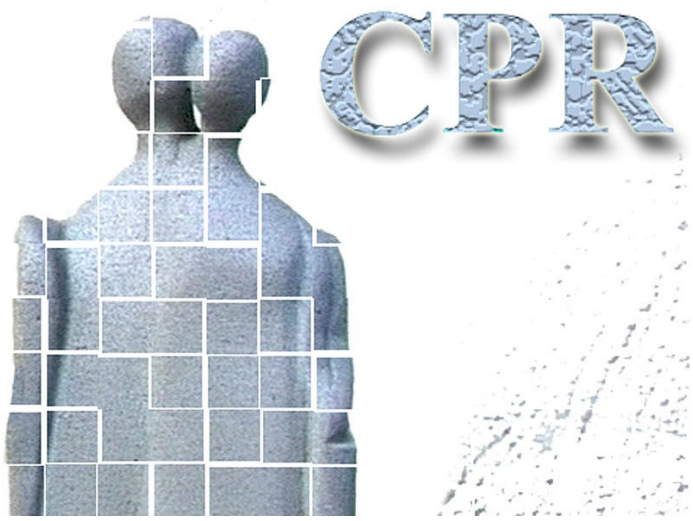


Manual

on

CPR services

Annex 8

GCTP standard etc.

The CPR Office

Datavej 20, PO Box 269, 3460 Birkerød, Denmark

Emailcpr@cpr.dk. Telefax +45 45 82 51 10. Website: www.cpr.dk

Table of contents

1. Introduction

1.1 Identification

General Communication Transfer Protocol (GCTP) is a data transport format to communicate data between two programs. The standard is currently aiming at solving the need for data communication between a hosted server and a PC client.

1.2 Objective

The objective of this document is to describe the GCTP format in such a manner that the reader is able to decode data from the GCTP format. This document is also to enable implementation of programs which are able to write and receive GCTP.

1.3 Requirements

Communication between the client and the server is based on an ascii text string.
The method by which the text string is communicated has no effect on this standard. The text string can be communicated via a file, as data in a cgi field or by any other form of communication.
We divide the text string into blocks using a number of tags, which mark the beginning of a block and the end of a block.

It should be noted that this document does not show XML-header, XML-command statements and the block that refers to XML-namespace. These are mentioned in Annex 5 about logon and general use of CPR services.

Using tags to control the text string does not require specification of lengths of blocks or fields. However, the description enables introduction of length descriptions on all tags, if required.

We do not communicate the field type descriptions in the actual data string. The field types must be known by the client as well as the server. The field types are given on the basis of the reference of the field.
Server and client must agree in advance what fields to communicate in each transaction.
The fields do not need to be in a specific order.

1.4 Limitations

Data in the value fields cannot contain the characters ascii x3E(>) or ascii x22(”). If these characters occur, they must be sent twice without intermediate characters.

1.5 Abbreviations and definitions

1.5.1 Abbreviations

This section contains a list in alphabetical order of abbreviations used in this document.

CSC	Computer Science Corporation
KSS	Quality management system
GCTP	General Communication Transfer Protocol

1.6 References

1.6.1 Formal references

GCTP is included as a block in the XML document of XML ver 1.0 http://www.w3.org/TR/REC-xml. (REC-xml).

2. Tags and blocks

2.1 Tags

All tags are inserted between the set of characters '<>'. A tag has a name and can have one or more attributes. The attributes consist of an attribute name followed by a equals sign (=) which again is followed by attribute data. Attribute data is specified in double quotation marks (“), attribute data can be characters, between the attributes there must be at least one blank space (ascii x20).

Example 1. *A tag with a name and 2 attributes:*

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

2.2 StartTag

A block has a StartTag which is the block name. The block name is a type of description which tells what type of data follows and how it is organised.

Example 2. *Here a new block starts of the type PersonData:*

```
<PersonData>
```

2.3 EndTag

Blocks which have content always have an EndTag which ends the block. The EndTag has the same name as the block, however with a prefixed ‘/’.

Example 3. *Here the block PersonData ends:*

```
</PersonData>
```

2.4 Blocks

Blocks are defined using tags. A block is defined using the tag name, which then is the block name. The block name starts just after the starting <, and must end with at least one blank space. The block name can occur in a long or a short version. The short version is used if the block name occurs in a great number, e.g. in connection with search results. The block name implicitly provides the organisation of data, i.e. what attribute names this tag has, and what the block can contain between its two tags.

Example 4. *Long block name:*

```
<Field attributter>
    Block contents....
</Field>
```

Example 5. *Short block name:*

```
<F attributter>
    Block contents....
</Fd>
```

The following example shows the block with the block name kvit. Kvit has no attributes, but can contain two fields with ”KvitType” and “KvitNr” respectively as references.

Example 6. *Kvit:*

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

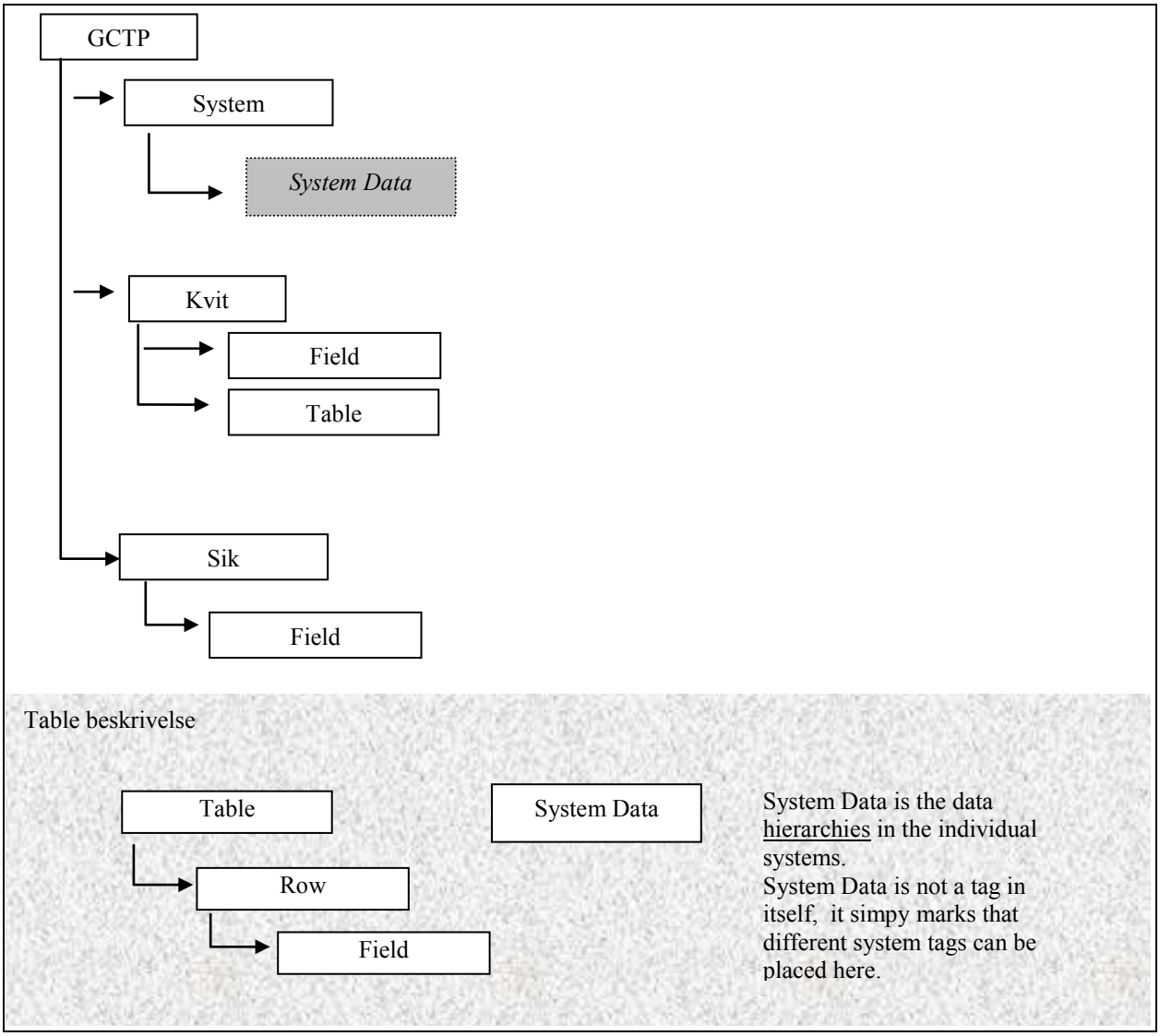
2.5 Empty block

If a block has no content, it is not necessary to send an endtag. An empty block is described by prefixing the end tag character '>' with '/'. Empty blocks are used when fields are sent. Field data is transported as attributes to the field.

Example 7. *The block Field has no content and is therefore ended with this tag />:*

```
<Field />
```

3. Logical hierarchy



4. General tags and attributes

These tags are generally used by all systems using GCTP.

If an attribute is specified in brackets, it means that the attribute is optional.

4.1 GCTP

Long block name: GCTP

Short block name: Not defined.

The GCTP tag is always the first tag in a field. This tells us that data exists in the GCTP tags up to the GCTP's endtag.

Attribute	Name	Set of values	Description
v	Version	Text	GCTP version used by this GCTP value.

Example 8. *:*

```
<Gctp v="1.0">
  ……
</Gctp>
```

4.2 System

Long block name: System
Short block name: Not defined.

The system tag is used for information about what system you intend to run against. This tag enables a file to contain data for different systems. Data for the individual systems are grouped under the same system tag. The individual system determines how data is organised under its system tag. The individual system is described as extensions of the GCTP format. In these extensions, the systems' tag and data organisation are described.

Attribute	Name	Set of values	Description
r	System name	Text	Describes the system which is to be worked with on the host.

Example 9. *:*

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

4.3 CprServiceHeader

Long block name: CprSystemheader
Short block name: Not defined.

The CprServiceHeader tag is used for information about what mode the incident is in. What incident is it about? This tag makes it possible to read what actions you can subsequently perform.

Example 10 describes the initiation of an incident. Example 11 shows replies with the actions allowed.

Example 10. *:*

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

Attribute	Name	Set of values	Description
u	W for pending or not present	Text	Describes the mode of the incident
r	Name of incident	Text	Name of incident
st	Status of incident	Text	Describes whether the incident is primary or secondary
a	Action	Text	Describes what action you are doing
ts	Timestamp	Text	Date and time-stamping

Example 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>
```

Possible values for the attribute actions:

Action V	Validation of incident
Action G	Saving incident
Action 1	Initiation from pending
Action 2	Save as draft
Action 3	Save and send for approval in pending
Action 4	Save and approve in pending
Action 5	Activation from pending
Action 7	Rejection

4.4 F(ield)

Long block name: Field

Short block name: F

Attribute	Name	Set of values	Description
r	Field reference	Text	Unique identification of the field. This field reference is known on both sides, and is used to identify the field, its type and set of values. The field reference is called client reference in the CPR system.
(v)	Value	Text	Actual data content in this field
(a)	Attribute	Text	This attribute can be used to give further information to this field. The set of values depends on the system used.
(a1)	Suppl attribute	Text	Supplementary attribute. This attribute can be used to give further information about this field. This may be information which cannot be found in the (a) attribute. The set of values depends on the system used. It is obvious to extend this with a2,a3….ax if a system requires this.
(t)	Text	Text	If data content in Value can be expressed by another text, this can be specified here. This can be used in connection withcode/clear text translations.
(ts)	Text short	Text	Same as 't', only with short version of the text.
(tl)	Text long	Text	Same as 't', only with a longer describing text.
(tm)	Text medium	Text	Same as 't', only with a medium-long describing text.
(e)	Error state	['0','1']	Describes whether this field has errors. • 0 No errors • 1 The field has errors

Example 12. *Long version field specified as "<Field ":*

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

Example 13. *Short version field is specified only as "<F ":*

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

4.5 Row

Long block name: Row.

Short block name: Not defined.

Row should be able to contain a row of data from a table. The usage attribute can determine whether this row is to be inserted, deleted, updated etc. The actual data fields are fields in the row block.
Row has been prepared to occur in a table block, and the interaction between table and row is the method to transmit a table between server and client.
Row is unlikely to appear without being in a table. Table is described as in point 4.6 of this document.

Attribute	**Name**	**Set of values**	**Description**
(u)	Usage	['C','U','D','M' 'REST']	What is going to happen with this row? 'C' Create, insert this as new row. 'U' Update, update this row. 'D' Delete, delete this row. 'M' Model row. This model is a model sent out by the server. The model row must be the first row in the table. The client uses this model row to see attributes of the individual columns if the client wants to insert new rows. Any default values and authority codes for new rows are also specified here. When M is used k must be empty. 'REST' More data exists in this table. This row is always the last in the table. It contains the key to the next row. This is done by scrolling. The client can use this key as a REST key in a new search, the host will then deliver data starting from this row. If this row is received, it means that the host has more rows for this table. If this row is not received, the host does not have more for this table.
(k)	Key	Text	Unique identification of precisely one row. This is forwarded from the server, and the server uses this row for the unique identification.

Example 14. *Table without 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>
```

Example 15. *Table with model row and rest row:*

```
 <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 informs that there are a total of 290 rows for this table. The model row shows that the first column is "FORNVN", the second column is "MELNVN" and the third column is "EFTERNVN". Middle name in the first row is empty, which is why this field is not sent. The last row is a REST row, and the key for this row can be used to tell from where the next search should begin.

4.6 Table

Long block name: Table.
Short block name: Not defined.

The table has been prepared to contain data in table format, i.e. in rows and columns. The table tag is used with the row tag. The table tag encircles the entire table. The individual rows are encircled by rowtag in the table tag.

Attribute	Name	Set of values	Description
(r)	Table name	Text	Table name of this table. This is a unique identification of this specific table type. Can be compared with the attribute r in Field.
(aia)	Total number	Numeric	Describes how many rows in total this search results in. Used by scrolling, so that the client can create scrollbars etc. from the beginning of the search
(min)	Min row	Text	The minimum number of rows this table can contain.

(max)	Max row	Text	The maximum number of rows this table can contain. Used by the server to tell how many rows the client can insert in this table.

Example 16.

```
<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

Long block name: Kvit.
Short block name: Not defined.

Confirmation from the server about the process of the requested action.

Attribute	**Name**	**Set of values**	**Description**
r	Type	[Dump,System, Anmrk,Afslut, Fejl,Ok]	Type of confirmation. Depending on the type, data can occur in the block.
(t)	Text	Text	Text of confirmation.
(v)	Value		Confirmation number for this type. Used for error number etc. Value depends on the system used.

Example 17.

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

Example 18. *When ending a CPR service, the following may occur:*

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

Example 19. *If there is a need to just send one row of error texts, the field FEJLTEXT is used in the table:*

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

4.8 SIK

Long block name: Sik.
Short block name: Not defined.

Syntax which is used in connection with logon

Example 20. *:*

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

Attribute	**Name**	**Set of values**	**Description**
ui	User id	Text	User id
(pw)	Password	Text	Password
(npw)	New password	Text	New password

Conversion of GCTP signs to valid XML signs

The GCTP is received as a text string from the CPR system. The text string is searched through for illegal signs and the illegal signs are replaced by legal signs in fields with illegal signs.

Signs to be replaced	**New sign to replace the illegal sign**
&plus;	+ eller &#002A;
&amp;	&# 0026;
&quot;	&# 0027;
&lt;	&#003C;
&gt;	&#003E;
&percnt;	&# 0025;

When the illegal signs have been replaced by legal signs, the edited XML string is processed by an XML parser.