

CardAccess[®]

Database en API beschrijving

CA3000 Versie 2.11



Inhoudsopgave

1	Database beschrijving CA3000	2
2	Tabelomschrijvingen	6
1.1	Kaarthouder tabel	6
1.1.1	Gegevens wijzigen via de kaarthouder tabel	6
1.1.2	Tabel beschrijving Badge	11
1.2	Transactie tabellen	14
1.2.1	Tabel beschrijving HostEvent, PendingAlert en Event	15
1.2.2	Uitleg belangrijke velden van de tabel HostEvent, PendingAlert en Event	16
1.2.3	Uitleg transacties van andere systemen importeren in de event tabel	17
1.3	Aanwezigheidslijst	18
1.3.1	Tabel beschrijving Inlist	18
1.4	Categorie counter	19
1.5	Server tabel	20
1.5.1	Kaarthouder gegevens aanpassen rechtstreeks via de server tabel	20
1.5.2	Direct besturen van deuren rechtstreeks via de server tabel	21
1.5.3	Direct besturen van relais rechtstreeks via de server tabel	21
1.6	Status tabel	21
1.6.1	Tabel beschrijving Status	22
1.6.2	Uitleg belangrijke velden van de status tabel	22
2.1	Aansturing van een externe plattegrond	23
3	API (Application Programming Interface) beschrijving	25
3.1	Mogelijkheden API	25
3.1.1	Configuratie Connection String	25
3.1.2	Badge Operation	25
3.1.3	Door Operation	26
3.1.4	Relay Operation	26
3.1.5	Reader Operation	27
3.1.6	Panel Operation	27
3.1.7	Inlist	28
4	Voorbeeld code	29
4.1.1	Configuratie Connection String	29
4.1.2	Badge Operation	30
4.1.3	Inlist Operation	30
4.1.4	Door Operation	31
4.1.5	Relay Operation	31
4.1.6	Reader Operation	32
4.1.7	Panel Operation	32
4.1.8	Input Operation	33
4.1.9	T&A Operation	33

Inleiding

Dit document beschrijft een aantal tabellen uit de CardAccess database die gebruikt kunnen worden om integraties met CardAccess te kunnen doen.

Daarnaast wordt de API (Application Programming Interface) beschreven)

Uitleg database

De door CardAccess toegepaste databases zijn SQL databases. Deze databases draaien op het Microsoft platform MS SQL 2005, MS SQL 2008 en MS SQL 2012. De volledige MS SQL versie of de MS SQL Express versie.

CardAccess creëert tijdens de installatie automatisch twee databases. De eerste is de zogenaamde Live database. Hierin bevinden zich alle configuratie instellingen zoals de panelen (centrales), de lezers (deuren), de tijdzones, de toegangsgroepen, de kaarthouders, de transacties etc.

Naast de zogenaamde Live database zal bij de installatie een tweede database worden aangemaakt, dit is de zogenaamde archief database.

Deze tweede database wordt gebruikt om transacties naartoe te archiveren vanuit de live database. Wanneer de applicatie wordt opgestart, en vervolgens op elk volle uur, wordt er in de Live database gekeken hoeveel transacties er zijn, alle transacties boven de ingestelde waarde (de oudste) zullen dan automatisch gearchiveerd worden naar de transactie database. Eventueel kunnen ook handmatig via de software alle configuratie instellingen vanuit de Live database naar de archief database gearchiveerd worden. Het archiveren van de Configuratie instellingen is een eenvoudige back-up mogelijkheid die we kunnen gebruiken om de laatste wijzigingen die gedaan zijn in de Live database heel eenvoudig ongedaan te kunnen maken.

Standaard blijven er 10.000 transacties in de live database staan. Oudere transacties zullen steeds naar de archief database gearchiveerd worden. Deze waarde is instelbaar via het menu systeem instellingen binnen de applicatie. Indien de archief database een bepaalde grootte heeft bereikt zal het systeem automatisch een nieuwe archief database aanmaken. Deze zal dan vervolgens automatisch gebruikt worden voor de archivering van transacties. Deze nieuwe archief database en de oude archief database worden beide benaderd bij zoek opdrachten (Historie). Via het menu systeem instellingen kan een oude database worden los gekoppeld waarna deze verplaatst kan worden naar een ander medium zoals bijvoorbeeld een Cd-rom of DVD.

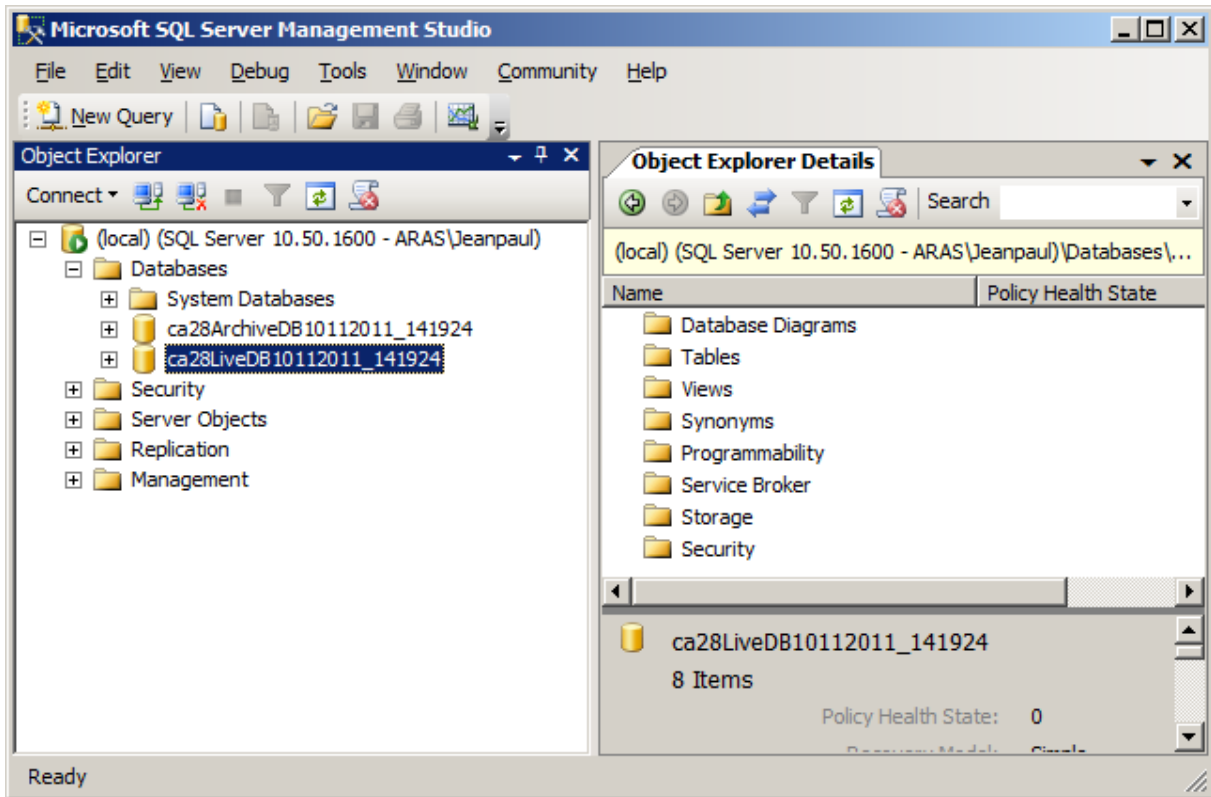
API

Mits voorzien van de juiste licentie is in de CA3000 software een API driver beschikbaar. Dit is een dll bestand wat zich in de CardAccess3000 map bevindt. Met behulp van deze API kunnen andere applicaties CardAccess functionaliteiten integreren.

In deze handleiding staat een voorbeeld programma waarin alle beschikbare functies van de API.dll zijn geïntegreerd. Daarnaast staat er voorbeeld code in deze handleiding.

1 Database beschrijving CA3000

Onderstaand scherm toont de beide databases die tijdens de installatie zijn aangemaakt.



Beide databases bevatten dezelfde tabellen. Indien men gegevens gaat invoeren zullen deze in de zogenaamde Live database terecht komen.





























































































Er zijn meerdere manieren om een koppeling te maken met CA3000. Verder stelt CardAccess een API driver beschikbaar waarmee de meeste functies zijn aan te sturen en te bedienen.















































Daarnaast is er een import export programma beschikbaar waarmee transacties automatisch geëxporteerd kunnen worden en kaarthouder gegevens automatisch geïmporteerd kunnen worden.

Het is tevens mogelijk om bepaalde zaken zoals bijvoorbeeld kaarthouder informatie rechtstreeks te wijzigen in de database via ODBC.

De volgende schermafdruk toont de tabellen binnen de Live database.

- [-] Tables
 - + System Tables
 - + dbo.AccGrp
 - + dbo.AccGrp_Badges
 - + dbo.AckAllPending
 - + dbo.ActiveLinks
 - + dbo.alarmLink_ScriptLinks
 - + dbo.alarmLink_scripts
 - + dbo.AlertSounds
 - + dbo.ALPEventTypes
 - + dbo.APBAreas
 - + dbo.APBTreeDetail
 - + dbo.APBTreeMaster
 - + dbo.ArchiveSettings
 - + dbo.AstaLog
 - + dbo.Badge
 - + dbo.BadgePhoto
 - + dbo.badgingPackages
 - + dbo.BuildInfo
 - + dbo.CategoryCounters
 - + dbo.Cctv
 - + dbo.CentralStationAutho
 - + dbo.CentralStationEvent
 - + dbo.CentralStationLinks
 - + dbo.CentralStationServer
 - + dbo.Client
 - + dbo.Com
 - + dbo.Commun
 - + dbo.CompanyMasterList
 - + dbo.DatabaseList
 - + dbo.DBAudit
 - + dbo.Delete_Badges
 - + dbo.DeviceIcon
 - + dbo.DvrCamera
 - + dbo.DvrConfig
 - + dbo.DvrInput
 - + dbo.DvrReader
 - + dbo.dynMap
 - + dbo.dynMapPoints
 - + dbo.ElevAg
 - + dbo.EPI_CONFIG
 - + dbo.EPI_DBCONFIG
 - + dbo.EPI_FORMAT
 - + dbo.epiFiles
 - + dbo.Event
 - + dbo.EventPhotos

		dbo.EventType
		dbo.Facility
		dbo.Format
		dbo.GA_ALIAS
		dbo.GA_COLNAMES
		dbo.GA_FK
		dbo.GA_IMAGE
		dbo.GA_IMAGEINFO
		dbo.GA_LAYOUTFIELDS
		dbo.GA_LIST
		dbo.GA_LISTDATA
		dbo.GA_LOG
		dbo.GA_LOGARCHIVE
		dbo.GA_PMK
		dbo.GA_READER
		dbo.GA_TABLENAMES
		dbo.GatewayStatus
		dbo.globalVariables
		dbo.Groups
		dbo.Holiday
		dbo.HostBadgeUpdates
		dbo.HostEvent
		dbo.InList
		dbo.Input
		dbo.Link
		dbo.LockoutAreas
		dbo.LockStatus
		dbo.Log
		dbo.MAccGrp
		dbo.MActiveLinks
		dbo.Mail
		dbo.Map
		dbo.MapApb
		dbo.MapEventFilter
		dbo.MapPoints
		dbo.MCom
		dbo.messageTemplates
		dbo.MHoliday
		dbo.MLockoutAreas
		dbo.mobileDevices
		dbo.MTimezon
		dbo.napcoAlarmAreaTypeDef
		dbo.napcoAlarmPanelDef
		dbo.napcoConnectionTypes
		dbo.napcoDeviceStatus
		dbo.napcoDeviceStatusDefs

- +  `dbo.napcoEventDef`
- +  `dbo.napcoMPermissions`
- +  `dbo.napcoPanelAreaEventDef`
- +  `dbo.napcoPanelAreas`
- +  `dbo.napcoPanelEventDef`
- +  `dbo.napcoPanelRelayEventDef`
- +  `dbo.napcoPanelRelays`
- +  `dbo.napcoPanels`
- +  `dbo.napcoPanelUsers`
- +  `dbo.napcoPanelZoneEventDef`
- +  `dbo.napcoPanelZones`
- +  `dbo.napcoPermissions`
- +  `dbo.napcoServerCommandQueue`
- +  `dbo.napcoWorkstations`
- +  `dbo.Nodes`
- +  `dbo.Oper`
- +  `dbo.operatorInstructionLinks`
- +  `dbo.operatorInstructions`
- +  `dbo.OperFilters`
- +  `dbo.Panel`
- +  `dbo.Panel_Types`
- +  `dbo.PendingAlert`
- +  `dbo.Priv_Det`
- +  `dbo.Privileg`
- +  `dbo.Reader`
- +  `dbo.Relay`
- +  `dbo.Resp`
- +  `dbo.SchedEvent`
- +  `dbo.scriptCounters`
- +  `dbo.scriptingServers`
- +  `dbo.scriptWizard_Types`
- +  `dbo.scriptWizard_TypeStatus`
- +  `dbo.Server`
- +  `dbo.snppProviders`
- +  `dbo.Sphere`
- +  `dbo.Status`
- +  `dbo.System`
- +  `dbo.TARpt0D`
- +  `dbo.Template`
- +  `dbo.Timezone`
- +  `dbo.tmpAckEvents`
- +  `dbo.tmpBadge`
- +  `dbo.tmpEvent`
- +  `dbo.tmpEventPhotos`
- +  `dbo.Wireless`
- +  `dbo.Zones`

2 Tabelomschrijvingen

Hieronder staat de omschrijving per tabel. In deze lijst zijn alleen die tabellen opgenomen die voor integratiedoeleinden een toepassing hebben.

Name Table	Description Table
AccGrp	Rechten toegangsgroepen
AccGrp_Badges	Speciale toegangsgroep per kaart
APBAreas	Zones
Badge	Kaarthouders
BadgePhoto	Kaarthouder foto
CategoryCounters	Tellerwaarden
Databaselist	Archief Database instellingen
DBAudit	Operator historie
Event	Alle transacties
PendingAlert	Alle alarmen die vervolgens naar de event table worden verplaatst
Facility	Project codes
Holiday	Vakantiedagen
HostEvent	Alarm transacties
Inlist	Aanwezige kaarthouders
Input	Apart geconfigureerde ingangen
MAccGrp	Namen Toegangsgroepen
MTimezon	Tijdzone benamingen
Oper	Operators (beheerders)
Panel	Panelen (Centrales)
Reader	Lezers (Deuren)
Relay	Aparte relais
Server	Communicatie server
Status	Toont real-time status van devices
System	Systeem instellingen
Timezone	Tijdzone instellingen
TmpEvent	Tijdelijke opslag voor transacties

1.1 Kaarhouder tabel

De belangrijkste tabel voor integratie toepassingen is de kaarhouder tabel genaamd "Badge".

1.1.1 Gegevens wijzigen via de kaarhouder tabel



De belangrijkste tabel voor integratie toepassingen is de kaarhouder tabel genaamd "Badge".

De tabel Badge bevat alle kaarhouder data. Indien we hier rechtstreeks iets in willen wijzigen dienen we er rekening mee te houden dat de gewijzigde gegevens niet altijd zomaar verzonden worden naar de panelen. Bij het wijzigen van een kaarhoudernaam is dit ook niet nodig maar bij het geldig maken van een pas of bij het aanpassen van de toegangsgroep is dit wel nodig.

Hoe zorg je er nu voor dat de wijzigingen worden doorgevoerd naar de panelen?

Als je bij het wijzigen van een kaarhouder ervoor zorgt dat het veld Flag op 1 wordt gezet dan zal het record automatisch geladen worden. Dit veld vertelt de Host communicatiedriver dat er een kaarhouder record is gewijzigd of is toegevoegd. Elke twee tot vijf minuten zal de Host communicatiedriver dit veld controleren en indien dit veld op 1 staat zal het record naar de panelen worden geladen.

In de volgende schermafdruck staan de kolom (veld) namen van deze tabel beschreven.

JPLT.ca26LiveD...57 - dbo.Badge			
	Column Name	Data Type	Allow Nulls
	Facility	smallint	<input type="checkbox"/>
	Badge	bigint	<input type="checkbox"/>
	AGroup1	smallint	<input checked="" type="checkbox"/>
	AGroup2	smallint	<input checked="" type="checkbox"/>
	Pin	smallint	<input checked="" type="checkbox"/>
	Issue	smallint	<input checked="" type="checkbox"/>
	Enabled	bit	<input checked="" type="checkbox"/>
	Track	bit	<input checked="" type="checkbox"/>
	Resident	bit	<input checked="" type="checkbox"/>
	InitLoad	bit	<input checked="" type="checkbox"/>
	Shunt	bit	<input checked="" type="checkbox"/>
	DurUse	bit	<input checked="" type="checkbox"/>
	Escort	bit	<input checked="" type="checkbox"/>
	APBSet	smallint	<input checked="" type="checkbox"/>
	AccessTime	smallint	<input checked="" type="checkbox"/>
	ActvDate	datetime	<input checked="" type="checkbox"/>
	ExprDate	datetime	<input checked="" type="checkbox"/>
	GroupNo	smallint	<input checked="" type="checkbox"/>
	Embossed	int	<input checked="" type="checkbox"/>
	LastName	varchar(25)	<input checked="" type="checkbox"/>
	FrstName	varchar(25)	<input checked="" type="checkbox"/>
	Dept	varchar(50)	<input checked="" type="checkbox"/>
	Phone	varchar(30)	<input checked="" type="checkbox"/>
	Location	varchar(55)	<input checked="" type="checkbox"/>
	Vehicle	varchar(17)	<input checked="" type="checkbox"/>
	License	varchar(11)	<input checked="" type="checkbox"/>
	Contact	varchar(50)	<input checked="" type="checkbox"/>
	ContPhone	varchar(30)	<input checked="" type="checkbox"/>
	SSN	varchar(12)	<input checked="" type="checkbox"/>

JPLT.ca26LiveD...57 - dbo.Badge		
Column Name	Data Type	Allow Nulls
CoID	varchar(16)	<input checked="" type="checkbox"/>
Changed	datetime	<input checked="" type="checkbox"/>
Flag	bit	<input checked="" type="checkbox"/>
AlarmShunt	smallint	<input checked="" type="checkbox"/>
Remarks	text	<input checked="" type="checkbox"/>
BBPhotoFile	varchar(40)	<input checked="" type="checkbox"/>
BBChanged	varchar(25)	<input checked="" type="checkbox"/>
PointAccess	bit	<input checked="" type="checkbox"/>
SDate	datetime	<input checked="" type="checkbox"/>
UserField1	varchar(20)	<input checked="" type="checkbox"/>
UserField2	varchar(50)	<input checked="" type="checkbox"/>
UserField3	varchar(20)	<input checked="" type="checkbox"/>
UserField4	varchar(20)	<input checked="" type="checkbox"/>
UserField5	float	<input checked="" type="checkbox"/>
UserField6	float	<input checked="" type="checkbox"/>
UserField7	float	<input checked="" type="checkbox"/>
UserField8	float	<input checked="" type="checkbox"/>
UserField9	varchar(255)	<input checked="" type="checkbox"/>
UserField10	varchar(255)	<input checked="" type="checkbox"/>
UserField11	varchar(255)	<input checked="" type="checkbox"/>
UserField12	text	<input checked="" type="checkbox"/>
UseCount	smallint	<input checked="" type="checkbox"/>
AG1Date	datetime	<input checked="" type="checkbox"/>
AG2Date	datetime	<input checked="" type="checkbox"/>
LVPnl	smallint	<input checked="" type="checkbox"/>
LVRdr	smallint	<input checked="" type="checkbox"/>
LVDDate	datetime	<input checked="" type="checkbox"/>
BBTemplate	int	<input checked="" type="checkbox"/>
Gender	varchar(1)	<input checked="" type="checkbox"/>

JPLT.ca26LiveD...57 - dbo.Badge		
Column Name	Data Type	Allow Nulls
DateOfBirth	smalldatetime	<input checked="" type="checkbox"/>
MiddleName	varchar(30)	<input checked="" type="checkbox"/>
UserField13	varchar(50)	<input checked="" type="checkbox"/>
UserField14	varchar(50)	<input checked="" type="checkbox"/>
UserField15	varchar(50)	<input checked="" type="checkbox"/>
UserField16	varchar(50)	<input checked="" type="checkbox"/>
UserField17	varchar(50)	<input checked="" type="checkbox"/>
UserField18	varchar(50)	<input checked="" type="checkbox"/>
UserField19	varchar(50)	<input checked="" type="checkbox"/>
UserField20	varchar(50)	<input checked="" type="checkbox"/>
UserField21	varchar(50)	<input checked="" type="checkbox"/>
UserField22	varchar(50)	<input checked="" type="checkbox"/>
UserField23	varchar(50)	<input checked="" type="checkbox"/>
UserField24	varchar(50)	<input checked="" type="checkbox"/>
UserField25	varchar(50)	<input checked="" type="checkbox"/>
UserField26	varchar(50)	<input checked="" type="checkbox"/>
UserField27	varchar(50)	<input checked="" type="checkbox"/>
UserField28	varchar(50)	<input checked="" type="checkbox"/>
UserField29	varchar(50)	<input checked="" type="checkbox"/>
UserField30	varchar(50)	<input checked="" type="checkbox"/>
UserField31	varchar(50)	<input checked="" type="checkbox"/>
UserField32	varchar(50)	<input checked="" type="checkbox"/>
UserField33	varchar(50)	<input checked="" type="checkbox"/>
UserField34	varchar(50)	<input checked="" type="checkbox"/>
UserField35	varchar(50)	<input checked="" type="checkbox"/>
UserField36	varchar(50)	<input checked="" type="checkbox"/>
UserField37	varchar(50)	<input checked="" type="checkbox"/>
UserField38	varchar(50)	<input checked="" type="checkbox"/>
UserField39	varchar(50)	<input checked="" type="checkbox"/>

JPLT.ca26LiveD...57 - dbo.Badge			
Column Name	Data Type	Allow Nulls	
UserField40	varchar(50)		<input checked="" type="checkbox"/>
UserField41	varchar(50)		<input checked="" type="checkbox"/>
UserField42	varchar(50)		<input checked="" type="checkbox"/>
UserField43	varchar(50)		<input checked="" type="checkbox"/>
UserField44	varchar(50)		<input checked="" type="checkbox"/>
UserField45	varchar(50)		<input checked="" type="checkbox"/>
UserField46	varchar(50)		<input checked="" type="checkbox"/>
UserField47	varchar(50)		<input checked="" type="checkbox"/>
UserField48	varchar(50)		<input checked="" type="checkbox"/>
AGroup3	smallint		<input checked="" type="checkbox"/>
AGroup4	smallint		<input checked="" type="checkbox"/>
AGroup5	smallint		<input checked="" type="checkbox"/>
AGroup6	smallint		<input checked="" type="checkbox"/>
AG3Date	datetime		<input checked="" type="checkbox"/>
AG4Date	datetime		<input checked="" type="checkbox"/>
AG5Date	datetime		<input checked="" type="checkbox"/>
AG6Date	datetime		<input checked="" type="checkbox"/>
BadgeID	varchar(50)		<input checked="" type="checkbox"/>
EPI_FORMAT_ID	varchar(10)		<input checked="" type="checkbox"/>
APBArea	smallint		<input checked="" type="checkbox"/>
caObjectID	uniqueidentifier		<input type="checkbox"/>
HolidayCalendar	smallint		<input checked="" type="checkbox"/>
NAPCOPermissionID	int		<input checked="" type="checkbox"/>
MApbExempt	bit		<input checked="" type="checkbox"/>
Category1	bit		<input checked="" type="checkbox"/>
Category2	bit		<input checked="" type="checkbox"/>
Category3	bit		<input checked="" type="checkbox"/>
Category4	bit		<input checked="" type="checkbox"/>
Category5	bit		<input checked="" type="checkbox"/>

Category6	bit	<input checked="" type="checkbox"/>
Category7	bit	<input checked="" type="checkbox"/>
Category8	bit	<input checked="" type="checkbox"/>
Category9	bit	<input checked="" type="checkbox"/>
Category10	bit	<input checked="" type="checkbox"/>
Category11	bit	<input checked="" type="checkbox"/>
Category12	bit	<input checked="" type="checkbox"/>
Category13	bit	<input checked="" type="checkbox"/>
Category14	bit	<input checked="" type="checkbox"/>
Category15	bit	<input checked="" type="checkbox"/>
Category16	bit	<input checked="" type="checkbox"/>
AgencyCode	int	<input checked="" type="checkbox"/>
Series	smallint	<input checked="" type="checkbox"/>
PersonId	bigint	<input checked="" type="checkbox"/>
OrgCat	smallint	<input checked="" type="checkbox"/>
OrgId	int	<input checked="" type="checkbox"/>
OrgAssoc	smallint	<input checked="" type="checkbox"/>
		<input type="checkbox"/>
FirstIn	bit	<input checked="" type="checkbox"/>
ExcludeLockdown	bit	<input checked="" type="checkbox"/>
LastEditDate	datetime	<input checked="" type="checkbox"/>
CreationDate	datetime	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

1.1.2 Tabel beschrijving Badge

In de volgende tabel worden de velden van de kaarthouder (Badge) tabel beschreven.

Column Name	Dutch Translation	Description	Flag veld benodigd
Facility	Project code nummer	Dit nummer in combinatie met het kaartnummer maakt de kaart uniek	X
Badge	Kaartnummer	Kaartnummer van de kaart	X
AGroup1	Toegangsgroep 1 nummer	Bepaald de kaarthouderrechten. (Deuren)	X
AGroup2	Toegangsgroep 2 nummer	Idem AGroup1	X
AGroup3	Toegangsgroep 3 nummer	Idem AGroup1	X
AGroup4	Toegangsgroep 4 nummer	Idem AGroup1	X
AGroup5	Toegangsgroep 5 nummer	Idem AGroup1	X
AGroup6	Toegangsgroep 6 nummer	Idem AGroup1	X
PIN	PIN nummer	PIN code veld	X
Issue	Uitgave nummer	Geeft aan hoeveel kaarten met dit kaartnummer en Facility nummer er reeds	X

		zijn uitgegeven. Standaard is 0	
Enabled	Geldig	Kaart geldig als veld=1	X
Track	Gevolgde kaart	Kaarthouder die we willen volgen binnen het TC	
Resident		Kaart laden naar paneel	
InitLoad		Kaart bij download laden	
Shunt	Alarm schakelen	Ja=1	X
DurUse		Herhaal kaart gebruik	
Escort	Bezoekerskaart	Kaart die geëscorteerd moet worden	
APBSet	APB locatie	IN,UIT,nvt,Geen	X
AccessTime	Deurtijd	Apart per kaart in te stellen indien gewenst	X
ActvDate	Activeer datum	Datum Kaart Geldig	X
ExprDate	Kaart deactiveren	Datum Kaart deactiveren	X
GroupNo	Groepnummer	Administratieve groep waarin de kaarthouder eventueel thuis hoort	
Embossed	Opgedrukt nummer	Nummer wat op de kaart staat indien dit anders is als in de kaart	
Lastname	Achternaam	Achternaam kaarthouder	
FrstName	Voornaam	Voornaam kaarthouder	
Dept	Afdeling	Afdeling kaarthouder	
Phone	Telefoonnummer	Telefoonnummer waaronder kaarthouder is te bereiken	
Location	Locatie	Locatie waar kaarthouder werkzaam is	
Vehicle	Voertuig	Voertuig van kaarthouder	
License	Kenteken	Kentekenplaatnummer	
Contact	Contact	Contactpersoon kaarthouder	
ContactPhone		Tel. contactpersoon	
SSN	Social Security Number	Sofie nummer	
CoID	Contact ID	Contact ID nummer	
Changed	Datum record gewijzigd	Datum waarop het kaarthouder record is gewijzigd	
Flag	Mutatieveld	Dit veld geeft aan dat de kaarthouder is gewijzigd en dat de kaart naar het paneel geladen dient te worden	NVT
AlarmShunt		NVT	
Remarks	Opmerkingen	Opmerkingen die betrekking hebben op de kaarthouder	
BBPhotoFile	Foto ID nummer	Foto ID	
PointAccess		NVT	
Sdate		NVT	
Userfield1	Extra veld 1	Vrij te definiëren veld 1	
Userfield 2-48	Extra veld 2-48	Vrij te definiëren veld 2-48	
UseCount	Gebruikers teller	Hoe vaak de kaart gebruikt mag worden	
AG1Date	Verval Datum Toegangsgroep 1	Datum waarna Toegangsgroep 1 vervalt	
AG2Date	Verval Datum Toegangsgroep 2	Datum waarna Toegangsgroep 2 vervalt	

AG3Date	Verval Datum Toegangsgroep 3	Datum waarna Toegangsgroep 3 vervalt	
AG4Date	Verval Datum Toegangsgroep 4	Datum waarna Toegangsgroep 4 vervalt	
AG5Date	Verval Datum Toegangsgroep 5	Datum waarna Toegangsgroep 5 vervalt	
AG6Date	Verval Datum Toegangsgroep 6	Datum waarna Toegangsgroep 6 vervalt	
LVPnl	Laatste paneel	Laatst paneel waarop de laatste transactie plaats vond	
LVRdr	Laatste lezer	Laatst lezer waarop de laatste transactie plaats vond	
LVDate	Laatste Datum	Datum waarop de laatste transactie plaats vond	
BBTemplate	Template kaarthouder	Template waarop de kaarthouder afgedrukt kan worden	
Gender	Geslacht	Geslacht kaarthouder	
DateOfBirth	Geboorte datum	Geboorte datum kaarthouder	
Middlename	Toevoeging naam	Voorbeeld: George W. Bush. De W is de middle name.	
BadgeID		NVT	
EPI_FORMAT_ID		NVT	
APBArea	APB zone	Geeft aan in welke zone de kaarthouder zich bevindt	
caObjectID		NVT	
HolidayCalendar	Vakantie kalender	Geeft weer aan welke kalender deze kaart is gekoppeld	
NAPCOPermissionID	Alarm permissie	Alarm permissie	
MApbExempt		NVT	
Category1	Categorie 1	Geeft weer aan welke categorie deze kaart is gekoppeld	X
Category216	Categorie 2....16	Idem Category1	X
AgencyCode		NVT	
Series		NVT	
PersonId		NVT	
OrgCat		NVT	
OrgId		NVT	
OrgAssoc		NVT	

Belangrijke velden binnen de kaarthouder tabel zijn het kaartnummer, de toegangsgroep(en), het kaart geldig veld en het zogenaamde Flag veld.

Indien er mutaties in het kaarthouder record plaatsvinden die betrekking hebben op de toegangsrechten. Zoals het wijzigen van de kaart van geldig naar ongeldig of andersom dan dient de flag op 1 te worden gezet. Door het flag veld op 1 te zetten weet de communicatiedriver dat de kaart is gewijzigd en dat er een mutatie naar 1 of meerdere panelen verzonden dient te worden. Indien de toegangsgroep(en) van het kaarthouder record word(en) gewijzigd zal men ook de flag op 1 moeten zetten zodat de communicatiedriver weet dat de er wijzigingen hebben plaatsgevonden die van belang zijn voor de paneel data.

Velden als voor en achternaam kunnen worden gewijzigd zonder de flag op 1 te zetten omdat dit geen consequenties heeft voor de paneel data. Dit omdat dit soort gegevens niet in de panelen worden geladen.

De velden die gemerkt zijn met de X zijn de velden waarbij het flag veld op 1 gezet dient te worden bij een mutatie.

1.2 Transactie tabellen

In de transactie tabel komen alle transacties te staan. Onder transacties verstaan we kaarttransacties zoals kaart geweigerd op deze tijd, kaart geldig, kaart IN, kaart UIT, kaart Klok IN, paneel laden gestart, Ingang abnormaal, Uitgang Op, Deur open, Deur te lang open etc..etc.

We kunnen onder systeem instellingen een drempelwaarde (prioriteit) instellen. Alle transacties boven deze drempelwaarde (prioriteit) zullen niet in het alarm scherm terecht komen maar worden rechtstreeks naar de transactielijst verplaatst. Zoals we hieronder kunnen zien is het bovenste gedeelte van het scherm het transactiescherm en het onderste gedeelte het alarm scherm.


Foto	Klasse	Omschrijving	Locatie	Datum	Operator	Bevestigd
	Kaart geldig	Faber, Olivier	2-1 2-1	9-11-2012 16:19:02	Auto bevestigd	9-11-2012 16:19
	Paneel fout	Geen verbinding	02 Microtem	9-11-2012 16:19:24	Bevestigd	9-11-2012 16:19
	Paneel fout	Nu verbinding	02 Microtem	9-11-2012 16:19:55	Bevestigd	9-11-2012 16:20
	Paneel data	Laden gestart door paneel	02 Microtem	9-11-2012 16:19:55	Bevestigd	9-11-2012 16:20
	Paneel data	Laden gestart	02 Microtem	9-11-2012 16:19:56	Bevestigd	9-11-2012 16:20
	Paneel data	Laden compleet	02 Microtem	9-11-2012 16:20:04	Bevestigd	9-11-2012 16:20
	Kaart geweigerd op deze tijd	Faber, Olivier	2-1 2-1	9-11-2012 16:27:56	Bevestigd	9-11-2012 16:28
	Kaart geweigerd op deze tijd	Faber, Olivier	2-1 2-1	9-11-2012 16:27:59	Admin	9-11-2012 16:28
	Kaart geweigerd op deze tijd	Faber, Olivier	2-1 2-1	9-11-2012 16:28:00	Admin	9-11-2012 16:28
	Kaart geweigerd op deze tijd	Faber, Olivier	2-1 2-1	9-11-2012 16:28:01	Admin	9-11-2012 16:28
	Paneel data	Download Operator Request	02 Microtem	9-11-2012 16:28:16	Admin	9-11-2012 16:28
	Paneel data	Laden gestart	02 Microtem	9-11-2012 16:28:16	Admin	9-11-2012 16:28
	Paneel data	Laden compleet	02 Microtem	9-11-2012 16:28:25	Bevestigd	9-11-2012 16:28
	Kaart geldig	Faber, Olivier	2-1 2-1	9-11-2012 16:28:35	Auto bevestigd	9-11-2012 16:28
	Kaart geldig	Faber, Olivier	2-1 2-1	9-11-2012 16:28:41	Auto bevestigd	9-11-2012 16:28
	Kaart geldig	Faber, Olivier	2-1 2-1	9-11-2012 16:28:43	Auto bevestigd	9-11-2012 16:28
	Kaart geldig	Faber, Olivier	2-1 2-1	9-11-2012 16:28:45	Auto bevestigd	9-11-2012 16:28
	Kaart geldig	Faber, Olivier	2-1 2-1	9-11-2012 16:28:45	Auto bevestigd	9-11-2012 16:28
	Kaart geldig	Faber, Olivier	2-1 2-1	9-11-2012 16:28:47	Auto bevestigd	9-11-2012 16:28
	Paneel fout	Geen verbinding	03 Super-2	9-11-2012 17:06:37	Admin	12-11-2012 13:2
	Paneel fout	Geen verbinding	02 Microtem	9-11-2012 17:06:42	Admin	12-11-2012 13:2
	Video	DVR Recording Service Started	VNTC03-029	12-11-2012 13:27:49	Admin	12-11-2012 13:2
	Systeem fout	Poling Startup	VNTC03-029	12-11-2012 13:27:54	Admin	12-11-2012 13:2
	Paneel fout	Geen verbinding	02 Microtem	12-11-2012 13:28:03	Admin	12-11-2012 13:2
	Systeem fout	Wireless Socket Server Startup	VNTC03-029	12-11-2012 13:28:59	Bevestigd	12-11-2012 13:2
	Kaart geldig	test, jp	3-1 P3 Lezer 1 in	12-11-2012 13:31:10	Auto bevestigd	12-11-2012 13:3
	Kaart geldig	test, jp	4-2 P4 Lezer 2	12-11-2012 13:31:24	Auto bevestigd	12-11-2012 13:3
	Kaart geldig	test, jp	4-2 P4 Lezer 2	12-11-2012 13:31:24	Auto bevestigd	12-11-2012 13:3
	Kaart geldig	test, jp	4-2 P4 Lezer 2	12-11-2012 13:31:26	Auto bevestigd	12-11-2012 13:3
	Kaart geldig	test, jp	4-2 P4 Lezer 2	12-11-2012 13:31:28	Auto bevestigd	12-11-2012 13:3
	Kaart geldig	test, jp	4-2 P4 Lezer 2	12-11-2012 13:32:10	Auto bevestigd	12-11-2012 13:3

Klasse	Omschrijving	Locatie
Ingang abnormaal	4-5 Input 4-5	04 Super-2
Paneel data	Download Operator Request	04 Super-2
Paneel data	Laden gestart	04 Super-2

Technisch gezien komen alle transacties in de database binnen in de tabel HostEvent. Als het een alarm betreft wordt deze verplaatst naar het alarm scherm. Dit is een aparte tabel genaamd PendingAlert. Vervolgens gaan de transacties na een instelbare tijd naar het transactiescherm. Dit is een aparte tabel genaamd Event.

Transacties die binnen komen die geen alarm zijn gaan direct van de HostEvent tabel naar de Event tabel en slaan de PendingAlert tabel over.

Alle drie de tabellen HostEvent, PendingAlert en Event zijn hetzelfde wat betreft de opbouw. In de volgen de tabel worden de kolom (veld) namen van deze tabel getoond.

JPLT.ca26LiveD...57 - dbo.Event			Summary
Column Name	Data Type	Allow Nulls	
 Seqno	uniqueidentifier	<input type="checkbox"/>	
Priority	smallint	<input checked="" type="checkbox"/>	
Cat	smallint	<input checked="" type="checkbox"/>	
PnlNo	smallint	<input checked="" type="checkbox"/>	
EDate	datetime	<input checked="" type="checkbox"/>	
Number	smallint	<input checked="" type="checkbox"/>	
Status	smallint	<input checked="" type="checkbox"/>	
Facno	smallint	<input checked="" type="checkbox"/>	
Badge	bigint	<input checked="" type="checkbox"/>	
Class	varchar(48)	<input checked="" type="checkbox"/>	
Description	varchar(64)	<input checked="" type="checkbox"/>	
Name	varchar(48)	<input checked="" type="checkbox"/>	
Arch	smallint	<input checked="" type="checkbox"/>	
AckOpr	smallint	<input checked="" type="checkbox"/>	
AckTStamp	datetime	<input checked="" type="checkbox"/>	
Actions	text	<input checked="" type="checkbox"/>	
RespReq	bit	<input checked="" type="checkbox"/>	
Sphere	int	<input checked="" type="checkbox"/>	
Tag	int	<input checked="" type="checkbox"/>	
Pending	bit	<input checked="" type="checkbox"/>	
HasPhoto	bit	<input type="checkbox"/>	
UTCOffset	int	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

1.2.1 Tabel beschrijving HostEvent, PendingAlert en Event

In de volgende tabel worden de velden van deze tabel beschreven.

Column Name	Dutch Translation	Description
Seqno	Sequence nummer	Uniek record nummer
Priority	Prioriteit transactie	1 is belangrijk 99 onbelangrijk
Cat	Categorie	
PnlNo	Paneel nummer	Paneel waarop transactie plaats vond
EDate	Transactie datum	Datum waarop transactie plaats vond
Number	Nummer	Lezer, ingang, uitgang waarop transactie plaats vond
Status	Status veld	Toond status transactie
Facno	Project code nummer	
Badge	Kaartnummer	
Class	Typetransactie	Type ligt aan soort/type melding
Description	Omschrijving	Omschrijving transactie
Name	Naam	Naam kaarthouder
Arch	Bevestiging	Of de transactie door een operator is bevestigd. indien dit veld leeg is. is de transactie automatisch bevestigd.

AckOpr	Bevestigd door Operator	Geeft aan door welke operator de transactie is bevestigd
AckTStamp	Datum bevestiging	Geeft de datum weer waarop de transactie is bevestigd
Actions	Actie	Wat de actie van de operator is geweest
RespReq	respons vereist	Of er een respons vereist was
Sphere		NVT
Tag	Extra sub status veld	
Pending	Wachtend op bevestiging	Transactie wacht op bevestiging
HasPhoto		NVT
UTCOffset		Tijd in minute voor of na UTC tijd

1.2.2 Uitleg belangrijke velden van de tabel HostEvent, PendingAlert en Event

Cat = Type event
 Type 0 = badge
 1 = input
 2 = output
 4 = panel
 5 = reader

PnlNo = Panel number
 Number = Device number (reader, input, output number)

Status by badge (Cat 0) event, number ((1) 1=reader 1)

Status 0 = Kaart geldig (badge valid)
 Status 11 = Kaart ongeldig (badge violate void)
 Status 13 = Kaart ongeldig project code (badge violate facility)
 Status 2 = Kaart IN (badge Entry)
 Status 24 = Kaart UIT (badge Exit)
 Status 6 = kaart klok IN (Badge Clock-IN)
 Status 8 = kaart klok UIT (Badge Clock-OUT)
 Status 16 = Kaart ongeldig APB (Badge violate APB)

Status by input (cat 1) event, number ((1) 1=input 1)

Status 0 = Ingang normaal (input normaal)
 Status 1 = Ingang abnormaal (input abnormaal)

Status by output (cat 2) event, number ((1) 1=output 1)

Status 0 = Uitgang af (Output Off)
 Status 1 = uitgang op (Output On)

Status by panel (Cat 4) event, number (always 0)

Status 3 = System event
 Status 4 = Downloading firmware
 Status 5 = Download complete
 Status 7 = No response / now responding

We hebben de leverancier gevraagd status 7 op te splitsen in:

Status 7, Tag 1= No response
 Status 7, Tag 0= Now responding

Dit zal in een volgende versie geïmplementeerd worden.

Status by reader (cat 5) event, number ((1) 1=lezer1)

Status 1, Tag 1= Deur geforceerd (door forced open)
 Status 1, Tag 0= Deur nu gesloten (door now closed)
 Status 3 = Deur is te lang open (Door is open to long)
 Status 6 = Deur handmatig geopend (Door manual Unlock)
 Status 2 = Deur in vrije tijdzone (Door Free Access)

1.2.3 Uitleg transacties van andere systemen importeren in de event tabel.

Vanaf CA3000 versie 2.11.72 kunnen we transacties van andere systemen tonen in het CA3000 transactiescherm.

Hiervoor dienen we onderstaande Stored Procedure te draaien die een transactie kan toevoegen (insert) in de event tabel.

Vervolgens gebruiken we Command #60 (Zie hieronder) in the server tabel (gebruikmakend van "seqno" uitgang van de Stored Procedure) als een nieuw veld in the server tabel om de transactie te tonen.

Steps to broadcast Event to Event table:

- 1) Execute **stored procedure** (ca_sp_Asta_AddEvent.sp). Pass all the following values into this stored procedure.

Example1 - exec ca_sp_ASTA_AddEvent 21,4,0,@date,0,3,0,0,'SYSTEM','PIVDETECT',
 'PIV',0,2015,'1',1,0,0,0,0,@seqno output

SrvID	Record	Val1	Val2	Val3	Val4	Val5	UTCExecTime	GUID	Time
	60	1	0	0	0	0	0	seqno	NULL

New #60
Command

New field in the Server table to pass Guid
of Event record

1.3 Aanwezigheidslijst

In de tabel genaamd Inlist komen alle aanwezige kaarthouders te staan. De getoonde lijst kaarthouders wordt geüpdate als kaarthouders van de IN naar de UIT status veranderen en omgekeerd.

In de volgen de tabel worden de kolom (veld) namen van deze tabel getoond.

JPLT.ca26LiveD...57 - dbo.InList			
	Column Name	Data Type	Allow Nulls
▶	Name	varchar(64)	<input type="checkbox"/>
🔑	Badge	bigint	<input type="checkbox"/>
	Reader	varchar(64)	<input checked="" type="checkbox"/>
	PanelTime	datetime	<input checked="" type="checkbox"/>
	Sphere	int	<input checked="" type="checkbox"/>
	PnlNo	smallint	<input checked="" type="checkbox"/>
	RdrNo	smallint	<input checked="" type="checkbox"/>
🔑	Facility	smallint	<input type="checkbox"/>
	APBArea	smallint	<input checked="" type="checkbox"/>
	APBIn	bit	<input checked="" type="checkbox"/>
	TAIn	bit	<input checked="" type="checkbox"/>
	UTCOffset	int	<input checked="" type="checkbox"/>
	ZoneId	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

1.3.1 Tabel beschrijving Inlist

In de volgende tabel worden de velden van deze tabel beschreven.

Column Name	Dutch Translation	Description
Name	Naam	Naam Kaarthouder
Badge	Kaartnummer	Kaartnummer
Reader	Lezer	Lezer waarop de transactie plaats vond
PanelTime	Paneel tijd	Tijd wanneer de transactie plaats vond
Sphere		NVT
PnlNo	Paneel nummer	Paneel nummer waarop de transactie plaats vond
RdrNo	Lezer nummer	Lezer nummer waarop de transactie plaats vond
Facility	Project code	Project code kaart
APBArea	APB zone	Zone waarin de kaart zich bevindt
APBIn	APB IN	Was het een APB IN transactie
TAIn	Klok IN	Was het een Klok IN transactie
UTCOffset		NVT
ZoneId		Zone nummer waarin de kaart zich bevindt

Indien we via de database de APB status van een kaarthouder aanpassen wordt deze nog niet in het paneel gewijzigd. Indien dit nodig is dienen we onderstaand commando hiervoor te gebruiken.

Om kaart wijzigingen naar een paneel te verzenden dienen we onderstaand commando in te voeren in de tabel Server DB. Gebruik hiervoor de volgende waarden:

Record = 16

(Badge Edit command)

Val1 = Facility number
Val2 = Badge number

Het database veld wat wordt gebruikt voor APB is "APBSet".

De waarden die in dit veld kunnen worden ingevuld zijn:

0 = IN
1 = OUT
2 = SET Next use
3 = Ignore APB checking

Als de kaart (Badge) is benoemd als een voertuig kaart , dan dienen we bit2 van dit APBSet veld te gebruiken.

Als voorbeeld, als we een voertuig kaart geen last van APB te laten ondervinden moeten we APBSet op 7 zetten.

Voor APB aanpassingen gebruikt de applicatie commando 16 (with Facility, badge parameters), deze wordt dan verwerkt als een normale kaart wijziging.

De In-list wordt bij een APB of Zone APB steeds ge update door een aantal SQL Update statements.

1.4 Categorie counter

Om handmatig een categorie op een bepaalde waarde in te stellen gebruiken we onderstaand commando. Dit commando plaatsen we in de Server DB. Zie hoofdstuk 1.5.

Record = 26
Val1 = Panel Number (0 = all Panels)
Val2 = Cat Counter number (0 = all 16 categories)
Val3 = Value to set (0 – 65000)
Val4 = condition to set
1 = set if current value is less than the new value
2 = set if current values is greater than the new value
3 = Unconditionally set the new value
4 = Reset count to zero (0),Val3 is ignored.

Voorbeeld:

```
case 0:                □- NOT USED
  status = STS_NEX;
  break;

  /* IF CATEGORY LESS THAN NEW VALUE */
case 1:
  flag = (value > NV.category[i].counter);
  break;

  /* IF CATEGORY GREATER THAN NEW VALUE */
case 2:
  flag = (NV.category[i].counter > value);
  break;

  /* UNCONDITIONAL SET COUNT */
case 3:
  flag = TRUE;
  break;

  /* RESET CAT COUNTER (NEW) */
case 4:
  NV.category[i].counter = 0;
  break;
}
```

1.5 Server tabel

In de Server tabel zet het systeem gegevens waarmee de communicatiedriver kan zien wat er geladen dient te worden naar de panelen.

In de volgen de tabel worden de kolom (veld) namen van deze tabel getoond.

JPLT.ca26LiveD...57 - dbo.Server			
	Column Name	Data Type	Allow Nulls
▶	SrvId	int	<input type="checkbox"/>
	Record	smallint	<input checked="" type="checkbox"/>
	Val1	int	<input checked="" type="checkbox"/>
	Val2	bigint	<input checked="" type="checkbox"/>
	Val3	int	<input checked="" type="checkbox"/>
	Val4	int	<input checked="" type="checkbox"/>
	Val5	int	<input checked="" type="checkbox"/>
	TimStamp	datetime	<input checked="" type="checkbox"/>
	UTCExecTime	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

In de volgende tabel worden de velden van deze tabel beschreven.
De werking van deze tabel is als volgt.

De CardAccess GUI weet precies wanneer hij een aangepast record in de server tabel moet plaatsen. Al er een record in de server tabel wordt gezet, dan wordt dit record binnen 8 seconden verwerkt.

1.5.1 Kaarthouder gegevens aanpassen rechtstreeks via de server tabel

Indien we snel een kaarthouder willen verwerken kunnen we ook direct gegevens naar de server tabel wegschrijven.

```
Record      =      16
  Val1      =      <facility number>
  Val2      =      <badge number>
  Val3      =      0      <_--- Panel number, 0 = send to all panels>
Timestamp  =      current time
```

LET OP!

Als VAL3 gevuld wordt met 0, dan wordt de kaart naar alle panelen wordt toegezonden.
Als je vervolgens bij Val2 kaartnummer 0 invoert dan worden alle kaarten gewist.

Met het onderstaande voorbeeld commando laden we kaart 1234 met project/facility code 1 naar alle panelen.

```
Insert into server (record, val1, val2, val3)
Values (16, 1, 1234, 0)
```

Om een kaart te verwijderen. Plaats het volgende commando in de server tabel.

Record = 7 // Delete command
 Val1 = 0 // 0 is a Badge delete
 Val2 = Badge Number
 Val3 = Facility number

1.5.2 Direct besturen van deuren rechtstreeks via de server tabel

Onderstaande opdrachten zijn te gebruiken om deuren en relais rechtstreeks via de database open te sturen. Dit gaat via de tabel genaamd Server.

Om een deur te bedienen:

Record = 2
 Val1 = Panel number (not the address)
 Val2 = Command
 Val3 = Reader # (1-8)

Commands: bits 0-3
 0 = No change
 1 = Manual Lock
 2 = Manual momentary Unlock
 3 = Manual unlock and Hold Unlocked

bits 4-7
 0 = No change
 1 = Enable Reader
 2 = Disable Reader

Met het onderstaande voorbeeld commando openen we lezer 8 op paneel 4 volgens de ingestelde deur open tijd (momentary Unlock).
 Insert into server (record, val1, val2, val3)
 Values (2, 4, 2, 8)

1.5.3 Direct besturen van relais rechtstreeks via de server tabel

Om een relais te bedienen:

Record = 4
 Val1 = Panel number (not the address)
 Val2 = Command
 Val3 = Relay bits

Commands: bits 0-3
 0 = No change
 1 = Momentary ON
 2 = Hold ON
 3 = Turn OFF Relay

Relay bits
 Val3 0-73 => 1 = relay 1, 2 = relay 2, ... 73 = relay 73

Met het onderstaande voorbeeld commando openen we relais 73 op paneel 4 volgens de ingestelde relais open tijd (momentary Unlock).
 Insert into server (record, val1, val2, val3)
 Values (4, 4, 1, 73)

1.6 Status tabel

In de Status tabel worden de status van een device weergegeven.

In de volgen de tabel worden de kolom (veld) namen van deze tabel getoond.

JPLT.ca26LiveD...57 - dbo.Status			
	Column Name	Data Type	Allow Nulls
▶	Panel	smallint	<input type="checkbox"/>
▶	Type	smallint	<input type="checkbox"/>
▶	Device	smallint	<input type="checkbox"/>
	Status	smallint	<input checked="" type="checkbox"/>
	SDate	datetime	<input checked="" type="checkbox"/>
	Flag	bit	<input checked="" type="checkbox"/>
	Positn	smallint	<input checked="" type="checkbox"/>
	State	smallint	<input checked="" type="checkbox"/>
	GroupNo	int	<input checked="" type="checkbox"/>
	Version	varchar(20)	<input checked="" type="checkbox"/>
	MaxCards	int	<input checked="" type="checkbox"/>
	UsedCards	int	<input checked="" type="checkbox"/>
	ExpMem	int	<input checked="" type="checkbox"/>
	MaxXact	int	<input checked="" type="checkbox"/>
	MaxAG	int	<input checked="" type="checkbox"/>
	UsedAG	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

1.6.1 Tabel beschrijving Status

In de volgende tabel worden de velden van deze tabel beschreven.

Column Name	Dutch Translation	Description
Panel	Paneel	Paneel waarop het device zit
Type	Type	Type ingang
Device	Device type	Lezer, ingang, uitgang etc
Status	Status	Hier wordt de status van het device weergegeven
SDate		Datum/tijd transactie
Flag		NVT
Positn		NVT
State		NVT
GroupNo	Groep nummer	Groep nummer
Version	Versie	Versie paneel
MaxCards	Maximum kaarten	Aantal kaarten die in het paneel kunnen
UsedCards	Ingevoerde kaarten	Aantal kaarten die in het paneel zitten
ExpMem	Extra geheugen	Of het paneel is voorzien van extra geheugen
MaxXact	Maximum aantal transacties	Maximum aantal transacties dat het paneel kwijt kan
MaxAG	Maximum aantal toegangsgroepen	Maximum aantal toegangsgroepen dat het paneel kwijt kan
UsedAG	Aantal ingevoerde toegangsgroepen	Aantal ingevoerde toegangsgroepen in het paneel

1.6.2 Uitleg belangrijke velden van de status tabel

Paneel = Het software matige paneel nummer
 Type 0 = reader status
 1 = input status

2 = output status
 3 = nvt
 4 = Paneel status

Device = Reader, input, output nummer
 Bij paneel status (type 4) geeft hij hier '0' aan.

Status = reader (0)
 1 = deur is dicht
 2 = deur is geforceerd/ te lang open
 3 = deur in free access
 4 = deur staat op open houden
 9 = deur is open gestuurd

Let op! De melding "deur is geforceerd/ te lang open" worden alleen ververst als het status scherm in CardAccess open staan.

Status = input (1)
 17 = normaal
 18 = abnormaal
 19 = lijnfout (sluiting / open)

Status = output (2)
 24 = normaal (af)
 25 = abnormaal (op)

Status = paneel (4)
 11 = abnormaal (geen verbinding)
 17 = normaal (verbinding)

2.1 Aansturing van een externe plattegrond

De symbolen van panelen, readers, inputs en outputs worden bij het opstarten van de plattegrond gesynchroniseerd met de status tabel. Vervolgens worden de symbolen aangestuurd door transacties in de PendingAlert tabel.

Indien u symbolen van een ander programma wilt aansturen kunt u dit op dezelfde manier doen.

Een tweede manier is om de Socket Alert Notifier te gebruiken van CardAccess. Dit is een programma wat we via CardAccess kunnen activeren waarmee we elke transactie via TCP/IP naar een bepaald IP adres en poortnummer kunnen versturen. Informeer bij ARAS naar de mogelijkheden.

The screenshot displays the CardAccess 3000 software interface. The main window is titled "CardAccess 3000 (West-Europa (standaarttijd)) - Host PC". The menu bar includes "Systeem", "Sturing", "Toegang", "Administratie", "Configuratie", "Lockdown sturing", "Toon", and "Help". The toolbar contains icons for "Kaarthouder...", "Tijd zones...", "Toegang...", "Rapporten", "Status...", "Deuren...", "Relais...", "Links...", "Activiteiten links...", "Toon DVR...", and "Log uit...".

The "Transacties:" section features a toolbar with "Bladeren", "Reageren", "Plattegrond", "Video", "Sturing", and "Recent", along with a dropdown menu "Als Local Time". Below this is a table with columns: "Foto", "Klasse", "Omschrijving", "Locatie", "Datum", "Operator", and "Bevestigd". The table lists various transactions, including "Kaart geweigerd op deze tijd", "Paneel data", "Kaart geldig", "Paneel fout", "Systeem fout", "Video", "Ingang abnormaal", and "Deur nu dicht".

A "Single-Map Monitoraras bbg" window is overlaid on the table, showing a floor plan map. A red 'X' icon on the map indicates an alarm. A tooltip for this icon displays the following information: "P4 Lezer 1", "11/12/2012 01:35:04 PM", "DOOR FORCED OPEN", and "4-1 P4 Lezer 1". The map window also includes "Zoom In", "Zoom Out", "Disable Alert PopUp", "Zoom", and "Search" buttons.

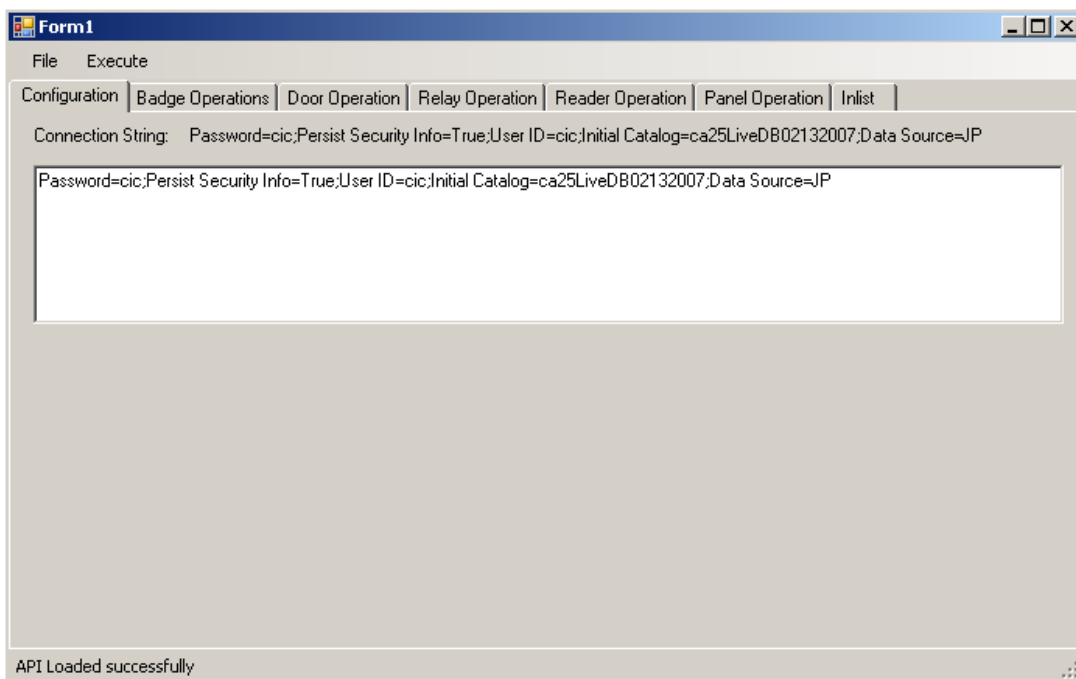
At the bottom left of the interface, there is a section labeled "Hangende alarmen" with a checkmark icon.

3 API (Application Programming Interface) beschrijving

3.1 Mogelijkheden API

- Kaarhouder mutaties
- Deur besturing
- Relais besturing
- Lezer besturing
- Paneel besturing
- Inlist

3.1.1 Configuratie Connection String



3.1.2 Badge Operation

Form1

File Execute

Configuration Badge Operations Door Operation Relay Operation Reader Operation Panel Operation Inlist

Badge:

Facility:

First Name:

Last Name:

Middle Initial:

AgNo1:

AgNo2:

AgNo3:

AgNo4:

AgNo5:

AgNo6:

Operation:

- Insert Badge
- Delete Badge
- Update Badge
- Enable badge
- Disable Badge

API Loaded success

3.1.3 Door Operation

Form1

File Execute

Configuration Badge Operations Door Operation Relay Operation Reader Operation Panel Operation Inlist

Panel #:

Door #:

Time:

Operation:

- No Change
- Lock Door
- Pulsed Unlock
- Hold Unlock

API Loaded successfully

3.1.4 Relay Operation

Form1

File Execute

Configuration | Badge Operations | Door Operation | Relay Operation | Reader Operation | Panel Operation | Inlist

Panel #: Paneel 1

Relays:

<input type="checkbox"/> 1	<input type="checkbox"/> 16	<input type="checkbox"/> 31	<input type="checkbox"/> 46	<input type="checkbox"/> 61
<input type="checkbox"/> 2	<input type="checkbox"/> 17	<input type="checkbox"/> 32	<input type="checkbox"/> 47	<input type="checkbox"/> 62
<input type="checkbox"/> 3	<input type="checkbox"/> 18	<input type="checkbox"/> 33	<input type="checkbox"/> 48	<input type="checkbox"/> 63
<input type="checkbox"/> 4	<input type="checkbox"/> 19	<input type="checkbox"/> 34	<input type="checkbox"/> 49	<input type="checkbox"/> 64
<input type="checkbox"/> 5	<input type="checkbox"/> 20	<input type="checkbox"/> 35	<input type="checkbox"/> 50	<input type="checkbox"/> 65
<input type="checkbox"/> 6	<input type="checkbox"/> 21	<input type="checkbox"/> 36	<input type="checkbox"/> 51	<input type="checkbox"/> 66
<input type="checkbox"/> 7	<input type="checkbox"/> 22	<input type="checkbox"/> 37	<input type="checkbox"/> 52	<input type="checkbox"/> 67
<input type="checkbox"/> 8	<input type="checkbox"/> 23	<input type="checkbox"/> 38	<input type="checkbox"/> 53	<input type="checkbox"/> 68
<input type="checkbox"/> 9	<input type="checkbox"/> 24	<input type="checkbox"/> 39	<input type="checkbox"/> 54	<input type="checkbox"/> 69
<input type="checkbox"/> 10	<input type="checkbox"/> 25	<input type="checkbox"/> 40	<input type="checkbox"/> 55	<input type="checkbox"/> 70
<input type="checkbox"/> 11	<input type="checkbox"/> 26	<input type="checkbox"/> 41	<input type="checkbox"/> 56	<input type="checkbox"/> 71
<input type="checkbox"/> 12	<input type="checkbox"/> 27	<input type="checkbox"/> 42	<input type="checkbox"/> 57	<input type="checkbox"/> 72
<input type="checkbox"/> 13	<input type="checkbox"/> 28	<input type="checkbox"/> 43	<input type="checkbox"/> 58	<input type="checkbox"/> 73
<input type="checkbox"/> 14	<input type="checkbox"/> 29	<input type="checkbox"/> 44	<input type="checkbox"/> 59	
<input type="checkbox"/> 15	<input type="checkbox"/> 30	<input type="checkbox"/> 45	<input type="checkbox"/> 60	

Operation: Pulse On

API Loaded success

3.1.5 Reader Operation

Form1

File Execute

Configuration | Badge Operations | Door Operation | Relay Operation | Reader Operation | Panel Operation | Inlist

Panel #: Paneel 1

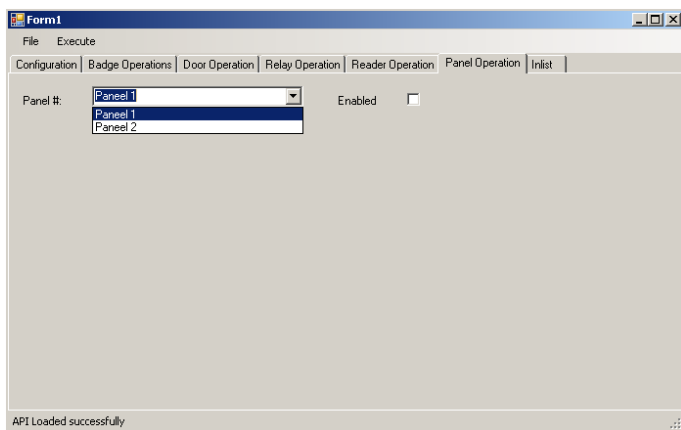
Reader: Hoofdentree In

Enabled

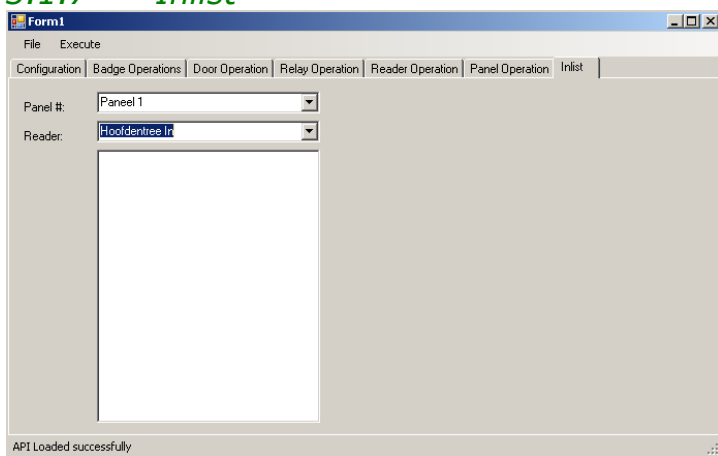
Enabled

API Loaded successfully

3.1.6 Panel Operation



3.1.7 Inlist



4 Voorbeeld code

4.1.1 Configuratie Connection String

The **_Initca3000** function initializes the ca3000 API for use by the host application.

```
int _Initca3000();
```

Parameters:

Return Values:

If the function succeeds, the return value is 0.

If the function fails, the return value is a negative number.

See Also:

_DeInitca3000, _GetConnectStr, _GetErrorString

The **_DeInitca3000** function de-initializes the ca3000 API. This function must be called before unloading the API dll to avoid memory leaks.

```
int _Initca3000();
```

Parameters:

N/A

Return Values:

If the function succeeds, the return value is 0.

If the function fails, the return value is a negative number.

See Also:

_Initca3000

The **_GetConnectStr** return the connection string that will be used by the API..

```
int _GetConnectStr (char* CS,
                  DWORD CSLen,
                  Bool Net);
```

Parameters:

CS

Returns the connection string that will be used by the API. *Required field.*

CSLen

Length of the allocated space for *CS*. *Required field.*

Net

Specifies if the calling application is a .Net application or not.

Return Values:

If the function succeeds, the return value is 0.

If the function fails, the return value is a negative number.

See Also:

_Initca3000, _GetErrorString

4.1.2 Badge Operation

The **_BadgeOperation** function adds/deletes/modifies Badges.

```
Int _BadgeOperation(WORD Facility,
                   __int64 Badge,
                   char* FirstName,
                   char* LastName,
                   char* MI,
                   WORD AGNo1,
                   WORD AGNo2,
                   WORD AGNo3,
                   WORD AGNo4,
                   WORD AGNo5,
                   WORD AGNo6,
                   WORD Operation);
```

Parameters:

Facility

Facility number. Value between 0 – 10. *Required field.*

Badge

Badge number. Value any positive integer. *Required field.*

FirstName

First name.

LastName

Last name. *Required field.*

MI

Middle initial.

AGNo1 - AGNo6

Access groups 1 through 6.

Operation

Intended operation. *Required field.*

0 = Add, 1 = Delete 2 = Modify, 3 = Enable
4 = Disable

Return Values:

If the function succeeds, the return value is number of row effected.

If the function fails, the return value is a negative number.

See Also:

_DeInitca3000, _GetErrorString

4.1.3 Inlist Operation

The **_InlistQuery** function returns the list of badges that are IN based on the Panel and Reader.

```
int _InlistQuery (WORD Panel,
                 WORD Reader,
                 __int64* BadgeList,
                 WORD* ResultSize);
```

Parameters:

Panel

Panel number. *Required field.*

Reader

Reader number. *Required field.*

BadgeList

Pointer to an array. *Required field.*

ResultSize

Length of the allocated space for *BadgeList*. If the size is smaller than the list, the function will return the actual size of the result set. *Required field.*

Return Values:

If the function succeeds, the return value is 0.

If the function fails, the return value is a negative number.

See Also:

_DeInitca3000, _GetErrorString

4.1.4 Door Operation

The **_DoorOperation** function allows manual door control.

```
Int _DoorOperation(WORD Panel,
                  WORD Door,
                  WORD Time,
                  WORD Operation);
```

Parameters:

Panel

Panel number. *Required field.*

Door

Reader number (1- 8). *Required field.*

Time

Duration (seconds: 0-59; minutes: 60 – 120 -> 1-60). *Required field.*

Operation

- 1 = Lock door
- 2 = Pulsed unlock
- 3 = Hold unlocked

Return Values:

If the function succeeds, the return value is 0.

If the function fails, the return value is a negative number.

4.1.5 Relay Operation

The **_RelayOperation** triggers a relay or a group of relays.

```
Int _RelayOperation(WORD Panel,
                   WORD Operation,
                   WORD* Relays,
                   WORD RelayCount
                   WORD Time);
```

Parameters:

Panel

Panel number. *Required field.*

Operation *Required field.*

- 1 = Pulse On
- 2 = Hold On
- 3 = Turn Off

Relays

List of relays that will be acted upon. *Required field.*

RelayCount

Number of relays in *Relays*. *Required field.*

Time

Used by times relay operations (0..3600 in Seconds)

Return Values:

If the function succeeds, the return value is 0.

If the function fails, the return value is a negative number.

4.1.6 *Reader Operation*

The **_ReaderOperation** function enables or disables a reader.

```
int _ReaderOperation(WORD Panel,  
                    WORD Reader,  
                    BOOLEAN Enabled);
```

Parameters:

Panel

Panel number. *Required field.*

Reader

Reader number. *Required field.*

Enable

Value *True* or *False* to enable/disable reader. *Required field.*

Return Values:

If the function succeeds, the return value is 0.

If the function fails, the return value is a negative number.

4.1.7 *Panel Operation*

The **_PanelOperation** function enables or disables a panel.

```
int _PanelOperation(WORD Panel,  
                   BOOL Enabled);
```

Parameters:

Panel

Panel number. *Required field.*

Enabled

Status. *Required field.*

Return Values:

If the function succeeds, the return value is 0.

If the function fails, the return value is a negative number.

4.1.8 *Input Operation*

The **_InputOperation** function enables or disables an input.

```
int _InputOperation(WORD Panel,
                  WORD Input);
```

Parameters:

Panel

Panel number. *Required field.*

Input

Input number. *Required field.*

Return Values:

If the function succeeds, the return value is Input Status.

If the function fails, the return value is a negative number.

4.1.9 *T&A Operation*

The **_TAQuery** function returns the Time and Attendance record for a badge.

```
int _TAQuery(WORD Facility,
             __int64 Badge,
             SYSTEMTIME StartTime,
             SYSTEMTIME EndTime,
             SYSTEMTIME* TimeList,
             WORD* ResultSize);
```

Parameters:

Facility

Panel number. *Required field.*

Badge

Badge number. *Required field.*

StartTime

Start of time range to report on. *Required field.*

EndTime

End of time range to report on. *Required field.*

TimeList

List of time pairs for T&A activity with in *StartTime* and *EndTime* range. *Required field.*

ResultSize

Size of allocated space for *TimeList*. If the size is smaller than the actual result set then the actual result size is reported. *Required field.*

Return Values:

If the function succeeds, the return value is 0.

If the function fails, the return value is a negative number.