

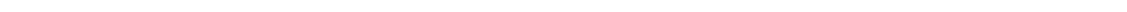
---

# CORE UDPS documentation

Version 1.10  
Januari 2019



HEALTHCARE  
SOLUTIONS



## Versie beheer

Versie	Datum	Auteur	Omschrijving
1.4	28-2-2014	Harmen Schut	Diverse wijzingen en aanvullingen. Zie document "Verschil CoreUDPS 1.3 en 1.4"
1.5	20-10-2015	Harmen Schut	Attribute reden in par.2.7.
1.6	08-06-2017	Harmen Schut	New triggers in par 3.9.
1.7			Distributielijst aangepast
1.8	04-01-2018	Harmen Schut	Functie cipavraag synchroon (in 3.1)
1.9	29-10-2018	Harmen Schut	Functie vertaalconclusie Retrospectief aanleveren BSN (2.7) CIS functies verwijderd Distributielijst aangepast
1.10	24-12-2018	Harmen Schut	Functie ppm_definitietabel

## Distributie

Naam	Functie	Organisatie
Hannelore Hofhuis	Directeur	Stichting PALGA
Richard Heutinck	Directeur Operations	DT Healthcare Solutions
Fons Geerts	Projectleider (Kern-UDPS)	DT Healthcare Solutions
Paul Hölzenspies	Projectleider (Sympathy)	DT Healthcare Solutions
Henk ten Hove	Product development	DT Healthcare Solutions
Jan Oosting	Bioinformaticus	LUMC
Richard Mertens	Senior consultant	MIPS

## Table of contents

Versie beheer .....	2
Distributie .....	2
1 1 The Core-UDPS .....	4
1.1 Excerpts to PALGA National Databases .....	4
1.2 Excerpt error handling .....	7
1.3 Checking diagnosis lines.....	8
1.4 Patient query to PZVDB .....	8
1.5 Follow-up and first incidence .....	8
1.6 Maintenance .....	9
2 The Core UDPS XML link with System-X .....	10
2.1 General.....	10
2.2 The XML pathology report "rapport" .....	11
2.3 The berichten and bericht elements .....	16
2.4 The element "antwoord" .....	17
2.5 The element "creatie" .....	18
2.6 The element "verwijder" .....	19
2.7 The element "wijziging" .....	20
2.8 The element "pfcontrole" .....	24
2.9 The element "vraag" .....	24
2.10 The element "parsepp" .....	28
2.11 The element "functie" .....	29
3 Functions.....	31
3.1 Function "cipavraag" .....	31
3.2 Function "cipareult" .....	32
3.3 Function "incidentie" .....	32
3.4 Function "followup" .....	33
3.5 Function "consultuitslag" .....	34
3.6 Function "spoolfiles" .....	35
3.7 Function "triggers".....	36
3.8 Function "vriesmateriaal" .....	38
3.9 Function "digitalebeelden" .....	38
3.10 Function "vertaalconclusie" .....	38
3.11 Function "ppm_definitietabel" .....	39
3.12 The element "drcvraag" .....	40
3.13 DRC controles.....	42
4 Permissions .....	44
5 DTD .....	45
6 UDPS dataset .....	50

# 1 1 The Core-UDPS

UDPS is a shortcut for ‘Uniform Decentraal PALGA Systeem’ the uniform local PALGA computer system that is placed in laboratories for clinical pathology to facilitate the local medical administration, and to send excerpts of pathology reports to the national PALGA database. The functionality of UDPS is (intentionally) limited to the medical administration. It was never intended that the system should provide full support for the primary laboratory process or provide a full grown management information system.

The term ‘uniform’ means (here) that all UDPS systems have the same nationally defined dataset (possibly with some local extensions). It is the intention of PALGA to ‘upgrade’ the current national database to a distributed one, using the UDPS systems with this common dataset in a virtual private network.

That dataset is described in a separate document ‘UDPS-dataset’

For some laboratories the functionality of UDPS is too limited, and they (plan to) use some other (third party) pathology system. For these laboratories PALGA provides a ‘small’ version of UDPS, called ‘Kern-UDPS’ or Core-UDPS that is not much more than an interface to the national PALGA database now, for sending excerpts to the database and for putting patient queries to the database. In the future it may be part of the National Pathology distributed database however. The third party system communicates with Core-UDPS using the standard UDPS XML server. This service (server) and its XML is documented separately in ‘UDPS-XML-publ.doc’.

The Core-UDPS can also be used exclusively for checking diagnosis lines, without focusing on reporting to the national PALGA database (see 1.3 and 3.9)

This document describes how the third party system should use this XML interface and how it can guarantee that correct excerpts of all (relevant) pathology reports are sent to the national database.

System maintenance on Core-UDPS is automated and/or done remote by the service provider as far as possible. What tasks are left to a local system administrator, possibly supported by the third party system, is documented here as well.

The third party pathology system will be called ‘System-X’ in the rest of this document, the Core-UDPS is simply often called UDPS. The national PALGA databases will be called PZVDB/RDS (Patient history Query Database and Datastore).

## 1.1 Excerpts to PALGA National Databases

When a report is finished on System-X, it sends a relevant part of it to UDPS using the XML ‘creatie’ and ‘wijziging’ orders. The attribute ‘status’ of the element ‘wijziging’ should be ‘8’. That status indicates to UDPS that an excerpt of this report must be sent to PZVDB/RDS. At least the following sections (‘rubriek’ elements) should be included:

- datumontvangst (2)
- naamman (1)
- naamvrouw (1)
- voorletters
- geslacht
- geboortedatum (2)
- geboorteeeuw (2)
- leeftijd (2)
- geboorteplaats
- geboorteland
- woonplaats
- postcode (2)
- rz
- tv
- hf
- diag1 – diag12 (3)
- qual1 – qual 4
- klinischegegevens
- macroscopie
- microscopie
- conclusie (2b)
- epicrise (2b)
- protocollair (5)
- protocoldata (5)
- bsnummer (2a)
- toestemmingcipa (2a)
- CRIS (4)

Ad (1): at least one of these must be filled.

Ad (2): required.

Ad (2a): required in future

Ad (2b): S-reports: one of the two is required, otherwise conclusie is required.

Ad (3): at least one is required.

Ad (5): when a national palgaprotocol was used for reporting

Ad (4): only B-reports, CRIS is a set of cervix cytology items, packed in a special 'container' section. PZVDB/RDS expects the following items:

- versie (default 3)
- cr\_door (1)
- cr\_instrument (1)
- cr\_aspect (1)
- cr\_aanleiding (1)
- cr\_klachten (1)
- cr\_ingrepen (1)

- cr\_initiatief (2)
- cr\_datumlm (1)
- cr\_patroon (1)
- cr\_anticonceptie (1)
- cr\_hormoongebruik (1)
- cr\_zwangerschap (1)
- cr\_duur (1a)
- cr\_herhaling (1)
- cr\_k (1)
- cr\_o (1)
- cr\_p (1)
- cr\_a (1)
- cr\_c (1)
- cr\_b (1)
- cr\_t (1b)

Ad (1): required by PZVDB/RDS in CRIS versions > 2

Ad (1a): required if cr\_zwangerschap 1, 2, 3 or 4

Ad (1b): required if cr\_b 2 or 3

Ad (2): not in CRIS version 3, only in 1 and 2.

See the UDPS dataset documentation for meaning and values of these items.

Three other sections may be sent to PZVDB/RDS as well ('vrij1', 'vrij2' and 'vrij3' may map to locally defined (short) sections of the report.)

If that is the case these sections should be included.

If, in a Core-UDPS, a report is sent in status 8, and –after the change- the report does not fully meet the requirements for PZVDB/RDS, the report ('wijziging') will be refused (nack) with an appropriate error message ('fout' element). The PZVDB/RDS requirements checks in the Core-UDPS include a DRC (diagnosisline) check.

If a report is sent with a section (rubriek) that is a date, the format and content will be checked as well, even if attribute 'soort' is not 'datum'. If the content is not empty or

an existing date, the report ('wijziging') will be refused (nack) with an error message ('fout' element).

SystemX must handle error messages (nack) from the UDPS XML server, bringing them to the attention of the system administrator if needed.

When UDPS sends the excerpt to PZVDB/RDS, it sets the status of the report (from 8) to 9. If UDPS notices that the report is not complete enough for PZVDB/RDS or that the content is wrong for some reason it will not send the excerpt and it will set the status to 4 (instead of 8.)

Some time later UDPS will receive result messages from PZVDB/RDS concerning excerpts that has been sent.

If PZVDB/RDS indicates that something is wrong with the content of an excerpt, UDPS will reset the status of the report to 8 again.

If everything is OK, than UDPS will set the report in archive status ('A'), moving the report from the working directories to an archive file.

Core-UDPS itself is capable of handling extended characters. The XML messages must indicate the proper encoding, or else 'UTF-8' is expected. In diagnosis lines (rubriek) extended characters are converted to a corresponding ascii character: the thesaurus does not contain any extended character).

## 1.2 Excerpt error handling

Due to user or system errors some reports will be refused by PZVDB/RDS or will even never be sent there. If nothing is done to restore these errors PZVDB/RDS will be incomplete and the working directories will get overloaded, bringing UDPS to a halt. The maximum number of reports in the UDPS working directories is set to 20.000, but the number of reports in the working directories should be kept well below that, and UDPS does provide means to handle errors:

- SystemX can fetch the error (and OK) messages from PZVDB/RDS (and other system message files) using the XML 'functie' order 'spool files'. The PZVDB/RDS results can be found in the spool file 'datacom'. SystemX should present (the content of) these files to the system administrator or some other executive that can restore errors.
- SystemX should periodically monitor the (changes of) status of reports in UDPS as long as this status is not set to 'A'. It can do so using the XML 'vraag' order or using an SQL database query.

Prior to sending the report, SystemX can issue a **pfcontrole** order. UDPS will react to this as if it were a 'wijziging' whis status 8 (complete) and report any omissions. The content of this element is like that of a 'wijziging'. There will not be any change in the report on disk though.

### 1.3 Checking diagnosis lines

Many PZVDB/RDS error messages may be prevented by using the ‘PALGA Diagnoseregule Controle Module’, that checks the content ‘diag’ and ‘qual’ sections. That can be done using the XML ‘functie’ order ‘drcvraag’. The result provides possible alternatives for unknown diagnosis terms.

PZVDB/RDS does not handle extended characters, the Core-UDPS will convert them to sensible ascii characters. The same is done when checking the diagnosislines using ‘drcvraag’.

### 1.4 Patient query to PZVDB

Via UDPS SystemX can query PZVDB for excerpts belonging to some patient. It can use the XML ‘functie’ orders ‘cipavraag’ and ‘ciparesult’ to do so, but it must create a report on UDPS first using the ‘creatie’ and ‘wijziging’ first. Such a report should contain the following sections:

- datumontvangst (2)
- naamman (1)
- naamvrouw (1)
- voorletters
- geslacht
- geboortedatum (2)
- geboorteplaats
- woonplaats

Ad (1): at least one of these must be filled.

Ad (2): required.

If the status of this report is set to 0, UDPS will automatically post a patient query to PZDVB during the following night (and set the status to 7, or to 4 if the query was refused for some reason.) Using ‘cipavraag’ is not necessary than, but can speed up things.

### 1.5 Follow-up and first incidence

Every weekend Core-UDPS produces two result sets:

- follow: follow-up information triggered by the content of ‘tv’, time-delay, and ‘hf’, repeat-factor, sections in excerpts, and
- incid: first cancer incidence messages, triggered by excerpts that were sent to PZVDB during the previous week.

UDPS fetches these results automatically, they should be available on Monday morning.



In UDPS these result sets consist of a number file of triggering reports (excerpts) and with each triggering report a result file containing the result of a national patient query for the patient in the triggering report.

System-X can use the XML 'functie' orders 'followup' and 'incidentie' to fetch this information from UDPS. See the XML documentation. The report number files 'incid' and 'follow' contain the most recent report numbers (these files are overwritten during the weekend. The files 'f.xxxx' and 'i.xxxx' (xxxx is a time stamp here) are number files belonging to previous result sets (follow-up and incidence respectively).

## 1.6 Maintenance

UDPS is designed to operate in an environment where no professional system administrator is available. It periodically does several system checks, and any irregularities are sent to the remote service provider using email over a dedicated VPN connection. This mechanism is of course dependent on hardware. The VPN connection should always be connected and never be switched off. The service provider may want to remote log in and will ask the local system administrator to open a VPN connection then.

In most cases the backup of the UDPS is stored on a disk offered by de ICT department of the hospital.

If no external location of disk space is available, then the local system administrator has to change backup tapes daily and check the spool files, directly on UDPS or through SystemX (using the XML 'functie' order 'spoolfiles'). These spool files contain messages concerning the daily workflow such as errors messages from PZVDB/RDS in.

System messages usually are in the directory /var/log/:

- messages: general system messages
- dps: general UDPS messages
  
- rapporten: updates of (pathology) reports
  
- rootmail: mail sent to the service provider

In UDPS the reports are stored in flat files in the file system. The database is laid over them and offers indexing and query facilities. In a working directory tree the reports are stored that have an active status (0-8), one report per file. Since the number of files should not grow to big, causing loss of performance, this is limited to 20.000. If the number of working directory reports exceeds this number, creating new reports and also updating reports will fail.

Core-UDPS rejects reports if it can be expected that sending them to PZVDB/RDS will fail. But even then it is possible that some reports are accepted, and subsequent sending to PZVDB/RDS fails due to software errors or thesaurus changes.

If the sending program reports an error in the excerpt, such as a misspelled diagnosis line, the report will be reset to status 8, and will be resent to PZVDB/RDS over and over. Such errors must be ‘repaired’, and **some user must see these errors** (found in the ‘datacom’ spool file) and correct them.

Once a report has been successfully sent to PZVDB/RDS, it gets status 9 and, in the following weekend, is moved to archive, files that contain all reports of one year and one kind of investigation. The status then becomes ‘A’. If for some reason a report is moved back to the working directory, or a new version is sent by System-X, a new version of the report is created. The old version will stay, unchanged, in the archive. The new version id will be ‘B’ and subsequently ‘C’ etcetera.

When a report has been created in Core-UDPS in status 0 the status changes to 7 after a query to PZVDB. This report will be removed after some month if the status is still 7.

## 2 The Core UDPS XML link with System-X

This chapter is the technical description of the XML and HTTP based client-server link that can be used to exchange data between the Core UDPS and other systems (System-X).

It describes the situation where the UDPS is the server and System-X is the client.

### 2.1 General

A System-X will (HTTP) post an XML message to the UDPS, and, if the server in the UDPS is up and running, it will receive an XML message back from the UDPS. The message layout is described (and prescribed) by the XML document type definition or DTD “berichten.dtd”, which can be found on the UDPS with the URL “*http://UDPSname/xml/dtd/berichten.dtd*”. See also the chapter DTD.

On the UDPS, the server can be used through the Apache HTTP web server on port 80 with the URL “*http://UDPSname/xmlserver*”. In this URL and in the URL of the DTD “*UDPSname*” is the name or IP address with which the UDPS is locally known. The client must identify itself using Basic Authentication and must have a user id and password on the UDPS. As a preferred alternative, a secure connection can be used. In any case the client must have permission for any use of this link. These permissions are tied to the user id, or to the system id if a secure connection is used.

The message from the client to the server can be seen as a bundle of orders or requests, the message back from server to the client is an answer, which can be an acknowledge or an error message, or can contain data, such as pathology reports or tables.

At this moment the following orders have been defined for UDPS:

1. **query**, select one or more pathology reports and deliver them or deliver some set of fields.
2. **creation**, create some pathology report in the UDPS
3. **deletion**, remove some pathology report from the UDPS
4. **update**, add to or update fields in a pathology report.
5. **print**, start some predefined selection/print procedure
6. **diagnosis check**, check diagnosis lines.
7. **function**, perform available function on the UDPS.

The implementation of these orders will be described in following chapters. Since the exchange of data often will imply exchanging pathology reports, first the XML representation of a report, the “rapport” element, will be described.

Generally, if an order implies the update of a report, and the server runs into some error, the update is skipped completely, and the answer will contain an error element, “fout”. If the order implies sending data to the client, the effect of an error is not well defined. Probably some data will be sent, together with some “fout” elements.

## 2.2 The XML pathology report ”rapport”

The XML representation of a pathology report in connection with UDPS is called “rapport”. The DTD can be found here: “<http://UDPSname/xml/dtd/rapport.dtd>”.

The following simple example describes this “rapport”:

```
<rapport id="T03-00001" mode="compleet" status="0">
  <rubriek id="1" naam="datumontvangst" soort="datum">20030102</rubriek>
  <rubriek id="2" naam="patientnummer" titel="Pat.nr.">12345678901</rubriek>
  <rubriek id="22" naam="klinischegevens">
    <span id="crreg">
      <reg>Fixed format line          1.</reg>
      <reg>Fixed formatted line       2.</reg>
      <reg>Another formatted line     3.</reg>
    </span>
  </rubriek>
  <rubriek id="56" naam="conclusie" titel="Konklusie">
    <par>This is line 1 of the conclusion.</par>
    <par>This is line 2 of the conclusion.</par>
    <par>This is line 3 of the conclusion.</par>
  </rubriek>
</rapport>
```

The obligatory attribute ”id” of a ”rapport”, the report name, consists of three elements.

The first letter determines the kind of investigation, the two-digit number after that is the year of the date of receipt, and the five-digit number at the end is a serial number.

The attribute ”mode” is an indication about what kind of ”report” this is:

- ”compleet”, a complete report,

- "partieel", only some "rubriek" fields (elements) of the report included,
- "nummer", no "rubriek" fields (elements) included here,
- "excerpt", the report is in fact an excerpt from the national PALGA database,
- "na", the report is not available or nor existing at all.

The attribute "**status**" signals the administrative status of the report in UDPS; usually "0" means it has just been created, "A" archive status, and other statuses something in between.

Additional "rapport" attributes may be:

- **so**, "soort onderzoek", kind of investigation
- **nummer**, the serial number part of the report name
- **jaar**, the year part of the report name
- **versie**, the version of the report, default first version "A"
- **autorisator**, the user id of the pathologist who authorized the report
- **autts**, the timestamp of authorization
- **autwegts**, the timestamp of removal of authorization
- **autweghaler**, the user id of the user who removed the authorization
- **autwegvrager**, the id of the pathologist who requested removal of authorization
- **autwegreden**, a code for the reason of removal of authorization
- **autmutaard**, a code for the type of change between authorizations
- **autmutpointer**, a pointer to a previous version of the report.

In the UDPS the fields in a report are called "rubriek". A report may contain any number of "rubriek" fields. The name and meaning of these fields are defined locally in UDPS with parameters. There may be differences between different kinds of investigation; a rubriek with the same "naam" (name) may have a different "id" (number) from one kind of investigation to another, and certainly from one laboratory (UDPS) to another. Many rubriek names are common, but some exist only in one or two UDPS.

The XML communication of "rubriek" fields therefore is basically based on the name, the rubriek attribute "naam". The attribute "id" can be missed. The attribute "titel" (title) just gives some additional information on how this rubriek is called in a data-entry screen. It will not always be present.

The "**rubriek**" attributes are:

- **id**, a number
- **naam**, the name or a synonym
- **titel**, the data-entry name
- **soort**, the type of rubriek, which may be
  - code, the rubriek contains a code
  - datum, this is a date of format "yyyymmdd"
  - datumtijd, a date and time of format "yyyymmddhhmmss"
  - dpsdatum, a date of format "dd-mm-yy"

- tekst, a series of ASCII characters (text)
- lang, a long rubriek containing "par" and/or "span" elements.  
Default is "tekst".
- **mode**, what to do with it, which can be
  - default, the action depends on "soort",
  - aanvullen, add the content to existing content
  - overschrijven, overwrite existing content with new one
  - niet\_overschrijven, do not overwrite, leave existing content in place.  
Default is "default"
- **waarde**, the code value if mode="code"
- **uitvoerder**, an id of the pathologist
- **datum**, the date of (the creation of) this rubriek, format "yyyymmdd"
- **assistent**, an id of an resident pathologist
- **supervisor**, an id of an supervisor to the resident
- **secr**, an id of an employee who did typing or correcting work
- **datumsecr**, the date this typing or correcting was done, format "yyyymmdd"

None of these attributes is required, but normally you would expect at least "naam" or "id". Most date-type rubriek fields have "soort=datum".

A "rubriek" element can contain data (text) or some other elements. The value of a "rubriek" field that has a limited length will be text in the data part of the "rubriek" element. The lines of a "rubriek" field with an unlimited length will be in "par" elements. Lines that have been pre-formatted will be in "reg" elements surrounded by a "span" element. This is explained in more detail in the paragraph "The element wijziging".

A special type of a "rubriek" with unlimited length is called "container rubriek". Such a "rubriek" field always starts with the string "CONTAINER", and will contain any number of lines that can be seen as (comma-separated) records in a database table. In the "rubriek" element they are represented by the "container" element. A "rubriek" element may contain one "container" element, and then no other elements are expected. The "rapport" DTD is not very accurate about this.

The "container" element can contain zero or more "key" elements. These "key" elements can contain one or more "item" elements. For example:

```
<container>
  <key id="CONTAINER">
    <item waarde="0" id="CONTAINER">
      <item waarde="1" id="Field1">
        <item waarde="2" id="Field2">
          </key>
        <key id="Key1">
          <item id="Field1" waarde="K1.f1"/>
          <item id="Field2" waarde="K1.f2"/>
        </key>
        <key id="Key2">
          <item id="Field1" waarde="K2.f1"/>
          <item id="Field2" waarde="K2.f2"/>
        </key>
      </key>
    </container>
```

This represents the following table:

CONTAINER	Field1	Field2
Key1	K1.f1	K1.f2
Key2	K2.f1	K2.f2

The first row contains titles of the fields (columns) and the first column contains the keys with which the records (rows) can be found.

In the "container" element the "key" element with attribute "id" "CONTAINER" represents the title row. The "waarde" attribute is the field number of the field with the title that is in attribute "id".

The "container" element has no attributes. The "key" element has only one attribute: "id". The "item" element is more complex, and is similar to the "rubriek" element. It has the following attributes:

- **id**, required, that is the fieldname of the item.
- **titel**, that is the title that is used in a data-entry screen.
- **soort**, the data-type, which can be
  - **code**, the attribute "waarde" will contain a code; a value belonging to that code may be in the data part of the (item) element.
  - **datum**, the attribute "waarde" will contain a date of format "yyyymmdd".
  - **tekst**, the attribute "waarde" will contain a string value.

Default is "tekst"

- **waarde**, this attribute contains the (main) value of the item.
- **extra**, this attribute contains some extra value. This is used to specify some extra value belonging to a code. If "waarde" "9" means "something different, namely", than in "extra" the different thing can be specified. In this case the data part of the element could be "something different, namely: something else". In XML:

```
<item id="what" soort="code" waarde="9" extra="something else">
something different, namely: something else</item>
```

One special "container rubriek" is the one containing CRIS3 records. For this a special XML element has been created, "cris", that is used as in the following example:

```
<rubriek naam="cris">
  <cris versie="3" type="klin">
    <crisitem id="1" naam="datumstrijk" waarde="20030513" soort="datum"/>
    <-- more crisitems -->
  </cris>
  <cris versie="3" type="kopac">
    <crisitem id="1" naam="k" waarde="1" soort="code"/>
  </cris>
</rubriek>
```

The "cris" element can only contain "crisitem" elements, and has attributes:

- **type**, "klin" or "kopac", indicates the type of crisitems to be expected.
- **versie**, a version number, is always "3" at this point of time.

The "crisitem" element can contain data, and has attributes:

- **id**, required, that is the fieldname of the item.
- **naam**, the name of this CRIS3 item
- **soort**, the data-type, which can be
  - "code", the attribute "waarde" will contain a code; a value belonging to that code may be in the data part of the (item) element.
  - "datum", the attribute "waarde" will contain a date of format "yyyymmdd".
  - "tekst", the attribute "waarde" will contain a string value.

Default is "tekst"

- **waarde**, this attribute contains the (main) value of the item.
- **extra**, this attribute contains some extra value. This is used to specify some extra value belonging to a code.

### **Kopieontvanger**

The content of the pseudo rubriek 'kopieontvanger' will be placed in de 'copiedata' container if it contains references to other laboratories and (their) report numbers like this:

```
<rubriek naam="kopieontvanger">lab24, T13-12345; lab25, T13-23456 T13-23457</rubriek>
```

Sets of laboratory references are separated with ';'. A laboratory reference consist of a laboratory code (labN), a ',' and one or more report references separated by spaces.

UDPS will send (XML CDA) reports to these laboratories by email then if also de rubriek 'soortaanvraag' is filled with 'consult tbv derden' and if some system configuration has been done.

So much for the "rapport" element and the elements it may contain.

## 2.3 The berichten and bericht elements

In the UDPS System-X XML link messages are exchanged of which the root element is called "berichten", which is in fact Dutch for "messages".

A "**berichten**" element can (only) contain any number of "bericht" elements; "bericht" is Dutch for (a single) "message". The "berichten" element may have two attributes: "id" and "aantal". The attribute "**id**" can be used for future reference, attribute "**aantal**" may be used to state the number of "bericht" elements in this "berichten".

The "**bericht**" element may have three attributes: "id", "aan" and "van". The attribute "**id**" is required and is used as a reference (id) in the "bericht" in the answer message. The attribute "**aan**" is a sort of "to" header, and "**van**" a "from" header. The server will exchange the content of "aan" en "van" in the answer message.

So, if the client sends this message to the server:

```
<berichten id="bers" aantal="2">
  <bericht id="b1" aan="U" van="X"> ... </bericht>
  <bericht id="b2" aan="U" van="X"> ... </bericht>
</berichten>
```

then the server will respond with the following message:

```
<berichten id="bers" aantal="2">
  <bericht id="b1" aan="X" van="U"> ... </bericht>
  <bericht id="b2" aan="X" van="U"> ... </bericht>
</berichten>
```

The id in the "bericht" and "berichten" in the answer message from the server refers directly to the id of the corresponding elements in the request message of the client.

The "bericht" element is a container for orders and answers, and it may contain any number of elements that represent an order ("vraag", "creatie", "verwijder", "wijziging", "drcvraag", "functie") in the message from the client, and just as many elements that represent an answer ("antwoord") in the message from the server to the client:

```
<bericht id="b1">
  <vraag id="v1"> ... </vraag>
</bericht>

<bericht id="b1">
  <antwoord id="v1"> ... </antwoord>
</bericht>
```

Before describing the order elements in later paragraphs, the "antwoord" element will be described in the next paragraph.



## 2.4 The element "antwoord"

An "antwoord" element in the answer message has a one to one relation with an order element in the request message. Every order leads to one answer and the id of the answer (attribute "id" of "antwoord") is the same as the id of the order (attribute "id" of the order element).

In "antwoord" the attribute "**id**" is required, and is the same as the id of the triggering order. The attribute "**type**" signals success. If the execution of the order for some reason failed, the "type" will be "nack". Generally one or more "fout" elements will be included in "antwoord" then, containing error messages. If execution succeeded then "type" will be "ack", unless the order implied the sending of data from the server to the client, in which case "type" will be "data".

The element "antwoord" can contain zero or more of the following elements:

- **fout**. This error element has attribute "id" which determines the type of error. The content of this element is the error message.
- **waarschuwing**. This warning element has attribute "id" which determines the type of warning. The content of this element is the warning message.
- **tekst**. This text element can contain one or more "span" and "par" elements, and has attributes "id" and "type". This may be used in the setting of some "functie".
- **rapporten**. This container element contains zero or more "rapport" elements.
- **tabellen**. This container element contains zero or more "tabel" elements (described later).
- **drc**. This element contains one or more "diagnose" elements (described later).
- **result**. This container element contains zero or more "rapport" elements, that are used to carry the result of a query to the national PALGA database, or "fout" elements, signaling errors. Attributes of "result" are "id", the report name of the triggering report, "aantal", the number of "rapport" elements, and "type" which can be "follow", "incid", "xrp" or (default) "result".
- **gebruikers**. This element is can contain zero or more 'gebruiker' elements.
- **groepen**. This element can contain zero or more 'groep' elements.

## 2.5 The element "creatie"

With the creation order "creatie" a new pathology report can be created on UDPS. The client must have the right permissions though. This element "creatie" knows the following attributes:

- **id.** This attribute is required, and will be referenced in the answer.
- **rapport.** This is the report name (Xyy-nnnnn) of the report to be created and is required.
- **status.** This is the administrative status of the report to be created, a single character, default "0".
- **datumontvangst.** This is the date of receipt of the report to be created. This date must have the format "yyyymmdd" and the year in it must match the year part of the report name. Default the current system date is used.

A "creatie" element may contain zero or more "rubriek" elements. Only if a new report can be created, these "rubriek" fields are inserted to the report. If such report exists, and therefore cannot be newly created, then insertion of these "rubriek" fields is skipped, and the report is left unchanged.

If the client wants to insert these fields anyway, it should include a "wijzigen" order element right after the "creatie". Then "wijzigen" is independent of the success or failure of the "creatie".

The following example illustrates how to create new reports in UDPS.

```
<berichten id="bers">
  <bericht id="b1" aan="UDPS" van="X">
    <creatie id="c1" rapport="T03-00001" status="1"
      datumontvangst="20030102"/>
    <creatie id="c2" rapport="T03-00001"/>
    <creatie id="c3" rapport="T03-90001">
      <rubriek naam="patientnummer">1234567</rubriek>
    </creatie>
  </bericht>
</berichten>
```

The server will respond:

```
<berichten id="bers">
  <bericht id="b1" aan="X" van="UDPS">
    <antwoord id="c1" type="ack"/>
    <antwoord id="c2" type="nack">
      <fout id="creatie_a">Rapport bestaat al T03-00001</fout>
    </antwoord>
    <antwoord id="c3" type="ack"/>
  </bericht>
</berichten>
```

## 2.6 The element "verwijder"

With the delete order "verwijder" a pathology report can be removed from UDPS. The client must have the right permissions though. This element "verwijder" knows the following attributes:

- **id**. This attribute is required, and will be referenced in the answer.
- **rapport**. This is the report name (Xyy-nnnnn) of the report to be created and is required.

There are several restrictions on removing a report from UDPS due to the fact that the existence of the report may have led to consequences: a report may have been sent to the requester, a billing record may have been sent, an excerpt may have been sent to the national database where it can not be removed. The following restrictions are implemented:

- The administrative status must be "0" or "7", indicating a newly created report.
- The report version must be "A", indicating the first (and only) version.
- The report may not be authorized.
- The status byte "uitslag" must not be "g" or "G"
- The status byte "cisuitslag" must not be "g" or "G"
- The status byte "factuur" must not be "g" or "G"
- The status byte "cipa" must not be "g" or "G"

The "wijzigen" order can be used of course to change the report prior to the "verwijder" order. If a report has been sent to the national database it should not be removed from UDPS. It can be "emptied" though by putting some predefined content in it. See the "green book" (het groene boekje). Here is an example of the "verwijder" order:

```
<berichten>
  <bericht id="b1" van="X" aan="U">
    <verwijder id="vw1" rapport="T03-00001"/>
  </bericht>
</berichten>
```

## 2.7 The element "wijziging"

Before describing how to update reports using this XML link, one thing must be made clear. At this point of time, when "rubriek" fields are updated using this XML link, the client is fully responsible for the internal integrity of the report. In a data-entry setting changing one "rubriek" field may be refused due to restrictions or may lead to an automatic update of some other "rubriek" field. This XML link does not support those functions. In UDPS however there is another server that does support these functions, the server for the graphical client. Probably in the future updates using this link will internally be redirected to the other server. As a result of that, the behavior of this link will change: the effect on a report can be different and other error message can be expected.

With the order element "wijziging" a client can change a report on the server. The element can contain any number of "rubriek" and "statusbyte" elements. See the paragraph about the pathology report for a description of "rubriek". The following attributes belong to "wijziging":

- **id**. This attribute is required and will be referenced in the answer message.
- **rapport**. This required attribute (Xyy-nnnnn) is the report name of the report that must be changed.
- **status**. The administrative status the report should have on the server after the changes have been committed, a single character. Default the status does not change.
- **mode**. This attribute is default 'update'. On a Core-UDPS if it is 'update-aut' the report will have the label 'authorized' (statusbyte 'eind' = 'j').
- **reden**. If this attribute is 'bsn-update' the report will be updated in de PALGA databases, but it will not be sent to some third parties like "DICA" and "IKNL".  
In that case 'postcode' and 'woonplaats' should not be included in the message.
- 

A "wijziging" can only be applied to an existing report (on the server). If the report does not exist on the server, or if some other failure occurred, the "antwoord" will have a type "nack", en will contain a "fout" element. Otherwise success is indicated by "antwoord" type being "ack". The following example will illustrate this.

```
<berichten>
<bericht id="b1" van="X" aan="U">
  <wijziging id="w1" rapport="T03-00001">
    <rubriek naam="patientnummer">9000001</rubriek>
    <rubriek naam="naamman">Superman</rubriek>
  </wijziging>
  <wijziging id="w2" rapport="T03-00001">
    <rubriek naam="patientnummer">9000001</rubriek>
    <rubriek naam="naamman">Superman</rubriek>
  </wijziging>
  <wijziging id="w3" rapport="T03-00002">
```

```

    <rubriek naam="patientnummer">9000001</rubriek>
    <rubriek naam="naamman">Superman</rubriek>
  </wijziging>
</bericht>
</berichten>

```

The server will respond:

```

<berichten>
<bericht id="b1" van="U" aan="X">
  <antwoord id="w1" type="ack"/>
  <antwoord id="w2" type="nack">
    <fout id="rubriek_so_na">Rubriek niet gedefinieerd voor
      soort onderzoek patientnummer</fout>
  </antwoord>
  <antwoord id="w3" type="nack">
    <fout id="rap_norap">Rapport niet gevonden T03-00002</fout>
  </antwoord>
</bericht>
</berichten>

```

In this "wijziging" context, the "rubriek" element may have an attribute "**mode**" which may be "default", "aanvullen", "overschrijven", or "niet\_overschrijven".

The effect of this "mode" is dependent on the type of "rubriek" field, not so much the "soort" that was stated as an attribute, but the real type as defined in the server or found in the report in the server. A "rubriek" field is defined as "kort" (short) if it may contain one line of text with a certain maximum length. If the maximum length is defined as "0" it may contain an (practically) unlimited number of lines of unlimited length. This type of "rubriek" field is called "lang" (long).

Default a "rubriek" field that is "kort" will be overwritten by the content of a "rubriek" element. The mode "aanvullen" will not have effect, the "rubriek" field will be overwritten anyway; it is impossible to append to a "rubriek" field that is "kort". The mode "niet\_overschrijven", Dutch for "do\_not\_overwrite", has the effect that a "rubriek" field that was not empty is left unchanged.

If a "rubriek" field is defined "lang", default the content of a "rubriek" element is appended to the field, with an empty line inserted if the field was not empty. If mode is "aanvullen", append, then an extra line "Aanvulling d.d. *date*" is inserted before the appended text. If mode is "overschrijven" then the content of the "rubriek" element will replace the content of the "rubriek" field. Mode "niet\_overschrijven" has no effect because it is the default here.

If the client orders "aanvullen" or "overschrijven" with respect to a "rubriek" field that was empty, a warning is issued, using the element "waarschuwing", which is like the error element "fout", but is meant less severe. Some other situations may lead to warnings as well.

If the "rubriek" elements attribute "soort" is "code", the value of the attribute "waarde" will be put into the "rubriek" field. If the element has a text content, that will be appended to it, separated by ": ", the "rubriek" field will be "code: text".

If the "rubriek" element's attribute "soort" is "datum", the content will be transformed to a "dpsdatum" and then put into the "rubriek" field.

If , in addition to that attribute "naam" is "geboortedatum" (date of birth), the "rubriek" field "geboorteeeuw" (century of birth) will be set as well, but not the "rubriek" field "leeftijd" (age) that in a data-entry setting usually is calculated comparing date of birth with the date of receipt. So the client should also send a "rubriek" "leeftijd" in that case.

If the "rubriek" element contains "par" or "span" elements, these will be unpacked.

A "**par**" will be one line in the "rubriek" field, followed by a (hard) line separator. Upon display and printing these lines can be formatted, inserting (soft) line separators and using a variable width font.

A "**span**" will contain zero or more "reg" elements. This is intended for pre-formatted lines. In the "rubriek" field each "reg" will lead to one line, followed by a (hard) line separator. But these lines will be surrounded by some mark-up that has the effect that printing and displaying programs do not try to reformat these lines, and that fixed width font will be used for printing and display.

A "rubriek" element may have some attributes stating "who has done it, and when". Some "rubriek" fields of type "lang" have associated "rubriek" fields of type "kort". The UDPS server will intelligently try to update these associated fields if the corresponding attributes exist in the "rubriek" element.

The element "wijziging" can also contain element "**statusbyte**". This is meant to change status bytes in the "rubriek" field "statusrubriek". In a table in UDPS (autopositie) status bytes have been defined with their position in "statusrubriek", usually "eind", "uitslag" and "factuur", and sometimes "assistent" and "deel". Other status bytes are rare. Such an element could be: "<statusbyte naam="assistent" waarde="n"/>".

The attributes of the element "statusbyte" are:

- **id**. The value of this attribute is the number of the "rubriek" field in which a byte must be set. A client should probably not use this id, but rely on UDPS for determining this number, or set a different "rubriek" field name in the attribute "rubriek".
- **naam**. This is the name of the statusbyte, "factuur", "uitslag" etcetera.
- **waarde**. This is the value that the byte must get, a character like "j".
- **rubriek**. With this attribute some other "rubriek" field can be assigned to contain the statusbyte. The default is "statusrubriek".

In UDPS the statusbyte "eind" cannot be set this way. This authorization status is primarily an attribute (AUTTS) of the report, residing in "rubriek" field 0

("rubriek0"). The statusbyte "eind" in "statusrubriek" is for historical and practical reasons still there, but as a derived value.

In UDPS permissions (parameters) determine which client can change which reports and which "rubriek" fields within those reports. Trying "wijziging" with a report name or "rubriek" name that does not match the permission will lead to receiving a "nack" with a "fout". The report will be left untouched.

The next example illustrates adding BSN and related data to existing reports. Here the attribute 'reden' has value '**bsn-update**'. The report will be updated in de PALGA databases, but it will not be sent to some third parties like "DICA" and "IKNL". In this case 'postcode' and 'woonplaats' are not (and must not be) included in the message.

```
<?xml version="1.0" encoding="utf-8"?>
<berichten>
  <bericht van="LMS" aan="kernDPS" id="353">
    <wijziging rapport="S16-00009" id="1" reden="bsn-update">
      <rubriek naam="geslacht">V</rubriek>
      <rubriek naam="voorletters">D</rubriek>
      <rubriek naam="naamvrouw">Duck</rubriek>
      <rubriek naam="vvnaamvrouw"/>
      <rubriek naam="naamman">Slager</rubriek>
      <rubriek naam="vvnaamman">de</rubriek>
      <rubriek naam="geboortedatum" soort="datum">19690809</rubriek>
      <rubriek naam="bsnummer">600012345</rubriek>
      <rubriek naam="statusbsn">ZIS</rubriek>
    </wijziging>
  </bericht>
</berichten>
```

## 2.8 The element "pfcontrole"

The order element 'pfcontrole' can be used to check whether a 'wijziging' with the same data in Core-UDPS and status '8' would be successful. In fact, it is implemented as such a 'wijziging' order, the error messages are the same, or the acknowledge is the same, but then the changes will not be saved in the UDPS database. Example:

Order:

```
<pfcontrole id="opdracht-id-1" rapport="T10-00001" >
  <rubriek naam="patientnummer">9000001</rubriek>
  <rubriek naam="naamman">Superman</rubriek>
</pfcontrole>
```

Answer:

```
<antwoord id="w1" type="ack"/>
```

Or something like:

```
<antwoord id="w2" type="nack">
  <fout id="identificer">message</fout>
</antwoord>
```

## 2.9 The element "vraag"

The order element "vraag", Dutch for "question", gives a client the opportunity to query the server for pathology reports. Several ways exist to do that; here are some examples (leaving the "bericht(en)" bit aside):

```
<vraag id="v1" rapport="T03-00001" geaut="beide"/>
```

```
<vraag id="v2" rapport="T03-00001" geaut="nee">
  <rubriek naam="statusrubriek"/>
  <rubriek naam="datummacroscopie"/>
  <rubriek naam="datumconclusie"/>
</vraag>
```

```
<vraag id="v3" type="nummers">
  <selectie laagste="T02-00001" hoogste="T02-99999"/>
  <selectie naam="codeaanvrager" waarde="HUI"/>
</vraag>
```

The first "vraag" (v1) could result to the answer:

```
<antwoord id="v1">
<rapporten aantal="1">
  <rapport id="T03-00001" status="0">
    <rubriek id="1" naam="datumontvangst"
      soort="datum">20030102</rubriek>
    <!--any non-empty "rubriek" will follow -->
  </rapport>
</rapporten>
</antwoord>
```



The second "vraag" (v2) could, provided that the report in question has not yet been authorized, result to the answer:

```
<antwoord id="v2">
<rapporten aantal="1">
  <rapport id="T03-00001" status="0">
    <rubriek id="44" naam="datummacroscopie"
      soort="datum">20030103</rubriek>
    <rubriek id="56" naam="statusrubriek">nnngn_____</rubriek>
  </rapport>
</rapporten>
</antwoord>
```

Only the "rubriek" fields that were element of the "vraag" are delivered in any resulting "rapport".

The third "vraag" (v3) could result to the answer:

```
<antwoord id="v3">
  <rapporten aantal="3">
    <rapport id="T02-05678" status="A"/>
    <rapport id="T02-06231" status="A"/>
    <rapport id="T02-09770" status="A"/>
  </rapporten>
</antwoord>
```

The "vraag" attribute "type" has value "nummers", and so the "rubriek" elements of "rapport" are suppressed, only the report numbers and status are shown.

The element "vraag" can have the following attributes:

- **id.** This attribute is required and will be referenced in the resulting "antwoord".
- **type.** This can be "nummers" or (default) "rapporten". If "rapporten" then "rubriek" elements will be included in "rapport" elements. If "nummers" then only the "rapport" element tag will be shown.
- **geaut.** This is a selection on the authorization status of reports. The value can be "ja" (yes), "nee" (no) or "beide" (both). Default is "ja", and then only authorized reports are shown. If "no" then only reports are shown that have not yet been authorized. If "both" all reports are shown.  
The client must have a permission to receive certain reports at all, and a separate permission to receive certain non-authorized reports. These permissions use pattern matching related to the report name, so some reports may, and some may not be shown.
- **rapport.** This attribute can be set with a report name if only one report is ordered in each "vraag".
- **max.** The value is the maximum number of reports to be delivered. The default is 500.
- **maxz.** The value is the maximum number of reports being searched. That means that a first selection using a database key, will deliver at maximum "maxz" reports on which further selections are committed.

As we have seen, a "vraag" element should either have a "rapport" attribute, or contain one or more "selectie" elements, that are used to perform a selection of reports.

The element "**selectie**" has attributes, but no further content. Attributes are:

- **laagste**. The "lowest" report name. Report names are alphabetically compared to this value.
- **hoogste**. The "highest" report name. Report names are alphabetically compared to this value.
- **eerste**. The first date of receipt of the selection of reports. A report with a date before this value is not selected. The format is "yyyymmdd".
- **laatste**. The last date of receipt of the selection of reports. A report with a date after this value is not selected. The format is "yyyymmdd".
- **status**. The administrative status of the report must have a value that is in the string that is the value of this attribute. For example, "0715" means status "0", "7", "1" or "5".
- **type**. This can be "attribuut", "key" or (default) "rubriek". If "key" then this item is considered to be a key to the database and is used for the primary search. If "attribute" then this item is a "rubriek0" attribute of a report, and if "rubriek" the item is considered to be a "rubriek" field.
- **naam**. The value of this attribute is the name of the "rubriek" field, report attribute or database key
- **operator**. This attribute defines the method with which the report values will be compared to the "selectie" attribute "waarde" value.
- **waarde**. This attribute is the value with which the content of the reports "rubriek", attribute or key is compared.

Different attributes may be in the same "selectie" element. If two or more "selectie" elements are found, these selections will have an "AND" relation to each other. Of course different selections involving anything other than report name, status and date of receipt must be in different "selectie" elements.

Every set of "selectie" elements must at least include one "key" selection, that is used the primary set of reports to which further selections are committed. The attributes "laagste", "hoogste", "eerste", "laatste" and "status" implicitly are key selections; report number, status and date of receipt are always database keys.

If some other key is defined through attribute "type" that one will be used for the primary selection:

Else, if "status" is defined, the administrative status is used.

Else, if "laagste" and/or "hoogste" are defined, the report name is used.

Else, if "eerste" and/or "laatste" is defined, the date of receipt will be used.

Else, the "antwoord" element will have type "nack" and a "fout" element, saying that no key was defined.

More than one explicit key will also lead to an error message.

The "**operator**" attribute has no effect on keys. A key is always searched comparing strings up to the length of the "waarde" attribute. If it is looking for "jan", the server will find "jan", "jans", "jansson" and "january", but not "mmm" or "ja".

For other types the "operator" attribute may be:

- **gelijk**, equals, the report item must be equal to the attribute "waarde" value.
- **ongelijk**, not equals, the report item must be something different then the attribute "waarde" value.
- **groter**, greater, the value of the report item must be greater then the value of the attribute "waarde"
- **kleiner**, smaller, the value of the report item must be smaller then the value of the attribute "waarde"
- **groter-of-gelijk**, greater or equal, the value of the report item must be greater then or equal to the value of the attribute "waarde".
- **kleiner-of-gelijk**, smaller or equal, the value of the report item must be smaller then or equal to the value of the attribute "waarde".
- **matcht**, matches, the value of the report item must match the value of the attribute "waarde"
- **matcht-niet**, not matches, the value of the report item must not match the value of the attribute "waarde"

## 2.10 The element "parsepp"

The order element "parsepp" contains a "rapport" element with a "rubriek" element with attribute "naam" equal to "microscopie". The content of this "rubriek" is parsed and, if it contains valid protocol data, as a result an "antwoord" is produced that includes a "rapport" with two "rubriek" elements: "protocolnaam" and "protocoldata".

If it contains protocol data that are not valid, the "antwoord" will contain a "fout". If it did not contain any protocol data, the antwoord will be empty.

Order:

```
<berichten>
<bericht profile="pcp" id="bericht-id-1" van="X" aan="DPS">
<parsepp id="opdracht-id-3">
<rapport><rubriek naam="microscopie" soort="lang"><par>De eerste twee regels
zijn vrije tekst.</par><par>Niet alles is protocollaire verslaglegging
immers.</par><par>protocollaire verslaglegging
okselklierdissectie</par><par>zijdigheid : rechts</par><par>status na swk :
negatief</par><par>aantal swk : 0</par><par>aantal macrometastasen in swk :
1</par><par>aantal micrometastasen in swk : 0</par><par>aantal klieren:
12</par><par>aantal klieren met geïsoleerde tumorcellen : 0</par><par>aantal
klieren met micrometastasen van 0.2 tot 2 mm : 1</par><par>aantal klieren met
metastasen groter dan 2 mm : 0</par><par>topklier positief : nee</par><par>massale
extranodale groei : nee</par><par>einde protocollaire verslaglegging
okselklierdissectie</par></par></rubriek></rapport>
</parsepp>
</bericht>
</berichten>
```

Result:

```
<berichten><bericht id="bericht-id-1" aan="X" van="DPS">
<antwoord type="nack" id="opdracht-id-3"><fout id="f1">Het aantal positieve
schildwachtklieren kan niet groter zijn dan het totale aantal
schildwachtklieren.</fout></antwoord>
</bericht></berichten>
```

Or with a slightly different (corrected) order:

```
<berichten><bericht id="bericht-id-1" aan="X" van="DPS"><antwoord type="ack"
id="opdracht-id-3"><rapport id="X00-00001" mode="update">
<rubriek naam="protocolnaam">okselklierdissectie</rubriek>
<rubriek naam="protocoldata">
<pitem id="versie" vartype="">02062011</pitem>
<pitem id="zijdigheid" vartype="string">rechts</pitem>
```

```

<pitem id="status na swk" vartype="string">negatief</pitem>
<pitem id="aantal swk" vartype="integer">0</pitem>
<pitem id="aantal macrometastasen in swk" vartype="integer">0</pitem>
<pitem id="aantal micrometastasen in swk" vartype="integer">0</pitem>
<pitem id="aantal klieren" vartype="integer">12</pitem>
<pitem id="aantal klieren met geïsoleerde tumorcellen" vartype="integer">0</pitem>
<pitem id="aantal klieren met micrometastasen van 0.2 tot 2 mm" vartype="integer">1</pitem>
<pitem id="aantal klieren met metastasen groter dan 2 mm" vartype="integer">0</pitem>
<pitem id="topklier positief" vartype="boolean">Nee</pitem>
<pitem id="massale extranodale groei" vartype="boolean">Nee</pitem>
</rubriek>
</rapport></antwoord></bericht></berichten>

```

Or with order:

```

<berichten>
<bericht profile="pcp" id="bericht-id-1" van="X" aan="DPS">
<parsepp id="opdracht-id-3">
<rapport><rubriek naam="microscopie" soort="lang"><par>De eerste twee regels
zijn vrije tekst.</par><par>En dit is nog &#xE9;&#xE9;n opmerking buiten het
protocol om.</par></rubriek></rapport>
</parsepp>
</bericht>
</berichten>

```

The result is:

```

<berichten><bericht id="bericht-id-1" aan="X" van="DPS"><antwoord type="ack"
id="opdracht-id-3"></antwoord></bericht></berichten>

```

## 2.11 The element “functie”

The order element ”functie” gives a flexible way of ordering all kinds of different things to the server, like serving and updating of tables, serving results of other links or even directly using these links, where the server translates this XML protocol to the protocol used by that link.

Currently the following functions have been defined in Core-UDPS:

- cipavraag, do a query to the national PALGA database.
- ciparesult, deliver results from a query to the national PALGA database.
- incidentie, retrieve information on first (cancer) incidence, number file names or content
- followup, retrieve follow-up information, number file names or content
- triggers, retrieve information on occurrence of changes in reports
- spoolfiles, retrieve UDPS system message files

- drcvraag, check diagnosis lines using PALGA Diagnoseregel Controle Module

These functions will be described in detail later. Other function may be added, in UDPS and as locally defined functions.

The element "functie" has the following attributes:

- **id**. This element is required and will be referenced in the "antwoord" element.
- **naam**. This element is required. It is the name of the function to be performed.
- **rapport**. This is the report name of the report that will be used in the function. For some function this attribute may be required, for others it is meaningless. In some functions this is not a fully qualified report name, but a pattern with which report names are matched.

A "function" element usually will contain one or more "**argument**" elements. These elements provide arguments to the function that could not as flexible have been implemented as attributes or otherwise. These arguments are just strings that must be interpreted by the function.

An "argument" element has one attribute, "**id**", which has not been used yet in any function. The content is used as argument string.

The content of the resulting "antwoord" is dependant on the function that has been called.

## 3 Functions

### 3.1 Function "cipavraag"

The function "cipavraag" (patient) queries the national CIPA database:

```
<functie naam="cipavraag" id="opdracht-id-1">
  <rapporten>
    <rapport id="T01-00001"/>
    <rapport id="B01-00001"/>
  </rapporten>
</functie>
```

uses the patient information in the specified reports for the query, and

```
<functie naam="cipavraag" id="opdracht-id-2" rapport="B02-00001"/>
```

uses the patient information in report B02-00001.

The result might be:

```
<antwoord mode="ack" id="opdracht-id-1">
  <waarschuwing id="w_cipavraag">Waarschuwing: geslacht leeg
    in T01-00001</waarschuwing>
</antwoord>
<antwoord mode="ack" id="opdracht-id-2"/>
```

The actual results of the patient query to the PZVDB can be fetched later with function "ciparesult".

This function can also be called in a synchronous mode:

```
<functie naam="cipavraag" id="op-id-2" synchroon="1" rapport="X18-12345">
  <argument>geboortedatum=27-12-60</argument>
  <argument>geboorteeeuw=19</argument>
  <argument>geslacht=v</argument>
  <argument>naamvrouw=dame</argument>
  <argument>naamman=</argument>
  <argument>voorletters=A.B.C.</argument>
  <argument>bsnummer=999999199</argument>
</functie>
```

In that case the answer will be that of the next function 'ciparesult'. The rapport attribute must be filled with a reference that has the format of a report but it should not be the name of an existing report unless intended.

### 3.2 Function "ciparesult"

The function "ciparesult" delivers the result of the patient query to the national PALGA database. For example, the order

```
<functie id="f1" naam="ciparesult" rapport="T03-00001"/>
```

delivers the result from PZVDB for (the patient in) report T03-00001, and the order

```
<functie id="f2" naam="ciparesult" rapport="T02-123"/>
```

gives the results for reports T02-12300 up to T02-12399, and the order

```
<functie naam="ciparesult" id="opdracht-id-1">
<rapporten>
  <rapport id="T01-00001"/>
  <rapport id="B01-00001"/>
</rapporten>
</functie>
```

gives the results belonging to the specified reports.

The "antwoord" element will contain zero or more "result" elements and/or "fout" elements. Such a "result" element is a container for "rapport" elements and "fout" elements. It has the following attributes:

- **id.** This is the report name of the triggering report, of which the patient data were used to query the national database.
- **aantal.** The number of "rapport" elements in this "result" element.
- **tijd.** The time of generation of the result.
- **type.** The type of result. This may be:
  - Result. The result of the patient query.
 At this point of time "result" is default, and the only implemented option.

### 3.3 Function "incidentie"

The function "incidentie" delivers the information on first (cancer) incidence that UDPS has received from the PZVDB. For example, the order

```
<functie id="f1" naam="incidentie"/>
```

delivers the first incidence information that was received from PZVDB last weekend:

```
<antwoord id="f1">
  <result id="T03-12345" type="incid" aantal="3"
    tijd="01-12-03 om 10 uur 44">
    ...
  </result>
</antwoord>
```



The content of the result will be ‘rapport’ elements, as in the example in the next paragraph ‘follow-up’.

The function:

```
<functie id="f2" naam="incidentie" type="files"/>
```

delivers the names of all related number files in a ”tabellen” element:

```
<antwoord mode="data" id="f2">
  <tabellen>
    <tabel id="/u/nrfiles">
      <item id="incid" waarde="1"/>
    </tabel>
  </tabellen>
</antwoord>
```

The ”argument” sub elements of ”vraag” can be used as (regular expression) selector:

```
<functie id="f3" naam="incidentie">
  <argument id="1">i\.20031001</argument>
  <argument id="1">i\.20031008</argument>
</functie>
```

will deliver the information belonging to the report numbers in the number files ”i.20031001” and ”i.20031008”. If there are more arguments, they have a ”or” relation with each other.

### 3.4 Function “followup”

The function ”followup” delivers the follow up information that UDPS has received from the PZVDB. The details are the same as in ”incidentie”, only do follow up number files start with ”f” instead of ”i”, and ‘follow’ instead of ‘incid’.

An example of a follow-up result:

```
<antwoord id="fup2">
<result id="B03-10633" type="follow" aantal="5" tijd="24-05-04 om 8 uur 49">
<rapport lab="Zwolle" id="T04-12345" mode="excerpt">
<rubriek naam="datumontvangst">26-03-2004</rubriek>
<rubriek naam="voorletter">N</rubriek>
<rubriek naam="geboorteplaats">onbekend</rubriek>
<rubriek naam="woonplaats">Datdorp</rubriek>
<rubriek naam="leeftijd">043</rubriek>
<rubriek naam="conclusie">
<par>Dikke naaldbipten mamma rechts: adenocarcinoom, graad 2, Her2 sterk
positief (3 plus). Oestrogenreceptor echter zwak positief, hooguit 2 plus.</par>
</rubriek>
<rubriek naam="vrij1">sophia</rubriek>
<rubriek naam="vrij3"></rubriek>
```

```

<rubriek naam="diag1">mamma*rechts*biopt*adenocarcinoom </rubriek>
</rapport>
<rapport lab="Lely" id="T04-98765" mode="excerpt">
<rubriek naam="datumontvangst">08-03-2004</rubriek>
<rubriek naam="voorletter">N</rubriek>
<rubriek naam="geboorteplaats">onbekend</rubriek>
<rubriek naam="woonplaats">Datdorp</rubriek>
<rubriek naam="leeftijd">043</rubriek>
<rubriek naam="conclusie">
<par>Biopt mamma rechts: geringe hoeveelheid materiaal zonder argumenten voor
maligniteit. Immuno volgt.</par>
<par></par>
<par>AANVULLEND BERICHT:</par>
<par>Immunohistochemisch evenmin argumenten voor maligniteit.</par>
<par>(12-03-04/JSvdL/bn)</par>
</rubriek>
<rubriek naam="rz">11</rubriek>
<rubriek naam="diag1">mamma*rechts*biopt*geen
maligniteit*immunohistochemie* geen goed materiaal</rubriek>
</rapport>
<rapport lab="" id="B03-11111" mode="excerpt">
<rubriek naam="datumontvangst">23-10-2003</rubriek>
<rubriek naam="voorletter">N</rubriek>
<rubriek naam="geboorteplaats">onbekend</rubriek>
<rubriek naam="woonplaats">Datdorp</rubriek>
<rubriek naam="leeftijd">043</rubriek>
<rubriek naam="conclusie">
<par>Pap 2. Abnormale plaveiselepitheelcellen.</par>
<par></par>
<par>Advies: Herhaling over 6 maanden. </par>
</rubriek>
<rubriek naam="_cris">versie=3, door=1:huisarts, instrument=1:cervexbrush,
aspect=1:normaal, aanleiding=4:vervolgonderzoek, klachten=0:geen,
ingrepen=0:geen, datumlm=20031001, patroon=1:regelmatig,
anticonceptie=1:hormonaal, hormoongebruik=0:geen, herhaling=2</rubriek>
<rubriek naam="_kopac">Kopacb: 49211B2</rubriek>
<rubriek naam="tv">30</rubriek>
<rubriek naam="hf">1</rubriek>
<rubriek naam="diag1">cervix*uitstrijk*systeem*ec+sqm*asp.ont*abn.plav*
g.a*g.a*pap2</rubriek>
</rapport>
</result>
</antwoord>

```

### 3.5 Function “consultuitslag”

The function “consultuitslag” delivers the URL’s of “consultuitslagen”, reports of pathology (consultation, revision, panel) performed in an external laboratory, that were sent to and stored on this UDPS.

```

<functie id="f1" naam="consultuitslag" rapport="T13-12345" type="b64"/>

<antwoord id="f1">
  <consultuitslagen rapport="T13-12345">
    <consultuitslag id=" lab_T13-11111A" url="/client/consult/T13-12345_lab_T13-11111A">
      <clinicalDocument>Here goes the base64 encoded XML CDA report
    </clinicalDocument>
    </consultuitslag>
    <!--more consultuitslag elements ->
    </consultuitslagen>
  </antwoord>

```

If the attribute “type” equals “b64” then the XML CDA reports are included (base64 encoded), else the “clinicalDocument” elements will be omitted.

The last character of the value of consultuitslag attribute “id” is the version of the report, where ‘A’ is the first ‘B’ the second etcetera.

The permission ‘functie\_consultuitslag’ that is required here, can also be used to gain access in a browser to the reports the URL points to.

### 3.6 Function “spoolfiles”

In DPS most important information for users that is generated by automatic processes is sent to 'spool files' that a user can print when the time is right.

System X can fetch these files from DPS using the XML server with the function 'spoolfiles'. Here is an example. The first 'functie' retrieves the id of all existing spool files. The second does two things:

- order the content of one file.
- order deletion of another file

The content of a file is packed in a 'tekst' element, which is like a long 'rubriek' in a 'rapport'. System X should delete spool files on DPS that were successfully sent over from DPS. Deletion is implemented as moving to another directory by the way.

If anything goes wrong, you get a 'fout' element within the resulting 'spoolfile' element.

Here is the example:

```

<?xml version="1.0" ?>
<berichten>
<bericht id="bericht-id-1" van="X" aan="DPS">
<functie naam="spoolfiles" id="opdracht-id-1">
</functie>
<functie naam="spoolfiles" id="opdracht-id-2">

```

```

<spoolfiles>
<spoolfile view="1" id="/var/spool/printers/1/cis317191900"
printer="1"/>
<spoolfile delete="1" id="/var/spool/printers/1/cis322131501"
printer="1"/>
</spoolfiles>
</functie>
</bericht>
</berichten>

```

The result may be:

```

<?xml version="1.0" ?>
<berichten>
  <bericht van="DPS" aan="X" id="bericht-id-1">
    <antwoord type="data" id="opdracht-id-1">
      <spoolfiles>
        <spoolfile id="/var/spool/printers/1/cis294131501"
printer="1"/>
        <spoolfile id="/var/spool/printers/1/cis301131501"
printer="1"/>
        <spoolfile id="/var/spool/printers/1/cis302131501"
printer="1"/>
        <spoolfile id="/var/spool/printers/1/cis317131501"
printer="1"/>
        <spoolfile id="/var/spool/printers/1/cis317191900"
printer="1"/>
      </spoolfiles>
    </antwoord>
    <antwoord type="data" id="opdracht-id-2">
      <spoolfiles>
        <spoolfile id="/var/spool/printers/1/cis317191900"
printer="1" view="1">
          <tekst>
            <span>
              <reg>first line</reg>
              <reg>next line</reg>
              <reg>more lines ..</reg>
            </span>
          </tekst>
        </spoolfile>
        <spoolfile delete="1"
id="/var/spool/printers/1/cis322131501"
printer="1">
          <fout id="file_na">Delete onmogelijk, file is niet
aanwezig: /var/spool/printers/1/cis322131501</fout>
        </spoolfile>
      </spoolfiles>
    </antwoord>
  </bericht>
</berichten>

```

### 3.7 Function “triggers”

The function “triggers” returns in the “antwoord” a list of events, notifications that a report has been changed. These notifications are called “triggers”, and in the “antwoord” element they are “trigger” elements packed in a “triggers” element like this:

```
<berichten>
  <bericht id="bericht-id-1" van="X" aan="DPS">
    <functie naam="triggers" id="opdracht-id-3"/>
  </bericht>
</berichten>
```

The server responds:

```
<?xml version="1.0" ?>
<berichten>
  <bericht van="DPS" aan="X" id="bericht-id-1">
    <antwoord type="ack" id="opdracht-id-3">
      <triggers>
        <trigger status="2" aut="d" rapport="T04-00001"
tijd="20040924165409"/>
        <trigger status="2" aut="" rapport="T04-00016"
tijd="20040928133328"/>
      </triggers>
    </antwoord>
  </bericht>
</berichten>
```

The attributes of the element “trigger” have the following meaning:

- rapport: the name of the report that has been changed
- status: the new administrative status of the report on disk (0-9,A)
- aut: the authorization status of the report:
  - “”: none, the report has not yet been authorized
  - “a”: the report is authorized
  - “d”: the report has been deauthorized
- tijd: the time of change (format: yyymmddhhmmss)
- consult: an electronic report of external consultation sent or received
  - “1”: sent
  - “2”: received
- result: result received from the PALGA PZVDB
  - “1”: query (pzv) sent
  - “2”: result received
- dica: status of reporting to DICA (dicastatus)
  - “F”: report sent
  - “G”: positive acknowledge received
  - “R”: refused, negative acknowledge received, indicating error
- iknl: status of reporting to IKNL (incidstatus)
  - “F”: report sent
  - “G”: positive acknowledge received
  - “R”: refused, negative acknowledge received, indicating error
- bvodk: status of reporting to BVO-DK (bvostatus)
  - “F”: report sent
  - “G”: positive acknowledge received
  - “R”: refused, negative acknowledge received, indicating error
  - “W”: BVO-DK signal: patient refused admission in BVO-DK
  - “U”: BVO-DK signal: patient (BSN) not known in BVO-DK
- bvobmhk: status of reporting to BVO-BMHK (bvostatus)

- “F”: report sent
- “G”: positive acknowledge received
- “R”: refused, negative acknowledge received, indicating error
- “W”: BVO-BMHK signal: patient refused admission in BVO-BMHK
- “U”: BVO-BMHK signal: patient (BSN) not known in BVO-BMHK
- lab2lab: a message has been sent or received via Lab2Lab
  - “1”: sent
  - “2”: received
- messagetype: the Lab2Lab MessageType
- messageid: the Lab2Lab MessageID

In case of a “lab2lab” trigger the reportid may not be known and the attribute “rapport” will be empty. The messageid can be used to retrieve the complete message (see Lab2Lab documentation).

Trigger information is sent only once to one client, it is deleted from the server as soon as the client has fetched it. The trigger mechanism can and must be set up separately for each client. Which events on reports trigger a trigger record is a matter of parameters on the server.

### 3.8 Function “vriesmateriaal”

The function ‘vriesmateriaal’ stores ‘ja’ or ‘nee’ in the ‘rubriek’ ‘vriesmateriaal’. This rubriek provides information on the existence of frozen material in relation to a report. A self explaining example:

```
<functie id="f1" naam="vriesmateriaal" rapport="T13-12345"
aanwezig="ja"/>
```

### 3.9 Function “digitalebeelden”

The function ‘digitalebeelden’ stores ‘ja’ or ‘nee’ in the ‘rubriek’ ‘digitalebeelden’. This ‘rubriek’ provides information on the existence of full slide images in relation to a report. A self explaining example:

```
<functie id="f1" naam="digitalebeelden" rapport="T13-12345"
aanwezig="ja"/>
```

### 3.10 Function “vertaalconclusie”

The function ‘vertaalconclusie’ returns (proposed) diagnosis lines and codes when feeded with ‘conclusie’ text. The following example will show this:

```
<berichten>
<bericht id="bericht-id-1" van="X" aan="DPS">
<functie naam="vertaalconclusie" id="opdracht-id-1">
<conclusie>Placenta, navelstreng en vliezen, kindgewicht 1990 gram:
```

```

Te klein (placentagewicht 344 gram &lt; P10).Coilingindex te hoog
( &gt; 0,3). Percentage uitvalhaarden &lt;5 (door infarcten).
</conclusie>
</functie>
</bericht>
</berichten>

```

The result will be:

```

<berichten><bericht id="bericht-id-1" aan="X" van="DPS">
<antwoord id="opdracht-id-3" type="data">
<conclusievertaling>
  <conclusie>Placenta, navelstreng en vliezen, kindgewicht 1990
gram:Te klein (placentagewicht 344 gram &lt; P10).Coilingindex te
hoog ( &gt; 0,3). Percentage uitvalhaarden &lt;5 (door infarcten).
</conclusie>
  <diagnose>
    <termen>placenta*navelstreng*spontaan verloren*vliezen</termen>
    <codes>T88000*T88800*P49990*T88210T88290</codes>
  </diagnose>
  <diagnose>
    <termen>kin*placenta*spontaan verloren*infarcten</termen>
    <codes>TY0210*T88000*P49990*M54700</codes>
  </diagnose>
</conclusievertaling>
</antwoord></bericht></berichten>

```

As you can see, the term ‘kin’ (chin) is not correct here. With simple texts this function will do a good job, with more complex ‘conclusie’ texts the resulting diagnosis may be empty or wrong.

The element <conclusie> may contain elements <par> as in other functions. In that case all text must be contained in <par>...</par> paragraphs.

### 3.11 Function “ppm\_definitietabel”

This function returns an XML version of the protocol definitions in “Definitietabel.csv”.

```

<?xml version="1.0" ?>
<berichten>
  <bericht id="bericht-id-4" van="X" aan="DPS">
    <functie naam="ppm_definitietabel" id="opdracht-id-4"/>
  </bericht>
</berichten>

```

This XML function will return something like:

```

<?xml version='1.0' encoding='UTF-8'?>
<berichten><bericht id="bericht-id-4" aan="X" van="DPS">
  <antwoord id="opdracht-id-4" type="data">
    <protocollen>
      <protocol>
        <id>appendix</id>
        <naam>Appendix</naam>
        <versie>3</versie>

```

```

    <pitem>
      <id>rapport</id>
      <naam>rapport</naam>
      <lengte>11</lengte>
      <type>Tekensreeks</type>
    </pitem>
    <pitem>
      <id>jaar</id>
      <naam>jaar</naam>
      <lengte>4</lengte>
      <type>HeelGetal</type>
    </pitem>
    <pitem>
      <id>pversie</id>
      <naam>pversie</naam>
      <lengte>6</lengte>
      <type>HeelGetal</type>
    </pitem>
    <pitem>
      <id>afstanddichtsbijzijnde</id>
      <naam>AfstandDichtsbijzijnde</naam>
      <lengte>150</lengte>
      <type>Tekensreeks</type>
    </pitem>
    <pitem>
      <id>afstanddichtsbijzijndeexact</id>
      <naam>AfstandDichtsbijzijndeExact</naam>
      <lengte>14</lengte>
      <type>Getal</type>
    </pitem>
    <!--more elements pitem -->
  </protocol>
  <!-- more elements protocol -->
</protocollen>
</antwoord>
</bericht></berichten>

```

### 3.12 The element "drcvraag"

The order element "drcvraag" can be used to check a set of diagnosis lines. It uses the PALGA module DRC to do the checking, and additionally gives spelling suggestions for any term that was not recognized by the DRC.

The "drcvraag" element may contain one or more "drc" elements, that contain one or more "diagnose" elements, each with one "dtermen" element containing a diagnosis line

The following example illustrates how to use this element. More information is available in the DRC documentation.

```

<drcvraag id="drcvraag-id-1">
  <drc id="d1" rapport="T01-00001" mode="1">
    <diagnose id="diag1">
      <dtermen>mamma*biopsie*g.a.</dtermen>
    </diagnose>
    <diagnose id="diag2">

```



```

    <dtermen>lymfklier*g.a.</dtermen>
  </diagnose>
</drc>
<drc id="d2" rapport="T01-00002" mode="1">
  <diagnose id="diag1">
    <dtermen>cervix*stens*geen afwijking</dtermen>
  </diagnose>
  <diagnose id="qual1">
    <dtermen>tumor*voorgeschiedenis</dtermen>
  </diagnose>
</drc>
</drcvraag>

```

### The answer will be (like):

```

<antwoord id="drcvraag-id-1">
  <drc rapport="T01-00001" id="d1" versie="1.1.2.1">
    <diagnose id="diag1">
      <dtermen>mamma*biopsie*g.a<dterm term="mamma" code="T04000" id="1"/>
      <dterm term="biopsie" code="P11400" id="2"/>
      <dterm term="g.a" code="M00100" id="3"/>
    </dtermen>
    <dcodes>T04000*P11400*M00100</dcodes>
  </diagnose>
  <diagnose id="diag2">
    <dtermen>lymfklier*g.a<dterm term="lymfklier" code="T08000" id="1"/>
    <dterm term="g.a" code="M00100" id="2"/>
  </dtermen>
  <dcodes>T08000*M00100*WRONGP</dcodes>
  <dfouten>
    <dfout id="15">Techniekterm ontbreekt (diag2).</dfout>
  </dfouten>
</diagnose>
</drc>
  <drc rapport="T01-00002" id="d2" versie="1.1.2.1">
    <dfouten>
      <dfout id="31">Standaard cervix/endom. codering ontbreekt.</dfout>
    </dfouten>
    <diagnose id="diag1">
      <dtermen>cervix*stens*geen afwijking<dterm term="cervix" code="T83000"
        id="1"/>
      <dterm fout="Onbekende term: stens (diag1)." term="stens" code="WRO
        NG!" id="2">
        <suggestie>stans</suggestie>
        <suggestie>stenose</suggestie>
        <suggestie>stent</suggestie>
        <suggestie>steen</suggestie>
        <suggestie>stein</suggestie>
        <suggestie>stevens</suggestie>
        <suggestie>stand</suggestie>
        <suggestie>stenen</suggestie>
        <suggestie>stress</suggestie>
      </dterm>
      <dterm term="geen afwijking" code="M00100" id="3"/>
    </dtermen>
    <dcodes>T83000*WRONG!*M00100*WRONGP</dcodes>
    <dfouten>
      <dfout id="10">Onbekende term: stens (diag1).</dfout>
      <dfout id="15">Techniekterm ontbreekt (diag1).</dfout>
    </dfouten>
  </drc>
</antwoord>

```

```

</diagnose>
<diagnose id="qual1">
  <dtermen>tumor*voorgeschiedenis<dterm term="tumor" code="M80011"
        id="1" alt="zwellung of neoplasma"/>
    <dterm term="voorgeschiedenis" code="Q00101" id="2"/>
  </dtermen>
  <dcodes>M80011*Q00101*WRQUAL</dcodes>
  <dfouten>
    <dfout id="11">Ongewenste term: tumor; advies: zwellung of
        neoplasma (qual1).</dfout>
    <dfout id="21">Geen qualifier term als eerste term (qual1).</dfout>
  </dfouten>
</diagnose>
</drc>
</antwoord>

```

### 3.13 DRC controles

In the PALGA module DRC the following verifications are executed.

1. The maximum amount of diagnosis lines within a set is 99.
2. All terms will be looked up in the PALGA thesaurus. If a term is still not found an error message will be given, and the code becomes 'WRONG!'. The first term sometimes will be joint together with the second one. This happens if the first term is 'huid', 'lymfklier', 'slijmvlies', 'slymvlies' of 'subcutis', and the combination of the first and the second term is found in the thesaurus.
  - An empty term will lead to code 'WRONGL'.
  - An unwanted term will lead to code 'WRONGW'.
  - A qualifier (code Q001..) in a diagnosis line leads to an extra code 'WRONGQ'.
  - If no topography (code T.....), then extra code 'WRONGT'.
  - If first topography not in first term: extra code 'WRONGV'.
  - In case of metastasis (code M[89]...6), but subsequently no topography: extra code 'WRONGM'.
  - In case of a penetration by tumor (code M80093), but subsequently no topography: extra code 'WRONGI'.
  - If no removal procedure : 'WRONGP'. A removal procedure starts with a code 'P10', 'P11', 'P065', 'P30', 'T88', 'T890', 'P49990' or 'T0X500'.
  - If no finding with histology: 'WRONGD'. Finding codes start with 'D', 'E', 'F' of 'M'.
  - In case of qualifiers:

- i. No joining of 2 terms
- ii. WRONG!, WRONGL, WRONGW.
- iii. If no qualifier (code Q001..) as first term: extra code 'WRQUAL'.
- iv. If no other term after qualifier: 'WREMTY'.

## 4 Permissions

Any action a client can (or cannot) perform using this XML link with the UDPS server is subject to permissions.

Usually a client must authenticate itself with a user id and a password. Then this user id is used as client id. If no user authentication is used, the clients system id is used as client id. This could be the case where a secure connection is used (HTTPS), where authentication is performed on system level.

If there is no client id, or the client id is not in the (first index of the) table "client", a "fout" element is sent back, requesting the client to "go away".

The permission value often is a pattern (regular expression), sometimes a (number) string. A pattern to be matched with a report name could for example be "[TS]": only histology (T) and section (S) reports. A pattern with which some "rubriek" names match: "m[ia]croscopie|conclusie". This way fine-tuning can be performed on what is permitted and what not for some particular user.

The following **global parameters** for a client are defined:

- https, if the value of this parameter is "nee" then a secure HTTPS connection is not required. The default will be "ja".

System-X can not set or change permissions. When the Core-UDPS is installed permission must be set for the user used (with profile 'standaard'):

```
- functie_ciparesult ja
- functie_cipavraag ja
- functie_followup ja
- functie_incidentie ja
- functie_spoolfiles ja
- functie_triggers ja
- vraag_rapport .*
- vraag_ongeacht .*
- wijziging_rapport .*
- wijziging_autraapport .*
- creatie_rapport .*
```

## 5 DTD

The document type definitions, on which these XML messages are based, can be found on any UDPS system with the URL "http://hostname/xml/dtd/dtdfile". In this URL "hostname" is the local name or IP address of this UDPS, and "dtdfile" is the filename of the file in which some DTD resides. The corresponding real filename and path of such a file is "/usr/prod/dpsweb/xml/dtd/dtdfile". If the content of such a DTD file is different from this documentation, the file is leading.

The following DTD files exist:

### berichten.dtd

```
<!ELEMENT berichten (bericht)*>
<!ATTLIST berichten id CDATA "" aantal CDATA #IMPLIED>
<!ENTITY % bericht SYSTEM "bericht.dtd">
%bericht;
```

### bericht.dtd

```
<!ENTITY % rapporten SYSTEM "rapporten.dtd">
%rapporten;
<!ENTITY % tabellen SYSTEM "tabellen.dtd">
%tabellen;
<!ENTITY % palgadrc SYSTEM "palgadrc.dtd">
%palgadrc;
<!ENTITY % permissies SYSTEM "permissies.dtd">
%permissies;
<!ENTITY % proc SYSTEM "proc.dtd">
%proc;

<!ELEMENT
bericht(creatie|verwijder|wijziging|vraag|functie|drcvraag|
antwoord|fout|permissies|groepen|gebruikers|passwd)*>
<!ATTLIST bericht
    id ID #REQUIRED
    van CDATA ""
    aan CDATA "" >
<!ELEMENT antwoord
(fout|waarschuwing|tekst|rapporten|tabellen|result|
groepen|gebruikers)*>
<!ATTLIST antwoord
    id CDATA #REQUIRED
    type ( ack | nack | data ) "data" >
<!ELEMENT waarschuwing (#PCDATA)>
<!ATTLIST waarschuwing
    id CDATA #REQUIRED
    rapport CDATA "" >
<!ELEMENT tekst (span | par)*>
<!ATTLIST tekst
    id CDATA ""
    type CDATA "">
<!ELEMENT vraag ( rubriek | selectie )*>
<!ATTLIST vraag
    id CDATA #REQUIRED
```

```

        type ( nummers | rapporten ) "rapporten"
        geaut ( ja | nee | beide ) "ja"
        max CDATA #IMPLIED
        maxz CDATA #IMPLIED
        rapport CDATA "" >
<!ELEMENT selectie EMPTY>
<!ATTLIST selectie
        laagste CDATA #IMPLIED
        hoogste CDATA #IMPLIED
        eerste CDATA #IMPLIED
        laatste CDATA #IMPLIED
        status CDATA #IMPLIED
        type (rubriek | attribuut | key) "rubriek"
        naam CDATA #IMPLIED
        operator (gelijk|ongelijk|groter|kleiner|groter-of-
gelijk|kleiner-of-gelijk| \
                |matcht|matcht-niet) "gelijk"
        waarde CDATA "">
<!ELEMENT creatie (rubriek)*>
<!ATTLIST creatie
        id CDATA #REQUIRED
        rapport CDATA #REQUIRED
        status CDATA #IMPLIED
        datumontvangst CDATA #IMPLIED >
<!ELEMENT functie ( argument | rapporten | tabellen )*>
<!ATTLIST functie
        id CDATA #REQUIRED
        naam CDATA #REQUIRED
        rapport CDATA ""
        status CDATA #IMPLIED >
<!ELEMENT argument (#PCDATA)>
<!ATTLIST argument
        id CDATA #REQUIRED>
<!ELEMENT wijziging ( rubriek | statusbyte )*>
<!ATTLIST wijziging
        id CDATA #REQUIRED
        rapport CDATA #REQUIRED
        status CDATA #IMPLIED
        mode CDATA ( update | update-aut ) "update" >
<!ELEMENT result (rapport | fout)*>
<!ATTLIST result
        id CDATA ""
        aantal CDATA #IMPLIED
        type ( follow | incid | result | xrp ) "result">

```

### **rapporten.dtd**

```

<!ELEMENT rapporten (rapport)*>
<!ATTLIST rapporten
        id CDATA ""
        aantal CDATA #IMPLIED>
<!ENTITY % rapport SYSTEM "rapport.dtd">
%rapport;

```

### **rapport.dtd**

```

<!ENTITY % item SYSTEM "item.dtd">
%item;
<!ELEMENT rapport ( rubriek | fout )*>
<!ATTLIST rapport
    id CDATA #REQUIRED
    keuze CDATA ""
    lab CDATA #IMPLIED
    so CDATA #IMPLIED
    jaar CDATA #IMPLIED
    nummer CDATA #IMPLIED
    status CDATA #IMPLIED
    versie CDATA #IMPLIED
    autorisator CDATA ""
    autts CDATA ""
    autwegts CDATA ""
    autweghaler CDATA ""
    autwegvrager CDATA ""
    autwegreden CDATA ""
    autmutaard CDATA ""
    autmutpointer CDATA ""
    mode ( compleet | partieel | excerpt | nummer |
na ) "compleet">
<!ELEMENT span (reg)*>
<!ATTLIST span class CDATA "" >
<!ELEMENT par (#PCDATA)>
<!ATTLIST par class CDATA "" >
<!ELEMENT reg (#PCDATA)>
<!ELEMENT fout (#PCDATA)>
<!ATTLIST fout id CDATA "" >
<!ELEMENT rubriek (#PCDATA | span | par | cris | container)*>
<!ATTLIST rubriek
    id CDATA ""
    naam CDATA ""
    titel CDATA ""
    soort ( code | datum | datumtijd | dpsdatum | tekst |
lang ) "tekst"
    mode ( default | aanvullen | overschrijven |
niet_overschrijven ) "default"
    waarde CDATA ""
    uitvoerder CDATA ""
    datum CDATA ""
    assistent CDATA ""
    supervisor CDATA ""
    secr CDATA ""
    datumsecr CDATA "" >
<!ELEMENT cris (crisitem)*>
<!ATTLIST cris
    type ( klin | uitslag ) "uitslag"
    versie CDATA "3">
<!ELEMENT crisitem (#PCDATA)*>
<!ATTLIST crisitem
    id CDATA ""
    naam CDATA ""
    soort ( code | datum | tekst ) "tekst"

```

```
        waarde CDATA ""
        extra CDATA "">
<!ELEMENT container (key)*>
<!ELEMENT key (item)*>
<!ATTLIST key
        id CDATA #REQUIRED >
<!ELEMENT statusbyte EMPTY>
<!ATTLIST statusbyte
        id CDATA ""
        naam CDATA #REQUIRED
        waarde CDATA #REQUIRED
        rubriek CDATA #IMPLIED >
```

### **result.dtd**

```
<!ELEMENT result (rapport | fout)*>
<!ATTLIST result
        id CDATA ""
        aantal CDATA #IMPLIED
        type ( follow | incid | result | xrp ) "result">
<!ENTITY % rapporten SYSTEM "rapporten.dtd">
%rapporten;
```

Other DTD files in the same directory don't have impact on this XML link.

In UDPS, the Ruby xserver does not check the messages against the DTD. It is intended to serve in a more or less private network with trusted clients that really should know how to produce a valid "berichten" element. Downloading the DTD and checking consumes precious time. And besides the Ruby XML parser in use cannot do checking at this point of time (may 2003).



**tabellen.dtd**

```

<!ELEMENT tabellen (tabel)*>
<!ATTLIST tabellen
    id CDATA ""
    aantal CDATA #IMPLIED>
<!ENTITY % tabel SYSTEM "tabel.dtd">
%tabel;

```

**tabel.dtd**

```

<!ENTITY % item SYSTEM "item.dtd">
%item;
<!ELEMENT tabel (tabel | item)* >
<!--
    xxx[a][b][c]=d wordt
    <tabel id="xxx">
        <tabel id="a">
            <tabel id="b">
                <item id="c" waarde="d"/>
            </tabel>
        </tabel>
    </tabel>
-->
<!ATTLIST tabel
    id CDATA #REQUIRED
    editable (all, ja, nee, hide) "ja"
    dim CDATA #IMPLIED
    type ( lokaal | landelijk | landelijk- |
landelijk+ ) "lokaal">

```

**item.dtd**

```

<!ELEMENT item (#PCDATA)*>
<!ATTLIST item
    id CDATA #REQUIRED
    titel CDATA ""
    editable ( ja | nee ) "ja"
    hide ( ja | nee ) "nee"
    soort ( code | datum | tekst ) "tekst"
    waarde CDATA ""
    extra CDATA "">

```

**palgadrc.dtd**

```

<!ELEMENT drcm ( drc )*>
<!ATTLIST drcm
    id CDATA #IMPLIED
    mode ( 0 | 1 ) "0">
<!ELEMENT drc ( dfouten?, diagnose* )>
<!ATTLIST drc
    id CDATA #IMPLIED
    rapport CDATA ""
    mode ( 0 | 1 ) "0">
<!ELEMENT diagnose ( dtermen, dcodes?, dfouten? )>
<!ATTLIST diagnose
    id CDATA #REQUIRED>
<!ELEMENT dtermen ( #PCDATA | dterm )*>
<!ELEMENT dterm ( #PCDATA | suggestie )*>
<!ATTLIST dterm
    id CDATA #REQUIRED
    nr CDATA #REQUIRED
    code CDATA #IMPLIED
    fout CDATA #IMPLIED
    alt CDATA #IMPLIED>
<!ELEMENT alt ( #PCDATA )>
<!ELEMENT suggestie ( #PCDATA )>
<!ELEMENT dcodes ( #PCDATA )>
<!ELEMENT dfouten ( dfout )*>
<!ELEMENT dfout ( #PCDATA )>
<!ATTLIST dfout
    id CDATA #REQUIRED>

```

**6 UDPS dataset**

For the UDPS dataset and national choicelists or tables see de Excel document “Core UDPS dataset dd.mm.jj.xls”