

---

## Dokumentlog

Dato	Version	Beskrivelse	Applikation version	Reference	Forfatter	Godkender
2015.11.30	2.0	Ny godkendelses-proces	MDS 4.1	K15 Transition	Oprydning CPRDOK	Godkendt af leverandør ifm. Transition
2017.06.27	2.1	Side 9, slettet kort version af Field attribut og slettet side 14 ”Konvertering af GCTP tegn til valide XML tegn”		SER-1322	Brian Høj Andersen	Ulla Victoria Krog
2020.01.02	3.0	Ref til CPR adresse mv fjernet		SER-3762	Søren Bækdal	Karin Schøtt



**Håndbog  
Til  
CPR services**

**Bilag 8  
GCTP-standard  
m.m.**

**CPR-kontoret**

Hjemmeside: [www.cpr.dk](http://www.cpr.dk)

*Flgsnr. 001*

*O:\GTS\CPR\CPRDOK\CPR-Opgaven\Ydelser\Udvikling og Rådgivning\Systemets eksterne grænseflader\Servicehåndbogen\CPR Servicehåndbog\Servicehaandbog\_bilag\_8.doc*

---

## Indholdsfortegnelse

<b>1. INDLEDNING</b> .....	<b>3</b>
1.1 Identifikation .....	3
1.2 Formål .....	3
1.3 Forudsætninger .....	3
1.4 Begrænsninger .....	3
1.5 Forkortelser og definitioner.....	3
1.6 Referencer.....	3
<b>2. TAGS OG BLOKKE</b> .....	<b>4</b>
2.1 Tags.....	4
2.2 StartTag.....	4
2.3 EndTag.....	4
2.4 Blokke.....	4
2.5 Tom Blok .....	5
<b>3. LOGISK HIRAKI</b> .....	<b>6</b>
<b>4. GENERELLE TAGS OG ATTRIBUTTER</b> .....	<b>7</b>
4.1 GCTP.....	7
4.2 System.....	7
4.3 CprserviceHeader .....	7
4.4 F(ield) .....	9
4.5 Row .....	10
4.6 Table.....	11
4.7 Kvit.....	12
4.8 SIK.....	13

## 1. Indledning

### 1.1 Identifikation

General Communication Transfer Protocol (GCTP) er et data transport format til kommunikation af data mellem to programmer. Standarden er p.t. målrettet mod at løse behovet for data kommunikation mellem en host baseret server og en pc baseret klient.

### 1.2 Formål

Formålet med dette dokument er at beskrive GCTP formatet på en sådan måde at læseren sættes istand i til at kunne afkode data fra gctp formatet. Det skal endvidere være mulig på baggrund af dette dokument at implementerer programmer der er isatnd til at skrive og modtage GCTP.

### 1.3 Forudsætninger

Kommunikationen mellem klienten og serveren baseres på en ascii tekststreng. Metoden, hvorved tekststrengen kommunikerer, har ingen betydning for denne standard, tekststrengen kan kommunikerer via en fil, som data i et cgi felt eller ved anden kommunikations form. Vi opdeler tekststrengen i blokke ved hjælp af en række tags, som markerer starten på en blok, samt en tilsvarende markering af blokkens afslutning.

Det bemærkes, at der i dette dokument ikke er vist XML-header, XML-commandstatements og blokken der henviser til XML-namespace. De fremgår af bilag 5 om logon og generel brug af CPR-services.

Ved at bruge tags til styring af tekst strengen, er det ikke nødvendigt at angive længder på blokke eller felter. Beskrivelsen åbner dog mulighed for at indføre længde beskrivelser på alle tags, hvis det skulle blive påkrævet.

Vi kommunikerer ikke felt type beskrivelserne med i selve datastrengen. Felt typerne skal være kendte af både klient og server. Felt typerne gives ud fra feltets reference. Server og klient skal altså på forhånd være enige om hvilke felter der kommunikerer i hver transaktion. Felterne behøver ikke at komme i bestemt række følge.

### 1.4 Begrænsninger

Data i value felterne kan ikke indholde karakterene ascii x3E(>) eller ascii x22("). Hvis disse karakterer forekommer skal de sendes to gange uden mellemliggende tegn.

### 1.5 Forkortelser og definitioner

#### 1.5.1 Forkortelser

Dette afsnit indeholder en liste i alfabetisk orden af forkortelser anvendt i dette dokumentet.

CSC	Computer Sciences Corporation
KSS	Kvalitet Styringssystemet
GCTP	General Communication Transfer Protocol

### 1.6 Referencer

#### 1.6.1 Formelle referencer

GCTP indgår som en blok i et XML-dokument af XML ver 1.0 <http://www.w3.org/TR/REC-xml>. (REC-xml).

## 2. Tags og blokke

### 2.1 Tags

Alle tags er indpakket i et sæt af karakterene '<>'. En tag har et navn og kan have en eller flere attributter. Attributter består af et attributnavn efterfulgt af et lighedstegn som efterfølges af attribut data. Attribut data angives i dobbelt citationstegn ("), attribut data kan kun være karakterer, mellem attributterne skal der mindst være et space tegn (ascii x20).

*Eksempel 1. En tag med navn og 2 attributter:*

```
<Navn Att1navn="AttData"  
Att2navn="AttData">
```

### 2.2 StartTag

En blok har en StartTag som giver bloknavnet. Bloknavnet er en type beskrivelse, som fortæller hvilken type af data, der nu følger, samt hvordan den er organiseret.

*Eksempel 2. Her starter en ny blok af typen PersonData:*

```
<PersonData>
```

### 2.3 EndTag

Blokke, der har et indhold, har altid en EndTag, som afslutter blokken. EndTaggen har samme navn som blokken, dog med en foranstillet skråstreg '/'.

*Eksempel 3. Her slutter blokken PersonData:*

```
</PersonData>
```

### 2.4 Blokke

Blokke defineres ved hjælp af tags. En blok defineres ved hjælp af tagnavnet, som herefter benævnes som bloknavn. Bloknavnet starter lige efter den startende <, og skal afsluttes med minimum et blank space. Bloknavnet kan forekomme i en lang og en kort version. Den korte version anvendes hvis bloknavnet forekommer i stort antal. f.eks ved søge resultater. Bloknavnet giver implicit organisationen af data, dvs. hvilke attribut navne har denne tag, og hvad kan blokken indeholde mellem sine to tags.

*Eksempel 4. Langt bloknavn:*

```
<Field attributter>  
  Blokindhold ....  
</Field>
```

*Eksempel 5. Kort bloknavn:*

```
<F attributter>  
  Blokindhold ....  
</Fd>
```

Her vises blokken med bloknavnet kvit. Kvit har ingen attributter, men kan indeholde to fields med henholdsvis "KvitType" og KvitNr som referencer.

*Eksempel 6. Kvit:*

```
<kvit>
  <Field r="KvitType" v="0"/>
  <Field r="KvitNr" v="0"/>
</kvit>
```

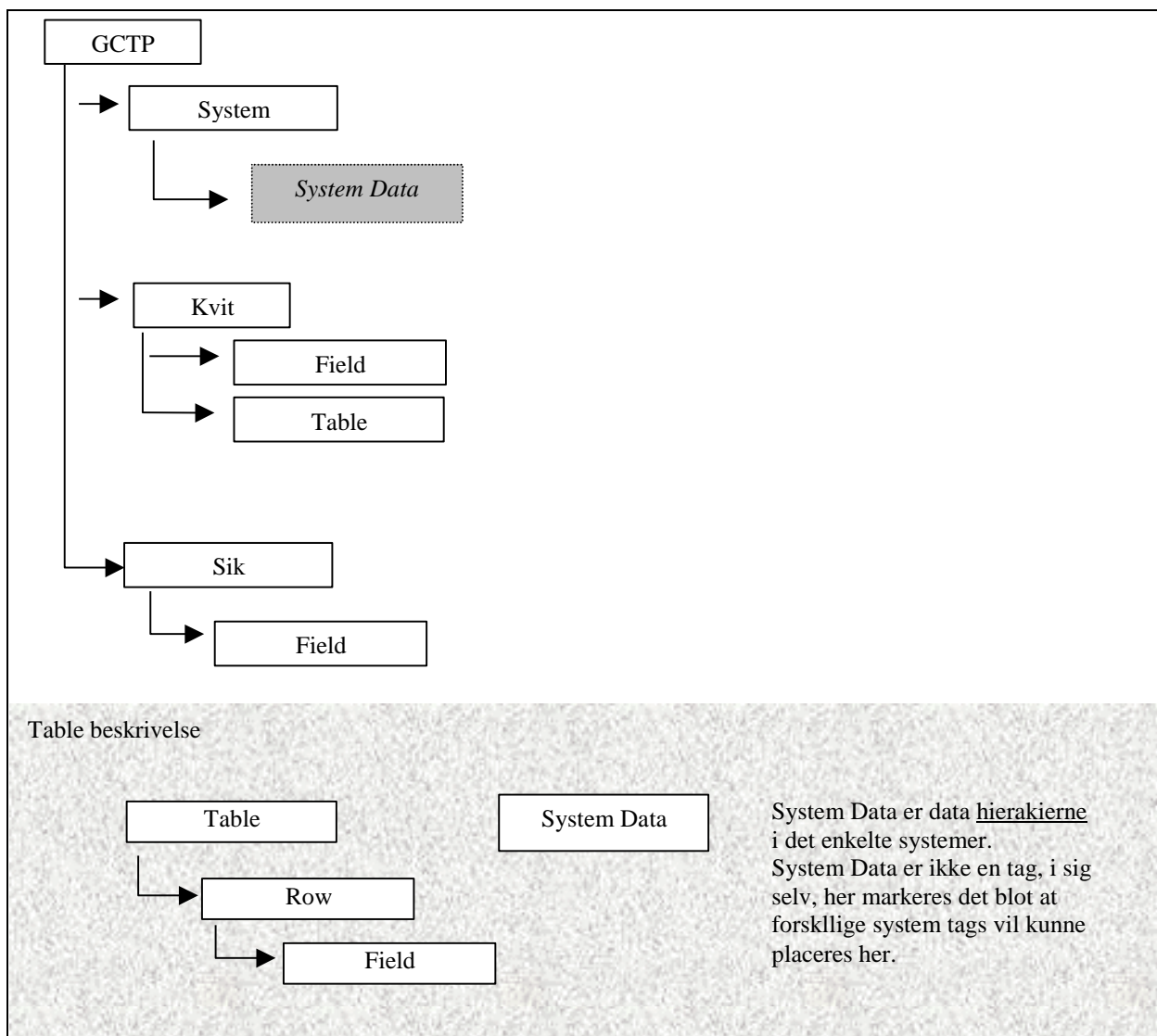
**2.5 Tom Blok**

Hvis en blok ikke har noget indhold, er det ikke nødvendigt at sende en endtag. En tom blok beskrives ved at det afsluttende tagmærke '>' foranstilles med '/'. Vi anvender tomme blokke, når vi sender felter. Felt data transporteres som attributter til feltet.

*Eksempel 7. Blokken Field har ingen indhold derfor afsluttes dette tag med />:*

```
<Field />
```

### 3. Logisk Hierarki



## 4. Generelle tags og attributter

Disse tags anvendes generelt af alle systemer som anvender GCTP.

Hvis en attribut angives i parentes, betyder det, at attributten er valgfri.

### 4.1 GCTP

Langt blok navn: GCTP

Kort blok navn: Ikke defineret.

GCTP taggen skal altid findes som den første tag i en fil. Herved fortælles det at, der frem til GCTP' slut tag findes data indkapslet GCTP tags.

Attribut	Navn	Værdisæt	Beskrivelse
v	Version	Tekst	GCTP version denne GCTP fil anvender.

Eksempel 8. :

```
<Gctp v="1.0">
.....
</Gctp>
```

### 4.2 System

Langt blok navn: System

Kort blok navn: Ikke defineret.

System taggen anvendes til oplysninger om hvilket system man agter at køre imod. Denne tag gør det muligt for en fil at indeholde data til forskellige systemer. Data til de enkelte systemer er grupperet under samme system tag. De enkelte systemer fastlægger, hvorledes data er organiseret under deres systemtag. De enkelte systemer er beskrevet som udvidelser af GCTP formatet, i disse udvidelser er systemernes tag og data organisation beskrevet.

Attribut	Navn	Værdisæt	Beskrivelse
r	System name	Tekst	Beskriver systemet, som der skal arbejdes med på hosten.

Eksempel 9. :

```
<System r="CprAjour">
.....
</System>
```



### 4.3 CprServiceHeader

Langt blok navn: CprSystemheader

Kort blok navn: Ikke defineret.

CprServiceHeader taggen anvendes til oplysninger om hvilken tilstand hændelsen er i. Hvilken hændelse det drejer sig om. Dette tag gør det muligt at aflæse hvilke aktioner man efterfølgende kan udføre.

Eksempel 10 beskriver initieringen af en hændelse, og Eksempel 11 viser svar med de tilladte aktioner.

*Eksempel 10. :*

```
<CprServiceHeader r="ADOPTI-I" st="P" u="V" a="I" ts="20050727094718410933">
```

Attribut	Navn	Værdisæt	Beskrivelse
u	V for vent eller ikke til stede	Tekst	Beskriver tilstanden på hændelsen
r	Hændelsesnavn	Tekst	Navnet på hændelsen
st	Hændelsesstatus	Tekst	Beskriver om hændelsen er primær eller sekundær
a	Aktion	Tekst	Beskriver hvilken aktion man er i gang med
ts	Timestamp	Tekst	Dato og tidsstempling

*Eksempel 11. :*

```
<CprServiceHeader r="ADOPTI-I" st="P" u="V" a="I" ts="20050727094718410933">
<Table r="Aktioner">
<Row>
<Field r="KODE" v="V" t="Validering"/>
</Row>
<Row>
<Field r="KODE" v="F" t="Fortryd"/>
</Row>
<Row>
<Field r="KODE" v="2" t="Gem kladde i vent"/>
</Row>
<Row>
<Field r="KODE" v="3" t="Gem og send til godk"/>
</Row>
</Table>
</CprServiceHeader>
```

Mulige værdier til attributen aktioner:

Aktion V	Validering af hændelse
Aktion G	Gemning af hændelse
Aktion 1	Initiering fra vent
Aktion 2	Gem som kladde
Aktion 3	Gem og send til godkendelse i vent
Aktion 4	Gem og godkend i vent
Aktion 5	Aktivering fra vent
Aktion 7	Afvisning

#### 4.4 F(ield)

Blok navn: Field

Attribut	Navn	Værdisæt	Beskrivelse
r	Field reference	Tekst	Unik identifikation af feltet. Denne felt reference kendes på begge sider, og anvendes til at identificerer feltet ,dets type og værdisæt. Felt referencen kaldes i Cpr systemet klient referencen.
(v)	Value	Tekst	Selve dataindholdet i dette felt
(a)	Attribute	Tekst	Denne attribut kan anvendes til at give yderligere informationer til dette felt. Værdisættet afhænger af hvilket system som anvendes.
(a1)	Suppl attribute	Tekst	Supplerende Attribut. Denne attribut kan anvendes til at give endnu yderligere oplysninger om dette felt, dette kan være oplysninger som ikke kan findes i (a)attributten. Værdisættet afhænger af hvilket system som anvendes. Det er oplagt at udvide denne med a2,a3...ax hvis et system finder behov for dette.
(t)	Text	Tekst	Hvis data indhold i Value kan udtrykkes ved anden tekst, kan denne tekst specificeres her. Dette kan anvendes ved kode/klartekst oversættelser.
(ts)	Text short	Tekst	Samme som 't', blot med kort version af teksten.
(tl)	Text long	Tekst	Samme som 't', blot med en længere beskrivende tekst.
(tm)	Text medium	Tekst	Samme som 't', blot med en mellem længde beskrivende tekst.
(e)	Error state	[ '0','1' ]	Beskriver om dette felt er fejlbehæftet. <ul style="list-style-type: none"> <li>• 0 Ingen fejl</li> <li>• 1 Der er fejl i feltet.</li> </ul>

Eksempel 12. Field angives som "<Field ">":

```
<Field r="CNVN_FORNVN" a="L" v="Peter"/>
```

## 4.5 Row

Langt blok navn: Row.

Kort blok navn: Ikke defineret.

Row er beregnet på at kunne indeholde en række af data fra en tabel. Ved hjælp af usage attributten kan det fastsættes, om denne række skal indsættes, slettes, opdateres etc. Selve datafelterne findes som fields inde i row blokken.

Row er forberedt til at forekomme inde i en table blok, og samspillet mellem table og row er netop metoden til at transmitere en tabel imellem server og klient.

Row vil næppe optræde uden at være indeholdt i en table. Table er beskrevet som i punkt af dette dokument.

Attribut	Navn	Værdisæt	Beskrivelse
(u)	Usage	['C','U','D','M' 'REST']	Hvad skal der ske med denne række 'C' Create, indsæt denne som ny række. 'U' Update, opdater denne række. 'D' Delete, slet denne række. 'M' Model række, denne række er en model som serveren sender ud. Modelrækken skal være den første række i tabellen. Klienten anvender denne til at se attributter til de enkelte kolloner, hvis klienten vil indsætte nye rækker. Her angives også eventuelle default værdier og myndighedskoder til nye rækker. Når M anvendes skal k være tom. 'REST' Der findes mere data til denne tabel. Denne række er altid den sidste i tabellen. Den indeholder nøglen til den næste række. Dette anvendes ved scrolling. Klienten kan anvende denne nøgle i en ny søgning som REST nøgle i en ny søgning, hosten vil så levere data startende fra denne række. Hvis denne række modtages betyder det at hosten har flere række til denne tabel, hvis den ikke modtages betyder det at hosten ikke har mere til denne tabel.
(k)	Key	Tekst	Unik identifikation af netop en række. Dette fremsendes fra serveren, og serveren anvender denne til den unikke identifikation.

**Eksempel 13. En tabel uden model row:**

```
<Table>
  <Row u="C" k="12345625122">
    <Field r="FORNVN" v="Peter"/>
    <Field r="MELNVN" v="Valby"/>
    <Field r="EFTERNVN" v="Larsen"/>
  </Row>
</Table>
```

**Eksempel 14. En tabel med model række & rest række:**

```
<Table aia="290">
  <Row u="M" >
    <Field r="FORNVN" a="S"/>
    <Field r="MELNVN"/>
    <Field r="EFTERNVN"/>
  </Row>
  <Row k="12345625122">
    <Field r="FORNVN" v="Peter"/>
    <Field r="EFTERNVN" v="Larsen"/>
  </Row>
  <Row k="12345625123">
    <Field r="FORNVN" v="Kurt"/>
    <Field r="MELNVN" v="Valby"/>
    <Field r="EFTERNVN" v="Hansen"/>
  </Row>
  <Row u="REST" k="12345625128"/>
</Table>
```

**aia** fortæller at der i alt er 290 rækker til denne tabel. Model rækken viser at første kolonne er "FORNVN", anden kolonne er "MELNVN" og tredje kolonne er "EFTERNVN". Det ses at mellemnavnet i første række er tomt hvorfor dette felt ikke sendes. Den sidste række er en "REST" række, dennes key kan anvendes til at fortæller hvor næste søgning skal starte fra.

## 4.6 Table

Langt blok navn: Table.

Kort blok navn: Ikke defineret.

Table er forberedt til at kunne indeholde data i tabel format, dvs i rækker og koller. Table tagget anvendes sammen med row tagget, Table tagget omkrænses hele tabellen. De enkelte rækker omkrænses af rowtag inde i tabel tagget.

Attribut	Navn	Værdisæt	Beskrivelse
(r)	Table name	Tekst	Tabel navnet til denne tabel. Dette er en unik identifikation af netop denne tabeltype. Kan sammenlignes med attributten r i Field.
(aia)	Antal i alt	Numerisk	Beskriver hvor mange rækker denne søgning i alt resulterer i. Anvendes ved scrolling således at klienten ved starten af søgningen kan opsætte scrolbars etc.
(min)	Min row	Tekst	Det minimale antal rækker som denne tabel kan indeholde.

(max)	Max row	Tekst	Det maksimale antal rækker denne tabel kan have. Anvendes af serveren til at fortælle hvor mange rækker klienten kan indsætte i denne tabel.
-------	---------	-------	--

**Eksempel 15.**

```
<Table r="NavneTab" mr="25">
  <Row k="12345625122">
    <Field r="CNVN_FORNVN" v="Peter"/>
    <Field r="CNVN_EFTERNVN" v="Larsen"/>
  </Row>
  <Row k="12345625122">
    <Field r="CNVN_FORNVN" v="Mette"/>
    <Field r="CNVN_EFTERNVN" v="Hansen"/>
  </Row>
</Table>
```

**4.7 Kvit**

Langt blok navn: Kvit.  
 Kort blok navn: Ikke defineret.

Kvitering fra server om forløbet af den ønskede handling.

Attribut	Navn	Værdisæt	Beskrivelse
r	Type	[Dump,System, Anmrk,Afslut, Fejl,Ok]	Typen af kvitering. Afhængig af type kan der forekomme data inde i blokken.
(t)	Tekst	Tekst	Kviterings tekst.
(v)	Værdi		Kviterings nummer for denne type. Anvendes blandt andet til fejl nummer.  Værdien afhænger af hvilket systemet der anvendes.

**Eksempel 16.**

```
<Kvit r="Fejl" t="person findes ikke" v="1022"/>
```

*Eksempel 17. Ved afslutning af en cpr service kan følgende forekomme:*

```
<Kvit r="Afslut" v="0"/>
  <Table>
    <Row>
      <Field r="PNR" v="1202553365"/>
      <Field r="ADRVN" v="Larsen,Poul"/>
    </Row>
    <Row>
      <Field r="PNR" v="0203613366"/>
      <Field r="ADRVN" v="Larsen,Emma"/>
    </Row>
  </Table>
</Kvit>
```

*Eksempel 18. Hvis der er behov for blot at sende en række fejl tekster anvendes feltet FEJLTEXT i en table:*

```
<Kvit r="Fejl" v="2001"/>
  <Table>
    <Row>
      <Field r="FEJLTEXT" v="Der må ikke angives
      personer med udenlands adresse"/>
    </Row>
    <Row>
      <Field r="FEJLTEXT" v="Addressen findes
      ikke"/>
    </Row>
  </Table>
</Kvit>
```

#### 4.8 SIK

Langt blok navn: Sik.  
Kort blok navn: Ikke defineret.

Syntaks som anvendes ved logon

*Eksempel 19. :*

```
<Sik ui="x3450" pw="xxxxxx" npw="xxxxxx"/>
```

Attribut	Navn	Værdisæt	Beskrivelse
ui	User id	Tekst	Bruger Id
(pw)	Password	Tekst	Kode ord
(npw)	New password	Tekst	Nyt kodeord