

IVERA-APP
Verkeersregelininstallaties
Versie 4.00
4 april 2018

Een uitgave van
Stichting Beheer IVERA protocol
Gorinchem, Nederland

Pub. No.: IVERA OD-VRI 4

Datum: 20 juli 2017

Titel: IVERA Objectdefinitie Verkeersregelininstallaties (versie 4)

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 693
4200 AR Gorinchem

© Copyright 2012-2017 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.....	8
2.	Inleiding	9
2.1	Afkortingen.....	10
3.	Systeem overzicht	11
3.1	IVERA verbindingen.....	11
3.1.1	Master to Slave (object management).....	11
3.1.2	Slave to Master (event- en logboek afhandeling).....	11
4.	IVERA-protocol.....	14
4.1	Inleiding.....	14
4.2	Master-Slave.....	14
4.3	OSI-model.....	16
4.4	Toepassingen, Applicaties en Automaten.....	16
4.5	IVERA objecten.....	17
4.6	Objectdefinitie	23
4.7	Objectsoorten.....	28
4.8	Gebruikers	30
4.9	Berichtdefinitie.....	31
4.10	Object Elementbereik.....	32
4.11	Data context diagrammen	34
4.11.1	Lezen van objecten.....	34
4.11.2	Schrijven van objecten	34
4.11.3	Schrijven van meerdere elementen.....	35
4.11.4	Lezen van objectattributen	35
4.11.5	Wijziging van objectattributen (optioneel).....	36
4.11.6	Master-slave synchronisatie.....	36
4.11.7	Gebeurtenis in de slave (trigger)	37
5.	Beheer van objecten.....	38
5.1	Homoniemen.....	38
5.2	Algemene objecten.....	39
5.3	Toepassings specifieke objecten	39
5.4	Applicatiespecifieke objecten	39
6.	Functionele omschrijving	40
6.1	Functionele modificatie van parameters (FMP)	40
6.2	Monitoring	41
6.3	Toevoegen extra elementen.....	41
6.4	Naamconventie objecten.....	41
7.	Dataformaten.....	44
7.1	Object type 0.....	44
7.2	Object type 1	45
8.	Management Interface TLC Faciliteiten	46
8.1	Object APPIFLOC.....	46
8.2	IVERA Gebruiker Beheer	48
8.2.1	Object USER	48
9.	Object definitie	50
9.1	Gebruikersgroepen.....	50
9.2	Overzicht alle objecten	50
10.	Datum en tijd	51

10.1	Algemeen datum en tijd.....	51
10.2	Objecten datum en tijd	52
10.2.1	TIJD	52
10.2.2	DATUM	53
11.	Identificatie	54
11.1	Algemeen identificatie	54
11.1.1	Object APPID.I.....	54
11.1.2	Object APPID.....	54
11.1.3	Object APPVER.I.....	55
11.1.4	Object APPVER.....	55
11.1.5	TID	56
11.1.6	Object TLC.I.....	56
11.1.7	Object TLC.....	56
11.1.8	Object RIS.I	57
11.1.9	Object RIS	57
11.1.10	YID.....	58
11.1.11	Applicatie specifieke events	58
12.	Toegang	60
12.1	Algemeen toegang	60
12.2	Objecten toegang.....	61
12.2.1	FTPUSER.I.....	61
12.2.2	FTPPASS	63
12.2.3	FTPLOCATION.....	64
12.2.4	DATACOM.I.....	64
12.2.5	DATACOM.....	67
12.2.6	LOGINNIVEAU	67
12.2.7	LOGIN.....	68
12.2.8	Datacommunicatie events	70
13.	VRI algemeen.....	71
13.1	Algemeen VRI algemeen.....	71
13.2	Objecten VRI algemeen (procesbesturing).....	72
13.2.1	CP.I.....	72
13.2.2	CP.....	73
13.2.3	CP.A	73
13.2.4	PP.I.....	73
13.2.5	PP	74
13.2.6	EXTRAINFO.I	74
13.2.7	EXTRAINFO.A.....	75
13.2.8	EXTRAINFOEXT	76
13.2.9	PAR.LB	76
13.2.10	PAR.LA	77
13.2.11	Programma-events	78
13.2.12	Resetevents.....	79
14.	Protocol	82
14.1	Algemeen protocol	82
14.2	Objecten protocol	83
14.2.1	ERROR.CODE	83
14.2.2	ERROR.INFO	83
14.2.3	ERROR.CMD.....	84
14.2.4	ABON.....	85
14.2.5	BB0	87
14.2.6	BB1	87
14.2.7	BBA0.....	88
14.2.8	BBA1.....	88
14.2.9	PING	89

15. Detectie	90
15.1 Algemeen detectie.....	90
15.2 Objecten detectie	90
15.2.1 D.I	90
15.2.2 TDB (alleen bij CCOL)	91
15.2.3 TDH (alleen bij CCOL)	91
15.2.4 TDH1 (alleen bij RWS-C)	92
15.2.5 TDH11 (alleen bij RWS-C)	92
15.2.6 TDH12 (alleen bij RWS-C)	92
15.2.7 TDH2 (alleen bij RWS-C)	93
15.2.8 TDH21 (alleen bij RWS-C)	93
15.2.9 TDH22 (alleen bij RWS-C)	93
16. Uitgangen	94
16.1 Algemeen uitgangen	94
16.2 Objecten uitgangen	94
16.2.1 U.I	94
16.2.2 U.A	95
16.2.3 T.U.A	95
16.2.4 U.LB	96
16.2.5 U.LA	96
17. Signaalgroepen.....	98
17.1 Algemeen signaalgroepen.....	98
17.2 Objecten signaalgroepen.....	100
17.2.1 SG.I.....	100
17.2.2 SGI.A	100
17.2.3 TSGI.A	101
17.2.4 SGI.LB	102
17.2.5 SGI.LA	102
17.2.6 TOR	103
17.2.7 TGOR	104
17.2.8 TGG	104
17.2.9 TGGL	104
17.2.10 TMGL.....	105
17.2.11 TGR	105
17.2.12 TVG.....	106
17.2.13 TVAG	106
17.2.14 TGL.....	106
17.2.15 TMG	107
17.2.16 TMG1	108
17.2.17 TMG2.....	108
17.2.18 TMG3.....	108
17.2.19 TMG4	108
17.2.20 TMG5.....	108
17.2.21 TMG6.....	108
17.2.22 LSGI.....	108
18. Regelapplicatie	110
18.1 Algemeen regelapplicatie	110
18.2 Objecten regelapplicatie	112
18.2.1 T.I.....	112
18.2.2 T.....	113
18.2.3 T.A	113
18.2.4 T.T.....	114
18.2.5 C.I (alleen bij CCOL).....	115
18.2.6 C (alleen bij CCOL).....	116
18.2.7 C.A (alleen bij CCOL)	116
18.2.8 C.T (alleen bij CCOL)	117

18.2.9	P.I.....	118
18.2.10	P.....	119
18.2.11	P.T	120
18.2.12	EGGP.I (alleen bij RWS-C)	121
18.2.13	EGGP (alleen bij RWS-C)	121
18.2.14	EGGP.T (alleen bij RWS-C)	122
18.2.15	S.I.....	123
18.2.16	S.....	124
18.2.17	S.T	125
18.2.18	KL.I (alleen bij RWS-C).....	126
18.2.19	KLB (alleen RWS-C).....	127
18.2.20	KLE (alleen RWS-C).....	128
18.2.21	CIFGUS	129
18.2.22	CIFWUS.....	129
18.2.23	CIFIS.....	130
18.2.24	CIFWPS.....	130
18.2.25	CIFGPS	131
18.2.26	CIFKLOK	132
18.2.27	CIFPARM1.....	133
18.2.28	CIFPARM2.....	133
18.2.29	PL.I.....	134
18.2.30	PLTXMAX	135
18.2.31	PLTPLON	135
18.2.32	PLTPLOFF.....	136
18.2.33	PLTXA	136
18.2.34	PLTXB	137
18.2.35	PLTXC	137
18.2.36	PLTXD	137
18.2.37	PLTXE	137
18.2.38	BL.A (alleen verplicht bij RWS-C, optioneel bij Ccol)	138
18.2.39	Commando's en events.....	139
19.	Events en Alarms.....	140
19.1	Algemeen Events en Alarms	140
19.2	Objecten Events en Alarms	141
19.2.1	EVENTLYST.I.....	141
19.2.2	EVENTLYST.INFO	142
19.2.3	Object APPFOUT.I	142
19.2.4	Object APPFOUT	143
19.2.5	Object APP.LA.....	143
19.2.6	Object APP.LB.....	143
19.2.7	Object APP.A.....	144
20.	Events	146
20.1	Categorien	146
20.1.1	I/O events	146
20.1.2	Program events	147
20.1.3	Supervisor events	147
20.1.4	Reset events.....	148
20.1.5	Commando events.....	148
20.1.6	Data communicatie events	149
21.	Bijlage: BNF-notatie.....	150

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 tot 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 9 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

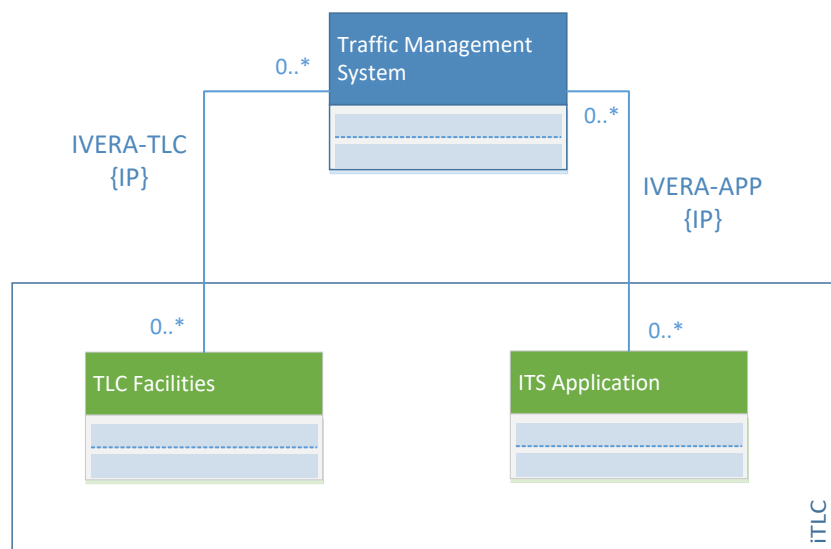
Voor beide IVERA versies is een aparte specificatie beschikbaar.

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.

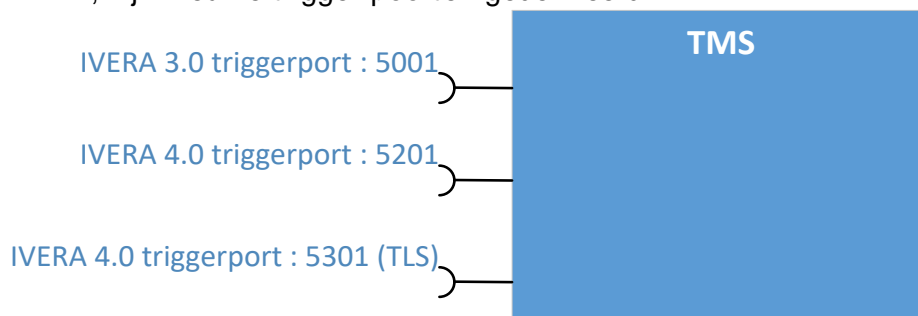
3.1.2 Slave to Master (event- en logboek afhandeling)

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 voor 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 }
```

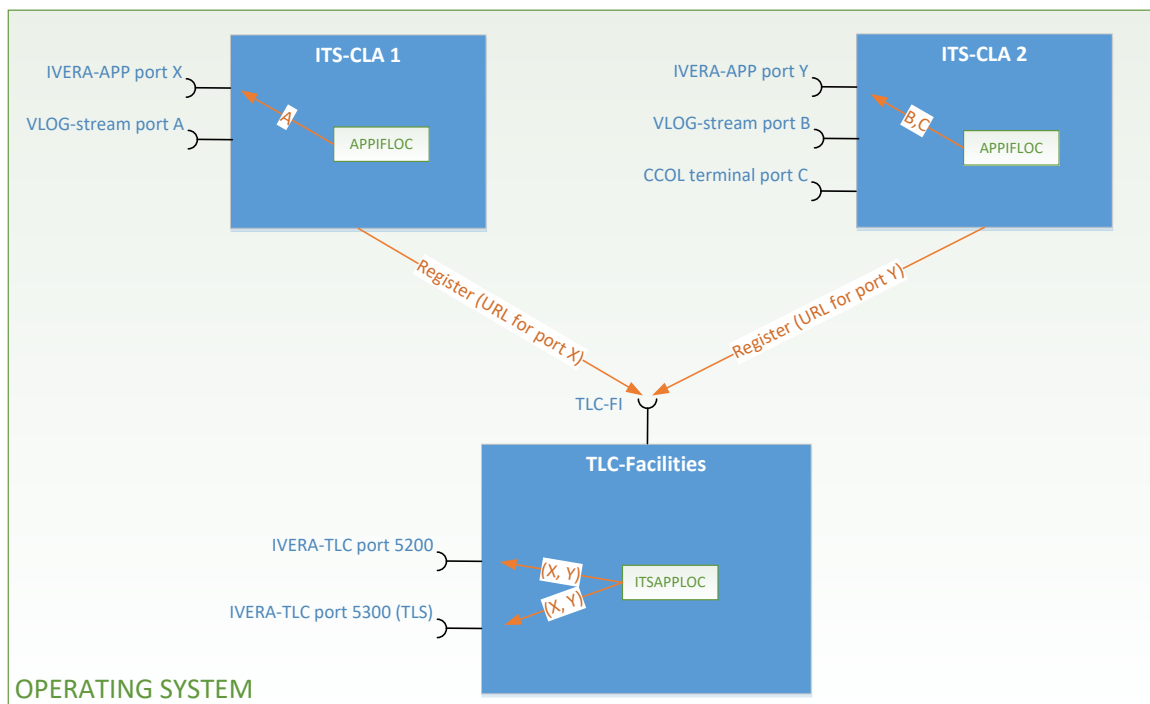
Voorbeeld:

```
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)

De onderstaande figuur beschrijft de wijze waarop een IVERA master kan de verbindingen kan opzetten voor een ITS Control Application (ITC-CLA):



Doordat IVERA-TLC beschikbaar is op een bekend adres/poort, kan dit door een IVERA master gebruikt worden om de gegevens van het IVERA object ITCAPPLOC op te vragen.

Dit object bevat per ITS-CLA 1 URL, verwijzend naar de management interface van de ITS-CLA.

Deze URL wordt uitgewisseld tijdens de registratie van de ITS-CLA bij de TLC-Facilities.

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

Als de management interface IVERA-APP is, dan kan de master het IVERA object APPIFLOC opvragen. Dit object bevat een lijst met URL's verwijzend naar alle andere interfaces van de betreffende ITS-CLA.

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
TGL	Een object dat de geeltijden van alle signaalgroepen bevat.
TOR	Een object dat alle ontruimingstijden bevat.
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 ITSAPLOC 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 applicaties bestaan zoals bij een verkeersregelininstallatie o.a. C-COL en RWS-C. Binnen een applicatie staat het vrij zelf objecten te definiëren. De naam van een applicatiespecifiek object begint altijd met de letter 'Y'.

Een applicatiespecifiek object is uniek gedefinieerd met: <TID, YID, ObjectNaam> waarbij: YID = applicatie-identificatie.

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 t.		X	
VRIPROGLYSTEXT	Uitgebreide programmalijs t		X	
BEDRIJF	Bedrijfstoestand m.b.t. de centrale.		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
BEDRIJF.I	Index object BEDRIJF		X	
FTPUSER.I	FTP-gebruikersnamen	X	X	
FTPPASS	FTP-passwords	X	X	
FTPLOCATION	FTP-locatie	X	X	
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 ingangstatus	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		

Name	Description ¹	IVERA-APP	IVERA-TLC	Mandatory
TMGL1	Maximum geeltijd (proces)		X	
TGR	Garantieroodtijd (appl)	X		
TGR1	Garantieroodtijd (proces)		X	
TVG	Vastgroentijd	X		
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	

Name	Description ¹	IVERA-APP	IVERA-TLC	Mandatory
TP.I	Index timers (proces)		X	
CP	Counter instelling (proces)	X		
CP.A	Lopende counter (proces)	X		
CP.I	Index counters (proces)	X		
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	Signaalgroepen 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 detectorschakelaar		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 inputschakelaar		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		

Name	Description ¹	IVERA-APP	IVERA-TLC	Mandatory
PLTPLON	inschakeltijd signaalplan (* TPL_on)	X		
PLTPLOFF	uitschakeltijd signaalplan (* TPL_off)	X		
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	
<i>DATACOM</i>	<i>Instelling datacommunicatie</i>	X	X	
DATACOM.I	Index object Datacom	X	X	
AUTHOG	Gebruikersnamen		X	
AUTHOP	Toegangscode		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	

Name	Description ¹	IVERA-APP	IVERA-TLC	Mandatory
ERROR.CMD	Het commando waar de error bij hoort	P	P	
NOODSTROOM	Instellingen voor de noodstroom voorziening		X	
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 21.

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 6.2 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
TGL	Geeltijden
TOR	Ontruimingstijden
P	Parameters
LSGE	Externe signaalgroep status (tekststring)
SGI.A	Interne 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. "SG.*" levert alle aan signaalgroepen 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.

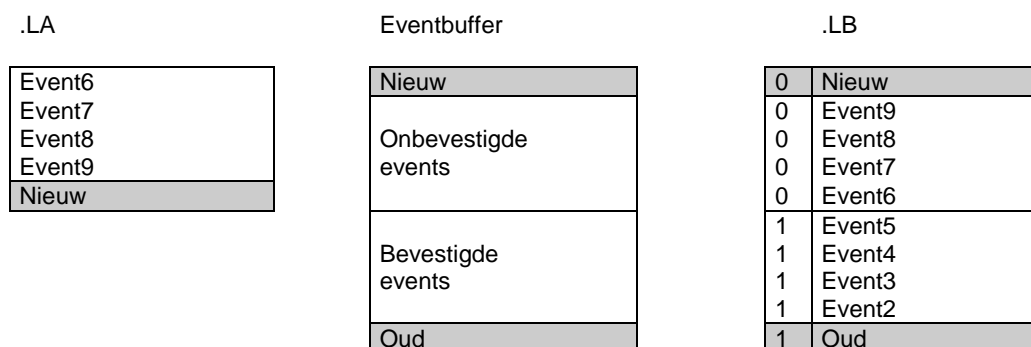
Een voorbeeld is het object “VRI.C”. Door het schrijven van een waarde naar dit object kunnen specifieke alarmen/events in een slave worden gereset.

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

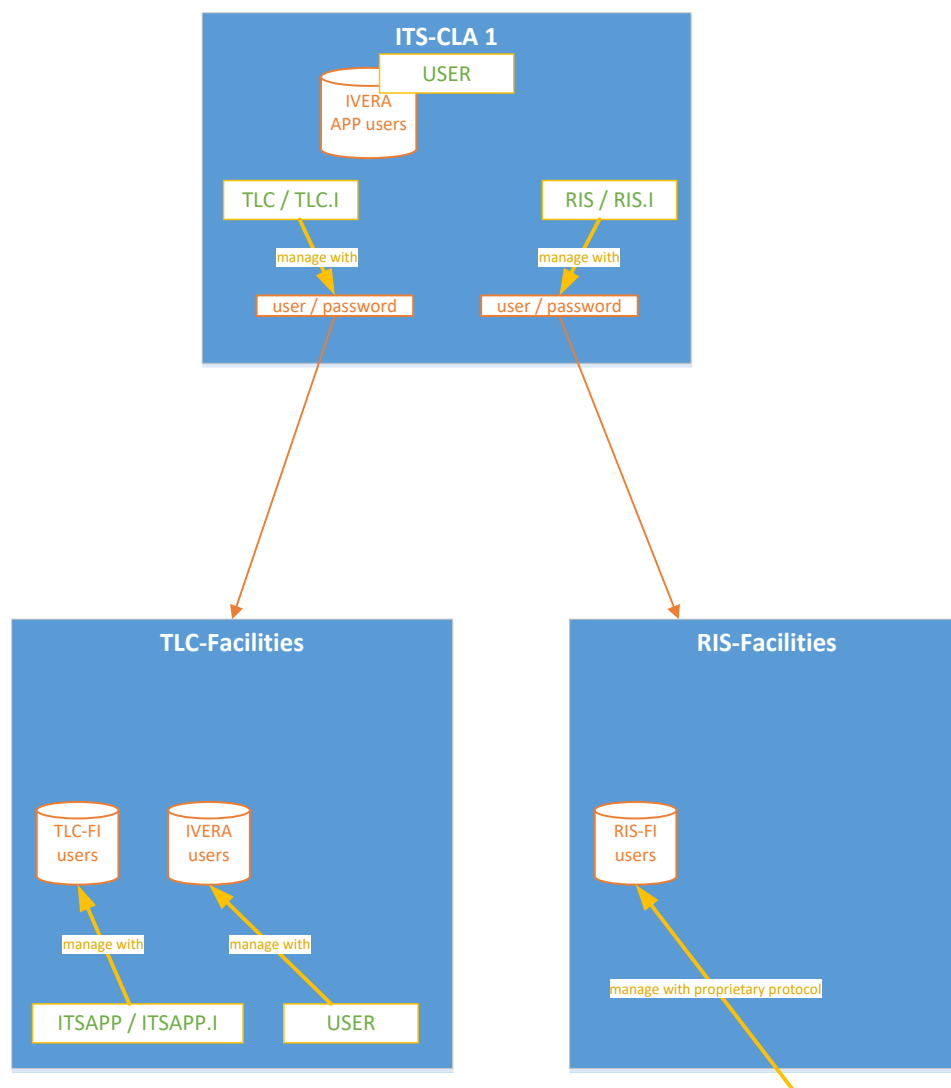
De UIC van het object DATACOM zal hiervoor worden gewijzigd van “6444” naar “6644”

In **Error! Reference source not found.**, 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:

```
ElementBereik dimensie 1, ElementBereik dimensie 2, .. , ElementBereik dimensie 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
TGL	TGL=3,3,3,3	Geeltijd van alle signaalgroepen
TGL/SG01-SG03	TGL/SG01-SG03=3,3,3	Geeltijd van SG01,SG02,SG03
TOR/SG01	TOR/SG01=-1,2,3,4	Ontruimingstijden van SG01
TOR/SG01,SG02	TOR/SG01,SG02=2	Ontruimingstijd SG01 -> SG02
@1#TGL	@1#=3,3,3,3	Geeltijd van alle signaalgroepen
@2#TGL/SG01-SG03	@2#=3,3,3	Geeltijd van SG01,SG02,SG03
@3#TOR/SG01	@3#=-1,2,3,4	Ontruimingstijden van SG01
@4#TOR/SG01,SG02	@4#=2	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
TGL/#0=3 TGL/SG02=9 TGL/SG02=4 TOR/SG01,SG02=2 PING/#0=5	TGL/#0=3 :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#TGL/#0=3 @2#TGL/SG02=9 @3#TGL/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#TGL=3 @2#TGL/*=3 @3#TGL/SG01-SG02=3 @4#TGL/SG01-SG03=3 @5#TGL/SG01-SG02=3,4 @6#TGL/SG01-SG03=3,4	@1#:E=14 @2#:A @3#:A @4#:E=16 @5#:A @6#:E=15	Bereik is niet volledig gespecificeerd Alle geeltijden naar 3 s. TGL/SG01=3 en TGL/SG02=3 Data niet geldig omdat gar. geeltijd SG03=4 s. TGL/SG01=3, TGL/SG02=4 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
TGL:N	TGL:N="TGL"	Objectnaam
TGL:T	TGL:T=0	Objecttype
TGL:E	TGL:E=4	4 signaalgroepen
TGL:U	TGL:U=6644	User Identificatie Control
TGL:L	TGL:L=1	Logboek aan
TGL:I	TGL:I="SG.I"	Index object(s)
TGL:W	TGL:W=1	Wijzigingsteller
TGL:MIN	TGL:MIN=2	Minimumwaarde
TGL:MAX	TGL:MAX=10	Maximumwaarde
TGL:IMIN	TGL:IMIN="TGGL"	Index minimum
TGL:IMAX	TGL:IMAX=""	Index maximum
TGL:S	TGL:S=1	Stapgrootte
TGL:A	TGL: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#TGL:L=1	@1#:A	Logboek aan
@2#TGL:A="L=1,IMIN=TGGL"	@2#:A	
SG.I:S=1	:E=19	Stapgrootte wordt niet ondersteund door tekstobjecten.

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 automaat 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)
 - Automaatspecifieke 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 toepassingspecifiek. 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 Applicatiespecifieke objecten

Voor de applicatiespecifieke objecten is de beginletter 'Y' gereserveerd. De namen van de applicatiespecifieke objecten worden door de leverancier van de applicatie bepaald.

De functie van een applicatiespecifiek object is eenduidig vastgelegd in combinatie met een unieke applicatie-identificatie (YID).

6. Functionele omschrijving

6.1 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
P.I	Vrije parameter namen
C.I	Teller namen
S.I	Schakelaar namen
T.I	Timer 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'.

De 12^e lege parameter naam wordt ingevuld met 'P_11'.

De 46^e lege tellernaam wordt ingevuld met 'C_45'.

De 43^e lege schakelaarnaam wordt ingevuld met 'S_42'.

De 4^e lege timer naam wordt ingevuld met 'T_3'.

6.2 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, LSGI 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.

6.3 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.

6.4 Naamconventie objecten

Voor de definitie en de naamgeving van objecten voor een verkeersregelinstallatie 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	KLB	Klok parameter 1	
L*	Lijndump	LSGI	Lijndump SG-toestand intern	
T*	Tijdstelling	TGL	Geeltijd instelling	
TD*	Tijdstelling detectie	TDB	Bezettijd voor aanvraag	
Y*	Applicatie specifiek	YID	Applicatie identificatie nummer	
BL	Blok of module	BL.A	Actueel bloknummer	BL en .A
C	Counter	C	Applicatie counter instelling	
P	Parameter	P	Applicatie parameter instelling	
S	Schakelaar	S	Applicatie schakelaar instelling	
SG	Signaalgroep	SGI.A	Actuele SG toestand intern	SGI en .A
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

Tabel 6.1 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
NUMKL	Aantal klokparameters (applicatie)
NUMS	Aantal schakelaars (applicatie)
NUMP	Aantal parameters (applicatie)
NUMT	Aantal timers (applicatie)
NUMC	Aantal counters (applicatie)
NUMU	Aantal uitgangen (proces + applicatie)
NUMI	Aantal ingangen (proces + applicatie)
NUMPL	Aantal signaalplannen (applicatie)
NUMEVENT	Aantal events
NUMINFO	Aantal automaat info regels
NUMCIFGUS	Aantal elementen in de CVN C-interface GUS buffer.
NUMCIFIS	Aantal elementen in de CVN C-interface IS buffer.
NUMCIFWPS	Aantal elementen in de CVN C-interface WPS buffer.
NUMCIFGPS	Aantal elementen in de CVN C-interface GPS buffer.
NUMCIFPARAM1	Aantal elementen in de CVN C-interface PARM1 buffer.
NUMCIFPARAM2	Aantal elementen in de CVN C-interface PARM2 buffer.
MAX_FLEN	Maximale lengte van functionele elementnamen.

Tabel 6.2 Constanten

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

7. Dataformaten

7.1 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
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
31	Signaalgroep toestand intern		
100	Element type		

Tabel 7.1 Data formaten object type 0
NB. Jaar is 4 cijfers.

Signaalgroep toestand intern (31)

De interne signaalgroep toestand is applicatiespecifiek, d.w.z. iedere applicatie kent zijn eigen interne signaalgroep toestanden. De volgende tabel geeft een overzicht van een groot aantal denkbare toestanden.

Waarde	Karakter	Toestand	C-COL	RWS-C
0	0	Recht op groen		ROG
1	1	Rood voor groen	RA	RVG
2	2	Garantiegroen		
3	3	Voorstartgroen	VS	
4	4	Vastgroen	FG	VG
5	5	Eerste VAG		1e VAG
6	6	Wachtgroen	WG	WG
7	7	Tweede VAG		2e VAG
8	8	Verlenggroen	VG	
9	9	Meeverlenggroen	MG	MVG
10	A	Veiligheidsgroen		
11	B	Garantiegeel		
12	C	Geel	FGL	VGL
13	D	Verlenggeel	VGL	VAGL
14	E	Garantierood		
15	F	Wachtrood		WR
16	G	Rood voor aanvraag	RV	
17	H	Gedoofd		
18	I	Gereserveerd		

Tabel 7.2 Signaalgroep toestand extern

7.2 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
31	Toestand van alle interne signaalgroep toestanden (algemeen).
40	Toestand van alle detectoren.
50	Toestand van overige ingangen of uitgangen.
100	Event
101	Parameterevent

Tabel 7.3 Data formaten objecttype

8. 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.

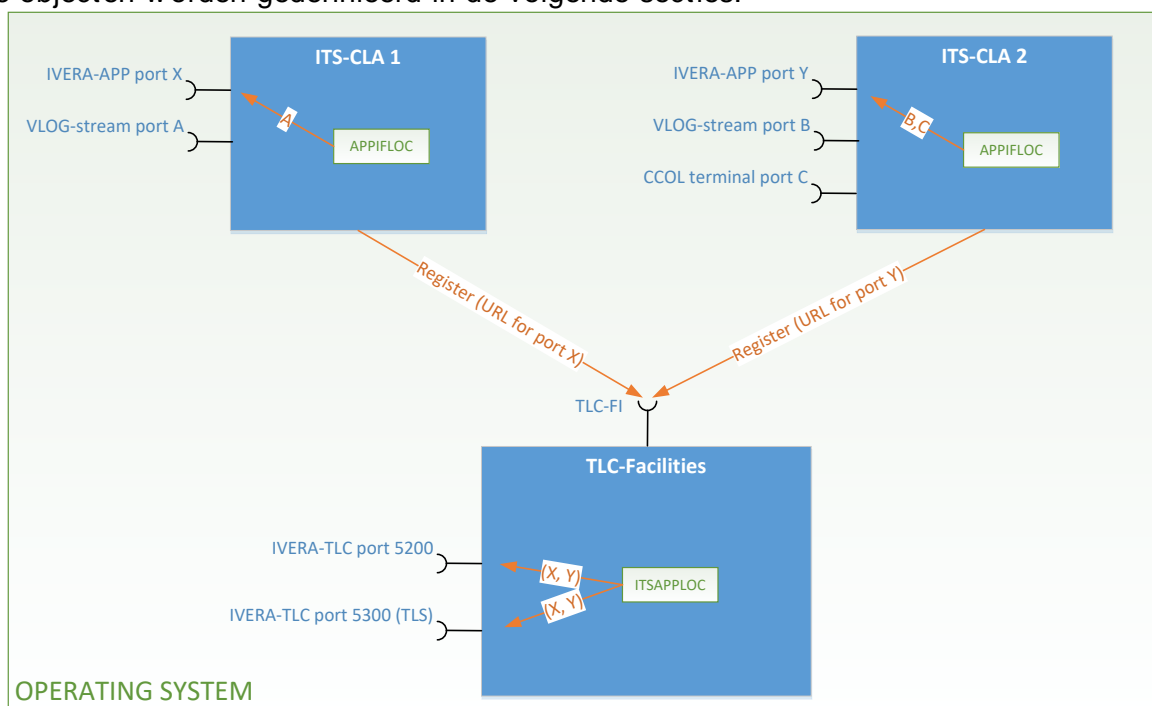
Aangezien de IVERA-TLC van de TLC-Faciliteiten beschikbaar is op een bekend address/port kan dit door een IVERA master gebruikt worden om de inhoud op te vragen van het IVERA Object 'ITSAPPLOC'.

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

8.1 Object APPIFLOC

Het object wordt gebruikt om van de ITS applicatie de beschikbare protocolen die een URI gebruiken op te vragen. Zie ook afbeelding: Afbeelding 4 resolving connection points of ITS-A

Attribuut	Type	Object	Omschrijving
N	1	APPIFLOC	Naam
O	1	Applicatie interface adressen (uri)	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
L	0		Logboek
W	0		Wijzigingsteller
E	0	NUMAPPIFLOC	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	401	Data-element formaat
S	0		Data-element stapgrootte
A	1		Overzicht alle attributen

Tabel 4 Object attributen APPIFLOC

Het Resourceidentificer format (401) dat voor dit object wordt gebruikt is gedefinieerd in 0.

Voorbeelden:

Een element lezen:

APPIFLOC/#1

APPIFLOC/#1= "ivera-app://10.10.39.40:5100,management interface"

Meerdere elementen lezen:

APPIFLOC

APPIFLOC= "ivera-app://10.10.39.40:5100,management interface","http://10.10.40.10,web interface

ccol://10.10.41.19:7000,command

parser","vlog://10.10.41.19:7001,verkeerslogging","","",""

Een leeg element lezen:

APPIFLOC/#3

APPIFLOC/#3= ""

8.2 IVERA Gebruiker Beheer

8.2.1 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 5 Object attributen USER

Format: IveraGebruiker (404), type 1

IveraGebruiker =
Gebruikersnaam + “,” + Gebruikersgroep [+ “,” + Wachtwoord + “,” + Nieuw1 + “,” + Nieuw2]

Gebruikersnaam = AsciiString
Gebruikersgroep = “1” | “2” | “3” | “4”
Wachtwoord = AsciiString
Nieuw1 = AsciiString
Nieuw2 = AsciiString

Gebruik: een element schrijven

USER/< element> = ”< gebruikersnaam> ,< gebruikersgroep> [,< wachtwoord> ,< nieuw1> ,< nieuw2>]”

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
< nieuw1> : nieuw wachtwoord voor deze gebruiker
< nieuw2> : nieuw wachtwoord voor deze gebruiker (moet overeenkomen met nieuw1)

Voorbeelden:

Een element schrijven:
USER/#2= "admin2,4,secret1,new Secret,new Secret"

Een element lezen:
USER/#2
USER/#2= "admin2,4"

9. Object definitie

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

9.1 Gebruikersgroepen

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

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

9.2 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 9.1 Object attributen

De tabel op de volgende bladzijden bevatten een overzicht van alle objecten, met uitzondering van de automaat (X..) en applicatie (Y..) specifieke objecten.

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.

10. Datum en tijd

10.1 Algemeen datum en tijd

Voor de datum en tijd zijn de volgende objecten gedefinieerd:

Object	Omschrijving
TIJD	Actuele systeemtijd
DATUM	Actuele systeemdatum

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.

Het instellen van de tijd vindt plaats met de objecten DATUM en TIJD.

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

10.2 Objecten datum en tijd

10.2.1 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 10.1 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 10.2 Voorbeelden TIJD

10.2.2 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 10.3 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 10.4 Voorbeelden DATUM

11. Identificatie

11.1 Algemeen identificatie

Voor de identificatie zijn de volgende objecten gedefinieerd:

Object	Omschrijving
TID	Toepassing identificatienummer
APPID.I	Index Identificatie van Applicatie
APPID	Identificatie van Applicatie
APPVER.I	Index versienummers
APPVER	Versies van de applicatie
TLC.I	TLC namen
TLC	TLC instellingen
RIS.I	RIS namen
RIS	RIS instellingen
YID	Applicatie identificatienummer

Tabel 3.14. Objecten identificatie

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

Versienummers worden toegepast om de applicatie te herkennen.

11.1.1 Object APPID.I

Het object APPID.I geeft de index namen van de applicatie identificatie:

Attribuut	Type	Object	Omschrijving
N	1	APPID.I	Naam
O	1	Index applicatie identificatie	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMAPPID	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 5 Object attributen

Het object APPID.I bevat een aantal elementen om de applicatie te beschrijven:

APPID.I		APP identificatie
0	LEVERANCIER	Partij die applicatie levert
1	ONTWERPER	Gegeven over ontwerper en of programmeur
2	KRP_NR	Kruispuntnummer
3	KRP_NAAM	Kruispuntnaam
4	PAK_TYPE	Naam van basispakket
5	APP_NAAM	Applicatiennaam
6	OMSCHRIJVING	Omschrijving van de applicatie
7	LEV7	Vrij in te vullen
8	LEV8	Vrij in te vullen
9	LEV9	Vrij in te vullen

Tabel 6 applicatie identificatie

11.1.2 Object APPID

Het object APPID geeft de applicatie identificatie:

Attribuut	Type	Object	Omschrijving
N	1	APPID	Naam
O	1	Applicatie identificatie	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMAPPID	aantal data-elementen
I	1	APPID.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 7 Object attributen

Voorbeeld:

Alle elementen lezen:

APPID

APPID= "LeverancierX","Klaas","KRP55","

Dorpstraat/Kerkstraat","CCOL","Applicatie","","","","",""

11.1.3 Object APPVER.I

Het object APPVER.I geeft de index namen van de applicatie versies:

Attribuut	Type	Object	Omschrijving
N	1	APPVER.I	Naam
O	1	Index versienummers	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMAPPVER	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 8 Object attributen

APPVER.I	Omschrijving
0	VERSIE
1	DATUM
2	LEV1
3	LEV2
4	LEV3

Tabel 9 Versie nummers

11.1.4 Object APPVER

Het object APPVER geeft de applicatie versies:

Attribuut	Type	Object	Omschrijving
N	1	APPVER	Naam
O	1	Versienummers	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMAPPVER	aantal data-elementen
I	1	APPVER.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 10 Object attributen

Voorbeeld:

Alle elementen lezen:

APPVER

APPVER= "Applicatie V1.0.0", "2017-04-27", "", "", ""

11.1.5 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 11.11 Object attributen

Voor dit toepassingsgebied is het bereik 1..9999 gereserveerd.

De TID bevat het versienummer van dit document "IVERA Objectdefinitie Verkeersregelinstallaties". Bijv. 301 is versie 3.01.

11.1.6 Object TLC.I

IVERA Object TLC.I geeft de namen van de kruispunten welke worden ondersteund door deze control applicatie.

De naam is geconfigureerd in de ITS applicatie en kan niet worden gewijzigd via het IVERA protocol. De naam van het kruispunt moet overeen komen met de kruispuntnaam welke is geconfigureerd in de TLC.

Attribuut	Type	Object	Omschrijving
N	1	TLC.I	Naam
O	1	Kruispuntnamen	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMTLC	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 Object attributen TLC.I

11.1.7 Object TLC

IVERA Object TLC kan worden gebruikt om de instellingen te beheren voor de verbinding naar de TLC Faciliteiten met de TLC-FI interface.

Attribuut	Type	Object	Omschrijving
N	1	TLC	Naam
O	1	TLC management	Omschrijving
T	1	1	Type
U	0	6444	User Identificatie Control
E	0	NUMTLC	aantal data-elementen
I	1	TLC.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	401	Data-element formaat
S	0		Data-element stapgrootte

Tabel 13 Object attributen TLC

Het bronidentificatie formaat (401) dat voor dit object wordt gebruikt is gedefinieerd in 0.

Let op dat het gebruikersnaam:wachtwoord gedeelte van de URI autorisatie component wordt gebruikt in dit object. Het wachtwoord wordt alleen gebruikt bij het schrijven van de URI. Het lezen van het object RIS zal een URI geven waarin het wachtwoord niet is opgenomen.

Voorbeelden:

- applicationX is de gebruikersnaam
- secret is het wachtwoord
- 10.10.39.40 is de hostnaam
- 11001 is het port nummer

Een element schrijven:

```
TLC/#1= "tlc-fis://applicationX:secret@10.10.39.40:11001"
```

Een element lezen:

```
TLC/#1
```

```
TLC/#1= "tlc-fis://applicationX@10.10.39.40:11001"
```

11.1.8 Object RIS.I

IVERA Object RIS.I geeft de functionele naam van de RIS.

Attribuut	Type	Object	Omschrijving
N	1	RIS.I	Naam
O	1	RIS namen	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMRIS	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 Object attributen RIS.I

11.1.9 Object RIS

IVERA Object RIS kan worden gebruikt om de instellingen voor de verbinding naar de RIS Facilites te beheren via de RIS-FI interface..

Attribuut	Type	Object	Omschrijving
N	1	RIS	Naam
O	1	RIS beheer	Omschrijving
T	1	1	Type
U	0	6444	User Identificatie Control
E	0	NUMRIS	aantal data-elementen
I	1	RIS.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	401	Data-element formaat
S	0		Data-element stapgrootte

Tabel 15 Object attributen RIS

Het bronidentificatie formaat (401) dat voor dit object wordt gebruikt is gedefinieerd in 0.

Let op dat het gebruikersnaam:wachtwoord gedeelte van de URI autorisatie component wordt gebruikt in dit object. Het wachtwoord wordt alleen gebruikt bij het schrijven van de URI. Het lezen van het object RIS zal een URI geven waarin het wachtwoord niet is opgenomen.

Voorbeelden:

Een element schrijven:

RIS/#1= "ris-fis://applicationX:secret@10.10.39.40:12001"

Een element lezen:

RIS /#1

RIS /#1= "tlc-fis://applicationX@10.10.39.40:12001"

11.1.10 YID

Het object YID geeft het automatisch identificatienummer weer:

Attribuut	Type	Object	Omschrijving
N	1	YID	Naam
O	1	applicatie 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 11.16 Object attributen

Binnen dit toepassingsgebied is het YID als volgt gedefinieerd:

Bereik	Gereserveerd voor
100.000 .. 199.999	C-COL
200.000 .. 299.999	RWS-C
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 11.17 YID

11.1.11 Applicatie specifieke events

Afhankelijk van het type applicatie is het mogelijk dat er events zijn die niet in een van de gedefinieerde categorieën vallen. De betekenis van applicatiespecifieke events

(200.000..299.999) is eenduidig vastgelegd in combinatie met het unieke applicatie-identificatienummer (YID).

12. Toegang

12.1 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
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 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.

12.2 Objecten toegang

12.2.1 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 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).

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	Lezen van de analyse informatie (MV

		opgeslagen.	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	<i>FAB1</i>		
7	<i>FAB2</i>		
8	<i>FAB3</i>		
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 12.2 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", "dumpinfo", "FAB1", "FAB2", "FAB3", "anonymous"	De gebruikersnamen FAB1, FAB2, FAB3 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 12.3 Voorbeelden FTPUSER.I

12.2.2 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 12.4 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 FTTPASS 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","Fdumpinfo","FFAB1","FFAB2","FFAB3","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 12.5 Voorbeelden FTPPASS

12.2.3 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 stapgrootte

Tabel 12.6 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","RESE RVE8","RESERVE9","/D2/We bServer"	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 directoy wordt gekozen

Tabel 12.7 Voorbeelden FTPLOCATION

12.2.4 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 12.8 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 12.9 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_RESPONS","RETRYTIJD","RETRYMAXIMUM","RESERVE14","RESERVE15"	Vraag de index van DATACOM op
DATACOM.I/#14	DATACOM.I/#14="TO_IVERA	Vraag het object DATACOM.I op

	_sessie"	
DATAKOM/TO_IV ERA_sessie=60	DATAKOM/TO_IVERA_sessi e=60	Stel de timeout in in het object DATAKOM in seconden

Tabel 12.10 Voorbeelden DATAKOM.I

12.2.5 DATAKOM

Het object DATAKOM geeft de instelling datacommunicatie weer:

Attribuut	Type	Object	Omschrijving
N	1	DATAKOM	Naam
O	1	Instelling datacommunicatie	Omschrijving
T	1	1	Type
U	0	6644	User Identificatie Control
E	0	16	aantal data-elementen
I	1	DATAKOM.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 12.11 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
DATAKOM/#0="12 34"	DATAKOM/#0="1234"	Wijzig de parameter 'telefoonnummer centrale'
DATAKOM/#1="19 2.168.42.1"	DATAKOM/#1="192.168.42.1 "	Wijzig de parameter 'IP_adres_centrale'
DATAKOM/#1	DATAKOM/#1="192.168.42.1 "	Verificatie dat waarde wordt overgenomen
DATAKOM/#0="+3 11234567890123"	:E=16	Wijzig de waarde van het eerste element uit object DATAKOM met een waarde > 16 digits, en teken(s)

Tabel 12.12 Voorbeelden DATAKOM

12.2.6 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 12.13 Object attributen

De volgende tabel bevat enkele voorbeelden:

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

Tabel 12.14 Voorbeelden LOGINNIVEAU

12.2.7 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 12.15 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.

12.2.8 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 12.16 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 storingslijst'. Indien dit event inactief wordt gemaakt door een ander event, is dit expliciet aangegeven met 'eventcode'.

13. VRI algemeen

13.1 Algemeen VRI algemeen

Voor de VRI algemeen zijn de volgende objecten gedefinieerd:

Object	Omschrijving
CP.I	Index counters (proces)
CP	Counter instelling (proces)
CP.A	Lopende counter (proces)
PP.I	Index parameters (proces)
PP	Parameter instelling (proces)
EXTRAINFO.I	Index extra info
EXTRAINFO.A	Informatieve actuele extra info string.
EXTRAINFOEXT	Informatieve toelichting extra info string.
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 CP.I, CP en CP.A

De counters in de procesbesturing bestaan uit: CP(counter instellingen), CP.I (Index) en CP.A (lopende counter). Naast de in de index-tabel gedefinieerde elementen kunnen er per installatie specifieke elementen worden toegevoegd.

Objecten PP.I en PP

De parameter instellingen in de procesbesturing bestaan uit: PP (parameter instelling) en PP.I (Index). Naast de in de 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.

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 13.2.9).

Programma events

Vanuit het regelprogramma worden foommeldingen 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 seriële koppeling;
- Onder- of bovenspanningsmelding.

13.2 Objecten VRI algemeen (procesbesturing)

13.2.1 CP.I

Het object CP.I geeft de index voor alle counters weer:

Attribuut	Type	Object	Omschrijving
N	1	CP.I	Naam
O	1	Index counters	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMCP	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).

CP.I	
APPLFOUT	Maximum aantal applicatiefouten
GUSWUS	Maximum aantal GUS-WUS fouten
FB	Maximum aantal rood fasebewakingsfouten
APPLTIJD	Maximum aantal executietijdoverschrijdingen door de applicatie.

Tabel 13.2 Counters procesbesturing

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
CP.I	CP.I="APPLFOUT", "GUSWU S", "FB", "APPLTIJD"	Vraag object CP.I op

CP.I/#0	CP.I/#0="APPLFOUT"	Vraag element uit object CP.I weer op ter controle
CP.I/#4	:E=12	Vraag een ongeldig element bereik op

Tabel 13.3 Voorbeelden CP.I

13.2.2 CP

Het object CP geeft de counter instelling voor alle counters weer:

Attribuut	Type	Object	Omschrijving
N	1	CP	Naam
O	1	counter instelling	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMCP	aantal data-elementen
I	1	CP.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 13.4 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
CP/#0	CP/#0=<waarde>	Vraag een element op uit object CP
CP/#0=6	:E=11	Wijzig de waarde van dit element (onjuist inlogniveau)
CP/#0=15	CP/#0=15	Wijzig de waarde van dit element

Tabel 13.5 Voorbeelden CP

13.2.3 CP.A

Het object CP.A geeft de lopende counter voor alle counters weer:

Attribuut	Type	Object	Omschrijving
N	1	CP.A	Naam
O	1	Lopende counter	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMCP	aantal data-elementen
I	1	CP.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 13.6 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
CP.A/#0	CP.A/#0=<waarde>	Opvragen van een element in CP.A object
CP.A:A	CP.A:A="N=CP.A,T=0,E=<N UMCP>,U=4444,I=CP.I,MIN= 0,S=1,F=1,O='Lopende counter (proces)'"	Opvragen van CP.A object attributen
CP.A:MIN	CP.A:MIN=0	Opvragen van een attribuut in CP.A object

Tabel 13.7 Voorbeelden CP.A

13.2.4 PP.I

Het object PP.I geeft de index voor alle parameters weer:

Attribuut	Type	Object	Omschrijving
N	1	PP.I	Naam
O	1	Index parameters	Omschrijving
T	1	1	Type

U	0	4444	User Identificatie Control
E	0	NUMPP	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).

PP.I	
GUSWUS	Maximum verschil tussen GUS en WUS (x100ms)
APPLTIJD	Maximum executietijd applicatie (x10ms)

Tabel 13.9 Parameters procesbesturing

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
PP.I	PP.I="GUSWUS","APPLTIJD"	Vraag object PP.I op
PP.I:A	PP.I:A="N=PP.I,T=1,E=<NUMPP>,U=4444,MAX=32,F=2,O='Index parameters (proces)'"	Opvragen van PP.I object attributen
PP.I/#0	PP.I/#0="GUSWUS"	Opvragen van een element in PP.I object
PP.I/#0="MijnIndex"	:E=11	Wijzigen van element niet mogelijk

Tabel 13.10 Voorbeelden PP.I

13.2.5 PP

Het object PP geeft de parameter instelling voor alle parameters weer:

Attribuut	Type	Object	Omschrijving
N	1	PP	Naam
O	1	parameter instelling	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMPP	aantal data-elementen
I	1	PP.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 13.11 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
PP:E	PP:E=<aantal pp>	Vraag het aantal elementen van object PP op
PP:A	PP:A="N=PP,T=0,E=<NUMPP>,U=6644,I=PP.I,MIN=0,S=1,F=1,O='Parameter instelling (proces)'"	Vraag de attributen op van object PP
PP/#0	PP/#0=<waarde>	Vraag een element uit object PP op

Tabel 13.12 Voorbeelden PP

13.2.6 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 13.13 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 13.14 Voorbeelden EXTRAINFO.I

13.2.7 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 13.15 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 °C"
 EXTRAINFOEXT/#1="Windsterkte en richting gemeten op 3m hoogte"
 EXTRAINFOEXT/#2="Aantal voertuigen per uur; rijsnelheid N299 N-richting"
 EXTRAINFOEXT/#3="Aantal voertuigen per uur; rijsnelheid N299 Z-richting"

EXTRAINFO.A/#0="22°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

13.2.8 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 13.16 Object attributen

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

13.2.9 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 13.17 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:uumss"
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 13.18 Voorbeelden PAR.LB

13.2.10 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 stapgrootte

Tabel 13.19 Object attributen

Zie de paragraaf 13.2.9 voor de formaat omschrijving.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
PAR.LA/#0	De VRI geeft een <ParameterEventBericht>	Vraag de waarde van het eerste onbevestigde logboekobject op.

	terug. Zie referentie.	
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 13.20 Voorbeelden PAR.LA

13.2.11 Programma-events

De programma-events omvatten algemene events en storingsen 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		Ja
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 13.21 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 trigger event in het object datacom zal bij het bereiken van de 90% grens een trigger event 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 optreedt 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 13.22 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'.

13.2.12 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 13.23 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."

14. Protocol

14.1 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 14.1 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.

14.2 Objecten protocol

14.2.1 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 14.2 Object attributen

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

Een voorbeeld is opgenomen in de paragraaf 14.2.3.

14.2.2 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 14.3 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 14.2.3.

14.2.3 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 14.4 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.

14.2.4 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 14.5 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

14.2.5 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 14.6 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 14.7 Voorbeelden BB0

14.2.6 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 14.8 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 14.9 Voorbeelden BB1

14.2.7 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 14.10 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 14.11 Voorbeelden BBA0

14.2.8 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 14.12 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 14.13 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=TGGL,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=TGOR, 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, O='Garantieontruimingstijd'"
BB0	BB0="TGL","TGGL","TOR","TGOR"

Tabel 3.8. Voorbeeld van objecten BB0 en BBA0.

14.2.9 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 14.14 Object attributen

Voorbeeld

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

Tabel 3.25. Voorbeeld PING

15. Detectie

15.1 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 haatmetingen binnen RWS-C. Deze vallen onder de index SG.I, vanaf RWS-C versie komende na 1.11.21 valt dit onder D.I.

Voor detectie zijn de volgende objecten gedefinieerd:

Object	Omschrijving
D.I	Index
TDB	Bezettijd voor aanvraag (alleen bij Ccol)
TDH	Hiaattijd (alleen bij Ccol)
TDH1	Actuele 1 ^e hiaattijd (alleen bij RWS-C)
TDH11	1 ^e hiaattijd periode 1 (alleen bij RWS-C)
TDH12	1 ^e hiaattijd periode 2 (alleen bij RWS-C)
TDH2	Actuele 2 ^e hiaattijd (alleen bij RWS-C)
TDH21	2 ^e hiaattijd periode 1 (alleen bij RWS-C)
TDH22	2 ^e hiaattijd periode 2 (alleen bij RWS-C)

Tabel 15.1 Detectie objecten

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.

15.2 Objecten detectie

15.2.1 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 15.2 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 '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"	Opragen toestand detector met index 0

Tabel 15.3 Voorbeelden index detectoren

15.2.2 TDB (alleen bij CCOL)

Het object TDB geeft de bezettijd weer van alle detectoren:

Attribuut	Type	Object	Omschrijving
N	1	TDB	Naam
O	1	Bezettijd voor aanvraag	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	2, getal met de eenheid 0.1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 15.4 Object attributen

De volgende tabel bevat enkele voorbeelden met element 021:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TDB/021=20	TDB/021=20	Bezettijd detector 021 ingesteld op 20, de aanvraag

Tabel 15.5 Voorbeelden bezettijden

15.2.3 TDH (alleen bij CCOL)

Het object TDH geeft de hiaattijd weer van alle detectoren:

Attribuut	Type	Object	Omschrijving
N	1	TDH	Naam
O	1	Hiaattijd	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	2, getal met de eenheid 0.1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 15.6 Object attributen

De volgende tabel bevat enkele voorbeelden met element 021.

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TDH/021=30	TDH/021=30	Hiaattijd detector 021 ingesteld op 30

Tabel 15.7 Voorbeelden hiaattijden

15.2.4 TDH1 (alleen bij RWS-C)

Het object TDH1 geeft de actuele 1^e hiaattijd weer van alle detectoren:

Attribuut	Type	Object	Omschrijving
N	1	TDH1	Naam
O	1	Actuele 1 ^e hiaattijd	Omschrijving
T	0	0	Type
U	0	6644	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 de eenheid 0.1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 15.8 Object attributen

De volgende tabel bevat enkele voorbeelden met element 021.

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TDH1/021	TDH1/021=30	Hiaattijd detector 021 is ingesteld op 30

Tabel 15.9 Voorbeelden hiaattijden

15.2.5 TDH11 (alleen bij RWS-C)

Het object TDH11 geeft de 1^e hiaattijd periode 1 weer van alle detectoren. Verwezen wordt naar paragraaf 15.2.4 voor de inhoudelijke omschrijving.

De volgende tabel bevat enkele voorbeelden met element 021.

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TDH11/021=30	TDH1/021=30	Hiaattijd detector 021 is ingesteld op 30
TDH11/021	TDH1/021=30	Opvragen huidige instelling.

Tabel 15.10 Voorbeelden hiaattijden

15.2.6 TDH12 (alleen bij RWS-C)

Het object TDH12 geeft de 1^e hiaattijd periode 2 weer van alle detectoren. Verwezen wordt naar paragraaf 15.2.4 voor de inhoudelijke omschrijving.

15.2.7 TDH2 (alleen bij RWS-C)

Het object TDH2 geeft actuele 2^e hiaattijd weer van alle detectoren. Verwezen wordt naar paragraaf 15.2.4 voor de inhoudelijke omschrijving.

15.2.8 TDH21 (alleen bij RWS-C)

Het object TDH21 geeft de 2^e hiaattijd periode 1 weer van alle detectoren. Verwezen wordt naar paragraaf 15.2.4 voor de inhoudelijke omschrijving.

15.2.9 TDH22 (alleen bij RWS-C)

Het object TDH22 geeft de 2^e hiaattijd periode 2 weer van alle detectoren. Verwezen wordt naar paragraaf 15.2.4 voor de inhoudelijke omschrijving.

16. Uitgangen

16.1 Algemeen uitgangen

Voor de 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)

Tabel 16.1 Uitgangen objecten

(overige) uitgangen

Voor de overige uitgangen zijn de volgende objecten gedefinieerd: U.A (actuele toestand), U.I (Index).

Voor de uitgangen zijn de toestanden, timer toestanden en logboeken beschikbaar.

16.2 Objecten uitgangen

16.2.1 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 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 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 16.3 Voorbeelden U.I

16.2.2 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 16.4 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 automatisch 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 16.5 Voorbeelden U.A

16.2.3 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 16.6 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 automatisch weergeeft waarden van type 0 (== getal)
TU.A:A	TU.A:A="N=TU.A,T=0,	Vraag attributen op van object TU.A

	E=<NUMU>,U=4444,I=U.I, MIN=0,S=1,F=2, O='Timer bezet/onbezet'	
TU.A/#0	TU.A/#0=<waarde_1>	Vraag een element op van het object TU.A

Tabel 16.7 Voorbeelden TU.A

16.2.4 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 16.8 Object attributen

Formaat: Toestand van overige uitgangen (50), type 1

Het object geeft een string met daarin een timestamp en voor iedere uitgang een karakter dat 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 16.9 Voorbeelden U.LB

16.2.5 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 16.10 Object attributen

Formaat: Toestand van overige uitgangen (50), type 1

Het object geeft een string met daarin een timestamp en voor iedere in/uitgang een karakter dat 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 16.11 Voorbeelden U.LA

17. Signaalgroepen

17.1 Algemeen signaalgroepen

De regeltechnische communicatie naar de weggebruiker vind plaats met behulp van signaalgroepen en uitgangsignalen. Dit hoofdstuk omschrijft alle objecten met betrekking tot de signaalgroepen. De uitgangssignalen worden onder het hoofdstuk 18, vanaf paragraaf 18.2.21 behandeld.

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. 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. Deze hiaatmetingen zijn in het 15 vanaf paragraaf 15.2.4 opgenomen.

Het hoofdstuk / deze paragraaf is opgedeeld meerdere onderdelen:

- De signaalgroep toestand objecten;
- De garantietijden objecten, onder te verdelen in applicatie- en procesbesturing;
- De regeltechnische 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
SGI.A	Actuele toestand intern (wachtrood, vastgroen, verlenggroen, etc).
TSGI.A	Lopende timer interne signaalgroep toestand.
SGI.LB	Logboek interne signaalgroep toestand (laatste xx veranderingen).
SGI.LA	Logboek interne signaalgroep toestand (onbevestigd).
TOR	Ontruimingstijden zoals gedefinieerd in het applicatiepakket (Alleen verplicht bij RWSC, optioneel bij CCOL).
TGOR	Garantie ontruimingstijden zoals gedefinieerd in het applicatiepakket.
TGG	Garantiegroentijd (appl)
TGGL	Garantiegeeltijd (appl) (Alleen verplicht bij RWSC, optioneel bij CCOL)
TMGL	Maximum geeltijd (appl)
TGR	Garantieroodtijd (appl)
TVG	Vastgroentijd
TVAG	Voertuigafhankelijk verlenggroen
TGL	Geeltijd
TMG	Actuele maximumgroentijd (alleen bij RWS-C)
TMG1	Maximumgroentijd 1 (alleen bij RWS-C)
TMG2	Maximumgroentijd 2 (alleen bij RWS-C)
TMG3	Maximumgroentijd 3 (alleen bij RWS-C)
TMG4	Maximumgroentijd 4 (alleen bij RWS-C)
TMG5	Maximumgroentijd 5 (alleen bij RWS-C)
TMG6	Maximumgroentijd 6 (alleen bij RWS-C)
LSGI	Lijndump SG-toestand (int)

Tabel 17.1 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 (TOR) waarbij geldt dat ontruimingstijden nooit lager ingesteld mogen worden dan de garantie ontruimingstijden.

De objecten TGG, TGGL en TGR bevatten de garantietijden zoals gedefinieerd in het applicatiepakket.

Regeltechnische tijden

Binnen het applicatiepakket zijn er een groot aantal tijdstellingen gerelateerd aan signaalgroepen, zoals o.a. vastgroentijd, maximum groentijd, etc. Voor een beschrijving van deze tijden wordt verwezen naar de documentatie van het applicatiepakket.

17.2 Objecten signaalgroepen

17.2.1 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 17.2 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 '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 17.3 Voorbeelden index signaalgroepen

17.2.2 SGI.A

Het object SGI.A geeft de interne signaalgroepstoestand weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	SGI.A	Naam
O	1	Signaalgroepstoestand (intern)	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	18	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	31	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 17.4 Object attributen

Formaat: Signaalgroepstoestand intern (31), type 0

De interne signaalgroepstoestand is applicatiespecifiek, d.w.z. iedere applicatie kent zijn eigen interne signaalgroepstoelstanden. De volgende tabel geeft een overzicht van een groot aantal denkbare toestanden.

Waarde	Karakter	Toestand	C-COL	RWS-C
0	0	Recht op groen		ROG
1	1	Rood voor groen	RA	RVG
2	2	Garantiegroen		
3	3	Voorstartgroen	VS	
4	4	Vastgroen	FG	VG
5	5	Eerste VAG		1e VAG
6	6	Wachtgroen	WG	WG
7	7	Tweede VAG		2e VAG
8	8	Verlenggroen	VG	
9	9	Meeverlenggroen	MG	MVG
10	A	Veiligheidsgroen		
11	B	Garantiegeel		
12	C	Geel	FGL	VGL
13	D	Verlenggeel	VGL	VAGL
14	E	Garantierood		
15	F	Wachtrood		WR
16	G	Rood voor aanvraag	RV	
17	H	Gedoofd		
18	I	Gereserveerd		

Tabel 17.5 Signaalgroepstoestand extern

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
SGI.A/#0	SGI.A/#0=<waarde>	Opvragen van een element in SGI.A object, waarbij <waarde> een getal is van 0 t/m 18
SGI.A:O	SGI.A:O="Signaalgroepstoestand (int)"	Opvragen van een attribuut in SGI.A object
@1#SGI.A:O="ok"	:E=11	Wijzigen van attribuut niet toegestaan.
SGI.A:O	SGI.A:O="Signaalgroepstoestand (int)"	Opvragen van attribuut

Tabel 17.6 Voorbeelden interne signaalgroepstoestand

17.2.3 TSGI.A

Het object TSGI.A geeft de timer van de interne signaalgroepstoestand weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TSGI.A	Naam
O	1	Timer signaalgroepstoestand (intern)	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 17.7 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TSGI.A	TSGI.A= <waarden van signaalgroepstoelstanden>	Lees het object TSGI.A
Opvragen van	Opvragen van TSGI.A object	Opvragen van een element in TSGI.A object

TSGL.A object attributen	attributen	
TSGL.A:A	TSGL.A:A="N=TSGL.A,T=0,E=<NUMSG>,U=4444,I=SG.I,MIN=0,S=1,F=2,O=Timer signaalgroepstatus (int)"	Opvragen van TSGL.A object attributen

Tabel 17.8 Voorbeelden timer interne signaalgroepstatus

17.2.4 SGI.LB

Het object SGI.LB geeft het interne signaalgroeplogboek weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	SGI.LB	Naam
O	1	Signaalgroeplogboek (intern)	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	31	Data-element formaat
S	0		Data-element stapgrootte

Tabel 17.9 Object attributen

Formaat: Toestand van alle interne signaalgroepstatus (31), 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 }
SGKarakter = /* zie tabel signaalgroepstatus (intern) */
HMS_timestamp = "hhmmss"
```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
SGI.LB	De VRI geeft als waarde: <toestand>, waarbij deze tussen (en inclusief) 0 en 18 ligt.	Vraag de waarde van SGI.LB op.

Tabel 17.10 Voorbeelden object SGI.LB

17.2.5 SGI.LA

Het object SGI.LA geeft het interne (onbevestigde) signaalgroeplogboek weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	SGI.LA	Naam
O	1	Signaalgroeplogboek (onbevestigd/intern)	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	31	Data-element formaat
S	0		Data-element stapgrootte

Tabel 17.11 Object attributen

Formaat: Toestand van alle interne signaalgroep toestanden (31), 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 }
SGKarakter = /* zie tabel signaalgroep toestand (intern) */
HMS_timestamp = "hhmmss"
```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
SGI.LA	De VRI geeft als waarde: <toestand>, waarbij deze tussen (en inclusief) 0 en 18 ligt.	Lees de toestand uit van het object signaalgroep toestanden intern onbevestigd (SGI.LA).
SGI.LA/#0=""		Maak het eerste object leeg.
SGI.LA	:E=17	Meermaals het object SGE.LA/#0="" opgeven resulteert in :E17, buffer is leeg.

Tabel 17.12 Voorbeelden object SGI.LA

17.2.6 TOR

Het object TOR geeft de ontruimingstijd (applicatie) weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TOR	Naam
O	1	Ontruimingstijd (applicatie)	Omschrijving
T	1	1	Type
U	0	6644	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 17.13 Object attributen

De mogelijkheden voor het adresseren van de ontruimingstijden zijn weergegeven in de volgende tabel.

De volgende tabel bevat enkele voorbeelden met elementen 01, 02, 03 en 04:

Methode	Omschrijving
TOR	Alle ontruimingstijden
TOR/*	Alle ontruimingstijden (gelijk aan TOR)
TOR/*,*	Alle ontruimingstijden (gelijk aan TOR)
TOR/SG03,SG02	element 9
TOR/SG01,*	elementen 0, 1, 2, 3
TOR/SG01	gelijk aan TOR/SG01, *
TOR/*,SG02	elementen 1, 5, 9, 13
TOR/SG01-SG03,SG01	elementen 0, 4, 8
TOR/SG02,SG02-SG03	elementen 5, 6
TOR/SG03,SG02-	elementen 9, 10, 11
TOR/SG01-SG02	elementen 0, 1, 2, 3 en 4, 5, 6, 7

Tabel 3.15. Voorbeelden van het lezen van een object met 2 dimensies

17.2.7 TGOR

Het object TGOR geeft de garantie ontruimingstijd (applicatie) weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TGOR	Naam
O	1	Garantie ontruimingstijd (applicatie)	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 17.14 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TGOR:E	TGOR:E=<aantal sg>,<aantal sg>	Vraag aantal elementen op
TGOR/#0	TGOR/#0=<waarde>,<waarde e>,...	Vraag een element op, waarde in automaat: alle TGOR met 1e SG
TGOR:I	TGOR:I=SG.I,SG.I	Vraag indexverwijzing op

Tabel 17.15 Voorbeelden object TGOR

17.2.8 TGG

Het object TGG geeft de garantie groentijd (applicatie) weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TGG	Naam
O	1	Garantie groentijd (applicatie)	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 17.16 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TGG:E	TGG:E=<aantal sg>	Aantal elementen opvragen
TGG/#0	TGG/#0=<waarde>	Een element opvragen
TGG:I	TGG:I=SG.I,	Indexverwijzing opvragen

Tabel 17.17 Voorbeelden object TGG

17.2.9 TGGL

Het object TGGL geeft de garantie geeltijd (applicatie) weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TGGL	Naam
O	1	Garantie geeltijd (applicatie)	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 17.18 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TGGL:E	TGGL:E=<aantal sg>	Aantal elementen opvragen
TGGL/#0	TGGL/#0=<waarde>	Een element opvragen
TGGL:I	TGGL:I=SG.I	Indexverwijzing opvragen

Tabel 17.19 Voorbeelden object TGGL

17.2.10 TMGL

Het object TMGL geeft de maximum geeltijd (applicatie) weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TMGL	Naam
O	1	Maximum geeltijd (applicatie)	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 17.20 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TMGL:E	De VRI geeft <aantalElementen> als waarde terug. Dit betreft het aantal data-elementen in object TMGL.	Vraag aantal data-elementen in object TMGL op.
TMGL/#0	De VRI geeft als waarde: <waarde> in een eenheid van 0.1 seconden.	Vraag de waarde van het object op.

Tabel 17.21 Voorbeelden object TMGL

17.2.11 TGR

Het object TGR geeft de garantie roodtijd (applicatie) weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TGR	Naam
O	1	Garantie roodtijd (applicatie)	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 17.22 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
-----------	------------------------------	--------------

TGR:E	TGR:E=<aantalElementen>	Aantal elementen opvragen
TGR/#0	TGR/#0=<waarde>	Een element opvragen
TGR:I	TGR:I=SG.I,	Indexverwijzing opvragen

Tabel 17.23 Voorbeelden object TGR

17.2.12 TVG

Het object TVG geeft de vastgroentijd weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TVG	Naam
O	1	Vastgroentijd	Omschrijving
T	1	0	Type
U	0	6644	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 17.24 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TVG	De VRI geeft als waarde: <waarde 1>, <waarde 2>, ... <waarde n>.	Vraag het aantal tijdsinstellingen van object TVG op.
TVG/#0	De VRI geeft als waarde: <waarde> in een eenheid van 0.1 seconden.	Vraag de waarde van het eerste TVG object op.

Tabel 17.25 Voorbeelden object TVG

17.2.13 TVAG

Het object TVG geeft de voertuigafhankelijk verlenggroentijd weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TVAG	Naam
O	1	Voertuigafhankelijk verlenggroentijd	Omschrijving
T	1	0	Type
U	0	6644	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 17.26 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TVAG	TVAG=60,60,60,...	Vraag object TVAG op
TVAG/#0=61	TVAG/#0=61	Wijzig een element uit TVAG object
TVAG/#0	TVAG/#0=61	Gewijzigde element ter controle opvragen

Tabel 17.27 Voorbeelden object TVAG

17.2.14 TGL

Het object TGL geeft de geeltijd weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
-----------	------	--------	--------------

N	1	TGL	Naam
O	1	geeltijd	Omschrijving
T	1	0	Type
U	0	6644	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 17.28 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TGL/#0	TGL/#0=<waarde>	Opvragen van een element in TGL object, waarbij <waarde> de geeltijd is van de eerste signaalgroep
TGL:A	TGL:A="N=TGL,T=0,E=<NUMSG>,U=6644,I=SG.I,MIN=-1,S=1,F=2,O='Geeltijd'"	Opvragen van TGL object attributen
TGL%	:E=10	Invalide syntax invoer TGL object

Tabel 17.29 Voorbeelden object TGL

Methode	Omschrijving
TGL	Alle geeltijden
TGL/*	Alle geeltijden (gelijk aan TGL)
TGL/#0	Geeltijd van element 0 (SG01)
TGL/#0-#3	elementen 0, 1, 2, 3
TGL/#2-	alle elementen vanaf 2 (2, 3)
TGL/SG01	Geeltijd SG01
TGL/SG01-SG04	Geeltijden van SG01 t/m SG04
TGL/SG03-	Geeltijden van SG03 en SG04
TGL/#1-SG04	Geeltijden SG02, SG03, SG04

Tabel 3.13. Voorbeeld lezen van objecten

17.2.15 TMG

Het object TMG geeft de actuele maximumgroentijd weer van alle signaalgroepen:

Attribuut	Type	Object	Omschrijving
N	1	TMG	Naam
O	1	Actuele maximumgroentijd	Omschrijving
T	1	0	Type
U	0	6644	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 17.30 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
TMG	De VRI geeft als waarde <waarde> als getal met de eenheid 0.1 seconden	Maximum groentijd opvragen
TMG/#0	TMG/#0=<waarde>	Maximum groentijd opvragen
TMG1:A	TMG1:A="N=TMG1,T=0,E=<NUMSG>,U=6644,I=SG.I,MIN=-1,S=1,F=2,O='Maximumgroen'"	Opvragen van TMG1 object attributen

	tijd 1"	
TMG1@	:E=10	Invalide syntax invoer TMG1 object

Tabel 17.31 Voorbeelden object TMG

17.2.16 TMG1

Het object TMG1 geeft de maximumgroentijd 1 weer van alle signaalgroepen. Verwezen wordt naar paragraaf 17.2.15 voor de inhoudelijke omschrijving.

17.2.17 TMG2

Het object TMG2 geeft de maximumgroentijd 2 weer van alle signaalgroepen. Verwezen wordt naar paragraaf 17.2.15 voor de inhoudelijke omschrijving.

17.2.18 TMG3

Het object TMG3 geeft de maximumgroentijd 3 weer van alle signaalgroepen. Verwezen wordt naar paragraaf 17.2.15 voor de inhoudelijke omschrijving.

17.2.19 TMG4

Het object TMG4 geeft de maximumgroentijd 4 weer van alle signaalgroepen. Verwezen wordt naar paragraaf 17.2.15 voor de inhoudelijke omschrijving.

17.2.20 TMG5

Het object TMG5 geeft de maximumgroentijd 5 weer van alle signaalgroepen. Verwezen wordt naar paragraaf 17.2.15 voor de inhoudelijke omschrijving.

17.2.21 TMG6

Het object TMG6 geeft de maximumgroentijd 6 weer van alle signaalgroepen. Verwezen wordt naar paragraaf 17.2.15 voor de inhoudelijke omschrijving.

17.2.22 LSGI

Het object LSGI geeft de lijndump SG-toestanden (intern) voor signaalgroepen weer:

Attribuut	Type	Object	Omschrijving
N	1	LSGE	Naam
O	1	Lijndump SG-toestand (intern)	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	31	Data-element formaat
S	0		Data-element stapgrootte

Tabel 17.32 Object attributen

Formaat: Toestand van alle interne signaalgroepstanden (31), 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 }
SGKarakter = /* zie tabel signaalgroepstand (intern) */
HMS_timestamp = "hhmmss"
```

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
LSGI	LSGI="HHMMSS:FF110F..."	Vraag de linedump van de interne signaalgroepstanden op. Het tijdstempel is actueel en de dump komt overeen met de interne toestand.
LSGI:A	LSGI:A="N=LSGI,T=1,E=1,U=4444,F=31,O=Lijndump SGtoestand (int)'"	Vraag de attributen op van het object LSGI
LSGI/#0	LSGI/#0="<HHMMSS>" + ":" + "<signaalgroepstandkarakters>"	Vraag een element op van het object LSGI

Tabel 17.33 Voorbeelden object LSGI

18. Regelapplicatie

18.1 Algemeen regelapplicatie

Voor de regelapplicatie zijn de volgende objecten gedefinieerd:

Object	Omschrijving
T.I	Index timers (applicatie)
T	Tijdinstelling (applicatie)
T.A	Lopende tijd (applicatie)
T.T	Type tijden
C.I	Index counter (applicatie) (alleen bij Ccol)
C	Counterinstelling (applicatie) (alleen bij Ccol)
C.A	Lopende counter (applicatie) (alleen bij Ccol)
C.T	Type counter (alleen bij Ccol)
P.I	Index parameters (applicatie)
P	Parameterinstelling (applicatie)
P.T	Type parameters (applicatie)
EGGP.I	Index EGG parameters (applicatie) (alleen bij RWS-C)
EGGP	EGG parameterinstelling (applicatie) (alleen bij RWS-C)
EGGP.T	Type EGG parameters (applicatie) (alleen bij RWS-C)
S.I	Index schakelaars (applicatie)
S	Schakelaar (applicatie)
S.T	Type schakelaars (applicatie)
KL.I	Index klokparameters (alleen bij RWS-C)
KLB	Klok parameter 1 (alleen bij RWS-C)
KLE	Klok parameter 2 (alleen bij RWS-C)
CIFGUS	CIF gewenste uitgangssturing
CIFWUS	CIF werkelijk uitgangssturing
CIFIS	CIF ingangsstatus
CIFWPS	CIF werkelijke programmastatus
CIFGPS	CIF gewenste programmastatus
CIFKLOK	CIF kalendertijd
CIFPARAM1	CIF parameter tabel 1
CIFPARAM2	CIF parameter tabel 2
PL.I	Index signaalplannen (alleen bij Ccol)
PLTXMAX	maximum waarde cyclustijd (*TX_max) (alleen bij Ccol)
PLTPLON	inschakeltijd signaalplan (*TPL_on) (alleen bij Ccol)
PLTPLOFF	uitschakeltijd signaalplan (*TPL_off) (alleen bij Ccol)
PLTXA	parameter vooruitschakelen (*TXA[]) (alleen bij Ccol)
PLTXB	parameter SG[] (*TXB[]) (alleen bij Ccol)
PLTXC	parameter EWG[] /SVG[] (*TXC[]) (alleen bij Ccol)
PLTXD	parameter EVG[] /SMG[] (*TXD[]) (alleen bij Ccol)
PLTXE	parameter EMG[] (*TXE[]) (alleen bij Ccol)
BL.A	Actueel blok / module / stage (alleen verplicht bij RWS-C, optioneel bij Ccol)

Tabel 18.1 Regelapplicatie objecten

Timers, counters, parameters en schakelaars

De tijdinstantellingen (in het applicatiepakket) bestaan uit: T (tijdinstantelling), T.I (Index) en eventueel T.A (lopende tijd).

NB. Het aantal elementen en de Index van de elementen zijn vastgelegd in de applicatie.

De counters (in het applicatiepakket) bestaan uit: C (counter instelling), C.I (Index) en eventueel C.A (lopende counter).

NB. Het aantal elementen en de Index van de elementen zijn vastgelegd in de applicatie.

De parameters (in het applicatiepakket) bestaan uit: P (parameter instelling) en P.I (Index).

NB. Het aantal elementen en de Index van de elementen zijn vastgelegd in de applicatie.

De schakelaars (in het applicatiepakket) bestaan uit: S (schakelaar instelling) en S.I (Index).

NB. Het aantal elementen en de Index van de elementen zijn vastgelegd in de applicatie.

In applicatiepakketten (o.a. RWSC) kunnen ook specifieke klokperiodes als parameter worden ingesteld. De klokperiodes bestaan uit: KLB (begin klokperiode), KLE (einde klokperiode) en KL.I (Index).

NB. Het aantal elementen en de Index van de elementen zijn vastgelegd in de applicatie.

De EGG parameters (in het applicatiepakket) bestaan uit: EGGP (parameterinstelling) en EGGP.I (index).

NB. Het aantal elementen en de Index van de elementen zijn vastgelegd in de applicatie.

Interface

De objecten CIFxxx zijn bedoeld voor monitoring van de CVN C-interface. De CVN C-interface is een interface tussen een verkeerskundige applicatiepakket, zoals CCOL en RWSC en de regeltoestel specifieke procesbesturing. Het aantal elementen van de objecten wordt bepaald door het applicatiepakket.

NB. In het IVERA protocol worden de CIF_IBER en CIF_UBER buffers niet ondersteund.

NB. Voor een beschrijving van de CVN C-interface wordt verwezen naar de CVN C-interface specificatie.

Signaalplannen

Binnen CCOL kan gebruik gemaakt worden van signaalplansturing. Met behulp van een aantal IVERA objecten kunnen de parameters voor de signaalplannen worden ingesteld. Hiervoor zijn de objecten PLx beschikbaar.

BL.A

Het actieve blok, module of stage (afhankelijk van het applicatiepakket) wordt weergegeven met het object BL.A.

18.2 Objecten regelapplicatie

18.2.1 T.I

Het object T.I geeft de index weer van alle timers:

Attribuut	Type	Object	Omschrijving
N	1	T.I	Naam
O	1	index timers (applicatie)	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMT	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 18.2 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 'T_(n-1)', waarbij (n-1) een getal weergeeft.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
T.I	T.I="<timer_1>","<timer_2>","...","<timer_n>"	Opvragen object T.I. Het resultaat mag in geen geval lege velden tonen.
T.I/#0	T.I/#0="<timer_1>"	Vraag een element op
T.I/#<laatsteElementWaarde+1>	:E=12	Vraag een ongeldig element bereik op

Tabel 18.3 Voorbeelden object T.I

18.2.2 T

Het object T geeft de tijdsinstelling weer van alle timers:

Attribuut	Type	Object	Omschrijving
N	1	T	Naam
O	1	tijdsinstelling (applicatie)	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMT	aantal data-elementen
I	1	T.I	Index verwijzing per dimensie
MIN	0	-1	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1	T.T	Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.4 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
T/#0	T/#0 = <waarde>	Opvragen van een element in T object
T/#0 = 1200	T/#0 = 1200	Wijzigen van element naar een waarde binnen data bereik
T/#0	T/#0 = 1200	Opvragen van element
T/#0 = <waarde>	T/#0 = <waarde>	Wijzigen van element naar waarde
T/#0 = -10	:E=16	Wijzigen van element naar een waarde buiten het data bereik

Tabel 18.5 Voorbeelden object T

18.2.3 T.A

Het object T.A geeft de lopende tijd weer van alle timers:

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

Tabel 18.6 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
T.A/#0=4.5	:E=11	Wijzigen niet toegestaan.
T.A/#0	T.A/#0=<waarde>	Vraag de waarde op. De VRI geeft <waarde> als waarde terug. Deze is opgeslagen als een getal met de eenheid 0.1 seconden.
T.A:E	De VRI geeft <aantalElementen> als waarde terug. Dit betreft het aantal data-elementen in object T.A.	Vraag alle data-elementen in object T.A op.

Tabel 18.7 Voorbeelden object T.A

18.2.4 T.T

Het object T.T geeft de type tijden weer van alle timers:

Attribuut	Type	Object	Omschrijving
N	1	T.T	Naam
O	1	Type tijden	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMT	aantal data-elementen
I	1	T.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	80	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	100	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.8 Object attributen

Formaat: Element type (100), type 0

In totaal worden er 7 bits gebruikt voor het element type.

De eerste 5 bits worden gebruikt volgens de volgende tabel

Waarde	Naam	Omschrijving
0	DEF_type	Geheel getal
1	TE_type	Tienden seconden
2	TS_type	Seconden
3	TM_type	Minuten
4	CT_type	Tellerwaarde
5	SW_type	Software schakelaar
10	MIN_type	Klokwaarde – minuten
11	UUR_type	Klokwaarde – uren
12	DGS_type	Klokwaarde – dagsoort
13	DAG_type	Klokwaarde – dag
14	MND_type	Klokwaarde – maand
15	TI_type	Klokwaarde – tijd: uren*100 + minuten
16	DA_type	Klokwaarde – datum: dag*100 + maand

Tabel 18.9 Bits elementtype

Het 7e bit wordt gebruikt om aan te geven of een element read-only is.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
T.T	T.T=<valt>,<valt>,...,<valt>	Opvragen van de types tijden.

Tabel 18.10 Voorbeelden object T.T

18.2.5 C.I (alleen bij CCOL)

Het object C.I geeft de index weer van alle counters:

Attribuut	Type	Object	Omschrijving
N	1	C.I	Naam
O	1	index counters (applicatie)	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMC	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 18.11 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 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 'C_(n-1)', waarbij (n-1) een getal weergeeft.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
C.I:O	C.I:O="Index counters (appl)"	Vraag omschrijving object C.I op
C.I	C.I="<counter_1>","<counter_2>","...","<counter_n>"	Vraag object C.I op, het resultaat mag in geen geval lege velden tonen.
C.I:A	C.I:A="N=C.I,T=1,E=<NUMC>,U=4444,MAX=32,F=2,O='Index counters (appl)'"	Vraag attributen op van object C.I

Tabel 18.12 Voorbeelden object C.I

18.2.6 C (alleen bij CCOL)

Het object C geeft de tijdsinstelling weer van alle counters:

Attribuut	Type	Object	Omschrijving
N	1	C	Naam
O	1	counterinstelling (applicatie)	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMC	aantal data-elementen
I	1	C.I	Index verwijzing per dimensie
MIN	0	-1	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1	C.T	Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.13 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
C:E	De VRI geeft als waarde <aantalCounters>, het aantal counters binnen het applicatiepakket is duidelijk.	Vraag op hoeveel counters er aanwezig zijn in het applicatiepakket.
C/#0	De VRI geeft als waarde <counter>, de waarde in de counter is bekend.	Vraag de waarde van counter nummer nul op.
C/<naam>=<waarde>	:E=11	Ken waarde van <waarde> toe aan waarde van <naam>
C/<naam>		Vraag deze vervolgens op ter controle.

Tabel 18.14 Voorbeelden object C

18.2.7 C.A (alleen bij CCOL)

Het object C.A geeft de lopende tijd weer van alle counters:

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

Tabel 18.15 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
C.A/#<aantal/2>	C.A/#<aantal/2>=<waarde>	Opvragen van een element uit object C.A, waarbij waarde gelijk is aan het index-nummer.
C.A/#<aantal/2>=<waarde2>	:E=11	Wijzigen van waarde van opgevraagde element

Tabel 18.16 Voorbeelden object C.A

18.2.8 C.T (alleen bij CCOL)

Het object C.T geeft de type tijden weer van alle counters:

Attribuut	Type	Object	Omschrijving
N	1	C.T	Naam
O	1	Type counters	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMC	aantal data-elementen
I	1	C.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	80	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	100	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.17 Object attributen

Formaat: Element type (100), type 0

In totaal worden er 7 bits gebruikt voor het element type.

De eerste 5 bits worden gebruikt volgens de volgende tabel

Waarde	Naam	Omschrijving
0	DEF_type	Geheel getal
1	TE_type	Tienden seconden
2	TS_type	Seconden
3	TM_type	Minuten
4	CT_type	Tellerwaarde
5	SW_type	Software schakelaar
10	MIN_type	Klokwaarde – minuten
11	UUR_type	Klokwaarde – uren
12	DGS_type	Klokwaarde – dagsoort
13	DAG_type	Klokwaarde – dag
14	MND_type	Klokwaarde – maand
15	TI_type	Klokwaarde – tijd: uren*100 + minuten
16	DA_type	Klokwaarde – datum: dag*100 + maand

Tabel 18.18 Bits elementtype

Het 7e bit wordt gebruikt om aan te geven of een element read-only is.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
C.T	C.T=<valc_1>,<valc_2>,...,<valc_n>	Opvragen types voor de counters.

Tabel 18.19 Voorbeelden object C.T

18.2.9 P.I

Het object P.I geeft de index weer van alle parameterinstellingen:

Attribuut	Type	Object	Omschrijving
N	1	P.I	Naam
O	1	index parameters (applicatie)	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMP	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 18.20 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 'P_(n-1)', waarbij (n-1) een getal weergeeft.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
P.I	P.I="<parameterNaam_1>","<parameterNaam_2>","...","parameterNaam_n>"	Opvragen object P.I op, het resultaat mag in geen geval lege velden tonen.

Tabel 18.21 Voorbeelden object P.I

18.2.10 P

Het object P geeft de tijdsinstelling weer van alle parameterinstellingen:

Attribuut	Type	Object	Omschrijving
N	1	P	Naam
O	1	parameterinstelling (applicatie)	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMP	aantal data-elementen
I	1	P.I	Index verwijzing per dimensie
MIN	0	-1	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1	P.T	Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.22 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
P/#0	P/#0=<waarde>	Opvragen van een element in P object
P:A	P:A="N=P,T=0,E=<NUMP>,U=6644,I=P.I,MIN=-1,S=1,F=1,O='Parameterinstelling (appl)'"	Opvragen van P object attributen
P}	:E=10	Foutieve syntax invoer P object

Tabel 18.23 Voorbeelden object P

18.2.11 P.T

Het object P.T geeft de type parameters weer van alle parameterinstellingen:

Attribuut	Type	Object	Omschrijving
N	1	P.T	Naam
O	1	Type parameters (applicatie)	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMP	aantal data-elementen
I	1	P.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	80	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	100	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.24 Object attributen

Formaat: Element type (100), type 0

In totaal worden er 7 bits gebruikt voor het element type.

De eerste 5 bits worden gebruikt volgens de volgende tabel

Waarde	Naam	Omschrijving
0	DEF_type	Geheel getal
1	TE_type	Tienden seconden
2	TS_type	Seconden
3	TM_type	Minuten
4	CT_type	Tellerwaarde
5	SW_type	Software schakelaar
10	MIN_type	Klokwaarde – minuten
11	UUR_type	Klokwaarde – uren
12	DGS_type	Klokwaarde – dagsoort
13	DAG_type	Klokwaarde – dag
14	MND_type	Klokwaarde – maand
15	TI_type	Klokwaarde – tijd: uren*100 + minuten
16	DA_type	Klokwaarde – datum: dag*100 + maand

Tabel 18.25 Bits elementtype

Het 7e bit wordt gebruikt om aan te geven of een element read-only is.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
P.T	P.T=<valp>,<valp>,...,<valp>	Opvragen types object P.

Tabel 18.26 Voorbeelden object P.T

18.2.12 EGGP.I (alleen bij RWS-C)

Het object EGGP.I geeft de index weer van alle EGG parameterinstellingen:

Attribuut	Type	Object	Omschrijving
N	1	EGGP.I	Naam
O	1	index EGG parameters (applicatie)	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMEGGP	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
EGGP.I	EGGP.I=<EGGparameter_1 >,"<EGGparameter_2>","...," <EGGparameter_n>"	Opvragen object EGGP.I, het resultaat mag in geen geval lege velden tonen.
EGGP.I/#0	EGGP.I/#0=<EGGparameter_1>"	Vraag een element op uit object EGGP.I
EGGP.I/#<laatsteElementWaarde+1>	:E=12	Vraag een ongeldig element bereik op

Tabel 18.28 Voorbeelden object EGGP.I

18.2.13 EGGP (alleen bij RWS-C)

Het object EGGP geeft de tijdsinstelling weer van alle EGG parameterinstellingen:

Attribuut	Type	Object	Omschrijving
N	1	EGGP	Naam
O	1	EGG parameterinstelling (applicatie)	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMEGGP	aantal data-elementen
I	1	EGGP.I	Index verwijzing per dimensie
MIN	0	-1	Minimum data-elementwaarde
MAX	0		Maximum data-elementwaarde
ITYPE	1	EGGP.T	Index data-element type
F	0	1, getal met eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.29 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
EGGP/<naam>=<waarde2>	EGGP/<naam>=<waarde2>	Wijzigen van waarde van opgevraagde naam.
EGGP:E	EGGP:E=<aantal eggp>	Aantal elementen opvragen
EGGP/#0	EGGP/#0=<waarde>	Een element opvragen

Tabel 18.30 Voorbeelden object EGGP

18.2.14 EGGP.T (alleen bij RWS-C)

Het object EGGP.T geeft de type parameters weer van alle EGG parameterinstellingen:

Attribuut	Type	Object	Omschrijving
N	1	EGGP.T	Naam
O	1	Type EGG parameters (applicatie)	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMP	aantal data-elementen
I	1	P.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	80	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	100	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.31 Object attributen

Formaat: Element type (100), type 0

In totaal worden er 7 bits gebruikt voor het element type.

De eerste 5 bits worden gebruikt volgens de volgende tabel

Waarde	Naam	Omschrijving
0	DEF_type	Geheel getal
1	TE_type	Tienden seconden
2	TS_type	Seconden
3	TM_type	Minuten
4	CT_type	Tellerwaarde
5	SW_type	Software schakelaar
10	MIN_type	Klokwaarde – minuten
11	UUR_type	Klokwaarde – uren
12	DGS_type	Klokwaarde – dagsoort
13	DAG_type	Klokwaarde – dag
14	MND_type	Klokwaarde – maand
15	TI_type	Klokwaarde – tijd: uren*100 + minuten
16	DA_type	Klokwaarde – datum: dag*100 + maand

Tabel 18.32 Bits elementtype

Het 7e bit wordt gebruikt om aan te geven of een element read-only is.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
EGGP.T	EGGP.T=<val_1>,<val_2>,... <val_n>	Opvragen de types voor de EGGP.

Tabel 18.33 Voorbeelden object EGGP.T

18.2.15 S.I

Het object S.I geeft de index weer van alle schakelaars:

Attribuut	Type	Object	Omschrijving
N	1	S.I	Naam
O	1	index schakelaars (applicatie)	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMS	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 18.34 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 'S_(n-1)', waarbij (n-1) een getal weergeeft.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
S.I	S.I ="naam","naam",..., "naam"	Vraag de indexnamen van de schakelaars op.
S.I:E	De VRI geeft als waarde terug: <aantalElementen>	Vraag het aantal data-elementen in object S.I op.

Tabel 18.35 Voorbeelden object S.I

18.2.16 S

Het object S geeft de instelling weer van alle schakelaars:

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

Tabel 18.36 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
S/<naam>	S/<naam>=<waarde>	Vraag de waarde van één van de de schakelaar op
S/<naam>=<waard e2>	S/<naam>=<waarde2>	Wijzig de waarde van de schakelaar een naam.
S/<naam>	S/<naam>=<waarde2>	Vraag de waarde uit de vorige stap op.

Tabel 18.37 Voorbeelden object S

18.2.17 S.T

Het object S.T geeft de type schakelaars weer van alle schakelaars:

Attribuut	Type	Object	Omschrijving
N	1	S.T	Naam
O	1	Type schakelaars (applicatie)	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMS	aantal data-elementen
I	1	S.I	Index verwijzing per dimensie
MIN	0	0	Minimum data-elementwaarde
MAX	0	80	Maximum data-elementwaarde
ITYPE	1		Index data-element type
F	0	100	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.38 Object attributen

Formaat: Element type (100), type 0

In totaal worden er 7 bits gebruikt voor het element type.

De eerste 5 bits worden gebruikt volgens de volgende tabel

Waarde	Naam	Omschrijving
0	DEF_type	Geheel getal
1	TE_type	Tienden seconden
2	TS_type	Seconden
3	TM_type	Minuten
4	CT_type	Tellerwaarde
5	SW_type	Software schakelaar
10	MIN_type	Klokwaarde – minuten
11	UUR_type	Klokwaarde – uren
12	DGS_type	Klokwaarde – dagsoort
13	DAG_type	Klokwaarde – dag
14	MND_type	Klokwaarde – maand
15	TI_type	Klokwaarde – tijd: uren*100 + minuten
16	DA_type	Klokwaarde – datum: dag*100 + maand

Tabel 18.39 Bits elementtype

Het 7e bit wordt gebruikt om aan te geven of een element read-only is.

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
S.T	S.T=<vasl_1>,<vals_2>,...,<vals_n> vals_# = 69 als schakelaar read-only is vals_# = 5 voor de overige schakelaars	Vraag de types op voor de schakelaars en controleer of deze of 5 of 69 zijn.
S:ITYPE	S:ITYPE="S.T"	Vraag de type attributen op en controleer deze met behulp van de Object Specificatie

Tabel 18.40 Voorbeelden object S.T

18.2.18 KL.I (alleen bij RWS-C)

Het object KL.I geeft de index weer van alle klokparameters:

Attribuut	Type	Object	Omschrijving
N	1	KL.I	Naam
O	1	index klokparameters	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMKL	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 18.41 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
KL.I	KL.I=" <waarde_1> ", <waarde_2> ", ..., <waarde_n> "	Opvragen van KL.I object, waarbij <waarde> gedefinieerd is in applicatiepakket. Het resultaat mag in geen geval lege velden tonen.

Tabel 18.42 Voorbeelden object KL.I

18.2.19 KLB (alleen RWS-C)

Het object KLB geeft de instelling weer van alle klokparameters:

Attribuut	Type	Object	Omschrijving
N	1	KLB	Naam
O	1	Klok parameter 1	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMKL	aantal data-elementen
I	1	KL.I	Index verwijzing per dimensie
MIN	0	-1	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 18.43 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
KLB/#0=1200	:E=11	Wijzigen van element KLB, inlogniveau 1.
KLB/#0	KLB/#0=<waarde>	Opvragen van element KLB
KLB:MIN	KLB:MIN=<waarde>	Opvragen van attribuut van KLB object

Tabel 18.44 Voorbeelden object KLB

18.2.20 KLE (alleen RWS-C)

Het object KLE geeft de instelling weer van alle klokparameters:

Attribuut	Type	Object	Omschrijving
N	1	KLE	Naam
O	1	Klok parameter 2	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMKL	aantal data-elementen
I	1	KL.I	Index verwijzing per dimensie
MIN	0	-1	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 18.45 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
KLE	KLE=<waarde>	Opvragen van KLE object
KLE:A	KLE:A="N=KLE,T=0,E=<NUMKL>,U=6644,I=KL.I,MIN=-1,MAX=2400,S=1,F=20,O='Klok parameter 2'"	Opvragen van KLE object attributen
KLE/#0	KLE/#0=<waarde>	Opvragen van een element in KLE object
KLE/#0 = 1200	KLE/#0 = 1200	Wijzigen van element naar een waarde binnen data bereik
KLE/#0		Opvragen van element

Tabel 18.46 Voorbeelden object KLE

18.2.21 CIFGUS

Het object CIFGUS geeft de CIF gewenste uitgangssturing weer van alle uitgangen:

Attribuut	Type	Object	Omschrijving
N	1	CIFGUS	Naam
O	1	CIF gewenste uitgangssturing	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMCIFGUS	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 18.47 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
CIFGUS	CIFGUS=0,1,0,0,...	Signaalgroep met index 1 is groen
CIFGUS	CIFGUS=4,4,4,4,...	Automaat is gedoofd
CIFGUS	CIFGUS=0,2,0,0,...	Signaalgroep met index 1 is geel

Tabel 18.48 Voorbeelden object CIFGUS

18.2.22 CIFWUS

Het object CIFWUS geeft de CIF werkelijke uitgangssturing weer van alle uitgangen:

Attribuut	Type	Object	Omschrijving
N	1	CIFWUS	Naam
O	1	CIF werkelijke uitgangssturing	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMCIFWUS	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 18.49 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
CIFGWS	CIFGWS=0,1,0,0,...	Signaalgroep met index 1 is groen
CIFGWS	CIFGWS=4,4,4,4,...	Automaat is gedoofd
CIFGWS	CIFGWS=0,2,0,0,...	Signaalgroep met index 1 is geel

Tabel 18.50 Voorbeelden object CIFWUS

18.2.23 CIFIS

Het object CIFIS geeft de CIF ingangstatus weer van alle ingangen:

Attribuut	Type	Object	Omschrijving
N	1	CIFIS	Naam
O	1	CIF ingangstatus	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMCIFIS	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 18.51 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
CIFIS/#0	CIFIS/#0=<willekeurige waarde>	Voer een leesactie uit op het object CIFIS
CIFIS/#0=112	:E=11	Voer een schrijfactie uit op het object CIFIS

Tabel 18.52 Voorbeelden object CIFIS

18.2.24 CIFWPS

Het object CIFWPS geeft de CIF werkelijke programmastatus weer:

Attribuut	Type	Object	Omschrijving
N	1	CIFWPS	Naam
O	1	CIF werkelijke programmastatus	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMCIFWPS	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 18.53 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
CIFWPS	Element 0 van object CIFWPS heeft de waarde CIF_STAT_GEDOOFD (=1)	Automaat is 'gedoofd'.
CIFWPS	Element 0 van object CIFWPS heeft de waarde CIF_STAT_KP (=2)	Automaat staat op 'knippen'.
CIFWPS	Element 0 van object CIFWPS heeft de waarde CIF_STAT_AR (=4)	Automaat staat op 'alles rood'.
CIFWPS	Nadat de fasebewaking is opgetreden heeft element 1 van CIFGPS de waarde CIF_FB_FOUT (=1)	Automaat is in fasebewaking gekomen.

Tabel 18.54 Voorbeelden object CIFWPS

18.2.25 CIFGPS

Het object CIFGPS geeft de CIF gewenste programmastatus weer:

Attribuut	Type	Object	Omschrijving
N	1	CIFGPS	Naam
O	1	CIF gewenste programmastatus	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMCIFGPS	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 staggrootte

Tabel 18.55 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
CIFGPS=1,0,0,0,0,0,0	Element 0 van object CIFWPS heeft de waarde CIF_STAT_GEDOOFD (=1)	Automaat is 'gedoofd' als gevolg van een programmawens.
CIFGPS=2,0,0,0,0,0,0	Element 0 van object CIFWPS heeft de waarde CIF_STAT_KP (=2)	Automaat staat op 'knipperen' als gevolg van een programmawens.
CIFGPS=4,0,0,0,0,0,0	Element 0 van object CIFWPS heeft de waarde CIF_STAT_AR (=4)	Automaat staat op 'alles rood' als gevolg van een programmawens.

Tabel 18.56 Voorbeelden object CIFGPS

18.2.26 CIFKLOK

Het object CIFKLOK geeft de CIF kalendertijd weer:

Attribuut	Type	Object	Omschrijving
N	1	CIFKLOK	Naam
O	1	CIF 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	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 18.57 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
CIFKLOK	CIFKLOK/#0=< UUMM>	Lees de tijd van de object CIFKLOK
CIFKLOK/#0="MA A 2007-11-09 23:43:12"	:E=11	Wijzigen van de tijd en datum van de object CIFKLOK niet toegestaan.

Tabel 18.58 Voorbeelden object CIFKLOK

18.2.27 CIFPARM1

Het object CIFGPS geeft de CIF parameters van tabel 1 weer:

Attribuut	Type	Object	Omschrijving
N	1	CIFPARM1	Naam
O	1	CIF parameter tabel 1	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMCIFPARM1	aantal data-elementen
I	1		Index verwijzing per dimensie
MIN	0	-4	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 18.59 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
cifparm1/#0=123	:E=11	Wijzig de eerste parameter van het object CIFPARM1

Tabel 18.60 Voorbeelden object CIFPARM1

18.2.28 CIFPARM2

Het object CIFGPS geeft de CIF parameters van tabel 2 weer:

Attribuut	Type	Object	Omschrijving
N	1	CIFPARM2	Naam
O	1	CIF parameter tabel 2	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	NUMCIFPARM2	aantal data-elementen
I	1		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 eenheid 1	Data-element formaat
S	0	1	Data-element stapgrootte

Tabel 18.61 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
Cifparm2/#0=123	:E=11	Wijzig de eerste parameter van het object CIFPARM2

Tabel 18.62 Voorbeelden object CIFPARM1

18.2.29 PL.I

Het object PL.I geeft de index van alle signaalplannen weer:

Attribuut	Type	Object	Omschrijving
N	1	PL.I	Naam
O	1	Index signaalplannen	Omschrijving
T	1	1	Type
U	0	4444	User Identificatie Control
E	0	NUMPL	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 18.63 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
PL.I	PL.I="PL1","PL2","PL3","PL4"	Opvragen object PL.I

Tabel 18.64 Voorbeelden object PL.I

18.2.30 PLTXMAX

Het object PLTXMAX geeft maximum waarde van de cyclustijd van alle signaalplannen weer:

Attribuut	Type	Object	Omschrijving
N	1	PLTXMAX	Naam
O	1	Maximum waarde cyclustijd (*TX_max)	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMPL	aantal data-elementen
I	1	PL.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 18.65 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
PLTXMAX/PL1	PLTXMAX/PL1=120	Opvragen cyclusduur PL1

Tabel 18.66 Voorbeelden object PLTXMAX

18.2.31 PLTPLON

Het object PLTPLON geeft de inschakeltijd signaalplan van alle signaalplannen weer.

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

Attribuut	Type	Object	Omschrijving
N	1	PLTPLON	Naam
O	1	Inschakeltijd signaalplan (*TPL_on)	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
L	0		Logboek
W	0		Wijzigingsteller
E	0	NUMPL	aantal data-elementen
I	1	PL.I	Index verwijzing per dimensie
MIN	0	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	1, getal met eenheid 1	Data-element formaat
S	0		Data-element stapgrootte
A	1		Overzicht alle attributen

Tabel 18.67 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
PLTPLON/PL1	PLTPLON/PL1=1	Opvragen inschakelmoment PL1

Tabel 18.68 Voorbeelden object PLTPLON

18.2.32 PLTPLOFF

Het object PLTPLOFF geeft de uitschakeltijd signaalplan van alle signaalplannen weer:

Attribuut	Type	Object	Omschrijving
N	1	PLTPLOFF	Naam
O	1	Uitschakeltijd signaalplan (*TPL_off)	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMPL	aantal data-elementen
I	1	PL.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 18.69 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
PLTPLOFF/PL1	PLTPLOFF/PL1=1	Opvragen uitschakelmoment PL1

Tabel 18.70 Voorbeelden object PLTPLOFF

18.2.33 PLTXA

Het object PLTXA geeft de parameter vooruitschakelen van alle signaalplannen per plan weer:

Attribuut	Type	Object	Omschrijving
N	1	PLTXA	Naam
O	1	Parameter vooruitschakelen (*TXA[])	Omschrijving
T	1	0	Type
U	0	6644	User Identificatie Control
E	0	NUMPL, NUMSG	aantal data-elementen
I	1	PL.I, 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 18.71 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
PLTXA/PL1,02	PLTXA/PL1,02=0	Opvragen TXA moment PL1 richting 02
PLTXA/PL1,02=5	PLTXA/PL1,02=5	Wijzigen TXA moment PL1 richting 02

Tabel 18.72 Voorbeelden object PLTXA

18.2.34 **PLTXB**

Het object PLTXB geeft de parameter SG[] van alle signaalplannen per plan weer. Verwezen wordt naar paragraaf 18.2.33 voor de inhoudelijke omschrijving.

18.2.35 **PLTXC**

Het object PLTXC geeft de parameter EWG[] / SVG[] van alle signaalplannen per plan weer. Verwezen wordt naar paragraaf 18.2.33 voor de inhoudelijke omschrijving.

18.2.36 **PLTXD**

Het object PLTXD geeft de parameter EVG[] / SMG[] van alle signaalplannen per plan weer. Verwezen wordt naar paragraaf 18.2.33 voor de inhoudelijke omschrijving.

18.2.37 **PLTXE**

Het object PLTXE geeft de parameter EMG[] van alle signaalplannen per plan weer. Verwezen wordt naar paragraaf 18.2.33 voor de inhoudelijke omschrijving.

18.2.38 BL.A (alleen verplicht bij RWS-C, optioneel bij Ccol)

Het object BL.A geeft het actuele blok / module of stage weer:

Attribuut	Type	Object	Omschrijving
N	1	BL.A	Naam
O	1	Actueel blok / module / stage	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		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 18.73 Object attributen

De volgende tabel bevat enkele voorbeelden:

Voorbeeld	Respons vanuit de applicatie	Omschrijving
BL.A	BL.A=2	Vraag het nummer van het actieve blok op.
BL.A/#0=2	:E=11	Wijzigen niet toegestaan.

Tabel 18.74 Voorbeelden object BL.A

18.2.39 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 = "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.

19. Events en Alarms

19.1 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
APPFOUT.I	Fouttoestand
APPFOUT	Index fouttoestand
APP.LA	Logboek met meldingen van ITS-Applicatie (onb)
APP.LB	Logboek met meldingen van ITS-Applicatie
APP.A	Actieve storingslijst

Tabel 19.1 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, verstuurt de slave een <BerichtSlaveTrigger>. De TriggerCode in dit bericht komt overeen met de eventcode zoals omschreven in het object 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 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 triggerlijst te vullen vanaf bijvoorbeeld de centrale. Er zijn NUMEVENT events aanwezig.

19.2 Objecten Events en Alarms

19.2.1 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 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).

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 19.3 Voorbeelden object EVENTLYST.I

19.2.2 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 19.4 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 19.5 Voorbeelden object EVENTLYST.INFO

19.2.3 Object APPFOUT.I

Het object APPFOUT.I geeft de index namen van de foutcode:

Attribuut	Type	Object	Omschrijving
N	1	APPFOUT.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 6 Object attributen

Het object APPFOUT bevat twee foutcodes. De gebruikte nummers zijn de event codes van object APP.LB:

APPFOUT.I		
0	FATAAL	Eerste fatale fout
1	MELDING	Laatste niet fatale fout

Tabel 7 Foutcode

19.2.4 Object APPFOUT

Het object APPFOUT geeft de actuele foutcode:

Attribuut	Type	Object	Omschrijving
N	1	APPFOUT	Naam
O	1	actuele foutcode	Omschrijving
T	1	0	Type
U	0	4444	User Identificatie Control
E	0	2	aantal data-elementen
I	1	APPFOUT.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 8 Object attributen

Voorbeeld:

Alle elementen lezen:

APPFOUT

APPFOUT= 0,2500

19.2.5 Object APP.LA

Het object APP.LA geeft het APP-logboek (onbevestigd):

Attribuut	Type	Object	Omschrijving
N	1	APP.LA	Naam
O	1	APP-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 9 Object attributen

19.2.6 Object APP.LB

Het object APP.LB geeft het APP-logboek:

Attribuut	Type	Object	Omschrijving
N	1	APP.LB	Naam
O	1	APP-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 10 Object attributen

19.2.7 Object APP.A

Het object APP.A bevat de lijst met actuele applicatie storingen:

Attribuut	Type	Object	Omschrijving
N	1	APP.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 11 Object attributen

Voorbeeld:

Een element lezen:

APP.A/#0

APP.A/#0= "20170426:175437,0,2500,"

20. Events

De objecten VRI.LA en VRI.LB bevatten alle relevante events die kunnen optreden in een TLC en zijn beschikbaar via de IVERA-TLC interface. De APP.LA en APP.LB objecten bevatten alle relevante events die kunnen optreden in een applicatie en zijn beschikbaar via de IVERA-APP interface.

Aangezien er een splitsing is in functionaliteit tussen IVERA-APP en IVERA-TLC, worden sommige events alleen op een van de interfaces verwacht, terwijl de andere events op beide interfaces verwacht kunnen worden.

Deze sectie bevat definities van nieuwe events, gewijzigde events en voor elk event een definitie op welke interface het kan worden verwacht.

Nieuwe of gewijzigde events zijn *schuingedrukt* gemarkeerd. De verwachte interface is gemarkeerd met (T) voor IVERA-TLC en/of (A) voor IVERA-APP.

Detail info en aanwezigheid in APP.A of VRI.A is weggelaten voor alle events die niet zijn gewijzigd (gemarkeerd met grijze cellen). Voor meer details van deze events zie **Error! Reference source not found., Error! Reference source not found..**

20.1 Categorieën

De volgende tabel geeft een overzicht van de verschillende event categorieën.

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

20.1.1 I/O events

Code	Omschrijving	Detailinfo	APP.A VRI.A	Interface (A/T)
1010	Lampfout			T
1020	Detectiefout			T
1030	Akoestischefout			T

20.1.2 Program events

Code	Omschrijving	Detailinfo	APP.A VRI.A	Interface (A/T)
2000	Programma event			A/T
2001	VRI status wijziging			T
2002	Programmaomschakeling			A/T
2003	Brugingreep			A
2004	Brandweeringreep			A
2005	AHOB melding			A
2500	Fasebewaking			A
2501	GUS-WUS fouten CVN C-interface			A
2502	Rekentijdproblemen			A
2503	Garantietijdonderschrijding			A
2504	Maximumtijdoverschrijding			A
2505	Start niet kunnen regelen door storing			A
2506	Einde niet kunnen regelen door storing			A
2510	Overig Logboek 90% vol grens bereikt.			A/T
2511	VRI.LA Logboek 90% vol grens bereikt.			T
2512	PAR.LA Logboek 90% vol grens bereikt.			A/T
2513	OV.LA Logboek 90% vol grens bereikt.			A
2514	<i>APP.LA Logboek 90% vol grens bereikt</i>			A
2600	Seriële koppeling - ontbreken levenssignaal.			A/T
2601	Seriële koppeling - geen communicatie.			A/T
2700	Onderspanningsmelding			T
2701	Bovenspanningsmelding			T
2702 ²	Telefoonnummer centrale kwijt			

20.1.3 Supervisor events

Code	Omschrijving	Detailinfo	APP.A VRI.A	Interface (A/T)
3000	Algemeen bewakerevent			T
3001	Conflict			T
3002	Lampfout			T
3003	Meer dan 1 kleur			T
3004	Geelknipperfout			T
3005	Garantietijdonderschrijding			T
3006	Maximumtijdoverschrijding			T
3007	Fout in eindschakelaar			T
3008	Witknipperfout			T
3009	Halfconflict OV			T
3010	Volgordebewaking			T

² Dit event is verouderd, van de iTLC wordt verwacht dat deze communiceert door middel van broadband technologie

20.1.4 Reset events

Code	Omschrijving	Detailinfo	APP.A VRI.A	Interface (A/T)
4000	Algemeen resetevent			A/T
4001	Reset van alle storingen			A/T
4002	Reset van detectiealarmen			T
4003	Reset van lampfouten			T
4004	Reset van applicatiefouten			A
4005	Reset van tellers			A
4006	Reset teller applicatiefouten			A
4007	Reset teller aantal GUS-WUS fouten			A
4008	Reset teller fasebewakingsfouten			A
4009	Reset teller executietijdoverschrijdingen			A
4010	Netspanning uitsterbericht			T
4011	Opstartbericht			T
4012	Deur open politie paneel			T
4013	Deur open wegbeheerder			T
4014	Deur open energie compartiment			T
4015	Testbericht noodkreetmelder			T
4016	Noodstroomvoedingbericht			A/T
4022	'Aanvraag toestemming lokaal' is gedaan door gebruiker bij VRI.			T
4023	'Aanvraag toestemming lokaal' is ingetrokken door gebruiker bij VRI.			T

20.1.5 Commando events

Deze events worden door een IVERA master gebruikt om commando's naar IVERA slaves te sturen door middel van het VRI.C object

Code	Omschrijving	Interface (A/T)
5001	Test putsarmatuur	T
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.	T
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.	T
5100..5199	Gewenste VRI-status vanuit centrale	T
5200..5299	Gewenste programmanummer vanuit centrale	T
5300..5399	Gewenste subprogrammanummer vanuit centrale	T
5990	Geeft VRI opdracht een warme herstart uit te voeren	A/T
9990	Geeft VRI opdracht een warme herstart uit te voeren (verouderd)	T

20.1.6 Data communicatie events

Code	Omschrijving	Detailinfo	APP.A VRI.A	Interface (A/T)
6000	Testtrigger			A/T
6001	Begin fysieke verbinding			A/T
6002	Einde fysieke verbinding			A/T
6003	Poging tot inbreuk IVERA			A/T
6004	Uitbellen naar centrale			A/T
6005	Login IVERA			A/T
6006	Logout IVERA			A/T
6012	Deur open politie paneel	0 = gesloten, 1 = open		T
6013	Deur open wegbeheerder	0 = gesloten, 1 = open		T
6014	Deur open energie compartiment	0 = gesloten, 1 = open		T
6023	Poging tot inbreuk TLC-FI			T
6025	TLC-FI verbonden			A/T
6026	TLC-FI verbroken			A/T
6027	Configuratiefout TLC-FI			A/T
6041	Ivera gebruiker aangemaakt			A/T
6042	Ivera gebruiker verwijderd			A/T
6043	Ivera gebruiker gewijzigd	1 = naam, 2 = wachtwoord, 3 = gebruikersgroep		A/T
6051	TLC-FI gebruiker aangemaakt			A/T
6052	TLC-FI gebruiker verwijderd			A/T
6053	TLC-FI gebruiker gewijzigd	1 = naam, 2 = wachtwoord, 3 = type		A/T
6061	RIS-FI gebruiker aangemaakt			A/T
6062	RIS-FI gebruiker verwijderd			A/T
6063	RIS-FI gebruiker gewijzigd	1 = naam, 2 = wachtwoord, 3 = type		A/T

21. 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 .. z

NB. De beschrijving is overgenomen uit “Brinch Hanssen on Pascal compilers”.