van de maximale dimensies; echter, de VRI kan in de objecteigenschappen andere dimensies aangeven.

Functies

Per circuit bestaat een koppeling met een functie. Alvorens het tellen kan worden gestart, zal per circuit de te gebruiken hardware (signaalgroep en/of detector) moeten worden gekozen alsook de functie waarvoor en de tijdstippen waarop wordt geteld. Er zijn functies voor het tellen van:

- * de verkeersintensiteit met behulp van detectoren;
- * het aantal roodovergangen per signaalgroep;
- * het aantal doorroodrijders (door het koppelen van een detector aan een signaalgroep).

Afhankelijk van de functie moeten de volgende gegevens worden ingevoerd:

x = minimaal opgeven, - niet benodigd

	functie	detector	signaalgroep
de verkeersintensiteit	I	x	-
het aantal roodovergangen	R	-	x
het aantal doorroodrijders	D	x	x

Wanneer de functie middels het IVERA-object TELINST gewijzigd wordt, zal foutcode :E=16 (...) geretourneerd worden, wanneer niet aan bovenstaande voldaan is (gelijktijdig wijzigen van functie en detector en/of signaalgroep is wel toegestaan).

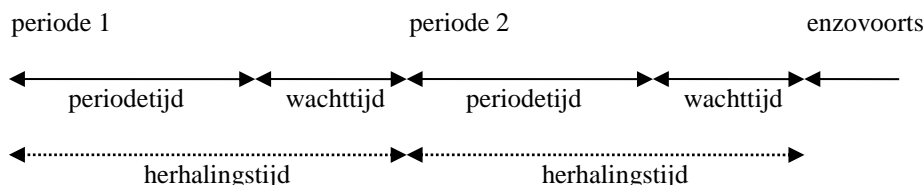
Hetzelfde geldt wanneer de waarde van de detector en/of de signaalgroep gewijzigd wordt.

Het gebruik van een combinatie van functies op een detector is mogelijk door een aantal circuits (met verschillend gedefinieerde functie) aan die detector te koppelen.

Telperioden

De telling start op het moment dat de interne klok van de VRI de opgegeven startdatum en -tijdstip bereikt. Als het tellen gestart is, wordt er vervolgens in maximaal 168 perioden geteld. De perioden kunnen worden afgewisseld met een wachttijd.

Periodeduur en wachttijd vormen samen de herhalingsperiode.



Indien de herhalingsperiode kleiner is dan de periodeduur, dan worden de perioden aansluitend doorgeteld.

Telinstellingen

Ongeacht de functie moeten per circuit de volgende gegevens worden ingevoerd:

- * datum: de begindatum van de telling.
- * tijd: de begintijd van de telling.
- * herhalingstijd: na de herhalingstijd wordt met een nieuwe telperiode aangevangen.
- * duur: gedurende dit tijdsinterval wordt er vanaf de aanvang van de telperiode geteld.
- * mode:
 - * stop teller: op dit telcircuit wordt niet geteld;
 - * 12-168 periodes tellen: de telling stopt na 12-168 periodes;
 - * continu tellen: de telling loopt continu door.

Met de grens roodstart wordt aangegeven hoeveel seconde na het rood worden van de gekoppelde signaalgroep de telling moet beginnen. Met de grens stop wordt aangegeven hoeveel seconde na het rood worden van de gekoppelde signaalgroep de telling moet eindigen. Als de signaalgroep binnen de gestelde tijd niet meer rood is stopt de telling automatisch. Wanneer tijdens het instellen de opgegeven roodstart-waarde groter is dan de stopwaarde, dan wordt fout :E16 geretourneerd.

Een instelling die ongewijzigd dient te blijven kan met een “-“ worden aangegeven. De VRI controleert of dat het aantal argumenten in de string klopt, zo niet dan geeft de VRI een error-melding.

De default-instelling (na een koude start van de VRI) van een telregister is:

```
telinst/#0="19930101,0,0,0,S,I,0,0,0,0"
```

Bij een starttijdstip in het verleden moet er nagegaan worden of de telling nog actief zou moeten zijn. Als dat het geval is, wordt de telling bij het begin van het volgende tellingtijdvak alsnog gestart.

Voorbeelden	
TELINST/#1="19970918,600,100,15,P,I,-,-,D02_1,-"	Instelling telcircuit 2.
TELINST/*="19970918,600,100,15,P,I,-,-,-,-"	Instelling alle telcircuits
TELINST/*="19970918,600,-,-,-,-,-,-,-"	Instelling datum en starttijd alle telcircuits

Tabel 23.3 Telcircuitinstellingen

23.2.2 TELDATA

Het object TELDATA geeft de data telprogramma voor alle telcircuits weer:

Attribuut	Type	Object	Omschrijving
N	1	TELDATA	Naam
O	1	data telprogramma	Omschrijving
T	1	1	Type
U	0	6664	User Identificatie Control
E	0	NUMTELDATA	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	201	Data-element formaat
S	0		Data-element stapgrootte

Tabel 23.4 Object attributen

Formaat: Data telprogramma (201), type 1

```
TELDATA
= periode
+ { “,” + IntegerWaarde }
```

De ‘periode’ komt overeen met de actuele periode (1..168) van het telcircuit indien het telcircuit actief is. Indien het tellen in het telcircuit niet actief is, is de waarde van ‘periode’ gelijk aan 0. Na de actuele periode volgen door middel van een komma gescheiden de telwaardes per periode. De teldata per circuit kan worden gewist door het schrijven naar het object “TELDATA”.

De eerste telling wordt weggeschreven in de resultaten van periode 1. Het telresultaat van de actieve periode kan een actueel getal bevatten; dat wil zeggen dat dan de telling direct wordt bijgewerkt (dus niet nadat de periode verlopen is). Een volgende telperiode wordt gestart als de opgegeven herhalingsstijd verstreken is, volgende tellingen worden in opvolgende periodetellers weggeschreven.

Als de 168^e periode is verlopen zal het telcircuit stoppen, tenzij opgegeven is dat er continu geteld moet worden (zie continu tellen).

Telresultaten worden pas overschreven op het moment dat de periode behorende bij een telresultaat gestart is. Resultaten worden wel verwijderd (op -1 gezet), wanneer een teller geherconfigureerd wordt. Bij storing van de detector waarop geteld wordt, wordt als resultaat in TELDATA -2 gegeven.

Telresultaten blijven na spanningsuitval en herstart bewaard.

Het aantal IntegerWaarden is afhankelijk van de dimensies van de telcircuits.

Continu tellen

Wanneer continu geteld wordt en het maximaal aantal perioden is geteld, dan zal de starttijd/-datum bij het ingaan van een nieuwe periode door de VRI aangepast worden. Deze nieuwe waarde is met “TELINST” op te vragen. Dus: de starttijd/-datum geeft altijd het tijdstip aan waarop periode 1 in TELDATA gestart is.

Wanneer bij continu tellen de 168^e periode afgelopen is, zijn er 2 mogelijkheden voor de 169^e periode:

1. De actieve periode wordt teruggezet naar 1; de 169^e periode overschrijft de eerste telperiode. De telwaarden van overige perioden blijven behouden. Bij de volgende perioden wordt de actieve periode telkens weer opgehoogd.
2. De actieve periode blijft continu op 168 staan na afloop van elke nieuwe periode. De telwaarden van alle perioden schuiven 1 positie op "naar links". Hierdoor verdwijnt de waarde van periode 1 en komt positie 168 vrij zodat de nieuwe telling hierin opgeslagen kan worden.

Merk op dat bij mogelijkheid 1 de starttijd/-datum elke 168 perioden aangepast wordt. Bij mogelijkheid 2 verandert deze na de 168^e periode elke periode weer.

Nadat de spanning uitgevallen is geweest, wordt de actieve periode zóveel perioden opgeschoven als de spanningsuitval heeft geduurd. Bij het bereiken van het maximaal aantal beschikbare perioden worden de starttijd en –datum aangepast. Spanningsuitval kan langer duren dan (maximaal aantal beschikbare perioden * MAX(herhalingstijd, periodeduur)).

Voorbeeld continu tellen (antwoorden uit VRI)

```
telinst/#0="20050628,1040,2,2,C,R,0,0,021,Ri-2"  
teldata/#0="1,0,8,12,14,11,15,11,16,11,17,11,18"  
telinst/#0="20050628,1040,2,2,C,R,0,0,021,Ri-2"  
teldata/#0="2,15,0,12,14,11,15,11,16,11,17,11,18"  
telinst/#0="20050628,1040,2,2,C,R,0,0,021,Ri-2"  
teldata/#0="3,15,16,0,14,11,15,11,16,11,17,11,18"  
telinst/#0="20050628,1040,2,2,C,R,0,0,021,Ri-2"  
teldata/#0="4,15,16,17,0,11,15,11,16,11,17,11,18"
```

Het hier beschreven gedrag is het gedrag dat de implementaties in versie 2 en hoger van IVERA vertonen. In sommige implementaties van IVERA 1.30 komen afwijkingen van dit gedrag voor.

23.2.3 TELMON

Het object TELMON geeft de actuele data telprogramma voor alle telcircuits weer:

Attribuut	Type	Object	Omschrijving
N	1	TELMON	Naam
O	1	Actuele data telprogramma	Omschrijving
T	1	1	Type
U	0	44444	User Identificatie Control
E	0	NUMTELDATA	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	202	Data-element formaat
S	0		Data-element stapgrootte

Tabel 23.5 Object attributen

Formaat: Actuele data telprogramma (202), type 1

```

TELMON
= datum + ","
+ tijd + ","
+ duur + ","
+ telwaarde

datum =
jaar * 10000 + maand * 100 + dag van de maand      ; actuele datum

tijd =
uur * 100 + minuut                                  ; actuele tijd

duur =
uur * 100 + minuut                                  ; tijd dat deze actuele telling al loopt

```

Na datum, tijd en duur volgt de actuele telwaarde van het telcircuit.
Als een telling niet loopt wordt in de actuele waarde -1 teruggegeven.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TELMON/#0	TELMON/#0=<jjjjmmdd>+","<hhmm>+","<hhmm>","<telwaarde>	Vraag een element op van object TELMON. Resultaat zinn de actuele waarden, waarbij waarden volgens object definitie formaat zijn.
TELMON:A	TELMON:A="N=TELMON,T=1,E=<NUMTELDATA>,U=4444,F=202,O='Actuele data telprogramma'"	Vraag attributen op van het object TELMON
TELMON/#0=<jjjjmmdd>+","<hhmm>+","<hhmm>","<telwaarde>	:E=11	Wijzigen niet toegestaan.

Tabel 23.6 Voorbeelden object TELMON

24. Events en Alarms

24.1 Algemeen Events en Alarms

Voor de Events en Alarms zijn de volgende objecten gedefinieerd:

Object	Omschrijving
EVENTLYST.I	Eventnummers als tekststring
EVENTLYST.INFO	Detailinformatie over het event
VRIFOUT.I	Index foutcodes
VRIFOUT	Actuele foutcode
VRIFSUB.I	Index subsystemen
VRIFSUB	Fout status van subsystemen.
VRI.LB	VRI-logboek
VRI.LA	VRI-logboek (onb)

Tabel 24.1 Events en Alarms objecten

In de verkeersregelininstallatie kunnen allerlei events (waaronder alarmen) optreden die aan de centrale moeten worden gemeld. In de centrale wordt aan de hand van de melding bepaald welke actie moet worden ondernomen. Verder moet de VRI de events opslaan zodat ze eventueel later kunnen worden uitgelezen voor analyse.

Binnen het IVERA protocol is het melden van events als volgt geregeld:

- In de VRI zijn objecten gedefinieerd die als queue fungeren (event objecten). Deze objecten bevatten alle nog niet door de centrale bevestigde events.
- Bij het optreden van een event dat wordt opgeslagen in het object VRI.LB, en dat in het DATACOM/#3 filter is opgenomen, stuurt de slave een <BerichtSlaveTrigger>. De TriggerCode in dit bericht komt overeen met de eventcode zoals omschreven in object het VRI.LB.
- Bij het optreden van ander type events wordt **geen** <BerichtSlaveTrigger> verzonden. De centrale moet in dit geval op regelmatige tijdstippen de bijbehorende event objecten lezen.
- Het bevestigen van events gebeurt door het schrijven naar de objecten.
- Indien er geen verbinding is met de centrale, zal de VRI afhankelijk van de aard van het event zelfstandig een verbinding met de centrale opbouwen.

NB. Zie IVERA functionele specificatie voor een voorbeeld van het lezen en bevestigen van events.

NB. Zie paragraaf "VRI commando's" voor het sturen van commando's naar een VRI en het resetten van events.

NB: Zie paragraaf "DATACOM" voor een beschrijving van het event filter.

Het EVENTLYST object kan worden gebruikt om een overzicht van de ondersteunde events op te vragen, met uitzondering van applicatiespecifieke events. Deze lijst kan worden gebruikt om de triggerlist te vullen vanaf bijvoorbeeld de centrale. Er zijn NUMEVENT events aanwezig.

24.2 Objecten Events en Alarms

24.2.1 EVENTLYST.I

Het object EVENTLYST.I geeft de eventnummers als tekststring weer:

Attribuut	Type	Object	Omschrijving
N	1	EVENTLYST.I	Naam
O	1	eventnummers als tekststring	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMEVENT	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 24.2 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
EVENTLYST.I	EVENTLYST.I="1010","1020","2001","2002",..., "<max_fabrikantspecifiek_event>"	Opvragen van de de lijst van beschikbare eventnummers.
EVENTLYST.I/#0	EVENTLYST.I/#0="1010"	Opvragen van het eerste eventnummer.

Tabel 24.3 Voorbeelden object EVENTLYST.I

24.2.2 EVENTLYST.INFO

Het object EVENTLYST.INFO geeft de eventnummers als tekststring weer:

Attribuut	Type	Object	Omschrijving
N	1	EVENTLYST.INFO	Naam
O	1	Detailinformatie over het event	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMEVENT	aantal data-elementen
I	1	EVENTLYST.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, ruwe tekst	Data-element formaat
S	0		Data-element stapgrootte

Tabel 24.4 Object attributen

Voorbeeld:
EVENTLYST.I

Element	Eventnummer
0	1010
1	1020
2	2001
3	2002
...	
NUMEVENT-1	<max_fabrikantspecifiek_event>

EVENTLYST.INFO

Element	Omschrijving
0	Lampfout
1	Detectiefout
2	VRI status wijziging
3	Programmaomschakeling
...	...
NUMEVENT-1	<Omschrijving max_fabrikantspecifiek_event>

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
EVENTLYST.INFO /#0	EVENTLYST.INFO/#0="Lampfout"	Opvragen aanvullende informatie betreffende een event
EVENTLYST.INFO /1010	EVENTLYST.INFO/1010="Lampfout"	

Tabel 24.5 Voorbeelden object EVENTLYST.INFO

24.2.3 VRIFOUT.I

Het object VRIFOUT.I geeft de index foutcode weer:

Attribuut	Type	Object	Omschrijving
N	1	VRIFOUT.I	Naam
O	1	index foutcode	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	2	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 24.6 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Het object VRIFOUT bevat twee foutcodes. De nummering van de foutcodes komt overeen met de nummering van de eventcodes in het object VRI.LB:

VRIFOUT.I		
0	FATAAL	Eerste fatale fout
1	MELDING	Laatste niet fatale fout

Tabel 24.7 Foutcode

FATAAL

Eerste fatale melding waardoor de VRI naar knippen of gedoofd is gegaan.

MELDING

Laatste niet fatale melding die is opgetreden in de VRI.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRIFOUT.I	VRIFOUT.I="FATAAL","MELDING"	Lees het object VRIFOUT.I
VRIFOUT.I/#0	VRIFOUT.I/#0="FATAAL"	
VRIFOUT.I/#1	VRIFOUT.I/#1="MELDING"	

Tabel 24.8 Voorbeelden object VRIFOUT.I

24.2.4 VRIFOUT

Het object VRIFOUT geeft de actuele foutcode weer:

Attribuut	Type	Object	Omschrijving
N	1	VRIFOUT	Naam
O	1	actuele foutcode	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	2	aantal data-elementen
I	1	VRIFOUT.I	Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0		Data-element stapgrootte

Tabel 24.9 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRIFOUT:A	VRIFOUT:A="N=VRIFOUT,T=0,E=2,U=44 44,I=VRIFOUT.I,F=1,O='Actuele foutcode'"	Opvragen attributen van het object VRIFOUT.

Tabel 24.10 Voorbeelden object VRIFOUT

24.2.5 VRIFSUB.I

Het object VRIFSUB.I geeft de index subsystemen weer:

Attribuut	Type	Object	Omschrijving
N	1	VRIFSUB.I	Naam
O	1	index subsystemen	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0	MAX_FLEN	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	2	Data-element formaat
S	0		Data-element stapgrootte

Tabel 24.11 Object attributen

Formaat: Indexnaam (2), type 1

Een index object bevat de functionele namen voor de elementen van andere objecten. Een indexnaam mag alleen letters, cijfers en underscores bevatten (zie de BNF definitie in Bijlage: BNF-notatie).

Het object VRIFSUB bevat een overzicht van de foutstatus van de diverse systeemdelen:

VRIFSUB.I		
0	DETECTIE	Status van het detectiesysteem.
1	LAMP	Status van lampen.
2	APPLICATIE	Status van het applicatieprogramma.
3	OV	Status van het selectieve detectiesysteem voor openbaarvervoer.
4	Reserve4	Gereserveerd.
5	Reserve5	Gereserveerd.
6	Reserve6	Gereserveerd.
7	Reserve7	Gereserveerd.
8	Reserve8	Gereserveerd.
9	Reserve9	Gereserveerd.

Tabel 24.12 Fouten in subsysteem

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRIFSUB.I	VRIFSUB.I="DETECTIE", "LAMP","APPLICATIE","OV", "Reserve4","Reserve5", "Reserve6","Reserve7", "Reserve8","Reserve9"	Opvragen van elementen in VRIFSUB.I object
VRIFSUB.I/#0	VRIFSUB.I/#0="DETECTIE"	
VRIFSUB.I/#1	VRIFSUB.I/#1="LAMP"	

Tabel 24.13 Voorbeelden object VRIFSUB.I

24.2.6 VRIFSUB

Het object VRIFSUB geeft de foutstatus van het subsysteem weer:

Attribuut	Type	Object	Omschrijving
N	1	VRIFSUB	Naam
O	1	Foutstatus van subsysteem	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	10	aantal data-elementen
I	1	VRIFSUB.I	Index verwijzing per dimensie
MIN	0	-1	Minimum data-elementwaarde
MAX	0	1	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	60	Data-element formaat
S	0		Data-element stapgrootte

Tabel 24.14 Object attributen

Formaat: Foutstatus (60), type 0

Een foutstatus geeft aan of zich in een bepaald deel van het regeltoestel een fout bevindt. Een fout kan worden gereset door het schrijven van een willekeurige waarde naar het bijbehorende element. Een fout wordt ook gereset als via het object VRI.C een resetcommando wordt gegeven voor alle storingsen.

Waarde	Omschrijving
-1	Foutstatus wordt door VRI niet ondersteund.
0	Geen fout.
1	Actuele of "latched" fout aanwezig.

Tabel 24.15 Foutstatus

N.B. De waarde -1 is gereserveerd voor het geval een VRI een bepaalde fout classificering niet wordt ondersteund.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRIFSUB/#0	VRIFSUB/#0 = <waarde>	Opvragen van een element in VRIFSUB object
VRIFSUB:A	VRIFSUB:A="N=VRIFSUB,T=0,E=10,U=444,I=VRIFSUB.I,MIN=-1,MAX=1,S=1,F=60,O='Fout status van subsystemen'"	Opvragen van VRIFSUB object attributen
VRIFSUB:O	VRIFSUB:O = "Fout status van subsystemen."	Opvragen van een attribuut in VRIFSUB object

Tabel 24.16 Voorbeelden object VRIFSUB

24.2.7 VRI.LB

Het object VRI.LB geeft het VRI-logboek weer:

Attribuut	Type	Object	Omschrijving
N	1	VRI.LB	Naam
O	1	VRI-logboek	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	0 .. 1000	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	100	Data-element formaat
S	0		Data-element stapgrootte

Tabel 24.17 Object attributen

Formaat: Event (100), type 1

Een eventobject bevat een event in de slave in een leesbare ASCII string. Een event bestaat uit een timestamp, een eventcode en een nadere detaillering van het event in tekst. De detailinformatie is afhankelijk van de eventcode.

```
EventBericht = TimeStamp + "," + Bevestigd + "," + EventCode + "," + DetailInfo
TimeStamp = "jjjjmmdd:uummss"
Bevestigd = 0 | 1
EventCode = IntegerWaarde
DetailInfo = Detailstring
```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
VRI.LB/#0	VRI.LB/#0="20140317:073800,0,1020,021,0,0,1,0,0,0,1,0"	Detectielus is defect. De detector meldt een hardwarestoring.

Tabel 24.18 Voorbeelden object VRI.LB

24.2.8 VRI.LA

Het object VRI.LA geeft het VRI-logboek (onbevestigd) weer:

Attribuut	Type	Object	Omschrijving
N	1	VRI.LA	Naam
O	1	VRI-logboek (onbevestigd)	Omschrijving
T	1	1	Type
U	0	6666	User Identificatie Control
E	0	0 .. 1000	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0		Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	100	Data-element formaat
S	0		Data-element stapgrootte

Tabel 24.19 Object attributen

Formaat: Event (100), type 1

Een eventobject bevat een event in de slave in een leesbare ASCII string. Een event bestaat uit een timestamp, een eventcode en een nadere detaillering van het event in tekst. De detailinformatie is afhankelijk van de eventcode.

```
EventBericht = TimeStamp + "," + Bevestigd + "," + EventCode + "," + DetailInfo
TimeStamp = "jjjjmmdd:uummss"
Bevestigd = 0 | 1
EventCode = IntegerWaarde
DetailInfo = Detailstring
```

Voorbeeld

Een voorbeeld van het lezen en bevestigen van events: In de slave zijn 5 events aanwezig in het event object "VRI.LA", te weten "melding 1" t/m "melding 5". De master kan alle aanwezige events lezen door het lezen van alle elementen van het object "VRI.LA". De slave antwoordt hierop met alle beschikbare events.

Master	Slave
VRI.LA	VRI.LA="melding 1","melding 2","melding 3","melding 4","melding 5"z

Tabel 24.20 Voorbeeld lezen van event object.

Nadat de master heeft gelezen, maar voor dat de master de events heeft bevestigd, ontstaat er een nieuw event; "melding 6". Het object VRI.LA bevat nu dus 6 elementen.

Onbevestigde events
"melding 1","melding 2","melding 3","melding 4","melding 5","melding 6"

Tabel 24.21 Data in object VRI.LA

De master kan de 5 gelezen events bevestigen door het schrijven van willekeurige data naar de gelezen elementen.

Master	Slave
VRI.LA/#0-#4=""	:A

Tabel 24.22 Voorbeeld bevestigen van events in een event object.

Na het bevestigen van de 5 gelezen events bevat het object nog 1 element.

Onbevestigde events

"melding 6"

Tabel 24.23 Data in object VRI.LA

Het nieuwe event, "melding 6" is niet verloren gegaan, doordat alleen de gelezen events worden bevestigd. Het commando bevestigen van alle events (VRI.LA/*="") wordt niet gebruikt, omdat dit tot gevolg heeft, dat ook alle nog niet gelezen events bevestigd worden.

Om de master in geval van veel events te waarschuwen is er een eventreeks 2510, 2511, 2512 en 2513 om te waarschuwen dat een logboek vol dreigt te raken.

NB. Indien een event-object vol is kunnen er events verloren gaan.

NB. Het is de taak van de master om regelmatig de informatie in het object te lezen.

25. Bijlage: BNF-notatie

In deze beschrijving van het IVERA-protocol wordt gebruik gemaakt van de Backus-Naur form (BNF) voor de beschrijving van het protocol. Hieronder volgt in het kort een uiteenzetting van deze notatie.

De beschrijving van het applicatieprotocol is opgebouwd uit zogenaamde BNF regels. Een BNF regel heeft het volgende formaat:

$$N = E$$

Waarbij N de naam is van een syntactische eenheid en E is een syntax expressie.

Een syntax expressie E heeft het formaat: $T_1 \mid T_2 \mid \dots \mid T_n$

T_1, T_2, \dots, T_n zijn syntax termen van E.

Een syntax term T heeft het formaat: $F_1 F_2 \dots F_n$

F_1, F_2, \dots, F_n zijn de syntax factoren van T. Een term definieert regels, waarbij een regel bestaat uit een regel met het formaat F_1 , gevolgd door een regel met het formaat F_2, \dots , gevolgd door een regel met het formaat F_n .

Een syntax factor F met het formaat: $[E]$

beschrijft een regel die leeg is of een regel met het formaat E (waarbij E een syntax expressie is).

Een syntax factor F met het formaat: $\{ E \}$

beschrijft een regel die bestaat uit nul of meer regels met het formaat E (waarbij E een syntax expressie is).

Een syntax factor F met het formaat

$$N$$

refereert naar een syntax regel genaamd N.

een syntax factor F met het formaat

$$"ab..z"$$

definieert de symbolen $ab \dots z$

NB. De beschrijving is overgenomen uit "Brinch Hanssen on Pascal compilers".