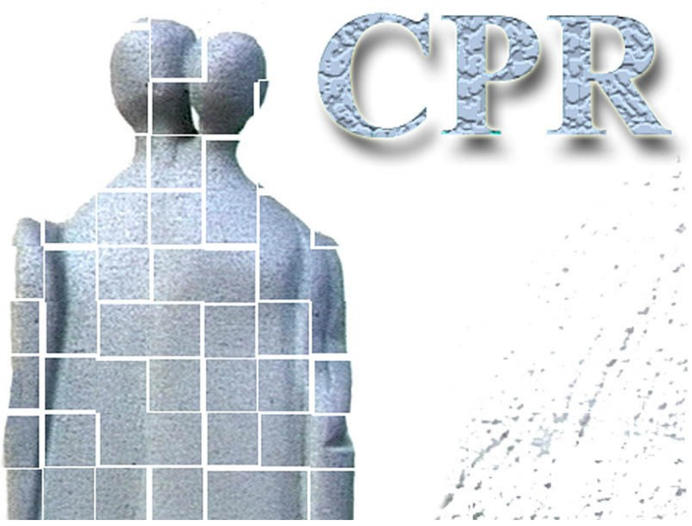

Manual

on

CPR services

Annex 6

Use of CPR search services, programming guidelines

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

2. Premise for the reader

The examples are easier to understand after having read the GCTP standard.

Standard concerning the general GCTP format

GCTP is the standard for communication between server and client which the CPR has chosen to use, i.e. because this ensures a dynamic interface between server and client.

3. Premise before calling services

Before beginning to call CPR services, the user needs a physical connection to the CPR and must be logged on to the CPR system as described in

Annex 4: Establishment of network connection to the CPR and
Annex 5: Establishment of connection to and logon to the CPR server.

The examples only describe the GCTP part of the communication, and not any HTTP headers and similar. This is considered to be part of the transport layer and this is irrelevant to the actual application.

It should also 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.

In terms of GCTP, all searches fall under the CprSoeg system.

Searches are described in a **Service specification** which describes the search, the keys the client must/can send to the server, and the data the server returns.

4. Security in connection with searches

Access to carry out searches in the CPR is controlled by the individual user ID.
Different people in the same authority can have different access (profiles) to search in the CPR.

5. General description of CPR search services

As described earlier, there are two types of CPR search services: look-up and search.

Look-up can be divided into two subtypes:

1. Simple look-up with only one hit
   E.g. "Current address"
   (displays the person's current address row as well as miscellaneous marks)

2. Look-up with several rows
   E.g. "Former addresses"
   (displays a list of a person's current and former addresses)

The actual searches all return several rows, but there are also two types here:

1. Limited searches with qualified keys or on small amounts of data
   E.g. "Authority search" (search on municipality codes and similar)

2. Broad searches on relatively large amounts of data
   E.g. "Date of birth search" or "Name search"

The following describes the four types of search in more detail.

6. Simple look-up

Communication is fairly simple in connection with simple look-up.
The service specification clearly states what keys the client is to call with and what fields the server returns.

When the client has received the data returned, the look-up can be forgotten.
The server has checked security (does the user have access to the service) and has logged the look-up (this user has looked up information on this/these person(s)).

7. Look-up with several rows

To maintain good performance, the CPR server returns no more than 20 rows to the client in connection with this type of look-up. Where requested, the client can set this maximum even lower (indicated by the key MAXA). This can be used if only the person's former address is to be displayed (first former address).

If the client needs this (indicated by the key AIA), the server can tell how many rows there are in total. With this knowledge, the client can set up scrollbars etc. correctly.

If there are more rows than those returned, the server also returns a "restart row" which only contains one key for the client to call with in order to get the next rows.

Therefore, the client must remember to save the restart key for any later use.

The service specification clearly states what keys the client is to call with and what fields the server returns. In some cases, this type of look-up will return some single fields to start with, e.g. address in connection with "Show all residents on the person's address".

8. Limited searches

The limited searches work in the same way as “Look-up with several rows” and are often just “broad searches” on a very small amount of data.

9. Broad searches

In broad searches, the look-up in the database is potentially so extensive that the keys are saved for the individual rows in a temporary search data store on the server.

In connection with a name search, for example, all civil registration numbers on those matching the search criteria are saved.
When/if the client asks for the next 20 persons on the list, the server only has to retrieve the keys in the search data store and read the persons’ data.

Otherwise, restart works in the same way as in “Look-up with several rows” and “Limited searches”.

The key to the search data store is part of the restart key, so the server can use this key to read the result of the search, if the client calls to read more than the rows returned.

When the client initiates such a search, the server will also return the key to the search data store as a single field (field reference "SEARCH_ID").
When the user leaves the list, the client **must** call a special CPR utility service (SOEGDONE) to have the search data store deleted.

Note that if the number of rows that exists is lower than the maximum number wanted (MAXA - perhaps defaulted to 20), a search data store will **not** be saved, as the client already has received all data.

10. Example 1: Show current address (simple look-up)

The service to display current address is ADRESSE3: Current address

The service specification states that the client MUST specify a civil registration number (PNR) and that with the AKX key, the client can specify whether undone and corrected rows should be included in the look-up.
In this look-up, AKX states whether the undone/corrected rows are to be included when setting up TIDLADRMRK.

If the client is assumed to want to include the undone/corrected rows when setting up TIDLADRMRK, the client's GCTP to the server will look as follows:

```
<Gctp v="1.0" ><System r="CprSoeg"><Service r="ADRESSE3"><CprServiceHeader
r="ADRESSE3"><Key><Field r="PNR"  v="1212121212"/><Field r="AKX" v="X"/></Key></
CprServiceHeader></Service></System></Gctp>
```

The more reader-friendly version:

```
<Gctp v="1.0" >
  <System r="CprSoeg">
    <Service r="ADRESSE3">
      <CprServiceHeader r="ADRESSE3">
        <Key>
          <Field r="PNR" v="1212121212"/>
          <Field r="AKX" v="X"/>
        </Key>
      </CprServiceHeader>
    </Service>
  </System>
</Gctp>
```

Note that:

- If the client does not want to take into account any undone/corrected rows
  <Field r="AKX" v="X"/>
  is left out. (not all searches/look-ups use AKX)

- The keys do not have to be in the order described in the service specification. As long as the GCTP structure is followed, the internal location of the individual fields is of secondary importance.

- The civil registration number here (1212121212) is an example. An actual search with this civil registration number will result in an error. All keys are validated before the searches are activated and the validation of PNR has to take place in the CPR.

In our example, the reply from the server will be as follows:

```
<Gctp v="1.0 env="Test" ><System r="CprSoeg"><Service r="ADRESSE3">
<CprServiceHeader r="ADRESSE3" ts="20010123201713711036"/><CprData u="O"><Rolle
r="HovedRolle"><Field r="TIDLADRMRK" v="X"/><Field r="TIDLKOMMRK" v="X"/><Field
r="SUPADRMRK" v="X"/><Field r="ADRKOD" v="1"/><Field r="TIMSTP"
v="20000525142701794751"/><Field r="SYSTEM_TS" v="20000525141925862898"/><Field
```

r="ADRTS" v="20000525141953813707"/><Field r="MYNKOD" v="0201" t="Allerød" tl="Allerød"/><Field r="VEJKOD" v="2073" t="Aarupvej" tm="Aarupvej"/><Field r="HUSNR" v="002"/><Field r="ETAGE" v="st"/><Field r="SIDEDOER"/>
<Field r="TILFLYKOMDATO" v="199902020000"/><Field r="BOLIGTYPKOD" v="0" t="Ukendt boligtype"/><Field r="POSTNR" v="3520" t="Farum"/><Field r="STARTDATO" v="200005201414"/><Field r="KOMKOD" v="0207" t="Farum" tl="Farum"/><Field r="FLYTPAMRK" v="X"/></Rolle></CprData></Service></System></Gctp>

The more reader-friendly version:

```
<Gctp v="1.0"  env="Test" >
 <System r="CprSoeg">
  <Service r="ADRESSE3">
   <CprServiceHeader r="ADRESSE3" ts="20010123201713711036"/>
   <CprData u="O">
    <Rolle r="HovedRolle">
     <Field r="TIDLADRMRK" v="X"/>
     <Field r="TIDLKOMMRK" v="X"/>
     <Field r="SUPADRMRK" v="X"/>
     <Field r="ADRKOD" v="1"/>
     <Field r="TIMSTP" v="20000525142701794751"/>
     <Field r="SYSTEM_TS" v="20000525141925862898"/>
     <Field r="ADRTS" v="20000525141953813707"/>
     <Field r="MYNKOD" v="0201" t="Allerød" tl="Allerød"/>
     <Field r="VEJKOD" v="2073" t="Aarupvej" tm="Aarupvej"/>
     <Field r="HUSNR" v="002"/>
     <Field r="ETAGE" v="st"/>
     <Field r="SIDEDOER"/>
     <Field r="TILFLYKOMDATO" v="199902020000"/>
     <Field r="BOLIGTYPKOD" v="0" t="Ukendt boligtype"/>
     <Field r="POSTNR" v="3520" t="Farum"/>
     <Field r="STARTDATO" v="200005201414"/>
     <Field r="KOMKOD" v="0207" t="Farum" tl="Farum"/>
     <Field r="FLYTPAMRK" v="X"/>
    </Rolle>
   </CprData>
  </Service>
 </System>
</Gctp>
```

Note especially the **SIDEDOER** field. This field has only been included here to show what an empty field **could** look like.
Normally such fields will not be sent, as they do not contain any information.
As the service specification shows, there are many more fields in ADRESSE3, but many of them cannot be displayed at the same time (as the person can **either** live abroad **or** in Denmark **or** be missing).
In certain situations, the server may decide to send empty fields in the GCTP string anyway; often for debugging purposes. Among other things, this is why it is necessary that the client can react reasonably to a **dynamic GCTP string.**

11. Example 2: Show the person's current and former addresses (look-up with list)

Initiating a list is done largely in the same way, only in this situation it is possible to ask the counter mentioned above, which will state how many rows the look-up will generate in total.
In this situation we asked to have the counter displayed with AIA=X.

```
<Gctp><System r="CprSoeg"><Service r="ADRESSE1"><CprServiceHeader r="ADRESSE1">
<Key><Field r="PNR" v="1212121212"/><Field r="AKX" v="X"/><Field r="MAXA" v="10"/>
<Field r="AIA" v="X"/></Key></CprServiceHeader></Service></System> </Gctp>
```

Reader-friendly version:

```
<Gctp>
 <System r="CprSoeg">
  <Service r="ADRESSE1">
   <CprServiceHeader r="ADRESSE1">
    <Key>
     <Field r="PNR" v="1212121212"/>
     <Field r="AKX" v="X"/>
     <Field r="MAXA" v="10"/>
     <Field r="AIA" v="X"/>
    </Key>
   </CprServiceHeader>
  </Service>
 </System>
</Gctp>
```

The server's response: (only the reader-friendly version is shown here)

```
<Gctp v="1.0"  env="Test"
 <System r="CprSoeg">
  <Service r="ADRESSE1">
   <CprServiceHeader r="ADRESSE1" ????????????????????????/>
   <CprData u="O">
    <Rolle r="HovedRolle">
     <Table r="ADRESSE" aia="20">
      <Row k="19430730120019950508100017300852">
       <Field r="TIMSTP" v="19950508100017300852"/>
       <Field r="SYSTEM_TS" v="19950508115740600010"/>
       <Field r="ADRTS" v="19950508100017300852"/>
       <Field r="MYNKOD" v="0267" t="Skovby" tl="Skovby"/>
       <Field r="ADRKOD" v="1"/>
       <Field r="STARTDATO" v="194307301200"/>
       <Field r="STARTDATOUSM" v="*"/>
       <Field r="HUSNR" v="020"/>
       <Field r="VEJKOD" v="2650" t="Roskildevej" tm="Roskildevej"/>
       <Field r="TILFLYKOMDTO" v="194307210000"/>
       <Field r="CONVN" v="c/o Olga Persersen"/>
       <Field r="CONVNTS" v="19981113090000000000"/>
       <Field r="KOMKOD" v="0265" t="Roskilde" tl="Roskilde"/>
       <Field r="POSTNR" v="4000" t="Roskilde"/>
       <Field r="BOLIGTYPKOD" v="1" t="Egentlig beboelse"/>
```

```
        <Field r="STADR" v="Roskildevej 20"/>
       </Row>
       <Row k="19430730120019910214163730671007194307301200A">
        <Field r="TIMSTP" v="19910214163730671007"/>
        <Field r="SYSTEM_TS" v="19910214122531900086"/>
        <Field r="ADRTS" v="19910214163730671007"/>
        <Field r="AKM" v="A" t="Fortrudt"/>
        <Field r="MYNKOD" v="0267" t="Skovby" tl="Skovby"/>
        <Field r="ADRKOD" v="2"/>
        <Field r="UDRLANDEKOD" v="5205" t="Atlantis" tl="Atlantis"/>
        <Field r="STARTDATO" v="194307301200"/>
        <Field r="SLUTDATO" v="194307301200"/>
        <Field r="SLUT_TS" v="19910214163730671007"/>
        <Field r="INDRLANDEKOD" v="0000" t="Ukendt Myndighed" tl="Ukendt Myndighed"/>
        <Field r="UDLANDSADR1" v="O-Adr-1"/>
        <Field r="UDLANDSADR2" v="O-Adr-2"/>
        <Field r="UDLANDSADR3" v="O-Adr-3"/>
        <Field r="UDLANDSADR4" v="O-Adr-4"/>
        <Field r="UDLANDADR_MYNKOD" v="0000"
                t="Ukendt Myndighed" tl="Ukendt Myndighed"/>
        <Field r="UDLANDADR_TS" v="19430730000000000000"/>
        <Field r="STADR" v="I udlandet"/>
       </Row>
       <Row k="19430730120019910112130549528586194307301200K">
        <Field r="TIMSTP" v="19910112130549528586"/>
        <Field r="SYSTEM_TS" v="19910112115740600004"/>
        <Field r="ADRTS" v="19910112130549528586"/>
        <Field r="AKM" v="K" t="Rettet"/>
        <Field r="MYNKOD" v="0267" t="Skovby" tl="Skovby"/>
        <Field r="ADRKOD" v="1"/>
        <Field r="STARTDATO" v="194307301200"/>
        <Field r="SLUTDATO" v="194307301200"/>
        <Field r="HUSNR" v="030"/>
        <Field r="VEJKOD" v="2650" t="Roskildevej" tm="Roskildevej"/>
        <Field r="TILFLYKOMDTO" v="194307210000"/>
        <Field r="KOMKOD" v="0265" t="Roskilde" tl="Roskilde"/>
        <Field r="POSTNR" v="4000" t="Roskilde"/>
        <Field r="BOLIGTYPKOD" v="1" t="Egentlig beboelse"/>
        <Field r="STADR" v="Roskildevej 30"/>
       </Row>
        .
        .
        <7 andre rækker>
        .
        .
       <Row u="REST" k="19301122120019910115100912813115193507220000"/>
      </Table>
     </Rolle>
    </CprData>
    <Kvit r="Ok" t="" v="0"/>
   </Service>
  </System>
</Gctp>
```

Note that each row has its own key (k=), and that the last row has usage = Restart (U=REST). This key is to be used if the client wants to read the last 10 rows, as seen in this example:

```
<Gctp>
 <System r="CprSoeg">
  <Service r="ADRESSE1">
   <CprServiceHeader r="ADRESSE1">
    <Key>
     <Field r="PNR" v="1212121212"/>
     <Field r="AKX" v="X"/>
     <Field r="MAXA" v="10"/>
     <Field r="REST" v="19301122120019910115100912813115193507220000"/>
    </Key>
   </CprServiceHeader>
  </Service>
 </System>
</Gctp>
```

In this example, the AIA key has been removed, as we already know from the first call that there are 20. Instead the restart key has been specified.
However, it is important to state the criteria keys again (here PNR and AKX).
If AKX=X was **not** stated in the restart call, this call will not contain the undone/corrected rows and therefore perhaps not return 10 rows as expected.

12. Example 3: Authority search (limited search)

In the following example, the client wants a list of authorities starting with “farum”.
The client has asked for 15 hits to be returned at a time, and for the total number (AIA=X) to be returned.

```
<Gctp v="1.0" >
 <System r="CprSoeg">
  <Service r="MYN2">
   <CprServiceHeader r="MYN2">
    <Key>
     <Field r="MYTE" v="farum"/>
     <Field r="AIA" v="X"/>
     <Field r="MAXA" v="15"/>
    </Key>
   </CprServiceHeader>
  </Service>
 </System>
</Gctp>
```

The server’s response can be as follows:

```
<Gctp v="1.0" env="Test" >
 <System r="CprSoeg">
  <Service r="MYN2">
   <CprServiceHeader r="MYN2" ts="20010130200740753336"/>
   <CprData u="O">
    <Rolle r="HovedRolle">
     <Table r="MYN" aia="2">
      <Row k="0207Farum">
       <Field r="CMYN_MYNKOD" v="0207" t="Farum" tl="Farum"/>
       <Field r="CMYN_MYNTYP" v="05"/>
       <Field r="CMYN_SLUTDATO"/>
      </Row>
      <Row k="7414Farum,Farum">
       <Field r="CMYN_MYNKOD" v="7414"
                 t="Farum,Farum" tl="Farum Sogn, Farum Kommune"/>
       <Field r="CMYN_MYNTYP" v="25"/>
       <Field r="CMYN_SLUTDATO"/>
      </Row>
     </Table>
    </Rolle>
   </CprData>
  </Service>
 </System>
</Gctp>
```

If the client only wanted to search for municipalities (and in this case only get one unique hit), the client would have to add the MYNT key as follows:

```
<Gctp v="1.0" >
 <System r="CprSoeg">
  <Service r="MYN2">
   <CprServiceHeader r="MYN2">
    <Key>
     <Field r="MYTE" v="farum"/>
     <Field r="MYNT" v="05"/>
     <Field r="AIA" v="X"/>
     <Field r="MAXA" v="15"/>
    </Key>
   </CprServiceHeader>
  </Service>
 </System>
</Gctp>
```

The response would look almost like the above, however AIA would be 1 and the row with "Farum sogn" would not be there.

13. Example 4: Date of birth search (broad search)

Search on date of birth is a potentially “extensive” search.
The server will therefore save the keys on the rows found (here civil registration numbers), so that only the keys already found have to be used to look up data, if the client asks for more than the first 20 persons.

Conversely, this means that the server has some storage which will just take up space when the user has finished using the result.
Therefore, the client must tell the server when the user has finished with the list.

This applies to all cases in which one or more rows have been returned to the client, but needless to say this is not necessary if the search does not provide a result.

The client’s request:

```
<Gctp v="1.0" >
 <System r="CprSoeg">
  <Service r="FODSOG2">
   <CprServiceHeader r="FODSOG2">
    <Key>
     <Field r="AIA" v="X"/>
     <Field r="FODT" v="19540322"/>
     <Field r="KOMK" v="0265"/>
    </Key>
   </CprServiceHeader>
  </Service>
 </System>
</Gctp>
```

The client wants to find a person who lives in Roskilde (municipality code 265) and who was born on 22 March 1954.

The server’s response can be as follows:

```
<Gctp v="1.0"  env="Test" >
 <System r="CprSoeg">
  <Service r="FODSOG2">
   <CprServiceHeader r="FODSOG2" ts="20010131155558793795"/>
   <CprData u="O">
    <Rolle r="HovedRolle">
     <Field r="SEARCH_ID" v="B553C318E0B9C000"/>
     <Table r="FOEDSOEG" aia="39">
      <Row k="1212121212B553C318E0B9C000">
       <Field r="PNR" v="1212121212"/>
       <Field r="ADRNVN" v="Riesen,Rie" t="Rie Riesen"/>
       <Field r="FOEDMYNTXT" v="Farum,Farum"/>
       <Field r="STADR" v="Grønnemosen 5"/>
       <Field r="POSTNR" v="4000" t="Roskilde"/>
       <Field r="KOEN" v="K"/>
      </Row>
      <Row k="1313131313B553C318E0B9C000">
       <Field r="PNR" v="1313131313"/>
```

```
        <Field r="ADRNVN" v="Rolfsen,Rolf" t="Rolf Rolfsen"/>
        <Field r="FOEDMYNTXT" v="Farum,Farum"/>
        <Field r="STADR" v="Grønnemosen 5 A"/>
        <Field r="POSTNR" v="4000" t="Roskilde"/>
        <Field r="KOEN" v="M"/>
       </Row>
       <Row k="1414141414B553C318E0B9C000">
        <Field r="PNR" v="1414141414"/>
        <Field r="ADRNVN" v="Piasen,Pia" t="Pia Piasen"/>
        <Field r="FOEDMYNTXT" v="Farum,Farum"/>
        <Field r="ADRBESKYT" v="1"
                t="Beskyt" tm="Navne-  og adresse besk." tl="Navne- og adresse beskyttelse"/>
        <Field r="STADR" v="Roskildevej 20"/>
        <Field r="POSTNR" v="4000" t="Roskilde"/>
        <Field r="KOEN" v="K"/>
       </Row>
       <Row k="1515151515B553C318E0B9C000">
        <Field r="PNR" v="1515151515"/>
        <Field r="ADRNVN" v="Persen,Per" t="Per Persen"/>
        <Field r="FOEDMYNTXT" v="Farum,Farum"/>
        <Field r="STADR" v="Roskildevej 21 B"/>
        <Field r="POSTNR" v="4000" t="Roskilde"/>
        <Field r="KOEN" v="M"/>
       </Row>
       <Row k="1616161616B553C318E0B9C000">
        <Field r="PNR" v="1616161616"/>
        <Field r="ADRNVN" v="Oliasen,Ole" t="Ole Oliasen"/>
        <Field r="CNVN_STATUS" v="80" ts="UDR" tl="Udrejst" t="Udrejst"/>
        <Field r="FOEDMYNTXT" v="Farum,Farum"/>
        <Field r="STADR" v="Roskildevej 25"/>
        <Field r="POSTNR" v="4000" t="Roskilde"/>
        <Field r="KOEN" v="M"/>
       </Row>
.
        .
        <16 andre rækker>
        .
        .
       <Row u="REST" k="2222222222B553C318E0B9C000"/>
      </Table>
     </Rolle>
    </CprData>
   </Service>
  </System>
</Gctp>
```

The client must save the value of SEARCH_ID. This value is used to tell the server when it is OK to delete the search data store. This may be because the user leaves the list or if the client has “read to the bottom”.

Note that if the number of rows that exist is lower than MAXA (perhaps defaulted to 20), a search data store will **not** be saved, as the client has already received all data. In this case SEARCH_ID will be blank, and will therefore usually not be sent out.