Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Finance FDF

**Federal Office of Information Technology, Systems, and Telecommunication FOITT**

PRO-IEW-EWM

2018

# EPR – Central Services

## Interface Documentation

| | |
|---|---|
| Customer | Federal Office of Public Health (FOPH) |
| Version | 1.0.22 |
| Version Date | 03.04.2018 |
| Authors | Louis Bernath, Reto Ghioldi |
| State | ☐ in progress ☐ in review ☒ reviewed ☐ approved |
| Classification | ☒ none ☐ internal ☐ classified |
| Document location | https://www.e-health-suisse.ch/gemeinschaften-umsetzung/umsetzung/programmierhilfen.html |

## Document history

| Version | Date | Changes | Authors |
|---|---|---|---|
| 1.0.21 | 27.02.2018 | Release version | FOITT, Louis Bernath, Reto Ghioldi |
| 1.0.22 | 03.04.2018 | Updated value set identifier for EprAuthorRole, | FOITT, Louis Bernath |

# Table of contents

# 1. Introduction

This document contains information about the EPR central services implementation.

## 1.1 Purpose of this document

This technical interface documentation of the EPR central services describes implementation specific details. Not implemented, optional interfaces are highlighted. Ambiguous definitions in the underlying regulations and standards are defined more precisely.

## 1.2 References

| Reference | Description |
|---|---|
| [IHE-HPD] | IHE IT Infrastructure - Supplement HPD Profile (http://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_Suppl_HPD.pdf) |
| [IHE-TF-2b] | IHE IT Infrastructure Technical Framework, Volume 2b (https://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_TF_Vol2b.pdf) |
| [BAG-A5] | National extensions to the IHE Technical Framework (https://www.bag.admin.ch/dam/bag/de/dokumente/nat-gesundheitsstrategien/strategie-ehealth/gesetzgebung-elektronisches-patientendossier/gesetze/SR%20816.111.1_ergaenzung-1-Anhang-5.pdf.download.pdf/SR%20816.111.1_Ergaenzung%201%20Anhang%205_DE.pdf) |
| [BAG-A5-S] | Schemas of the national extensions to the IHE Technical Framework (https://www.bag.admin.ch/dam/bag/de/dokumente/nat-gesundheitsstrategien/strategie-ehealth/gesetzgebung-elektronisches-patientendossier/gesetze/schemata-ergaenzung-1-anhang-5.zip.download.zip/Schemata%20Ergaenzung%201%20Anhang%205%20EPDV-EDI.zip) |
| [RFC4511] | Lightweight Directory Access Protocol (LDAP): The Protocol (https://www.ietf.org/rfc/rfc4511.txt) |
| [RFC4517] | Lightweight Directory Access Protocol (LDAP): Syntaxes and Matching Rules (https://tools.ietf.org/rfc/rfc4517.txt) |
| [RFC2798] | Definition of the inetOrgPerson LDAP Object Class (https://tools.ietf.org/rfc/rfc2798.txt) |
| [RFC4519] | Lightweight Directory Access Protocol (LDAP): Schema for User Applications (https://tools.ietf.org/rfc/rfc4519.txt) |
| [RFC2696] | LDAP Control Extension for Simple Paged Results Manipulation (https://www.ietf.org/rfc/rfc2696.txt) |
| [RFC2891] | LDAP Control Extension for Server Side Sorting |

| | |
|---|---|
| | (https://www.ietf.org/rfc/rfc2891.txt) |
| [RFC2985] | PKCS #9: Selected Object Classes and Attribute Types Version 2.0<br>(https://tools.ietf.org/rfc/rfc2985.txt) |
| [RFC7234] | Hypertext Transfer Protocol (HTTP/1.1): Caching<br>(https://tools.ietf.org/rfc/rfc7234.txt) |
| [RFC 4648] | The Base16, Base32, and Base64 Data Encodings<br>(https://tools.ietf.org/rfc/rfc4648.txt) |
| [W3C-SOAP12] | SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)<br>(https://www.w3.org/TR/soap12) |
| [ISO 21091] | Health informatics -- Directory services for healthcare providers, subjects of care and other entities, 2013<br>(https://www.iso.org/standard/51432.html) |
| [LDAP RCODE] | LDAP result code reference<br>(https://www.ldap.com/ldap-result-code-reference) |
| [SwissGov-PKI] | Swiss Government PKI<br>http://www.pki.admin.ch/ |
| [Digicert-RootC] | DigiCert Trusted Root Authority Certificates<br>https://www.digicert.com/digicert-root-certificates.htm |
| [EPR-WSDL] | EPR WSDL and schema files<br>https://www.e-health-suisse.ch/fileadmin/user_upload/Dokumente/2017/E/WSDL_Files.zip |
| [EPR-HPD-Schema] | EPR attribute and object class definitions for the HPD<br>https://www.e-health-suisse.ch/fileadmin/user_upload/Dokumente/2017/E/EPD_ZAD_HPD_Attribute.xlsx<br>or<br>https://www.e-health-suisse.ch/gemeinschaften-umsetzung/umsetzung/programmierhilfen.html<br>→ EPD_ZAD_HPD_Attribute.xlsx |
| [EPR-CPI-Schema] | EPR attribute and object class definitions for the CPI<br>https://www.e-health-suisse.ch/fileadmin/user_upload/Dokumente/2017/E/EPD_ZAD_CPI_Attribute.xlsx<br>or<br>https://www.e-health-suisse.ch/gemeinschaften-umsetzung/umsetzung/programmierhilfen.html<br>→ EPD_ZAD_CPI_Attribute.xlsx |
| [DSML] | Directory Services Markup Language (DSML), Version 2<br>http://www.oasis-open.org/committees/dsml/docs/DSMLv2.xsd |

## 1.3 Glossary

| Term | Definition |
|---|---|
| AD LDS | Microsoft Active Directory - Lightweight Directory Services<br>https://technet.microsoft.com/en-us/windowsserver/dd448612.aspx |
| CPI | Community Portal Index |
| DIT | Directory Information Tree |
| DN | Distinguished Name (LDAP) |
| DSML | Directory Service Markup Language |
| EPR | Electronic Patient Record |
| FOITT | Federal Office of Information Technology, Systems and Telecommunication |
| FOPH | Federal Office of Public Health |
| HPD | Healthcare Provider Directory |
| LDAP | Lightweight Directory Access Protocol |
| MDI | Metadata Index |
| PIDD | Provider Information Delta Download |
| RDN | Relative Distinguished Name (LDAP) |
| RFC | Requests for Comments |
| SVS | Sharing Value Sets |
| TLS | Transport Layer Security |
| WSG | Web Service Gateway<br>A web service specific proxy system. |
| XSD | XML Schema Definition |

# 2. System access

## 2.1 Authentication & Authorization

The system must be accessed by using TLS two way authentication. The client certificate is used to identify the requesting community and is only considered valid if it was issued by “Swiss Government Root CA II”. The certificate has to be still within the validity period. You will receive a connection reset of the TCP connection if you do not provide a valid certificate or do not provide a certificate at all.

You will receive an HTTP 400 with the following soap fault if the certificate is valid, but not yet configured for access on the edge servers:

```
<env:Code>
  <env:Value>env:Sender</env:Value>
</env:Code>
<env:Reason>
  <env:Text xml:lang="en-US">Rejected by policy (from client)</env:Text>
</env:Reason>
```

A SOAP fault with the sub code “InvalidSecurity” and an HTTP result code of 401 is returned if you provide a valid certificate that is configured on the edge servers, but is not (yet) known to the system:

```
<s:Code>
  <s:Value>s:Sender</s:Value>
  <s:Subcode>
    <s:Value xmlns:a="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">a:InvalidSecurity</s:Value>
  </s:Subcode>
</s:Code>
```

You will receive a SOAP fault with the sub code “FailedAuthentication” and an HTTP result code of 403 if you are authenticated successfully but are not authorized to perform the SOAP action (e.g. your community’s status is not active, see [EPR-CPI-Schema]):

```
<s:Code>
  <s:Value>s:Sender</s:Value>
  <s:Subcode>
    <s:Value xmlns:a="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">a:FailedAuthentication</s:Value>
  </s:Subcode>
</s:Code>
```

Note that if the interface or protocol (like DSML) is designed to provide its own error handling (like DSML result codes) the error is indicated leveraging the protocol (see section 3.2 Error Handling).

## 2.2 Certificate Trust Chain

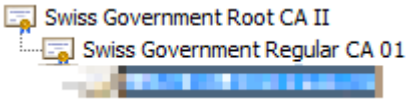
To allow mutual trust and establish a successful TLS connection you not only need the client certificate for authentication, but you also need to ensure that you trust the EPR server

certificate.

Depending on your validation method and if you do not use a commonly trusted list of root certificates (from Microsoft, Mozilla or Apple), you will need to download and install/import the necessary certificates into your trust store. All the certificate information can be downloaded from [SwissGov-PKI] (see Rootzertifikate->Swiss Government Root CA II) or [DigiCert-RootC].

### 2.2.1 Client certificate

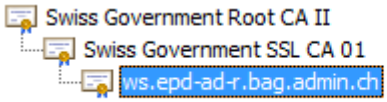
Please make sure the root and intermediate certificates for “Swiss Government Root CA II” and “Swiss Government Regular CA 01” are available in your trust store.




The client certificate ordering process is organized through the FOPH and is not part of this document.

### 2.2.2 Server certificate

Please make sure the root “Swiss Government Root CA II” and intermediate certificates for “Swiss Government SSL CA 01” are available in your trust store. Or take other measures to ensure the trust relationship to the server certificate.




## 2.3 Correlation Id’s

The system returns a correlation id for each request. Because there is both SOAP and HTTP support for certain operations a HTTP header was chosen. All operations return the HTTP header “epr-correlation-id”. The value of this header can be used to uniquely identify a request by the FOPH personnel and allow both tracing and reproduction of the request. This is especially helpful with the HPD feed, as this operation contains a batch of sub-operations.

We recommend that this correlation is produced when asking for support as it allows for easier reproduction of behaviour.

E.g. epr-correlation-id: d1795acc-9bec-490f-83c5-73bb5702a4ee

## 2.4 Endpoints and WSDL’s

All the necessary information for accessing the EPR services, like endpoint URLs, WSDL and Schemas are provided in [EPR-WSDL].

# 3. DSML

## 3.1 DSML implementation

Both the Healthcare Provider Directory (HPD) and the Community Portal Index (CPI) are implemented using the standard DSML interface. Exceptions are explicitly listed in the following sections.

### 3.1.1 Supported elements

#### 3.1.1.1 Non-supported DSML elements and types

The following elements are ignored:

- *DsmlMessage* (i.e. request) types *CompareRequest*, *AbandonRequest* and *ExtendedRequest, AuthRequest*
- *control* elements on all *DsmlMessage* types except the ones defined in 4.4.2.
- *xsd:anyURI* data type for the value of a *DsmlValue*

#### 3.1.1.2 Authentication

The *"authRequest"* element of a batch request is not evaluated. The community's identity is always defined by the (TLS) client certificate of the community.

#### 3.1.1.3 Binary values

Binary values, i.e. attributes of LDAP type Octet String, are expected to be correctly encoded as "*xsd:base64Binary"* data type (see [RFC 4648]).

#### 3.1.1.4 Encoding (Escaping)

Values and DNs are encoded/escaped on the server (e.g. "\0d" for a CR). Note that client side escaping using the backslash character ("\") will be double escaped.

### 3.1.2 Search Filter

Not all DSML search filter elements are supported by the HPD services. This is mostly a limitation of the underlying LDAP implementation.

- *ApproxMatch* is implemented as *EqualityMatch* (see [RFC4511], section 4.5.1.7.6).
- *ExtensibleMatch* is not implemented (see: [RFC4517], section 4). A *searchRequest* with an extensibleMatch in the filter leads into result code 53 (= Unwilling to Perform).
- An invalid filter results in result code 87 (= Filter Error)
- A missing filter results in a SOAP fault (i.e. a XML schema violation according to [DSML])
- The number of returned search result entries is limited (see limitations in section 3.2.7 Size limit of search requests).

To perform a full search without any entry filtering, the availability check for an always present attribute is recommended (i.e. a presence check of attribute *objectClass*).

### 3.1.3 Distinguished names (DNs)

Distinguished names (DNs) are very important in the LDAP world because they act as a unique identification of a single directory entry. HPD defines some constraints on the DN:

- Distinguished names are case insensitive.
- Control characters are not allowed (tabs, line breaks, etc.).
- Commas (",") are only allowed to separate relative distinguished names (RDNs).
- Equality signs ("=") are only allowed to separate attribute name and value.
- Leading and trailing whitespaces will be encoded.
- The following characters are allowed but get encoded/escaped to "\<char>":
  - #
  - " (quote)
  - ;
  - \
  - +
  - <
  - >

### 3.1.4 Type System

The following attribute types are used (including their optional shortcuts)

- ***Directory String (DString)*****:** Refers to the *Directory String* in UTF-8 encoding
- ***Octet String (OString)*****:** A binary value (base64 encoded in HPD transactions)
- ***Printable String (PString)*****:** A limited Latin character set (see [RFC4517], section 3.3.29)
- ***DN*****:** Distinguished name of another existing object in the HPD. A Directory String value.
- ***rDN:*** Relative DN of this object in the HPD. A Directory String value.
- ***OID:*** Object identifier of another directory element. A Directory String value.
- ***GeneralizedTime (GTime):*** UTC (GMT) timestamp in the format *“YYYYMMDDHHmmss.0Z”*

All types are standardized LDAP data types.

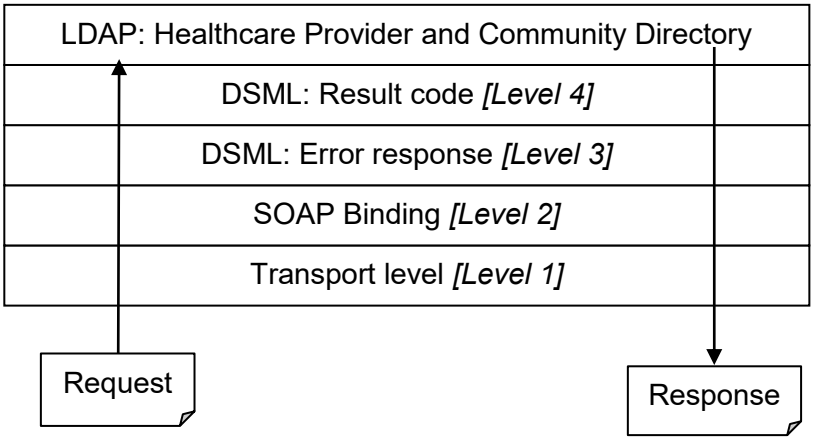
### 3.1.5 General attribute constraints

Mandatory attributes (= “must”) have to be provided. Otherwise an object class violation occurs (result code = 65). Empty attribute values or whitespaces are treated as if no value has been specified. Optional attributes (= “may”) are not required to execute the request.

# 3.2 Error Handling

## 3.2.1 General

Error handling in HPD takes place on several layers or levels of the processing stack.




The HPD service adapts the standardized errors of the different layers: HTTP Status codes, SOAP Faults, LDAP error types and result codes.

## 3.2.2 Level 1: Transport level

The system has not been reached at all and the error is not under the system's control.

**Examples:**

- Bad Endpoint URL at the client.
- Basic certificate based authentication failed (HTTP status code 401) because of an invalid chain of trust of the certificate.
- Request size is too large.

## 3.2.3 Level 2: SOAP Binding

The system has been reached but the SOAP protocol has been broken or some basic validation failed.

**Examples:**

- Malformed XML (see [W3CSOAP12] Chapter 5)
- WSDL/DSML schema violation
- Missing or unknown (authentication)
- Access not allowed to this client (authorization)
- Invalid SOAP action parameters (like unsupported request types in DSML batch)

**SOAP fault definition**

The following SOAP faults may be returned during validation:

| Scenario | Code value | Sub code value |
|---|---|---|
| XML schema violation | Sender | XML_SCHEMA_VIOLATION<br><br>*Namespace:*<br>urn:ch:admin:bag:epr:2017 |

### 3.2.4 Level 3: DSML request validation

The DSML batch has been processed corresponding to the specified batch processing settings but one or more <errorResponse> are returned. The requests with an errorReponse have not been executed against the directory.

**Examples:**

- Formal validation failed. The request could not be parsed and loaded into the system due to technical problems (like an invalid DN syntax or a non-existing DC).
- An addRquest with missing “objectClass” attribute or values.
- Object classes of the new entry do not match the allowed object classes of the selected OU.
- A mutative request (like an addRequest) in a query transaction (ITI-58).
- Connection error to the directory or database.
- Unexpected processing behaviour.

### 3.2.5 Level 4: LDAP execution errors

The DSML batch has been processed corresponding to the specified batch processing settings and an *xxxResponse* is returned (*xxx* matching to the executed request type).

The actual LDAP request has been executed but an error has occurred (i.e. the underlying LDAP server returned a result code <> "success").

The first detected error is returned (if multiple exist in the request) because the result code exists only once in the response.

**Examples:**

- Business rule validation failed. The request is structurally not allowed (like creating cross community relationships).
- Missing required attributes that are not validated by the EPR central services. The request execution resulted in a LDAP Schema violation (no such attribute, attribute is single valued and cannot contain multiple values).
- Data errors detected during the request execution (No such object/DN, object already exists, multi-valued attribute already contains this value.)
- Request and response limitations

### 3.2.6 General

Too large batch requests may lead to transfer errors. Hence it is a good practice to limit the request and response size. The following limitations exist in the EPR central services.

- All request sizes are limited to 100MByte on transport level (i.e. HTTP body).
- The maximum number of returned search entries in a query transaction is 1'000 (maximum page size).
- The maximum number of allowed requests in HPD feed batches is 1’000.

Please take note, that we are evaluating additional limitations to protect server resources and ensure optimal operation of the EPR central services.

### 3.2.7 Size limit of search requests

The query transactions for HPD and CPI return results up to the number specified in the requests *sizeLimit*. Except the specified size limit exceeds 1'000, the servers limit is respected. If your filter expression exceeds the size limit you will receive a LDAP result code is 4 (= size limit exceeded) instead of 0 (= success).

A client has to use the paged search control to get search entries if the result set is too large.

# 4. Healthcare Provider Directory (HPD)

The HPD interface is implemented following the IHE HPD Profile [IHE-HPD]. The interface provides three operations, of which two implement the elements of the DSML protocol (ITI-58 and ITI-59) and the third (CH:PIDD) that is defined by the EPR decree.

## 4.1 Batch processing

HPD ITI-58 and ITI-59 use the following behaviour for the DSML batch parameters:

- The *processing* attribute is ignored. Batch requests are always processed in sequential order[1].
- The *responseOrder* attribute is ignored. Batch requests are always processed in sequential order and the response elements have the same order as the original requests.
- The *onError* attribute is respected. The default is "exit" (if the attribute is not specified). With "exit" the batch requests is aborted on the first faulty request. When you request onError="resume" all requests are executed, even if one does result in an error.

## 4.2 Directory schema

The HPD base distinguished name / directory root is "dc=HPD,o=BAG,c=CH". All child elements of this node are considered a part or the HPD.

### 4.2.1 Versioning

Although the HPD is the master in the EPR central services, every community is free to have a local replicate of the directory. The provider information feed (ITI-59) and delta download (CH:PIDD) transactions can be used to synchronize the local replica with the master using DSML requests.

Hence it is very important that the master and the replicas use the same schema. Otherwise,

[1] **Performance note:** Clients may improve overall performance of request processing by sending multiple batches in parallel. They must ensure correctness of processing order over multiple batches on their own.

requests may result in a failure because of a structural difference in the underlying directory storage.

The master directory defines the LDAP schema and all clients must follow these definitions. Incompatible replicas may not be able to synchronize with the master directory.

In the EPR central services there is no explicitly versioned schema or corresponding version query operation available on machine to machine level.

### 4.2.2 Standard precedence

The object class and attribute schema is based on following standards. The following descending precedence is taken if the standards contradict each other:

1. IHE CH extensions [BAG-A5]
2. IHE HPD [IHE-HPD]
3. ISO 21091:2013 [ISO-21091]
4. *inetOrgPerson* [RFC-2798]
5. *organizationalPerson* and *person* [RFC-4519]
6. *naturalPerson* [RFC-2985]

The number of attributes has been reduced to the ones mentioned in the following standards:

- IHE CH extensions [BAG-A5]
- IHE HPD [IHE-HPD]

### 4.2.3 Object Classes and Organisational Units

The central services support three object classes defined in the LDAP schema. Each object class can be stored in its distinct OU container (see also 4.5.3.1 objectClass attributes)

| Element | Object class | Organisational Unit |
|---|---|---|
| Health professional | HCProfessional | HCProfessional |
| Health organisation | HCRegulatedOrganization | HCRegulatedOrganization |
| Relationships between organisations and professionals or organisations | groupOfNames | Relationship |

An overview of the complete LDAP schema is shown in appendix B. A complete list of object classes, attributes and their detailed definition and description can be obtained from [EPR-HPD-Schema].

## 4.3 Validations

Communities are intended to manage only their own data in the directory (multitenancy), although in general all HPD entries have a public visibility. The EPR central services contains validations that ensure the correctness of the directory data. Violation of the stated validation rules lead to a response with the corresponding result code. We use only standard result codes, even for implementation specific errors. For a list of common result codes see appendix A or for a complete list of the standard LDAP result codes see [LDAP RCODE].

Please note, that if not otherwise stated, all validations on distinguished names (name and value) and attribute name or values are case insensitive.

This means that the following DNs are equal:

```
UID=COMMUNITYA:1,OU=HCProfessional,DC=HPD,O=BAG,C=CH
uid=communitya:1,ou=hcprofessional,dc=hpd,o=bag,c=ch
```

And also the following attribute value pairs are equal:

```
hcProviderStatus=Active
HCPROVIDERSTATUS=ACTIVE
hcproviderstatus=active
```

# 4.4 Provider information query (ITI-58)

## 4.4.1 Supported request types

Only *searchRequest* entries are supported in a batch for the query transaction. The whole batch will be rejected with a SOAP fault if another request type is detected.

## 4.4.2 Supported control types

The control values must be delivered BER encoded as base64Binary type. A sort control is shown here.

**Example:**

```
<batchRequest xmlns="urn:oasis:names:tc:DSML:2:0:core">
       <searchRequest dn="DC=HPD,O=BAG,C=CH"
                     .scope="wholeSubtree"
                      derefAliases="neverDerefAliases"
                      sizeLimit="100">
              <control type="1.2.840.113556.1.4.473" criticality="true">
                     <controlValue xsi:type="xsd:base64Binary">
                            MIQAAAAUMIQAAAAOBAxIY0lkZW50aWZpZXI=
                     </controlValue>
              </control>
              <filter>
                     <present name="objectClass"/>
              </filter>
      </searchRequest>
</batchRequest>
```

A bad base64Binary encoded *controlValue* returns with an “internal server error” (status code 500). The whole request is treaded as malformed and rejected completely. Hence, the control’s *criticality* is not yet taken into account. The batch request (i.e. other requests in the batch) is not executed.

### 4.4.2.1 Restrictions

The control extension for paged results (4.4.2.2) is not compatible with the server side sorting control extension (4.4.2.3). Paging through the results will fail; only the first paged can be fetched.
We recommend to *not* combine these two control extensions in a single request and perform any combination of these two features on the client side instead.

#### 4.4.2.2 Paging

To get around the limitation of 1000 entries in one search, the standard paging mechanism is available.

The paging mechanism is called by adding a pagedResultsControl to the SearchRequest, conforming to [RFC2696]. The pagedResultControl has type “1.2.840.113556.1.4.319”.

The page size and a cookie have to set to the controlValue as a BER encoded base64Binary. On the first request (first page) the cookie has to be null, at all subsequent calls the cookie has to be the one returned in the result of the previous response. If the returned cookie is null again, this means that the last page has been returned. The actual search request should not change during paging.

**Example request:**

```
<searchRequest dn="DC=HPD,O=BAG,C=CH" scope="wholeSubtree"
               derefAliases="neverDerefAliases">
<control type="1.2.840.113556.1.4.319" criticality="true">
  <controlValue xsi:type="xsd:base64Binary">MIQAAAAFAgEHBAA=</controlValue>
</control>
  […]
</searchRequest>
```

According to [RFC2696] the contents of the control value is BER encoded with the following format “The searchControlValue is an OCTET STRING wrapping the BER-encoded version of the following SEQUENCE”.

```
searchControlValue ::= SEQUENCE {
        size            INTEGER (0..maxInt),
                                -- requested page size from client
                                -- result set size estimate from server
        cookie          OCTET STRING
}
```

In our example the controlValue “MIQAAAAFAgEHBAA=” contains the following data:
size = 7
cookie = null (always null for the first page)

**An example response with a paging control:**

```
<searchResponse>
  […]
  <searchResultDone>
    <control type="1.2.840.113556.1.4.319">
      <controlValue
xsi:type="xsd:base64Binary">MIQAAAF3AgEABIIBcAEAAABwAQAA/////TmSarNPDikDwvqZiKhY2PXDv
A5FaN5PUGCKNcMEn9gnb1GcmisgUO0SyXHFWbQjwAAAAABAAAAAAAAAE0CAAAcAAAABQAAAAIAAAAAAAAAAAA
AAUAAAAEAAEAuQEAALgBAAC5AQAAAAAAAKhX15tsnUlAmwdwLbOjkKUAAAAABAAAAAEAAAAAAAAAAAAAAP////
8IAAAABwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABAAAA/////0atE6ykdLw/NjMuYLLUW6dErinN
8y8ohN80SY4+trYgAAAAAH8CAAAAuAEAAAm5AQAARQIAAAlGAgAATQIAAAh/gAACTQAAAAAAAAAAAA/////w
AAAAAAAAAA/////w8AAAATAAAAFAAAAEFuY2VzdG9yc19pbmRleH8CAAAAuAEAAAm5AQAAAAAAAR/AgAAALgB
AAAJuQEAAP///////wAA</controlValue>
    </control>
    <resultCode code="0"/>
  </searchResultDone>
</searchResponse>
```

In the above result example the decoded value is:
size = 0

```
cookie      =     <binary cookie>
```

The returned cookie should be treated as an opaque structure and passed exactly as received from the server back with the next paging request.

Our system does not give estimates on the search requests filter. According to the [RFC2696] this functionality is optional. So, our implementation of the ITI-58 transaction will always return zero (0) for the field “size” in the control value of the search response.

([RFC2696]*: “In the control returned to the client, the size MAY be set to the server's estimate of the total number of entries in the entire result set. Servers that cannot provide such an estimate MAY set this size to zero (0).”*)

According to [RFC2696], in the last paged search response the control value’s “cookie” field will be *null* again.

([RFC2696]: *“The cookie MUST be set to an empty value if there are no more entries to return (i.e., the page of search results returned was the last), or, if there are more entries to return, to an octet string of the server's choosing, used to resume the search.”*)

The control is ignored if the page size is larger or equal to the overall size limit of the search request as the request can be satisfied in a single page. A size limit exceeded result code (= 4) is returned if the actual response contains more entries than the server limitations allow (see also section 3.2.7 Size limit of search requests) or the client’s size limit in the search request specifies.

### 4.4.2.3 Sorting

The Provider Information Directory Query transaction supports server side sorting. As specified by [RFC2891] you can provide a LDAP control with your search request. The concept is similar to the paging mechanism described in 4.4.2.2, except the format of the BER encoded data structure used as the control value.

**Example request:**

```
<searchRequest dn="DC=HPD,O=BAG,C=CH" scope="wholeSubtree"
               derefAliases="neverDerefAliases">
<control type=" 1.2.840.113556.1.4.473" criticality="true">
  <controlValue
xsi:type="xsd:base64Binary">MIQAAAAUMIQAAAAOBAxIY0lkZW50aWZpZXI=</controlValue>
</control>
  […]
</searchRequest>
```

There are some limitations to our LDAP implementation regarding sorting. First you can only sort on one single attribute and second you can not specify a matching rule id and are forced to leave that element empty (=null).

→ You will receive a “12: Unavailable Critical Extension” if you provide more than one field in the SortKeyList sequence (control value).

→ You will receive a “12: Unavailable Critical Extension” if you provide a MatchingRuleId in the control value.

## Behaviour

As it is impossible to provide a MatchingRuleId defining a localized ordering rule like French:Switzerland or German:Switzerland, the sorting is done in a language independent manner accordingly. This mean that in all orderings, neither the phonetic resemblance nor word stems have an influence on sorting.

Our experiments showed the following sorting behaviour:

### Directory String

- The sorting is done alphabetically, ascending or descending, depending on the reverseOrder Boolean specified in the control.
- Numbers are before letters.
- Lowercase letters are before uppercase letters.
- Lowercase and uppercase letters are kept together.
- Letters without accents are before letters with accents.
- Letters with and without accents are kept together.
- The accent ordering within the same letter is different to an ordering in Excel.
- Special characters from completely different cultures show up at the end.

**Example**

| **SortDescription-Asc** | **SortDescription-Desc** |
|---|---|
| en:01 | en:カbcdefg |
| en:0bcdefg | en:力bcdefg |
| en:9abcdefg | en:zz |
| en:aaaaaaa | en:Zaaaaaa |
| en:aAaaaaa | en:z |
| en:abcdefg | en:ýbcdefg |
| en:ABCDEFG | en:Ýbcdefg |
| en:àbcdefg | en:ÿbcdefg |
| en:Ábcdefg | en:waaaaaa |
| en:âbcdefg | en:vaaaaaa |
| en:Âbcdefg | en:ùbcdefg |
| en:zzaaaaa | en:977 |
| en:ZZaaaaa | en:97 |
| en:力bcdefg | en:9 7 |
| en:カbcdefg | en:0bcdefg |

**Octet String** (binaries)

- The sorting seems to be done according the Unicode value of the letter, ascending or descending, depending on the reverseOrder Boolean specified in the control.
- Numbers are before letters.
- Uppercase letters are before lowercase letters.
- Uppercase and lowercase letters are not kept together. First all uppercase letters, then all lowercase letters.
- Letters without accents are before letters with accents.
- Letters with and without accents are not kept together. First all letters without accents, then all letters with accent.

**Example**

| **SortDescription-Asc** | Original Text | **SortDescription-Desc** | Original Text |
|---|---|---|---|

| (binary) | | (binary) | |
|---|---|---|---|
| QUJjZGVm | ABcdef | w6RiY2RlZg== | äbcdef |
| QWJiZGVm | Abbdef | w6BiY2RlZg== | àbcdef |
| QWJjZGVm | Abcdef | w4RiY2RlZg== | Äbcdef |
| QkJjZGVm | Bbcdef | w4BiY2RlZg== | Àbcdef |
| YWJiZGVm | abbdef | YWJjZGVm | abcdef |
| YWJjZGVm | abcdef | YWJiZGVm | abbdef |
| w4BiY2RlZg== | Àbcdef | QkJjZGVm | Bbcdef |
| w4RiY2RlZg== | Äbcdef | QWJjZGVm | Abcdef |
| w6BiY2RlZg== | àbcdef | QWJiZGVm | Abbdef |
| w6RiY2RlZg== | äbcdef | QUJjZGVm | ABcdef |

**Generalized Time**

- Sorting is not possible. The LDAP System returns the error code 12:
  "An error occured while executing SearchRequest (request id='queryRequest valid', result code='UnavailableCriticalExtension') ... problem 5010 (UNAVAIL_EXTENSION)"

**DN**

- Sorting is not possible. The LDAP System returns the error code 12:
  "An error occured while executing SearchRequest (request id='queryRequest valid', result code='UnavailableCriticalExtension') ... problem 5010 (UNAVAIL_EXTENSION)"

### 4.4.3 Filter

The filter is evaluated structurally. Filters with unknown or undefined attributes result in an LDAP response code 16 (“No such attribute”).

### 4.4.4 Attribute selection

All entities (except for operational) attributes will be returned if there is no explicit attribute projection is provided in the search.

Only attributes explicitly listed in the documented LDAP schema (see section 4.2) can be projected.

## 4.5 Provider information feed (ITI-59)

### 4.5.1 Supported request types

Only *addRequest*, *modifyRequest*, *modDnRequest* and *delRequest* entries are supported in a batch for the feed transaction. The whole batch will be rejected with a SOAP fault if another request type is detected.

### 4.5.2 Distinguished names

The complete distinguished name represent the primary key of an entry in the LDAP. To allow multiple tenants to add and edit entries without conflicts, the following rules are enforced on all operations that manipulate distinguished names.

The distinguished name must be syntactically correct (e.g. no equals "=" or commas ","). → Violations lead to a response with result code 34: Invalid DN Syntax.

The correct rDN key must be used, that corresponds to the requested object class (as specified in chapter 4.2). → Violations lead to the result code 64: Naming Violation.

The distinguished name must be prefixed with the assigned "community prefix" from the CPI attribute "shcIssuerName" (section 5, Community Portal Index (CPI)). The rDN value format is

```
rdnValue    : = <prefix>:<id>“
prefix      : = CPI.shcIssuerName
id          : = <any valid character>
```

→ If you send requests with a prefix other than the one assigned to you, a response with the result code 50: Insufficient access rights will be returned.

Example, the following add request from a community with the prefix "CommunityA":

```
<addRequest
dn="uid=CommunityC:1,OU=HCRegulatedOrganization,DC=HPD,O=BAG,C=ch">
…
</addRequest>
```

Leads to the addResponse with the result code 50: Insufficient access rights:

```
<addResponse>
   <resultCode code="50"/>
</addResponse>
```

## 4.5.3 Attribute validation

In this chapter we provide information about the technically enforced validations on attribute values. According to underlying standards such as [IHE-HPD] or [ISO 21091] there are more format restrictions, but only the ones below are enforced in the EPR central services.

The rules are applied in all requests that directly or indirectly alter the mentioned attributes.

### 4.5.3.1 objectClass attributes

Entries in the directory organizational units (HcProfessional, HcRegulatedOrganization and Relationship) must fulfil the following constraints. These formal constraint will be validated for all mutative requests. Take note, that the optional object classes in the table below are inferred automatically (from the inheritance chain defined in the schema) if the caller omits those. But the caller is allowed to explicitly provide the complete inheritance chain if he wishes to do so. The only exception to this is the class "naturalPerson" which is auxiliary and optional on the HcProfessional.

| OU (Entity type) | Required object classes | Optional inherited object classes | Optional auxiliary object classes |
|---|---|---|---|
| **HcProfessional** | HpdProvider<br>HcProfessional | top<br>inetOrgPerson | naturalPerson |

| | | | |
|---|---|---|---|
| | | person<br>organizationalPerson | |
| **HcRegulatedOrganization** | HpdProvider<br>HcRegulatedOrganization | top<br>organization | uidObject |
| **Relationship** | groupOfNames | top | |

→ All add requests that omit the required object classes, provide not allowed object classes or add entries to an incorrect organizational unit will lead to result code 19: Constraint violation.
→ All modify requests that try to remove required object classes or provide not allowed object classes will lead to result code 19: Constraint violation.

### 4.5.3.2 Read-only and operational attributes

For all attributes that are not writable by the caller, but are calculated by the system, it is expected that all add or modify requests omit these attributes.

For all requests that try to add or modify one or more of the following attributes.
→ Violations lead to 19: Constraint Violation

```
hpdProvider.memberOf
top.createTimestamp
top.modifyTimestamp
```

### 4.5.3.3 Distinguished name references

The rules outlined in section 4.5.2 are also enforced when specifying a DN-reference by issuing an add or modify request (you can only reference entries of your own prefix).

The following table gives overview about the affected attributes of type “DN” and additional rules that apply.

| Attribute | Remarks |
|---|---|
| groupOfNames.owner | Only organizational provider can be referenced (OU=HCRegulatedOrganization) or community (OU=CHCommunity).[1] |
| groupOfNames.member | |
| HCProfessional.HcPracticeLocation | |
| HCRegulatedOrganization.ClinicalInformationContact | |
| HPDProvider.memberOf | Calculated inverse attribute of groupOfNames.member. Cannot be manipulated. |

[1] For building a tree of individual and organizational providers it does not make sense to have an individual provider referenced by groupOfNames.owner. It is therefore technically enforced that only references to organizations can be added or modified.
→ Violations will lead to 19: Constraint Violation

### 4.5.3.4 Metadata attributes

Attributes of this type are validated against a specific value set in the metadata index. The

format must be provided in the following format:

```
MdiCodeValue      := BAG:<CodeSystem>:<Code>[:<DisplayName>]
CodeSystem        := OID of the  code system
Code              := Code
DisplayName       := Display name of the code
```

→ All attribute values that do not correspond to the defined format, will lead to a result code 21: Invalid attribute syntax

| Attribute | Value set id |
|---|---|
| HCProfessional.HcProfession | 2.16.756.5.30.1.127.3.10.1.1.3 |
| HCRegulatedOrganization.HcSpecialisation and HCProfessional.HcSpecialisation | 2.16.756.5.30.1.127.3.10.1.18 |
| HCRegulatedOrganization.businessCategory | 2.16.756.5.30.1.127.3.10.1.11 |

For modifications on the above attributes each value is validated against the active version of the value set.
→ A 19: Constraint violation will be returned if the code system and code combination is not found.

`DisplayName` is optional and only allowed for attribute *HcSpecialisation* on *HCProfessional* and *HCRegulatedOrganization*.
→ All attributes beside HcSpecialisation will lead to a result code 21: Invalid attribute syntax if a DisplayName part is delivered in the value.

All metadata attributes are validated for uniqueness on a code level.
→ A 19: Constraint violation will be returned if the attribute values differ only on its optional display name part.

#### 4.5.3.5 Status attributes

The status attribute, defined on *HPDProvider.hpdProviderStatus*, is validated according to the IHE HPD profile [IHE-HPD]. This ensures all individual and organizational provider entries have a valid status.

| Provider type | Allowed status |
|---|---|
| Individual provider (HCProfessional) | Active<br>Inactive<br>Retired<br>Deceased |
| Organizational Provider (HCRegulatedOrganization) | Active<br>Inactive |

→ Violations lead to the result code 19: Constraint violation.

#### 4.5.3.6 Registration status attribute

The HC registration status attribute, defined on *HCProfessional.hcRegistrationStatus*, is validated although this attribute is not in use yet. Its value must be “unknown” (case insensitive). This ensures that all individual entries have the same fixed value for an undefined status.

→ Violations lead to the result code 19: Constraint violation.

#### 4.5.3.7 Gender attribute

The gender attribute, defined on naturalPerson, is validated according to the Swiss National Extensions and [RFC2985]. This ensures all individual provider entries have a valid gender.

| Provider type | Allowed status |
|---|---|
| Individual provider (HCProfessional) | m (male)<br>f (female) |

→ Violations lead to the result code 19: Constraint violation.

#### 4.5.3.8 Common name attribute

The common name (= *cn*) attribute on Person is validated according the structure specified in [ISO 21091], (*9.2.2.3, "Common Name"*). Only the comma separated structure is validated but not the actual content of the individual parts (i.e. surname, given names and UID):

```
Common-Name  ::= [Surname] ',' [Given-Names] ',' [UID]
Surname      ::= Cn-string
Given-Names  ::= Cn-string
UID          ::= Cn-string
Cn-string    ::= any string without ','
```

Whitespaces and empty values for `Surname`, `Given-Names` and `UID` are allowed. Only the person's first name should be used for `Given-Names`.

→ Violations lead to the result code 19: Constraint violation.

### 4.5.4 Referential integrity

For all attributes of type "DN" (see 4.2 Directory schema) the following referential integrity behaviour can be expected.

Consider element *A* referencing element *B* through attribute "ClinicalContactInformation" as an example for easier explanation.

If you delete element *B*, the following will happen. The value of *B* in the attribute *A.ClinicalContactInformation* will be deleted automatically. If *A.ClinicalContactInformation* contained only one value before the deletion of B, it will be empty/undefined. Otherwise *A.ClinicalContactInformation* will contain one value less, in this case the value "*B*".

If you rename the element *B* with a modDNRequest to a new name *C*, *A.ClinicalContactInformation* will be automatically updated with the new name *C*.

### 4.5.5 Request specifics

The following chapters describe behaviour that is specific to the implementation of the central service HPD. All not documented below can be expected to adhere to standard DSML/LDAP behaviour.

#### 4.5.5.1 ModDN request

ModDN requests will be refused if the DSML attribute *newSuperior* is set, as it is not allowed to move elements around in the DIT.

In ModDN requests only **relative** DN's like "newrdn="uid=CommunityC:00000001009"" are allowed in the attribute *newRdn*. If the **full** DN path (like *"newrdn="uid=CommunityC:00000001009,OU=HCRegulatedOrganization,DC=HPD,O=BAG,C=ch""*) is set, it leads to the resultCode 34 with the errorMessage *"Only RDNs are allowed as Attribute 'newrdn' value. Detected a full DN: 'uid=CommunityC:00000001009,OU=HCRegulatedOrganization,DC=HPD,O=BAG,C=ch'"*.

ModDN requests will be refused if the OU part of the DN contains an inexistent organization unit.

E.g. the request with the DN *"dn="uid=CommunityC:00000001099,OU=HCRegulatedOrganization,DC=HPD,O=BAG,C=ch""* will work whereas the request with the DN *"dn="uid=CommunityC:00000001099,OU=HCNotExisting,DC=HPD,O=BAG,C=ch""* will fail with the resultCode 50 and the errorMessage *"Entry can only be manipulated inside a valid OU. Check DN of requested entry: 'uid=CommunityC:00000001099,OU=HCNotExisting,DC=HPD,O=BAG,C=ch'."*

#### 4.5.5.2 Add request

Additions to groupOfNames entries under OU=Relationship are validated to ensure that if the attribute owner contains an OU=CHCommunity, the member attribute only can contain OU=HCRegulatedOrganizations.

→ All additions not conforming to this rule will result in code 19: Constraint violation

The owner attribute is mandatory and single valued. Multiple additions will result in code 20: Attribute or value exists, no additions will result in code 19: Constraint violation.

#### 4.5.5.3 Modify request

Modifications on groupOfNames entries under OU=Relationship are validated to ensure that only add or delete modifications can be performed.

→ All replace modifications will result in code 53: Unwilling to perform.

Modifications on groupOfNames entries under OU=Relationship are validated to ensure that if the attribute owner contains an OU=CHCommunity, the member attribute only can contain OU=HCRegulatedOrganizations.

→ All modifications not conforming to this rule will result in code 19: Constraint violation

The owner attribute is mandatory and single valued. Additions without prior delete will result in code 20: Attribute or value exists, delete without following adding will result in code 19: Constraint violation.

#### 4.5.5.4 Delete request

To avoid orphaned relationship entries the system validates for each organization to delete

that there is no owner reference of a relationship. When deleting entries of object class “HcRegulatedOrganization” you need to ensure that the elements are not referenced as a groupOfName.owner.

→ If the delete request concerns a HcRegulatedOrganization and the entry is referenced by one or more groupOfNames.owner you receive a result code 19: Constraint Violation.

## 4.6 Provider information delta download (CH:PIDD)

The provider information delta download transaction is used to synchronize a local replica of the directory with the master directory by fetching all changes at the master during a certain time span (at DSML request level).

Your own local changes which have already been synchronized back to the master directory using the feed operation may be of no interest for the client anymore. That's why a client can filter his own requests from the actual synchronization data with an additional PIDD parameter.

The client can use the DSML requests from the PIDD response and execute them against his local replica. Only successful DSML requests (resultCode = 0) from the feed operation appear in the PIDD.

The client himself has to keep track about the state of his replica, i.e. which requests from the PIDD have already been synchronized to the local replica.

The feed request execution time (server side) will act as a kind of directory version. Feed requests that have successfully been executed against the master directory appear immediately in the PIDD even if other requests in the same batch failed or have not been executed yet.

PIDD time values are always in UTC.

### 4.6.1 PIDD response

The PIDD response is a chronological list of DSML requests. The requests are grouped in batch requests. The *authRequest* of each of these batch requests indicates the responsible community for the original DSML request (CPI → *shcIssuerName* attribute, see [EPR-CPI-Schema]). The requestID of the original DSML requests is overwritten with the UTC execution timestamp of the request at the master directory (ISO 8601 format). A client that synchronizes without explicit *toDate* can use the last requestID timestamp as the new *fromDate* for the next synchronization run.

**Example:**

```
<downloadResponse>
  <batchRequest onError="resume">
    <authRequest principal="community1"/>
    <addRequest requestID="2018-03-12Z15:20:30.1234568Z"> ... </addRequest>
    <addRequest requestID="2018-03-12Z15:20:30.7765831Z"> ... </addRequest>
    <addRequest requestID="2018-03-12Z15:20:30.9692847Z"> ... </addRequest>
  </batchRequest>
  <batchRequest onError="resume">
    <authRequest principal="community2"/>
    <addRequest requestID="2018-03-15Z18:11:46.8745478Z"> ... </addRequest>
    <delRequest requestID="2018-03-15Z18:11:46.8745552Z"> ... </delRequest>
    <modifyRequest requestID="2018-03-15Z18:11:46.8745791Z"> ... </modifyRequest>
  </batchRequest>
</downloadResponse>
```

### 4.6.2 Time resolution

The time resolution for the execution timestamp at the master directory is on 7th fractional seconds precision. Requests for PIDD data with a higher precision for the time range will get

rounded (rounding to nearest 7th precision using half to even). This may result in rounding issues. It's not recommended to use a higher precision for the execution timestamp at the client than the master uses.

### 4.6.3 Batch settings

The DSML batches returned in the PIDD transaction always have the same batch settings:

- **responseOrder:** sequential (= DSML default)
- **processing:** sequential (= DSML default)
- **onError:** resume

# 5. Community Portal Index (CPI)

The CPI is a Swiss specific implementation. The interface provides one operation that implements the elements of the DSML protocol (CH:CPI)

## 5.1 Directory Schema

The CPI base distinguished name / directory root is “dc=CPI,o=BAG,c=CH”. All child elements of this node are considered a part or the CPI.

### 5.1.1 Versioning

Although the CPI is considered the master in the EPR central services, every community or vendor is free to have a local replicate of the directory. For synchronisation only the community information query (CH:CPI) can be used.

In the EPR central services there is no explicitly versioned schema or corresponding version query operation available on machine to machine level.

### 5.1.2 Standard precedence

The object class and attribute schema is based on the requirements of the Federal Office of Public Health, eHealth Suisse and software vendors:

1. IHE CH extensions [BAG-A5]

### 5.1.3 Object Classes and Organisational Units

The central services offer five object classes defined in the LDAP schema.

| Element | Object class | Organisational Unit |
|---|---|---|
| Community | CHCommunity | CHCommunity |
| XCA Initiating Gateway | CHXcaInitGw | CHEndpoint |
| XCA Responding Gateway | CHXcaRespGw | CHEndpoint |
| XCPD Initiating Gateway | CHXcpdInitGw | CHEndpoint |
| XCPD Responding Gateway | CHXcpdRespGw | CHEndpoint |
| Authorization Decision Provider Gateway | CHAuDecProv | CHEndpoint |
| Authorization Decision Consumer Gateway | CHAuDecCons | CHEndpoint |
| Assertion Provider | CHAssertProv | CHEndpoint |

An overview of the complete LDAP schema is shown in appendix B. A complete list of object classes, attributes and their detailed definition and description can be obtained from [EPR-CPI-Schema].

### 5.1.4 Gateway references and naming conventions

Each community has a set of gateway information for the inter-community communication. The semantical and technical information about the gateways and their functionality is not part of this document. Only the information structure is described further.

Each community of object class “CHCommunity” has attributes (e.g. shcXcaIniGW, see [EPR-CPI-Schema] Column “Datentyp=DN”) that reference the corresponding gateway, assertion provider or authorization decision elements in the organisational unit “CHEndpoint”. Furthermore, each gateway element is prefixed with the shcIssuerName of its community.

| CHCommunity DN reference | Naming convention |
|---|---|
| shcXcaIniGW | uid=<shcIssuerName>:XcaInitiatingGateway |
| shcXcaRespGW | uid=<shcIssuerName>:XcaRespondingGateway |
| shcXcpdIniGW | uid=<shcIssuerName>:XcpdInitiatingGateway |
| shcXcpdResGW | uid=<shcIssuerName>:XcpdRespondingGateway |
| shcAuDecProv | uid=<shcIssuerName>:AuthorizationDecisionProvider |
| shcAuDecCons | uid=<shcIssuerName>:AuthorizationDecisionConsumer |
| shcAsPrIsCrt | uid=<shcIssuerName>:AssertionProviderIssuerCertificate |

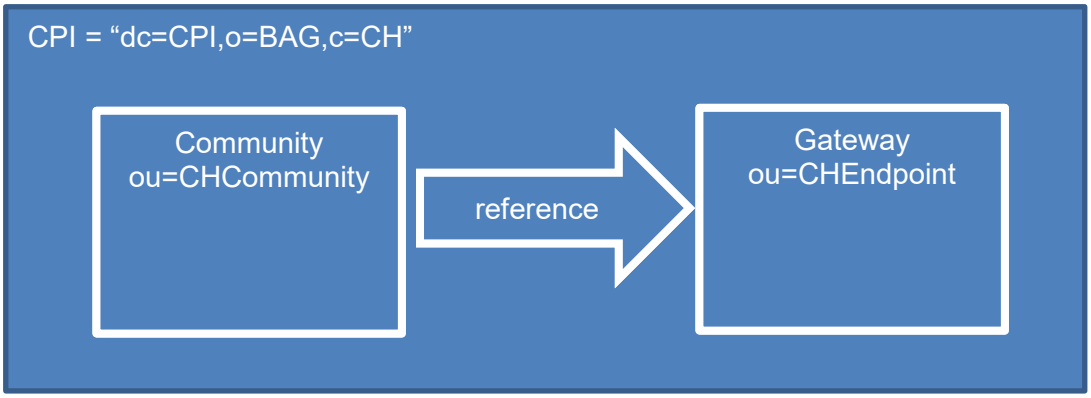
Note: The LDAP display name of the CPI attributes has a maximum length of 16 characters (eg. *shcXcaIniGW* for the *shcXcaInitiatingGateway* attribute). The complete LDAP-Mappings can be found in [EPR-CPI-Schema].

## 5.2 Community information query (CH:CPI)

The community portal index is implemented using a DSML interface. It uses the same DSML searchRequest protocol as is used for HPD search requests ((see 4.4 “Provider information query (ITI-58)”). The only difference is the use of different SOAP target and operation namespaces (see [EPR-WSDL]).

The only DSML element supported is the “*searchRequest”* on the CPI root or one it’s direct or indirect child elements. This means the CPI interface is read-only from a machine to machine perspective. The data can only be manipulated by the Federal Bureau of Public Health.

As mentioned in chapter 5.1.3, the community is structured in two separate organisational units underneath “dc=CPI,o=BAG,c=CH”.




You can query all community information by issuing a search request on the organisational unit “OU=CHCommunity,DC=CPI,O=BAG,C=ch” or by using the object class “CHCommunity” in a filter.

For all gateway information, the organisational unit “OU=CHEndpoint,DC=CPI,O=BAG,C=ch” can be queried or alternatively you could construct an object class filter containing all the object class values used in that OU.

To find the gateway information of a given community you can traverse the distinguished name reference of that community to its gateway elements or you can use the attribute shcIssuerName to filter the gateway elements; you could filter on the distinguished name only retrieving element that begin with '<shcIssuerName>:'. This would allow you to find all gateways of a given community without traversing the distinguished name reference attributes.

## 5.3 Community information delta download (CH:CIDD)

The CIDD serves the same purpose as the PIDD, but with community data. It provides the same structure in the form of a list of batchRequest elements that occurred for a given time interval. It can be used to replicate CPI data and detect changes without the need to compare CPI data with your local replica.

Although the interface is very similar, it uses its own namespace declarations for both request and response elements (see [EPR-WSDL]).

### 5.3.1 Request

There are two request parameters (compared to the three in the PIDD).

| Name | Description |
|---|---|
| **fromDate** | Required, lower interval boundary (date and time, inclusive). |
| **toDate** | Optional, upper interval boundary (date and time, inclusive). The current global EPR time will be used if not defined in the request. |

### 5.3.2 Response

The response consists of a sequence of DSML *batchRequest* elements. You can expect the mutative operations *addRequest*, *modifyRequest*, and *delRequest*. The grouping of the requests in multiple *batchRequest* elements expresses the different operations the administrator of the EPR central services executed against the CPI. Please be aware, that it is not guaranteed that a *batchRequest* contains a complete transaction of CPI data that will lead to a consistent state. If you set *toDate* explicitly it can easily happen that only half of the request that make a community complete are returned in the response (e.g. only modification of the community without the endpoint elements).

As with the PIDD, each request element contains the time of execution in the *requestID* attribute.

**Example:**

```
<downloadResponse xmlns="urn:ch:admin:bag:epr:2017">
  <batchRequest onError="resume" xmlns="urn:oasis:names:tc:DSML:2:0:core">
    <addRequest requestID="2017-12-11T11:55:31.7643345Z"
                dn="uid=test1,ou=chcommunity,dc=cpi,O=BAG,C=ch">
      <!-- details ommited -->
    </addRequest>
  </batchRequest>
  <batchRequest onError="resume" xmlns="urn:oasis:names:tc:DSML:2:0:core">
    <addRequest requestID="2017-12-12T08:09:52.7154691Z"
                dn="uid=test1:XcaInitiatingGateway,ou=chendpoint,dc=cpi,O=BAG,C=ch">
      <!-- details ommited -->
    </addRequest>
```

```
    <modifyRequest requestID="2017-12-12T08:09:52.7464564Z"
                   dn="uid=test1,ou=chcommunity,dc=cpi,O=BAG,C=ch">
      <!-- details ommited -->
  </modifyRequest>
  </batchRequest>
  <batchRequest onError="resume" xmlns="urn:oasis:names:tc:DSML:2:0:core">
    <modifyRequest requestID="2017-12-12T08:10:19.2713348Z"
                   dn="uid=test1,ou=chcommunity,dc=cpi,O=BAG,C=ch">
      <!-- details ommited -->
    </modifyRequest>
  </batchRequest>
  <batchRequest onError="resume" xmlns="urn:oasis:names:tc:DSML:2:0:core">
    <delRequest requestID="2017-12-12T13:35:53.2923242Z"
                dn="uid=test1:XcaInitiatingGateway,OU=CHEndpoint,DC=CPI,O=BAG,C=ch"/>
  </batchRequest>
</downloadResponse>
```

The CIDD uses the same time resolution (see **Time resolution**, 4.6.2) and batch settings (see 4.6.3) as the PIDD.

# 6. Metadata Index (MDI)

## 6.1 General

The Metadata Index is implemented with the SVS transaction ITI-48 (Retrieve Value Set) and ITI-60 (RetrieveMultiple Value Set) with the SOAP 1.2 and HTTP Binding.

## 6.2 Retrieve Value Set (ITI-48)

### 6.2.1 Request behaviour

The Retrieve Value Set request has only one mandatory parameter *id*. If you request a specific ValueSet but do not specify a *version* the MDI will chose the most recent version for you.

The most recent version is defined in ZAD as follows: the most recent (i.e. valid) *ValueSet* in respect of the metadata *EffectiveDate*. More precisely, the *ValueSet* with the highest *EffectiveDate* but where *EffectiveDate* is not higher than the current date (server time).

The request parameter *xml:lang* is supported, but will only produce results if left empty or the value “en-US” is supplied.

Following situation will lead to a SOAP fault:

- Missing elements in the SOAP body (e.g. *RetrieveValueSetRequest* or *ValueSet* element).
  Missing or misspelled *id* attribute on *ValueSet* element.
  Unknown value set id
  (*Note:* conforming to IHE SVS fault code → *“NAV”*)
  Unknown value set version
  (*Note:* conforming to IHE SVS fault code → *“VERUNK”*)
- Unknown or undefined language for a requested value set id and/or version

(Note: this behaviour is not specified by IHE SVS → *"LANGUNK"*)

Instead of setting directly the SOAP fault code to the specified IHE SVS fault code, the sub code is set. The top level fault code is always "Sender" which conforms to SOAP 1.2 standard (see [SOAP12], chapter 5.4.6).

So, MDI fault sub codes in ITI-48 are:

| Code | Reason text | Description |
|---|---|---|
| **NAV** | Unknown value set | IHE standard |
| **VERUNK** | Version unknown | IHE standard |
| **LANGUNK** | (Concept)Language '{language}' not supported. | CH extension |

The reason text is always in *"en-US"*, there's no multi language support in the error message.

### 6.2.2 Caching of the responses

According to the IHE SVS profile, the response data *cacheExpirationHint* is optional.

This cache hint is not supported by the current implementation (neither the SOAP nor the HTTP binding have currently support for it).

### 6.2.3 Response behavior

You can expect only active *ValueSet*'s returned.
The MDI will only return *Concept*'s in American English, so the *ConceptList* attribute *xml:lang* will always equal to "en-US".

You can expect following elements and attributes with non-empty values to be present in the response:

*ValueSet* (exactly one element)
*ValueSet/ConceptList* (exactly one element)
*ValueSet/ConceptList/@id*
*ValueSet/ConceptList/@version*
*ValueSet/ConceptList/@xml:lang*
*ValueSet/ConceptList/Concept* (one or more elements)
*ValueSet/ConceptList/Concept/@code*
*ValueSet/ConceptList/Concept/@codeSystem*
*ValueSet/ConceptList/Concept/@displayName*

## 6.3 Retrieve Multiple Value Set (ITI-60)

### 6.3.1 Request behaviour

Following filter behaviour is currently implemented for retrieving multiple value sets:

#### 6.3.1.1 Equal filter

A case-sensitive equal filter on the corresponding property.

#### 6.3.1.2 Contains filter

A case-insensitive contains filter on the corresponding property. This behaviour differs from the IHE SVS definition: POSIX regular expressions are not supported.

#### 6.3.1.3 Date filter

For filtering on date range two filter elements are provided. One to filter on the upper and one to filter on the lower end of the time interval. The date must be provided in xml date format. It will be silently ignored by the system if an additional time value is provided.

#### 6.3.1.4 Parameter list

From the following parameters, at least one must be specified. Otherwise the system will return a SOAP fault with the fault code “Sender” and no specific sub code.

| Element | Description |
|---|---|
| ID | An equal filter on *ValueSet.ID*. |
| DisplayNameContains | A case sensitive contains filter on *ValueSet.displayName*. |
| SourceContains | A case sensitive contains filter on *ValueSet.source*. |
| PurposeContains | A case sensitive contains filter on *ValueSet.purpose*. |
| DefinitionContains | A case sensitive contains filter on *ValueSet.definition*. |
| GroupContains | A case sensitive contains filter on *ValueSet.Group.displayName*. |
| GroupOID | An equal filter on *ValueSet.Group.ID*. |
| EffectiveDateBefore | Before or equal on *ValueSet.effectiveDate*. If a time is provided |
| EffectiveDateAfter | Equal or after on *ValueSet.effectiveDate* |
| ExpirationDateBefore | Before or equal on *ValueSet.effectiveDate*. If a time is provided |
| ExpirationDateAfter | Equal or after on *ValueSet.expirationDate* |
| CreationDateBefore | Before or equal on *ValueSet.expirationDate*. If a time is provided |
| CreationDateAfter | Equal or after on *ValueSet.creationDate* |
| RevisionDateBefore | Before or equal on *ValueSet.revisionDate*. If a time is provided |
| RevisionDateAfter | Equal or after on *ValueSet.revisionDate* |
| Format | This parameter is ignored. |

### 6.3.2 Response behaviour

Because there is no selection of a specific language (in contrast to the *“Retrieve Value Set”* transaction) and the IHE SVS allows only one ConceptList element per matched value set, the resulting concept display names will always be in English language.

You can expect following elements and attributes with non-empty values to be present in the response:

*DescribedValueSet* (exactly one element)
*DescribedValueSet/Source* (exactly one element)
*DescribedValueSet/Purpose* (exactly one element)
*DescribedValueSet/Definition* (exactly one element)
*DescribedValueSet/Status* (exactly one element)
*DescribedValueSet/Type* (exactly one element)
*DescribedValueSet/EffectiveDate* (exactly one element)

*DescribedValueSet/ConceptList* (exactly one element)
*DescribedValueSet/ConceptList/@id*
*DescribedValueSet/ConceptList/@version*
*DescribedValueSet/ConceptList/@xml:lang*
*DescribedValueSet/ConceptList/Concept* (one or more elements)
*DescribedValueSett/ConceptList/Concept/@code*
*DescribedValueSet/ConceptList/Concept/@codeSystem*
*DescribedValueSet/ConceptList/Concept/@displayName*
*DescribedValueSet/Group* (zero or more elements)
*DescribedValueSet/Group/@id*
*DescribedValueSet/Group/@displayName*
*DescribedValueSet/Group/@sourceOrganisation*
*DescribedValueSet/Group/Keyword* (zero or more elements)

## 6.4 Error Handling

### 6.4.1 General

Errors are returned as described in [IHE-TF-2b]. The following specific additional errors have been defined for the government central services.

### 6.4.2 Specific error: Query String violation

The system has been reached but the HTTP protocol has been broken or some basic validation for the services call failed. The processing is aborted in an early pipeline stage of the central services.

**Examples:**

- Malformed HTTP Query String (i.e. no well-formed list of key/value pairs for the request parameters)
- HTTP Query String Parameters do not match the endpoints specification (i.e. missing required or not specified parameters)

**HTTP Reponse definition**

The central services return a Bad Request status code (400) if the service contract, i.e. the request's query string specification, has been violated.

The following HTTP Warning header (see [RFC7234] section 5.5) are returned if a query string violation occurs:

| Scenario | Value |
|---|---|
| HTTP query string violation | 111 epr-cs "Bad request: <reason>" |

The content of the query string violation response is a SOAP Fault

| Scenario | Code value | Sub code value |
|---|---|---|
| HTTP query string violation | Sender | HTTP_QUERY_STRING_VIOLATION<br><br>*Namespace:*<br>urn:ch:admin:bag:epr:2017 |

# Appendix

## A. LDAP result codes

Standard LDAP result codes are returned by the EPR central services. The following table contains an extract of them explicitly mentioned in this interface documentation.

| Code | Description |
|---|---|
| 2 | *Protocol error*<br>Basic protocol standards are violated.<br>E.g.<br>• modDN request is missing the newRDN |
| 16 | *No such attribute*<br>Attribute is not defined on the entry's object class. |
| 19 | *Constraint violation*<br>Different server side enforced constraints were violated.<br>E.g.<br>• add to single valued attribute<br>• non-existing MDI code<br>• not allowed object class<br>• invalid rDN format<br>• not allowed references (wrong type)<br>• invalid HPD provider status format |
| 21 | *Invalid attribute syntax*<br>The attribute syntax was wrong.<br>E.g.<br>• MDI code format error |
| 32 | *No such object*<br>Entry does not exist in the directory. |
| 50 | *Insufficient access rights*<br>Modifications outside of the allowed organizational units.<br>Modifications outside of the assigned community domain (prefix). |
| 53 | *Unwilling to perform*<br>The request has a valid syntax but makes no sense or does not conform the intended use of the service.<br>E.g.<br>• A request is not fully specified (Add is missing the DN) |
| 65 | *Object class violation*<br>Missing required fields on an add request.<br>Missing specific attribute values that are required on the entry.<br>E.g.<br>• hcIdentifier constraints<br>• Gender constraints<br>• Owner constraints |
| 68 | *Entry already exists*<br>The DN of an add request already exists in the directory. |

| Code | Description |
|---|---|
| 87 | *Filter error*<br>The filter is not valid:<br>E.g.<br>• Logical And with just one operand |

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

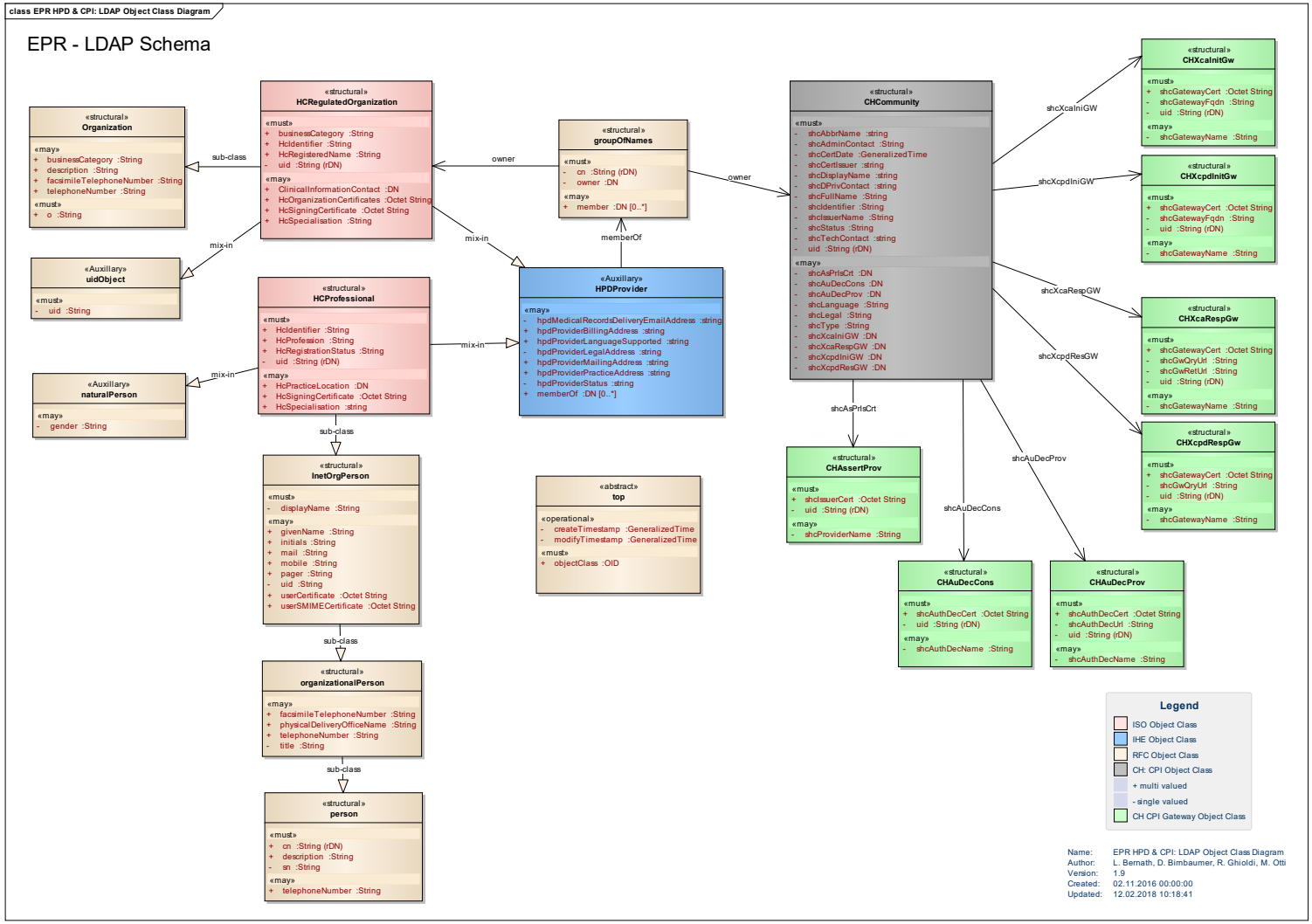
Swiss Confederation

Federal Department of Finance FDF

**Federal Office of Information Technology, Systems, and Telecommunication FOITT**

PRO-IEW-EWM

# B. LDAP Schema Overview



# C. Interface Changelog

Version 0.13.x

- HPD
  - If ModifyDN.NewRdn is missing: Not the whole batch is rejected. Further, result code 2 ("Protocol error") is returned instead of 67 ("Not Allowed on RDN").

Version 0.14.x

- HPD
  - *hcIdentifier* of hpdProvider class has changed to multi-valued attribute. Validation for organizations to have at least one attribute value with *"RefData:OID:"* prefix.
- CPI
  - New object classes for community gateway information: CHInitiatingGateway, CHRespondingGateway, CHAuthorizationDecisionGateway, CHAssertionProvider.
  - New and changed attributes for CHCommunity object class.
  - New OU=CHEndpoint to store community gateway information

Version 0.16.x

- HPD
  - New mandatory attribute HcRegistrationStatus on object class HCProfessional
  - Added correlation id in HTTP response header for support cases

Version 0.17.x

- HPD
  - uidObject is optional object class for HcRegulatedOrganization
- MDI
  - Clarify MDI behavior including what *“the most recent version of the Value Set”* means (see [IHE-TF-2b], section 3.48.4.1.2)
  - Minor XML schema fixes because implementation was based on incorrect IHE implementation material (SOAP actions, case sensitivity of id attribute, wrong HL7 type in ITI-48, etc.).

Version 0.18.x

- CPI
  - LDAP schema change for CPI: renamed/shortened object class and attribute names. See details in appendix B and [EPR-CPI-Schema]
  - Added CIDD (community information delta download) operation to CPI endpoint.
- DSML (CPI and HPD)
  - Search requests that lead to result code 4 (= size limit exceeded) include now results.
- MDI
  - Use “most recent” value set version instead of “active” (which is identical to the IHE terminology)

Version 0.19.x

- HPD
  - Attribute “o” of the object class “Organization” is mandatory (again).
  - Attribute “givenName” of the object class “inetOrgPerson” is optional (→ [RFC2798]).
  - Removed “name” and “distinguishedName” operational attributes from the allowed attributes of the “top” object class.
    This means, both attributes are not returned in HPD queries anymore.
  - Attribute “cn” of object class “Person” is validated.
- DSML
  - The filter element in a search request is no longer mandatory.
  - LDAP controls for paged results cannot be combined with server side sorting.

Version 0.20.x

- General
  - Naming cleanup on behalf of EPR AG TSI (EPD → EPR)
  - Added explicit XML schema validation of SOAP requests (HPD, CPI, MDI). Specified soap fault for schema violations.
- DSML
  - The filter element in a search request is mandatory again (according to DSML standard).
  -
- HPD
  - DisplayName can be delivered for HcSpecialisation attribute (metadata reference). Codes are validated for uniqueness in multi-value attributes.

Version 1.21.x

- General
  - First stable and feature complete release
- CPI
  - Community shcLanguage values are all lower-case.
  - Community shcType is an enumerated string value (see CPI profile).
  - Community shcType enumerated value “RootCommunity” renamed to “ReferenceCommunity”.
  - Each different configuration element got its distinct LDAP object class. To ease future updates to the schema without breaking changes.
  - XCPD responding gateway element no longer have a retrieve URL (attribute shcGwRetUrl)
  - Authorization Decision Consumer no longer have a URL (attribute shcAuthDecUrl)

Version 1.21.x

- HPD
  - The value set id used for validating the attribute HCProfessional.HcProfession was updated:
    from “2.16.756.5.30.1.127.3.10.1” to “2.16.756.5.30.1.127.3.10.1.1.3”.