

# EPP Handbuch

**Version 2.3.3**

23. Januar 2023

SWITCH

## Inhalt

1	Management Summary .....	4
2	Einleitung .....	5
2.1	EPP-Standards und rechtliche Grundlagen .....	5
2.2	Nutzungsbedingungen .....	6
3	Benutzung der EPP-Schnittstelle .....	7
3.1	Möglichkeiten der EPP-Schnittstelle .....	7
3.2	SWITCH-spezifische, generelle Punkte .....	8
3.3	Abbruch der Session .....	9
4	EPP-Commands .....	10
4.1	Protocol Commands .....	10
4.1.1	login .....	10
4.1.2	logout .....	12
4.1.3	hello .....	13
4.1.4	poll .....	13
4.1.5	Change-Poll Extension .....	18
4.1.6	Unhandled Namespace Extension .....	19
4.1.7	Abfrage Kontostand .....	19
4.2	Domain Commands .....	21
4.2.1	Allgemeines .....	21
4.2.2	DNSSEC .....	22
4.2.3	RGP .....	23
4.2.4	<i>domain:check</i> .....	23
4.2.5	<i>domain:info</i> .....	25
4.2.6	<i>domain:create</i> .....	28
4.2.7	<i>domain:delete</i> .....	29
4.2.8	<i>domain:transfer</i> .....	30
4.2.9	<i>domain:update</i> .....	33
4.3	Contact Commands .....	36
4.3.1	Allgemeines .....	36
4.3.2	<i>contact:check</i> .....	37
4.3.3	<i>contact:info</i> .....	38
4.3.4	<i>contact:create</i> .....	40
4.3.5	<i>contact:delete</i> .....	42
4.3.6	<i>contact:transfer</i> .....	42
4.3.7	<i>contact:update</i> .....	43
4.4	Host Commands .....	45

4.4.1	Allgemeines .....	45
4.4.2	<i>host:check</i> .....	46
4.4.3	<i>host:info</i> .....	47
4.4.4	<i>host:create</i> .....	48
4.4.5	<i>host:transfer</i> .....	48
4.4.6	<i>host:delete</i> .....	48
4.4.7	<i>host:update</i> .....	50
5	Testing.....	51
5.1	Test System.....	51
5.1.1	Voraussetzungen für das Test System.....	51
5.2	Testparcours.....	52
5.2.1	Voraussetzungen für den Testparcours.....	52
5.2.2	Ablauf Testparcours.....	53
6	Weitergehende Informationen .....	60
6.1	Web-Interface für EPP-Benutzer .....	60
	Appendix A: Abkürzungen / Begriffe .....	61

## 1 Management Summary

Das EPP-Protokoll (Extensible Provisioning Protocol) entspricht einem internationalen Standard für die Verwaltung von Domain-Namen. Mittels einer gesicherten, dauerhaften Verbindung können die notwendigen Informationen zwischen Registraren und der Registerbetreiberin ausgetauscht werden. Mit EPP erhalten Registrare die Möglichkeit, die Verwaltung von Domain-Namen in ihr bestehendes System zu integrieren.

Dieses Dokument enthält alle notwendigen Informationen für die Anbindung des EPP-Clients beim Registrar mit dem EPP-Server der Registry SWITCH. EPP wurde weitgehend nach internationalem Standard implementiert. SWITCH-spezifische Abweichungen vom EPP-Protokoll sind speziell gekennzeichnet: ➤

## 2 Einleitung

SWITCH ist die zuständige Stelle für die Registrierung und Verwaltung der country-code-Top-Level Domain-Namen (ccTLDs) der Schweiz (.ch) und Liechtensteins (.li). Eine zentrale Aufgabe ist die Registrierung von Second-Level Domain-Namen unter diesen ccTLDs, im nachfolgenden nur noch Domain-Namen genannt.

### 2.1 EPP-Standards und rechtliche Grundlagen

Dieses Handbuch basiert auf den nachfolgenden Dokumenten. Ausführliche Beispiele zu EPP-Commands sind in den untenstehenden RFC's aufgeführt:

Referenz	Dokument
[01]	RFC 5730: Extensible Provisioning Protocol (EPP) ( <a href="http://www.ietf.org/rfc/rfc5730.html">http://www.ietf.org/rfc/rfc5730.html</a> )
[02]	RFC 5731: Extensible Provisioning Protocol (EPP) Domain Name Mapping ( <a href="http://www.ietf.org/rfc/rfc5731.html">http://www.ietf.org/rfc/rfc5731.html</a> )
[03]	RFC 5732: Extensible Provisioning Protocol (EPP) Host Mapping ( <a href="http://www.ietf.org/rfc/rfc5732.html">http://www.ietf.org/rfc/rfc5732.html</a> )
[04]	RFC 5733: Extensible Provisioning Protocol (EPP) Contact Mapping ( <a href="http://www.ietf.org/rfc/rfc5733.html">http://www.ietf.org/rfc/rfc5733.html</a> )
[05]	RFC 5734: Extensible Provisioning Protocol (EPP) Transport Over TCP ( <a href="http://www.ietf.org/rfc/rfc5734.html">http://www.ietf.org/rfc/rfc5734.html</a> )
[06]	SWITCH Registrarvertrag inkl. Anhänge
[07]	Allgemeine Geschäftsbedingungen (AGB) für die Registrierung und Verwaltung von Domain-Namen unter „.ch“ und „.li“, Die jeweils gültige Version ist unter <a href="http://www.nic.ch">www.nic.ch</a> ersichtlich.
[08]	RFC 5910: Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP) ( <a href="http://www.ietf.org/rfc/rfc5910.html">http://www.ietf.org/rfc/rfc5910.html</a> )
[09]	RFC 3915: EPP Grace Period Mapping - Redemption Grace Period (RGP) ( <a href="http://www.ietf.org/rfc/rfc3915.html">http://www.ietf.org/rfc/rfc3915.html</a> )
[10]	RFC 8590: Change Poll Extension for EPP ( <a href="https://tools.ietf.org/html/rfc8590.html">https://tools.ietf.org/html/rfc8590.html</a> )
[11]	RFC 9038 Extensible Provisioning Protocol (EPP) Unhandled Namespaces ( <a href="https://www.ietf.org/rfc/rfc9038.html">https://www.ietf.org/rfc/rfc9038.html</a> )



- *Dieses Symbol kennzeichnet wichtige technische, SWITCH-spezifische Abweichungen vom internationalen EPP-Standard.*

Der Registrarvertrag und seine Anhänge haben in jedem Fall Vorrang gegenüber diesem Benutzerhandbuch.

## 2.2 Nutzungsbedingungen

Für den operativen Betrieb der EPP-Schnittstelle müssen folgende Punkte erfüllt sein:

- Unterzeichneter und gültiger Registrarvertrag
- Die von SWITCH freigeschaltete IP-Adresse Ihres EPP-Clients
- Testlauf erfolgreich bestanden (siehe Ziffer 5.3)

### 3 Benutzung der EPP-Schnittstelle

Für die Benutzung der EPP-Schnittstelle ist ein EPP-Client mit TLS-Verbindung und einer aktuellen und unterstützten Version von TLS erforderlich. **Gültig sind nur Versionen ab TLS 1.2.** Um eine TLS-Verbindung aufbauen zu können, muss die IP-Adresse des EPP-Clients bereits von SWITCH freigeschaltet sein. Sobald die TLS-Verbindung zustande gekommen ist, sind die EPP-Commands `login` und `hello` zugelassen. Alle anderen EPP-Commands sind erst nach erfolgreichem Login möglich. SWITCH bietet keinen Standard-EPP-Client und keine EPP-Bibliothek an. Es kann jeder EPP-Client verwendet werden, der die referenzierten Standards gem. Ziff. 2.1 unterstützt.

Zusätzlich zu den Standard-RFC muss ein EPP-Client entsprechend den Bestimmungen dieses Handbuchs und des Registrarvertrags (inkl. seiner Anhänge) konfiguriert werden.

Der EPP-Server unterstützt den Unicode-Zeichensatz in der UTF-8 Codierung (siehe Ziff. 3.2). Die Kommunikation vom EPP-Client zum EPP-Server erfolgt durch Commands im XML-Format. Der Server liefert unmittelbar eine dazugehörige Antwort zurück. Mehrere Commands hintereinander (Pipelining) werden einzeln und jeweils erst nach Antwort des vorangegangenen Commands verarbeitet.

#### 3.1 Möglichkeiten der EPP-Schnittstelle

Die EPP-Schnittstelle kennt zwei Hauptkategorien: Objekte und Commands (Aktionen). Objekte sind:

- domain
- contact
- host

Bei den Commands wird zwischen Protokoll Commands und objektspezifischen Commands unterschieden.

Die folgende Tabelle zeigt eine Übersicht der Protokoll Commands:

Command	Beschreibung
hello	Kontaktaufnahme
login	Anmeldung beim EPP-Server
logout	Abmeldung beim EPP-Server
poll	Abfragen von Mitteilungen aus der Pollqueue auf dem EPP-Server - der EPP-Benutzer erhält auf diesem Wege Benachrichtigungen von SWITCH.

Die folgende Tabelle zeigt eine Übersicht der objektspezifischen Commands:

Objektspezifische Commands		domain	contact	host
check	überprüft die Existenz eines	✓	✓	✓
info	liefert Informationen eines	✓	✓	✓
create	erstellt ein neues Objekt	✓	✓	✓
delete	löscht ein Objekt	✓	✓	✓
renew	erneuert ein Objekt	✗	-	-
transfer	transferiert ein Objekt	✓	✗	(implizit)
update	ändert die Angaben eines Objektes	✓	✓	✓

✓ umgesetzt  
 ✗ nicht umgesetzt  
 (implizit) dieser Command ist in EPP nicht vorgesehen.

## 3.2 SWITCH-spezifische, generelle Punkte

Gemäss Referenz (01) sind alle EPP-Commands „atomar“ und werden entweder komplett verarbeitet oder abgelehnt. Bei doppelter Übermittlung eines EPP-Commands wird auch der Zweite nochmals verarbeitet. Wenn ein Domain-Name bei der ersten Übermittlung bereits registriert wurde, liefert die zweite Übermittlung desselben Domain-Namens eine Fehlermeldung.

Bei einer Abfrage der Informationen eines Objektes (contact, domain und host) werden die von der Verordnung über Internet-Domains (VID, Art. 46) zur Veröffentlichung freigegebenen Daten zurückgegeben, ausser der EPP-Benutzer ist gemäss Registrarvertrag berechtigt, weitere Informationen einzusehen. Der Registrar sieht zum Beispiel das Abonnementsende (expiry date) bei den von ihm verwalteten Domain-Namen.

Generell werden nicht erforderliche und nicht gebrauchte Informationen, welche der EPP-Benutzer übermittelt, ignoriert.



- *Es werden keine durch den EPP-Client zu vergebende Status auf Objekten unterstützt wie z.B. <clientDeleteProhibited>, <clientTransferProhibited> etc.*
- *Es sind pro Registrar maximal drei Sessions gleichzeitig zum EPP-Server zugelassen.*
- *Generell werden folgende UTF-8 Zeichen unterstützt:*
  - *Basic Latin* *U+0020 - U+007E*
  - *Latin-1 Supplement* *U+00AE - U+00FF U+00A1 - U+00AC*
  - *Latin Extended-A* *U+0100 - U+017F*
  - *€-Euro-Zeichen* *U+20AC*
- *Die Datumsangaben erfolgen mit lokaler Zeit, auf Sekunden genau und mit der Differenz zu UTC. Im Beispiel Sommerzeit ist UTC also 2007-09-18Z12:32:00.*

Format:	yyyy-mm-ddThh:mi:ss+hh:mi (T
anstelle Z)	
Sommerzeit:	2007-09-18T14:32:00+02:00
Winterzeit:	2007-12-18T13:32:00+01:00

### 3.3 Abbruch der Session

Eine EPP-Session kann serverseitig aus folgenden Gründen abgebrochen werden:

- Die EPP-Session war länger als 3 Stunden inaktiv („session-timeout“)
- Die maximale Anzahl fehlgeschlagener EPP-Commands ist erreicht
- Ordentliche oder ausserordentliche Wartungsarbeiten am EPP-Server

## 4 EPP-Commands

Jeder Registrar hat eine maximale Anzahl EPP-Commands pro Minute zur Verfügung. Die Quota wird bestimmt von der Anzahl der verwalteten Domain-Namen. Abhängig von der Belastung des Registrierungssystems kann SWITCH diese Quota dynamisch erhöhen oder reduzieren. Ist die Quota erreicht, so verzögert der Server die Annahme der weiteren EPP-Requests.

Anzahl Domain-Namen	Commands pro Minute
< 1'000	50
bis 100'000	100
bis 200'000	150
bis 300'000	200
bis 400'000	250
bis 500'000	300
bis 600'000	350

### 4.1 Protocol Commands

Die Protocol-Commands werden zusätzlich zu den folgenden Angaben in Referenz (1) detailliert beschrieben.

#### 4.1.1 login

Eine EPP-Session wird aufgebaut durch

- a) eine TLS-Verbindung zum EPP-Server, und
- b) ein anschliessend erfolgreiches einloggen mit dem EPP `login` Command. Der Username wird durch Switch vergeben. Zulässige Zeichen siehe 3.2 mit Ausnahme Leerzeichen.

Der EPP-Server quittiert einen erfolgreichen Aufbau der TLS-Verbindung mit einem Greeting. Nach dem Verbindungsaufbau muss sich der Client noch mit dem `login` Command anmelden. Der EPP-Server antwortet darauf mit einer EPP-Response. Der Code 1000 bedeutet, dass die EPP-Session erfolgreich aufgebaut worden ist.

Nach erfolgreichem `login` steht in der EPP-Response, wie viele Nachrichten in der Pollqueue vorhanden sind, und worum es sich bei der ersten Nachricht handelt. Solange der EPP-Client kein `login` gesendet hat, wird nur `hello` unterstützt.

Mit `login` kann gleichzeitig das EPP-Login-Passwort geändert werden

```
<pw>old-password</pw>
<newPW>new-password</newPW>
```

Die Länge des EPP-Login-Passwort muss zwischen 10 - 16 Zeichen betragen und aus Klein- und Grossbuchstaben sowie mindestens einer Ziffer und einem Sonderzeichen<sup>1</sup> bestehen. Es gelten insbesondere die Bestimmungen gemäss AGB betreffend sorgfältiger Auswahl, Aufbewahrung und Geheimhaltung des Passwortes.

### Beispiel login Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <login>
      <clID>YOUR-CLID</clID>
      <pw>ABCDEF</pw>
      <options>
        <version>1.0</version>
        <lang>en</lang>
      </options>
      <svcs>
        <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>
        <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>
        <objURI>urn:ietf:params:xml:ns:host-1.0</objURI>
      </svcs>
    </login>
    <clTRID>ABC.1</clTRID>
  </command>
</epp>
```

### Unterstützte Extensions

```
<svcExtension>
  <extURI>urn:ietf:params:xml:ns:secDNS-1.1</extURI>
  <extURI>urn:ietf:params:xml:ns:rgp-1.0</extURI>
  <extURI>urn:ietf:params:xml:ns:changePoll-1.0</extURI>
  <extURI>https://www.nic.ch/epp/balance-1.0</extURI>
</svcExtension>
```

### Beispiel Login Response

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <msgQ count="1" id="1139047">
      <qDate>2007-09-26T00:00:00+02:00</qDate>
      <msg>Domain transfer completed successfully</msg>
    </msgQ>
    <trID>
      <clTRID>ABC.1</clTRID>
      <svTRID>20071008.13747.27297</svTRID>
    </trID>
  </response>
</epp>
```

<sup>1</sup> Erlaubte Sonderzeichen: + - % ( ) = . \_

*TEST-REGISTRAR-A hat eine unbestätigte Meldung in der Pollqueue.*

- • *Der EPP-Server **epp.nic.ch** wird mit einer TLS-Verbindung über Port 700 erreicht.*
- *Der EPP-Server lässt nur Verbindungen von Clients zu, deren IP-Adressen bei SWITCH registriert sind.*
- *Es wird ausschliesslich secDNS-1.1 unterstützt.*
- *Registrare können die DNSSEC Extension neu auch ohne Freischaltung durch SWITCH verwenden. Durch das verwenden der DNSSEC Extension beim Login verpflichtet sich der Registrar, DNSSEC über EPP auch zu unterstützen. Eingehende Transfers von Domain Namen mit DNSSEC werden nur erlaubt, wenn die DNSSEC Extension beim Login konfiguriert wurde.*

## 4.1.2 logout

Der EPP-Client muss seine Session mit dem `logout` Command beenden.

### Beispiel Logout Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <epp:command xmlns:epp="urn:ietf:params:xml:ns:epp-1.0">
    <epp:logout />
    <epp:clTRID>ABC.27</epp:clTRID>
  </epp:command>
</epp>
```

### Beispiel Logout Response

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1500">
      <msg lang="en">Command completed successfully; ending session</msg>
    </result>
    <trID>
      <clTRID>ABC.27</clTRID>
      <svTRID>20071008.13747.27299</svTRID>
    </trID>
  </response>
</epp>
```

## 4.1.3 hello

Der EPP-Client kann mit dem `hello` Command prüfen, ob die EPP-Session noch aktiv ist. Der EPP-Server antwortet auf ein `hello` Command mit dem Greeting.

### Beispiel hello Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <hello/>
</epp>
```

### Beispiel hello Response (Greeting)

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <greeting>
    <svID>SWITCH_EPP_Server</svID>
    <svDate>2007-10-09T08:44:14+02:00</svDate>
    <svcMenu>
      <version>1.0</version>
      <lang>en</lang>
      <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>
      <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>
      <objURI>urn:ietf:params:xml:ns:host-1.0</objURI>
      <svcExtension>
        <extURI>urn:ietf:params:xml:ns:secDNS-1.1</extURI>
        <extURI>urn:ietf:params:xml:ns:rgp-1.0</extURI>
        <extURI>urn:ietf:params:xml:ns:changePoll-1.0</extURI>
        <extURI>https://www.nic.ch/epp/balance-1.0</extURI>
        <extURI>urn:ietf:params:xml:ns:epp:unhandled-namespaces-1.0</extURI>
      </svcExtension>
    </svcMenu>
    <dcP>
      <access>
        <personalAndOther />
      </access>
      <statement>
        <purpose>
          <admin />
          <other />
          <prov />
        </purpose>
        <recipient>
          <ours />
          <public />
        </recipient>
        <retention>
          <legal />
        </retention>
      </statement>
    </dcP>
  </greeting>
</epp>
```

## 4.1.4 poll

Die Nachrichten, welche in der Pollqueue vorhanden sind, müssen nacheinander abgeholt und bestätigt werden. Sobald die erste Nachricht bestätigt wurde, kann die zweite abgeholt werden etc. Bestätigte Nachrichten werden aus der Pollqueue gelöscht.

→ Nicht bestätigte Nachrichten werden nach 30 Tagen aus der Pollqueue gelöscht und können nicht mehr abgeholt werden. Daher ist es wichtig, dass der Registrar die Pollqueue regelmässig abholt um nicht wichtige Informationen zu verpassen.

## Beispiel Poll Command zum Abrufen einer Meldung

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <poll op="req"/>
    <clTRID>Registrar_00_2</clTRID>
  </command>
</epp>
```

## Beispiel Poll Command zur Bestätigung einer Meldung

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <poll op="ack" msgID="1139047"/>
    <clTRID>Registrar_00_3</clTRID>
  </command>
</epp>
```

## Beispiel Poll Response zur Bestätigung einer Meldung

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <msgQ count="0" id="1139047"/>
    <trID>
      <clTRID>Registrar_00_3</clTRID>
      <svTRID>20071008.13689.27043</svTRID>
    </trID>
  </response>
</epp>
```

#### 4.1.4.1 Domain Transfer Completed Message

Diese Meldung wird dem bisherigen Registrar nach dem Abschluss eines Transfers in die Pollqueue gestellt.

Attribut	Bemerkung
domain:name	Domain-Name
domain:trStatus	“serverApproved”
domain:reID	contact:id des Antragstellers
domain:reDate	Datum des Antrags
domain:acID	”NULL”
domain:acDate	Datum der Ausführung
domain:exDate	Wird weggelassen.

#### Beispiel Poll Response nach erfolgreichem Transfer

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="1139047">
      <qDate>2007-09-26T00:00:00+02:00</qDate>
      <msg> Domain transfer completed successfully</msg>
    </msgQ>
    <epp:resData xmlns:epp="urn:ietf:params:xml:ns:epp-1.0">
      <domain:trnData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>yourname.ch</domain:name>
        <domain:trStatus>serverApproved</domain:trStatus>
        <domain:reID>RegistrarB</domain:reID>
        <domain:reDate>2007-09-18T22:43:00+02:00</domain:reDate>
        <domain:acID>NULL</domain:acID>
        <domain:acDate>2007-09-18T22:43:00+02:00</domain:acDate>
      </domain:trnData>
    </epp:resData>
    <trID>
      <clTRID>Registrar_00_2</clTRID>
      <svTRID>20071008.13688.27039</svTRID>
    </trID>
  </response>
</epp>
```

#### 4.1.4.2 Domain Delete Message

Diese Meldung wird in die EPP poll queue vom Registrar gestellt, wenn eine manuelle Löschung von einem Domain Namen durch die Registry stattgefunden hat. Eine solche Löschung kann unter speziellen Umständen vorkommen z.B. bei Domain Namen Missbrauch.

Attribut	Kommentar
domain:name	Domain Name
domain:upDate	Ausführungs Datum
domain:clID	Client ID vom Registrar des Domain-Namens

#### Beispiel einer Poll Response nach Löschung durch die Registry

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="46535949">
      <qDate>2019-01-28T16:14:47+01:00</qDate>
    </msgQ>
    <resData>
      <domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>delete-mell.ch</domain:name>
        <domain:roid>D6006352-SWITCH</domain:roid>
        <domain:status s="serverHold" lang="en" />
        <domain:status s="serverRenewProhibited" lang="en" />
        <domain:status s="serverTransferProhibited" lang="en" />
        <domain:status s="inactive" lang="en" />
        <domain:registrant>CH-MYTECH</domain:registrant>
        <domain:clID>Test-Registrar-X</domain:clID>
        <domain:upDate>2019-01-28T16:14:47+01:00</domain:upDate>
      </domain:infData>
    </resData>
    <extension>
      <changePoll:changeData xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0">
        <changePoll:operation>delete</changePoll:operation>
        <changePoll:date>2019-01-28T16:14:47+01:00</changePoll:date>
        <changePoll:svTRID>20190128.34733373</changePoll:svTRID>
        <changePoll:who>SWITCH manual delete</changePoll:who>
        <changePoll:reason>domain name abuse</changePoll:reason>
      </changePoll:changeData>
    </extension>
  </trID>
  <clTRID>ABC-12345</clTRID>
  <svTRID>20190128.75290441.758467418</svTRID>
</trID>
</response>
</epp>
```

#### 4.1.4.3 „Automated DNSSEC Provisioning“ Message

Diese Meldung wird in die EPP poll queue von einem Registrar gestellt, wenn eine Änderung an den DS Records von einem Domain Namen durch den “Automated DNSSEC Provisioning” Prozess stattgefunden hat.

Sie finden weitere Informationen über die verschiedenen Varianten, wie diese Meldung ausgeliefert wird, und wie man sie für Test-Zwecke auf unserer Test-Umgebung auslösen kann im Dokument „Guidelines for CDS processing at SWITCH“ (nur auf Englisch verfügbar):

[https://www.nic.ch/export/shared/.content/files/SWITCH\\_CDS\\_Manual\\_en.pdf](https://www.nic.ch/export/shared/.content/files/SWITCH_CDS_Manual_en.pdf)  
und auf der “CDS Status Check” Webseite:  
<https://www.nic.ch/de/faqs/dnssec/cds/>

Attribut	Kommentar
changePoll:reason	Hinweis auf Einrichtung, Änderungen oder Löschung der DS records
changePoll:who	SWITCH CDS: Hinweis auf «Automated DNSSEC provisioning» Prozess Änderung
changePoll:date	Ausführungs-Datum

#### Beispiel einer Poll Meldung nach Änderung durch den «Automated DNSSEC provisioning» Prozess

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="120030094">
      <qDate>2018-06-19T00:00:00+02:00</qDate>
    </msgQ>
    <epp:resData xmlns:epp="urn:ietf:params:xml:ns:epp-1.0">
      <domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>rollover-aa48.ch</domain:name>
        <domain:roid>D5920069-SWITCH</domain:roid>
        <domain:status s="ok" />
        <domain:registrant>123536</domain:registrant>
        <domain:clID>cds message</domain:clID>
        <domain:update>2018-06-21T11:49:54+02:00</domain:update>
      </domain:infData>
    </epp:resData>
    <extension>
      <changePoll:changeData xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0">
        <changePoll:operation>update</changePoll:operation>
        <changePoll:date>2018-06-21T11:49:54+02:00</changePoll:date>
        <changePoll:svTRID>20180621.34663867</changePoll:svTRID>
        <changePoll:who>SWITCH CDS: see https://www.nic.ch/de/faqs/dnssec/cds/</changePoll:who>
        <changePoll:reason>Rollover of DNSSEC Digest</changePoll:reason>
      </changePoll:changeData>
      <secDNS:infData xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
        <secDNS:dsData>
          <secDNS:keyTag>20011</secDNS:keyTag>
          <secDNS:alg>8</secDNS:alg>
          <secDNS:digestType>2</secDNS:digestType>
          <secDNS:digest>AAAA9AB3E7D203FFF7923B8773599E248717F1DC79A9BEF09D8981B13AB7A049E</secDNS:digest>
        </secDNS:dsData>
        <secDNS:dsData>
          <secDNS:keyTag>12123</secDNS:keyTag>
        </secDNS:dsData>
      </secDNS:infData>
    </extension>
  </response>
</epp>
```

```

        <secDNS:alg>7</secDNS:alg>
        <secDNS:digestType>4</secDNS:digestType>
<secDNS:digest>AAAA54840FBBB6F4270F8B6D8C06C6A2B3152E55D2E9F81132130E507829B6D24FA56A4E074B4692DDC46F512B0
48AAC</secDNS:digest>
        </secDNS:dsData>
    </secDNS:infData>
</extension>
<trID>
    <clTRID>EA03_RY_10_2_C</clTRID>
    <svTRID>$svTRID$</svTRID>
</trID>
</response>
</epp>

```

## 4.1.5 Change-Poll Extension

Bei einigen durch SWITCH vorgenommenen Änderungen an Domain Namen senden wir eine EPP Poll Meldungen zur Information an den Registrar.

Diese beinhalten den aktualisierten Domain Status (<epp:resData><domain:infData><domain:status>) sowie die ChangePoll Extension.

Die Change-Poll Extension ist eine Erweiterung der EPP Poll Meldungen um ergänzende Informationen über Änderungen an EPP Objekten zu liefern. (was, wann, wer und warum)

Die betroffenen Anwendungsfälle sind:

- Löschungen (Manuell oder durch unseren Malware-Prozess)
- Aktivierung / Deaktivierung von Domain Namen (Manuell oder durch unseren Malware-Prozess)
- Änderungen an den Name-Servern
- Neu setzen vom Transfercode
- CDS siehe Kapitel 4.1.4.3 Automated DNSSEC Provisioning

Die Implementation der ChangePoll beruht auf <https://tools.ietf.org/html/rfc8590>

→ Für eine einfachere automatisierte Verarbeitung empfehlen wir die Change-Poll Extension beim EPP Login zu deklarieren.

Andernfalls wird das XML Element mit dem Namespace "urn:ietf:params:xml:ns:changePoll-1.0" gemäss RFC 9038 Unhandled Namespaces im <extValue> eingebaut. Siehe auch Kapitel Unhandled Namespaces Extension.

### 4.1.6 Unhandled Namespace Extension

EPP, wie in RFC 5730 definiert, umfasst ein Verfahren um die während einer Sitzung gültigen Objekte (Domain, Host, Contact) und Dienst-Erweiterungen (RGP, DNSSEC, etc.) zu bestimmen. Die Objekte und Erweiterungen werden mit Namensraum-URI's identifiziert.

Wenn ein Objekt oder eine Dienst-Erweiterung vom Server unterstützt wird (im Server-Greeting aufgeführt) aber vom Client noch nicht (nicht im Login Command aufgeführt), dann ist sein Namensraum ein "unhandled namespace". RFC 9038 Unhandled Namespaces definiert eine Praxis für den Server um Daten von "unhandled namespaces" konform zu RFC 5730 an den Client zu senden.

Somit wird der Client über die Existenz solcher Namensräume informiert und sie können eventuell zu einem späteren Zeitpunkt noch automatisiert verarbeitet werden.

Ein Server der diese Praxis implementiert hat, führt `<extURI>urn:ietf:params:xml:ns:epp:unhandled-namespaces-1.0</extURI>` im Server Greeting auf.

Ein Client der `<extURI>urn:ietf:params:xml:ns:epp:unhandled-namespaces-1.0</extURI>` im EPP Login aufführt, signalisiert damit, dass er die Server Antworten auf Unhandled Namespaces untersucht.

### 4.1.7 Abfrage Kontostand

Mit untenstehendem Command kann der aktuelle Kontostand des Registrars ermittelt werden. Da es sich um eine Extension handelt, muss dies beim Login angegeben werden, damit die Response regulär erstellt werden kann. Falls man die Extension beim Login nicht spezifiziert, erhält man folgende Response:

```
<result code="2307">
  <msg lang="en">Unimplemented object service</msg>
</result>
```

Das Schema für Command und Response ist unter folgendem Link verfügbar:  
<https://www.nic.ch/epp/balance-1.0>

#### Beispiel Command Abfrage des Kontostandes

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <info>
      <balance:info
        xmlns:balance="https://www.nic.ch/epp/balance-1.0"/>
    </info>
    <clTRID>ABC</clTRID>
  </command>
</epp>
```

## Beispiel Response mit Angabe des Kontostandes

```
<epp xmlns="urn:iETF:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <infData xmlns="https://www.nic.ch/epp/balance-1.0">
        <balance>1234.50</balance>
        <currency>CHF</currency>
      </infData>
    </resData>
    <trID>
      <clTRID>ABC</clTRID>
      <svTRID></svTRID>
    </trID>
  </response>
</epp>
```

## 4.2 Domain Commands

### 4.2.1 Allgemeines

Die `domain` Commands werden zusätzlich zu den folgenden Angaben in Referenz (2) detailliert beschrieben.

- Der Halter eines Domain-Namens wird im Feld `domain:registrant` gespeichert.
- Der Domain-Namen-Transfercode darf nur dem Halter des Domain-Namens ausgehändigt werden.



- *In den EPP-Spezifikationen wird ein `renew` Command erwähnt. Dieser ist nicht implementiert, da ein serverseitiges `renew` durchgeführt wird. Ein `renew` wird mit dem Fehlercode 2101 („Unimplemented command“) abgelehnt.*
- *Einem Domain-Namen kann optional maximal ein Kontakt `type="tech"` zugeordnet werden.*

Domain Attribut	mandatory	Update möglich	Bemerkung
<code>name</code>	Yes	No	
<code>roid</code>	Yes	No	Dnnnnnnnn-SWITCH, supplied by Registry
<code>status</code>	No	No	
<code>registrant</code>	Yes	Yes	<code>contact:id</code>
<code>contact type=tech</code>	No	Yes	<code>contact:id</code>
<code>hostObj</code>	No	Yes	
<code>clID</code>	Yes	No	Von Registry gesetzt
<code>crID</code>			Wird weggelassen.
<code>crDate</code>		No	Von Registry gesetzt
<code>exDate</code>		No	Von Registry gesetzt
<code>upID</code>			Wird weggelassen.
<code>upDate</code>		No	Wird weggelassen.
<code>trDate</code>	No	No	Wird weggelassen.
<code>authinfo</code>	No	Yes	Leerzeichen “;” und “,” sind nicht erlaubt.

## 4.2.2 DNSSEC

DNSSEC kann optional verwendet werden und dient zur Signierung von Domain-Namen.

DNSSEC Attribut	mandatory	Bemerkung
alg	Yes	Algorithmus 8, 10, 13, 14, 15, 16
digestType	Yes	2 oder 4
digest	Yes	Max. 96 Charakter
flags	No	256 oder 257
protocol	No	3
keyTag	Yes	0-65535 Wertebereich
pubKey (DNSKEY)	No	Max. 4096 bits



- Der Registrar muss für die Nutzung von DNSSEC nicht mehr separat freigeschaltet werden. *Beim Login sollte DNSSEC (urn:ietf:params:xml:ns:secDNS-1.1) nur deklariert werden, wenn der Registrar DNSSEC über EPP unterstützt..*
- SWITCH macht keine Delegationschecks und überprüft nicht, ob signierte Domain-Namen erreichbar sind.
- *keyData* Einträge sind optional und werden in der Datenbank gespeichert, falls mitgesendet.
- Wenn *keyData* verwendet wird, sind die Attribute *flags*, *protocol*, *alg*, *pubKey string size* obligatorisch.
- Nur *secDNS1.1 DS-Data Interface* wird unterstützt (Referenz 8)
- Maximal 20 DNSSEC-Einträge pro Domain-Name
- Die Attribute `<secDNS:update urgent="true">` und `<secDNS:maxSigLife>` gemäss *rfc5910* werden nicht unterstützt. Requests die diese Attribute beinhalten werden mit dem ResultCode "2102 Unimplemented option" beantwortet.
- Bitte beachten Sie, dass die Algorithmen 5 (RSASHA1), 7 (RSASHA1-NSEC3-SHA1) und der DigestTyp 1 (SHA1) von SWITCH nicht mehr unterstützt werden.

### 4.2.3 RGP

RGP kann optional verwendet werden und dient zur Wiederherstellung von gelöschten Domain-Namen.



- *Beim Login muss RGP deklariert werden: urn:ietf:params:xml:ns:rgp-1.0*
- *Wenn der Domain-Name dem abfragenden Registrar gehört, dann kann mit dem Command `domain:info` abgefragt werden, ob der Domain-Name in der redemption Period ist*
- *Es gibt keinen pendingRestore RGP-Status. Wenn der restore Command erfolgreich war, dann geht der EPP-Status des Domain-Namens auf „ok“.*
- *Das optionale `rgp:report` Element im `rgp:restore` Element wird ignoriert.*
- *Ein separater `rgp:report` Command führt zu einem Fehler, da es kein pending Restore gibt.*

### 4.2.4 `domain:check`

Mit `domain:check` können Domain-Namen auf ihre Verfügbarkeit überprüft werden. Der Server antwortet mit registriert / nicht registriert. Massenabfragen sind gemäss Referenz (06) nur beschränkt zugelassen.



- *Es können maximal 10 Domain-Namen pro Command gleichzeitig abgefragt werden, ansonsten antwortet der Server mit dem Fehlercode 2308 („Data management policy violation“)*

#### Beispiel `domain check` Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <check>
      <domain:check xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>yourname.ch</domain:name>
        <domain:name>yourname.li</domain:name>
        <domain:name>studen-be.ch</domain:name>
        <domain:name>fuerstentum.li</domain:name>
      </domain:check>
    </check>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

### Beispiel domain check Response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:chkData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:cd>
          <domain:name avail="0">yourname.ch</domain:name>
          <domain:reason>In use</domain:reason>
        </domain:cd>
        <domain:cd>
          <domain:name avail="1">yourname.li</domain:name>
        </domain:cd>
        <domain:cd>
          <domain:name avail="0">studen-be.ch</domain:name>
          <domain:reason>City-domainname</domain:reason>
        </domain:cd>
        <domain:cd>
          <domain:name avail="0">fuerstentum.li</domain:name>
          <domain:reason>Legal reasons</domain:reason>
        </domain:cd>
      </domain:chkData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>20071009.14301.29655</svTRID>
    </trID>
  </response>
</epp>
```

## 4.2.5 domain:info

Mit `domain:info` werden die Daten des Domain-Namens ausgegeben (siehe Ziff. 3.2). Zusätzlich zu den Standardangaben wird im Feld `<domain:clID>` die ID des „sponsoring Client“, also des aktuellen Registrars angezeigt.

Die Felder `<domain:registrant>`, `<domain:exDate>` und `<domain:contact>` (falls vorhanden) werden nur ausgegeben, wenn

- der Domain-Name vom abfragenden Registrar verwaltet wird oder
- ein gültiger Transfercode mitgeliefert wird, gemäss folgendem Beispiel.

### Beispiel domain info Command mit Transfercode

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <command>
    <info>
      <domain:info
        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>fremder-domain.ch</domain:name>
        <domain:authInfo>
          <domain:pw>authinfo_pwd</domain:pw>
        </domain:authInfo>
        </domain:info>
      </info>
      <clTRID>rq0dr</clTRID>
    </command>
  </epp>
```

### Beispiel domain info Response

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>fremder-domain.ch</domain:name>
        <domain:roid>D7033341-SWITCH</domain:roid>
        <domain:status s="inactive" lang="en" />
        <domain:registrant>CH-HOLDER-ID</domain:registrant>
        <domain:contact type="tech">CH-TECH-ID</domain:contact>
        <domain:clID>Test-Registrar-X</domain:clID>
        <domain:exDate>2021-11-09T23:59:59+01:00</domain:exDate>
      </domain:infData>
    </resData>
    <trID>
      <clTRID>ABC</clTRID>
      <svTRID>20201110.118733861.1213851368</svTRID>
    </trID>
  </response>
</epp>
```

Bei einem gültiger Transfercode, erhält man als Response: `<result code="1000">`

Falsche Angaben im optionalen `<domain:authInfo>` Element führen immer zum Return Code: `<2202 Invalid authorization information>`

Ein im `<domain:pw>` mitgeliefertes roid Attribut wird immer ignoriert. Der Wert von `<domain:pw>` muss sich immer auf den abgefragten Domain-Namen beziehen und darf nicht leer sein.

## Beispiel domain info Command eigenverwaltete Domain-Namen

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <info>
      <domain:info xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>yourname.ch</domain:name>
      </domain:info>
    </info>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

## Beispiel domain info Response

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>yourname.ch</domain:name>
        <domain:roid>D2586061-SWITCH</domain:roid>
        <domain:status s="ok"/>
        <domain:registrant>5527981</domain:registrant>
        <domain:contact type="tech">5527981</domain:contact>
        <domain:ns>
          <domain:hostObj>ns1.yourname.ch</domain:hostObj>
          <domain:hostObj>ns2.yourname.ch</domain:hostObj>
        </domain:ns>
        <domain:clID>3703709</domain:clID>
        <domain:exDate>2009-01-31T00:00:00+01:00</domain:exDate>
      </domain:infData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>20080529.27664.913190</svTRID>
    </trID>
  </response>
</epp>
```

## Bei DNSSEC signierten Domain-Namen

```
<extension>
  <secDNS:infData
    xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
    <dsData xmlns="urn:ietf:params:xml:ns:secDNS-1.1">
      <keyTag>20530</keyTag>
      <alg>13</alg>
      <digestType>2</digestType>
      <digest>CAFFEEBABE00D87A0147EFE9877
        AB7335206ABFF6FA730BD6239D65CBAC7E768</digest>
      <keyData>
        <flags>257</flags>
        <protocol>3</protocol>
        <alg>13</alg>
        <pubKey>Lltqdb8EDyeQ/8RyJL1faHGKBw
          LPTYfZLV8vRN3opa4uY+nyhPghl
          EEBqDLdCo3ZpywJf1W39vwl03PZJvSzg==</pubKey>
      </keyData>
    </dsData>
  </secDNS:infData>
</extension>
```

*Abfrage eines Domain-Namens, der vom abfragenden Registrar verwaltet wird.*

### Example of a domain info response with "redemptionPeriod" status

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>switch02.ch</domain:name>
        <domain:roid>D5874403-SWITCH</domain:roid>
        <domain:status s="serverHold" lang="en" />
        <domain:status s="serverRenewProhibited" lang="en" />
        <domain:status s="serverTransferProhibited" lang="en" />
        <domain:registrant>SWITCH01</domain:registrant>
        <domain:ns>
          <domain:hostObj>ns1.switch02.ch</domain:hostObj>
          <domain:hostObj>ns2.switch02.ch</domain:hostObj>
        </domain:ns>
        <domain:host>ns1.switch02.ch</domain:host>
        <domain:host>ns2.switch02.ch</domain:host>
        <domain:clID>Test-X</domain:clID>
        <domain:upDate>2018-01-09T09:06:48+01:00</domain:upDate>
        <domain:exDate>2019-01-08T23:59:59+01:00</domain:exDate>
      </domain:infData>
    </resData>
    <extension>
      <rgp:infData xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0">
        <rgp:rgpStatus s="redemptionPeriod" lang="en" />
      </rgp:infData>
    </extension>
    <trID>
      <clTRID>DS03_RY_00_2_C</clTRID>
      <svTRID>20180109.75043211.757740352</svTRID>
    </trID>
  </response>
</epp>
```

*Abfrage eines Domain-Namens, der vom abfragenden Registrar verwaltet wird.*

## 4.2.6 domain:create

Mit `domain:create` werden Domain-Namen registriert. Vorgängig müssen alle Kontakte (Halter und technischer Kontakt) erfasst werden, falls sie noch nicht bestehen. Name-Server müssen ebenfalls vorgängig registriert werden, damit Sie dem Domain-Namen zugeordnet werden können.



- `<admin>` wird nicht unterstützt.
- `<billing>` wird nicht unterstützt (der Rechenkontakt ist immer der Registrar).

### Beispiel domain create Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <domain:create xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>yourname.ch</domain:name>
        <domain:ns>
          <domain:hostObj>ns1.yourname.ch</domain:hostObj>
          <domain:hostObj>ns2.yourname.ch</domain:hostObj>
        </domain:ns>
        <domain:registrant>HOLDERCONTACT</domain:registrant>
        <domain:contact type="tech">TECHCONTACT</domain:contact>
        <domain:authInfo>
          <domain:pw/>
        </domain:authInfo>
      </domain:create>
    </create>
    <c1TRID>ABC-12345</c1TRID>
  </command>
</epp>
```

### DNSSEC Zusatz

```
<extension>
  <secDNS:create xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
    <secDNS:dsData>
      <secDNS:keyTag>20530</secDNS:keyTag>
      <secDNS:alg>13</secDNS:alg>
      <secDNS:digestType>2</secDNS:digestType>
    <secDNS:digest>CAFFEEBABE00D87A0147EFE9877
      AB7335206ABFF6FA730BD6239D65CBAC7E768</secDNS:digest>
    <secDNS:keyData>
      <secDNS:flags>256</secDNS:flags>
      <secDNS:protocol>3</secDNS:protocol>
      <secDNS:alg>13</secDNS:alg>
      <secDNS:pubKey>Lltqdb8EDyeQ/8RyJl1faHGKBw
        LPTYfZLV8vRN3opa4uY+nyhPgHl
        EEqDLdCo3ZpywJf1W39vw103PZJvSzg==</secDNS:pubKey>
    </secDNS:keyData>
    </secDNS:dsData>
  </secDNS:create>
</extension>
```

keyData ist optional. Bei Mitsenden muss die Berechnung des pubKey zum digest führen.

### Beispiel domain create response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>yourname.ch</domain:name>
        <domain:crDate>1999-04-03T22:00:00.0Z</domain:crDate>
        <domain:exDate>2001-04-03T22:00:00.0Z</domain:exDate>
      </domain:creData>
    </resData>
    <trID>
      <c1TRID>ABC-12345</c1TRID>
      <svTRID>54321-XYZ</svTRID>
    </trID>
  </response>
</epp>
```

## 4.2.7 domain:delete

Mit `domain:delete` werden Domain-Namen per sofort gelöscht und setzten sie in die Übergangsfrist. Ein Domain-Name kann jederzeit unter Einhaltung der Regeln aus Referenz (06) durch den berechtigten Registrar gelöscht werden. Ein gelöschter Domain-Name kann erst nach der Übergangsfrist gemäss Referenz (07) wieder neu registriert werden.

Ein Domain-Name kann gelöscht werden, auch wenn noch untergeordnete Name-Server (subordinate hosts) vorhanden sind. Diese bleiben weiterhin registriert, können aber nicht mehr verwendet werden (siehe Ziff. 4.4).

### Beispiel domain delete Command

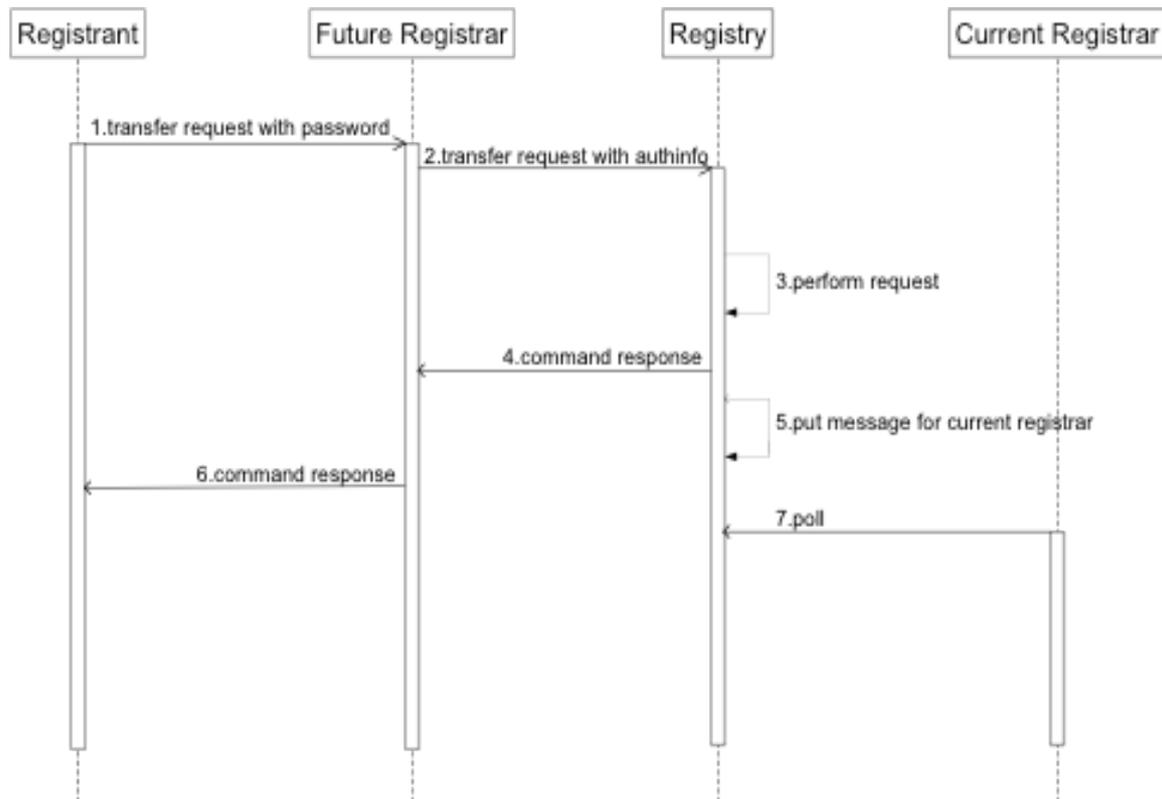
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <delete>
      <domain:delete
        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>yourname.ch</domain:name>
      </domain:delete>
    </delete>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

### Beispiel domain delete Response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>54321-XYZ</svTRID>
    </trID>
  </response>
</epp>
```

## 4.2.8 domain:transfer

Der Halter kann seinen Domain-Namen von einem Registrar seiner Wahl verwalten lassen. Mit `domain:transfer` wird der Domain-Name transferiert. Falls bei einem zu transferierenden Domain-Namen untergeordnete Name-Server vorhanden sind, werden diese ebenfalls transferiert.



Der vollständige Domain-Transfer-Prozess in sieben Schritten:

### 1. Transfer request with password

Der Halter muss bei seinem aktuellen Registrar einen Transfercode für seinen Domain-Namen. Sobald er den Transfercode erhalten hat, kann er diesen dem neuen Registrar weiterreichen.

### 2. transfer request with authinfo

Der neue Registrar sendet den `domain:transfer` Antrag mit passendem Transfercode via EPP an SWITCH.

### 3. perform request

Der EPP-Server prüft den Antrag und transferiert den Domain-Namen, falls der Transfercode korrekt ist. Allfällige, untergeordnete Name-Server werden ebenfalls transferiert (siehe Ziff. 4.4.1). Nach dem Transfer ist der Domain-Name im Domain-Namen-Bestand des neuen Registrars. Die Contact-IDs werden bei einem Transfer nicht mitgeliefert und müssen vom neuen Registrar durch eigene Contact-ID's ersetzt werden.



Mit `domain:update` muss der neue Registrar eine von ihm erstellte Contact-ID für den Halter (technischer Kontakt nicht zwingend) eintragen, bevor weitere Updates am Domain-Namen zugelassen werden. Gültige Commands sind bis dahin nur `check`, `info`, `delete` und `transfer`.

### 4. command response

Der EPP-Server sendet dem neuen Registrar eine Bestätigungsmeldung (EPP-Response).

### 5. put message for current registrar

SWITCH stellt dem bisherigen Registrar eine neue Nachricht in die Pollqueue, die den Transfer des Domain-Namens an einen neuen Registrar oder SWITCH bestätigt.

### 6. command response

Der neue Registrar kann dem Halter mitteilen, dass der Transfer vollzogen wurde. SWITCH tritt in der Regel nicht mit dem Halter in Kontakt, solange der Kunde bei einem Registrar ist, ausser in den in Referenz (06) erwähnten Fällen.

### 7. poll

Der bisherige Registrar kann die unter 5. generierte Nachricht in seiner Pollqueue abrufen.



- *Nach erfolgreichem Transfer wird der Status `<serverTransferProhibited>` gesetzt. Der Domain-Name kann während diesem Status nicht zu einem anderen Registrar transferiert werden. Dieser Status dauert 60 Tagen und wird anschliessend automatisch aufgehoben.*
- *Das Transfer Query Command wird nicht benötigt, da der Transfer bei gültigem Transfercode sofort ausgeführt wird. Auf einen Transfer Query Command antwortet der Server mit dem Fehler 2101 („Unimplemented command“).*
- *Der Transfer eines DNSSEC-signierten Domain-Namens zu einem Registrar der DNSSEC nicht unterstützt ist nicht möglich. In diesem Fall muss der bisherige Registrar DNSSEC vor dem Transfer entfernen.*

## Beispiel domain transfer Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <transfer op="request">
      <domain:transfer
        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>yourname.ch</domain:name>
        <domain:authInfo>
          <domain:pw>ABCDEF</domain:pw>
        </domain:authInfo>
        </domain:transfer>
      </transfer>
      <clTRID>ABC-12345</clTRID>
    </command>
  </epp>
```

## Beispiel domain transfer Response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:trnData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>yourname.ch</domain:name>
        <domain:trStatus>serverApproved</domain:trStatus>
        <domain:reID>TEST-REGISTRAR-A</domain:reID>
        <domain:reDate>2007-12-06T16:23:52+01:00</domain:reDate>
        <domain:acID>TEST-REGISTRAR-A</domain:acID>
        <domain:acDate>2007-12-06T16:23:52+01:00</domain:acDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>20071206.91098.282082</svTRID>
    </trID>
  </response>
</epp>
```

#### 4.2.9 *domain:update*

Mit `domain:update` werden Änderungen am Domain-Namen vorgenommen. Auch der Transfercode (`<domain:authInfo>`) lässt sich damit erstellen, der dem Halter weitergegeben werden kann, um den Domain-Namen zu transferieren.

Wenn der `domain:update` Command keine Felder ändert, wird die Fehlermeldung 2308 („Data management policy violation“) zurückgegeben.

Falls zum Zeitpunkt des `domain:update` Commands SWITCH-Contact-ID's eingetragen sind (weil zuvor ein Transfer stattgefunden hat), muss der Registrar mit `domain:update` seine eigenen Contact-IDs eintragen. Falls nur die Contact-ID für den Halter ersetzt wird, bleibt der technische Kontakt leer. Ein allfälliger technischer Kontakt wird gelöscht.



- `<admin>` *und* `<billing>` *werden ignoriert*
- *Der Transfercode* `<domain:authInfo>` *muss folgenden Regeln entsprechen:*
  - *6 – 60 Zeichen (Zeichensätze wie unter Ziff. 3.2)*
  - *Keine Leerzeichen, Komma, Strichpunkt*

## Beispiel domain update Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <update>
      <domain:update
        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>yourname.ch</domain:name>
        <domain:add>
          <domain:ns>
            <domain:hostObj>ns.yourns.ch</domain:hostObj>
          </domain:ns>
          <domain:contact
            type="tech">NEWTECHCONTACT</domain:contact>
          </domain:add>
          <domain:rem>
            <domain:contact
              type="tech">OLDTECHCONTACT</domain:contact>
            </domain:rem>
          <domain:chg>
            <domain:authInfo>
              <domain:pw>2BARfoo</domain:pw>
            </domain:authInfo>
          </domain:chg>
        </domain:update>
      </update>
      <clTRID>Test3</clTRID>
    </command>
  </epp>
```

### DNSSEC Extension

#### DNSSEC Daten hinzufügen:

```
<extension>
  <secDNS:update
    xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
    <secDNS:add>
      <secDNS:dsData>
        <secDNS:keyTag>16915</secDNS:keyTag>
        <secDNS:alg>13</secDNS:alg>
        <secDNS:digestType>2</secDNS:digestType>
        <secDNS:digest>CAFFEEBABE00D87A0147EFE9877
          AB7335206ABFF6FA730BD6239D65CBAC7E768</secDNS:digest>
      </secDNS:dsData>
    </secDNS:add>
  </secDNS:update>
</extension>
```

#### DNSSEC Daten ersetzen (alle DS-Records werden überschrieben):

```
<extension>
  <secDNS:update
    xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
    <secDNS:rem>
      <secDNS:all>true</secDNS:all>
    </secDNS:rem>
    <secDNS:add>
      <secDNS:dsData>
        <secDNS:keyTag>16915</secDNS:keyTag>
        <secDNS:alg>13</secDNS:alg>
        <secDNS:digestType>2</secDNS:digestType>
        <secDNS:digest>CAFFEEBABE00D87A0147EFE9877
          AB7335206ABFF6FA730BD6239D65CBAC7E768</secDNS:digest>
      </secDNS:dsData>
    </secDNS:add>
  </secDNS:update>
</extension>
```

#### DNSSEC Daten entfernen (Alle DS-Records die den mitgelieferten Attributen entsprechen werden gelöscht):

```
<extension>
  <secDNS:update
    xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
    <secDNS:rem>
      <secDNS:dsData>
        <secDNS:keyTag>16915</secDNS:keyTag>
        <secDNS:alg>13</secDNS:alg>
        <secDNS:digestType>2</secDNS:digestType>
        <secDNS:digest>CAFFEEBABE00D87A0147EFE9877
          AB7335206ABFF6FA730BD6239D65CBAC7E768</secDNS:digest>
      </secDNS:dsData>
    </secDNS:rem>
  </secDNS:update>
</extension>
```

## DNSSEC Extension

Alle DNSSEC Daten entfernen:

```
<extension>
  <secDNS:update xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
    <secDNS:rem>
      <secDNS:all>true</secDNS:all>
    </secDNS:rem>
  </secDNS:update>
</extension>
```

## Beispiel Domain Update Command mit einem RGP Restore Request

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <command>
    <update>
      <domain:update xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>yourname.ch</domain:name>
        <domain:chg/>
      </domain:update>
    </update>
    <extension>
      <rgp:update xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0" xsi:schemaLocation="urn:ietf:params:xml:ns:rgp-1.0
rgp-1.0.xsd">
        <rgp:restore op="request"/>
      </rgp:update>
    </extension>
  </command>
</epp>
```

oder (ohne xsi):

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <update>
      <domain:update xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>ugha.ch</domain:name>
        <domain:chg />
      </domain:update>
    </update>
    <extension>
      <rgp:update xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0">
        <rgp:restore op="request" />
      </rgp:update>
    </extension>
  </command>
</epp>
```

## Beispiel Domain Update Response

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>Test3</clTRID>
      <svTRID>20071207.92654.285546</svTRID>
    </trID>
  </response>
</epp>
```

## 4.3 Contact Commands

### 4.3.1 Allgemeines

Die `contact` Commands werden zusätzlich zu den folgenden Angaben in Referenz (3) detailliert beschrieben. Nur die gesetzlich erlaubten Daten werden veröffentlicht (siehe Ziff. 3.2). Der Umgang der übrigen Einsichtsnahmen und Bearbeitung richtet sich nach Referenz (06).

contact Attribut	mandatory	Update möglich	Bemerkung
id	Yes	No	Server unique id, ASCII-Grossbuchstaben, Ziffern und ”-“, supplied by Registrar
roid	Yes	No	Cnnnnnnnn-SWITCH, supplied by Registry
status	Yes	No	Linked, ok, serverDeleteProhibited, serverUpdateProhibited
name type=”loc”	Yes	Yes	Name Vorname
org type=“loc“	No	Yes	
street	Yes	Yes	
street	No	Yes	
street	No	Yes	
city	Yes	Yes	Maximale Länge : 30 Zeichen
sp	No	Yes	Kanton, Bundesland, Staat
pc	No	Yes	
cc	Yes	Yes	ISO.3166.1997
voice	No	Yes	
fax	No	Yes	
email	Yes	Yes	
clID	Yes	No	Von Registry gesetzt
crID		No	Von Registry gesetzt
crDate		No	Von Registry gesetzt
upID		No	Wird weggelassen.
upDate		No	Wird weggelassen.
trDate		No	Wird weggelassen.
authinfo			Wird ignoriert. ”not supported“
disclose			Error 2308

## 4.3.2 *contact:check*

Mit `contact:check` können Kontakte auf Ihre Existenz überprüft werden. Der Server antwortet mit `registriert` / `nicht registriert`. Massenabfragen sind gemäss Referenz (06) für den darin angegebenen Zweck und in Rücksprache mit SWITCH erlaubt.

### Beispiel `contact check` Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <check>
      <contact:check
        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
        <contact:id>AVAILABLE-CONTACT-ID</contact:id>
        <contact:id>NOT-AVAILABLE-CONTACT-ID</contact:id>
      </contact:check>
    </check>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

### Beispiel `contact check` Response

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <contact:chkData xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
        <contact:cd>
          <contact:id avail="1">AVAILABLE-CONTACT-ID</contact:id>
        </contact:cd>
        <contact:cd>
          <contact:id avail="0">NOT-AVAILABLE-CONTACT-ID</contact:id>
          <contact:reason>In use</contact:reason>
        </contact:cd>
      </contact:chkData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>20071207.92655.285552</svTRID>
    </trID>
  </response>
</epp>
```

### 4.3.3 *contact:info*

Mit `contact:info` werden Informationen zu Kontakten abgefragt.

`contact:info` Abfragen werden nur erfolgreich beantwortet, wenn eines der folgenden Kriterien erfüllt ist:

- der abgefragte Kontakt wird vom abfragenden Registrar verwaltet (sponsoring client)
- die Abfrage enthält ein gültiges `<contact:authInfo>` Element wie folgt:

#### Beispiel `contact info` Command mit `<contact:authInfo>` Element

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <info>
      <contact:info xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
        <contact:id>CH-TECH-ID</contact:id>
        <contact:authInfo>
          <contact:pw roid="D7033341-SWITCH">authinfo_pwd</contact:pw>
        </contact:authInfo>
      </contact:info>
    </info>
    <clTRID>ABC</clTRID>
  </command>
</epp>
```



Das Attribut `roid="D7033341-SWITCH"` im `<contact:pw>` Element muss vorhanden sein und bezieht sich auf einen Domain-Namen bei dem der abgefragte Kontakt als `<domain:registrant>` oder als `<domain:contact>` eingetragen ist. (siehe 4.2.5 `domain:info`) Der Wert vom `<contact:pw>` Element (hier `authinfo_pwd`) muss der korrekte Transfercode vom durch das `roid` Attribut bezeichneten Domain-Namen sein. Der entsprechende Domain-Name darf nicht in der RGP Phase sein. Sind die Angaben im optionalen `<contact:authInfo>` falsch, wird 2201 Authorization Error retourniert.

## Beispiel contact info Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <info>
      <contact:info
        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
        <contact:id>TEST-REGISTRAR-C-5</contact:id>
      </contact:info>
    </info>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

## Beispiel contact info Response

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <contact:infData xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
        <contact:id>TEST-REGISTRAR-C-5</contact:id>
        <contact:roid>C10577393-SWITCH</contact:roid>
        <contact:status s="ok"/>
        <contact:status s="linked"/>
        <contact:postalInfo type="loc">
          <contact:name>Test User3</contact:name>
          <contact:org>SWITCH</contact:org>
          <contact:addr>
            <contact:street>Test department 3</contact:street>
            <contact:street>Teststrasse 3</contact:street>
            <contact:city>Aarau</contact:city>
            <contact:pc>5000</contact:pc>
            <contact:cc>CH</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice>+41.22222555</contact:voice>
        <contact:email>test2@test1.ch</contact:email>
        <contact:clID>TEST-REGISTRAR-C</contact:clID>
        <contact:crID>TEST- REGISTRAR-C</contact:crID>
        <contact:crDate>2007-12-07T10:42:59+01:00</contact:crDate>
      </contact:infData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>20071207.92656.285558</svTRID>
    </trID>
  </response>
</epp>
```

*Abfrage eines Kontaktes, der vom abfragenden Registrar verwaltet wird.*

## 4.3.4 *contact:create*

Mit `contact:create` werden Contact-IDs generiert. Jeder Registrar muss eigene Contact-IDs für die von ihm verwalteten Domain-Namen erstellen und bearbeiten. Dies ist besonders bei einem Transfer wichtig, da am Domain-Namen vor dem Einsetzen eigener Contact-IDs keine Änderungen vorgenommen werden können.



- `<contact:postalInfo type="loc">` Nur der Adresstyp "loc" (localized) wird unterstützt. Der Adresstyp "int" (international) (7-bit ASCII-Zeichen) wird nicht zugelassen. Ein Command mit diesem Adresstyp wird zurückgewiesen. Werden Attribute mit `type="int"` und `type="loc"` gleichzeitig geschickt, wird der Adresstyp mit `type="int"` ignoriert.
- `<contact:street>` Eine Adresse kann bis zu drei Zeilen für die Strasse (Adresse) haben, muss jedoch mindestens ein `<contact:street>` Element enthalten. In diese Elemente ist ein allenfalls bestehendes Postfach einzugeben.
- `<contact:disclose>` wird nicht unterstützt.
- `<contact:email>` Die E-Mail-Adresse des Halters muss bei Kontakten angegeben werden. Die E-Mailadresse des Registrars darf nicht in diesem Feld erscheinen.
- `<contact:pw>` wird von SWITCH nicht verwendet und nicht abgespeichert. Die Authentifizierung erfolgt beim EPP-Login.
- `<contact:id>` muss auf dem EPP-Server eindeutig („server-unique“) sein und mindestens einen ASCII-Grossbuchstaben enthalten. Ziffern und Bindestrich sind erlaubt.

### Beispiel `contact create` Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <contact:create xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
        <contact:id>TEST-REGISTRAR-C-5</contact:id>
        <contact:postalInfo type="loc">
          <contact:name>Test User2</contact:name>
          <contact:org>SWITCH</contact:org>
          <contact:addr>
            <contact:street>Test department 2</contact:street>
            <contact:street>Teststrasse 2</contact:street>
            <contact:city>Aarau</contact:city>
            <contact:pc>5000</contact:pc>
            <contact:cc>CH</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice>+41.22222555</contact:voice>
        <contact:email>test2@test1.ch</contact:email>
      <contact:authInfo>
        <contact:pw/>
      </contact:authInfo>
    </contact:create>
  </create>
  <clTRID>Test2</clTRID>
</command>
</epp>
```

## Beispiel contact create Response

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <contact:creData xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
        <contact:id>TEST-REGISTRAR-C-5</contact:id>
        <contact:crDate>2007-12-07T11:29:51+01:00
        </contact:crDate>
      </contact:creData>
    </resData>
    <trID>
      <clTRID>Test2</clTRID>
      <svTRID>20071207.944.74886</svTRID>
    </trID>
  </response>
</epp>
```

### 4.3.5 *contact:delete*

Mit `contact:delete` können Contact-IDs gelöscht werden, sofern sie nirgends verknüpft sind (Status „linked“). Es dürfen nur selbst verwaltete Contact-IDs gelöscht werden. Der Registrar muss seine nicht mehr verwendeten Contact-IDs löschen.

#### Beispiel `contact delete` Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <delete>
      <contact:delete
        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
        <contact:id>TEST-REGISTRAR-C-6</contact:id>
      </contact:delete>
    </delete>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

#### Beispiel `contact delete` Response

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>20071207.92657.285565</svTRID>
    </trID>
  </response>
</epp>
```

### 4.3.6 *contact:transfer*

Der `contact:transfer` Command wird nicht unterstützt und mit dem Fehlercode 2101 abgelehnt.

### 4.3.7 *contact:update*

Mit `contact:update` werden Contact-IDs aktualisiert. Hierbei gelten dieselben Bedingungen wie bei `contact:create`. Die Adresse `<contact:addr>` kann nur als ganzer Block geändert werden.



- `<contact:postalInfo type="loc">` Nur der Adresstyp "loc" (localized) wird unterstützt. Der Adresstyp "int" (international) (7-bit ASCII-Zeichen) wird nicht zugelassen. Ein Command mit diesem Adresstyp wird zurückgewiesen. Werden Attribute mit `type="int"` und `type="loc"` gleichzeitig geschickt, wird der Adresstyp mit `type="int"` ignoriert.
- `<contact:street>` Eine Adresse kann bis zu drei Zeilen für die Strasse (Adresse) haben, muss jedoch mindestens ein `<contact:street>` Element enthalten. In diese Elemente ist ein allenfalls bestehendes Postfach einzugeben.
- `<contact:disclose>` wird nicht unterstützt.
- `<contact:email>` Die E-Mail-Adresse des Halters muss bei Kontakten angegeben werden. Die E-Mailadresse des Registrars darf nicht in diesem Feld erscheinen.
- `<contact:pw>` wird von SWITCH nicht verwendet und nicht abgespeichert. Die Authentifizierung erfolgt beim EPP-Login.
- `<contact:id>` muss auf dem EPP-Server eindeutig („server-unique“) sein und mindestens einen ASCII-Grossbuchstaben enthalten. Ziffern und Bindestrich sind erlaubt.

## Beispiel contact update Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <update>
      <contact:update
        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
        <contact:id>TEST-REGISTRAR-C-5</contact:id>
        <contact:chg>
          <contact:postalInfo type="loc">
            <contact:org/>
            <contact:addr>
              <contact:street>124 Example Dr.</contact:street>
              <contact:street>Suite 200</contact:street>
              <contact:city>Dulles</contact:city>
              <contact:sp>VA</contact:sp>
              <contact:pc>20166-6503</contact:pc>
              <contact:cc>US</contact:cc>
            </contact:addr>
          </contact:postalInfo>
          <contact:voice>+1.7034444444</contact:voice>
          <contact:fax/>
          <contact:authInfo>
            <contact:pw/>
          </contact:authInfo>
        </contact:chg>
      </contact:update>
    </update>
    <clTRID>freechoice</clTRID>
  </command>
</epp>
```

## Beispiel contact update Response

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>freechoice</clTRID>
      <svTRID>20071207.92690.285728</svTRID>
    </trID>
  </response>
</epp>
```

## 4.4 Host Commands

### 4.4.1 Allgemeines

Alle `host` Commands werden zusätzlich zu den folgenden Angaben in Referenz (4) detailliert beschrieben. Es werden die folgenden Hosts (Name-Server) unterschieden:

<b>gültige interne Name-Server unter .ch/.li</b>	<b>ungültige interne Name-Server unter .ch/.li</b>	<b>externe Name-Server nicht unter .ch/.li</b>
Endet auf .ch oder .li, übergeordneter (superordinate) Domain-Name ist registriert	Endet auf .ch oder .li, übergeordneter (superordinate) Domain-Name ist nicht registriert	Name-Server, der nicht auf .ch oder .li endet
<code>ns1.iamregistered.ch</code>	<code>ns1.iamnotregistered.ch</code>	<code>ns1.yourname.com</code>
Domain-Name <code>iamregistered.ch</code> ist registriert	Domain-Name <code>iamnotregistered.ch</code> ist nicht registriert	Unabhängig davon, ob <code>yourname.com</code> registriert ist oder nicht

**Gültige interne Name-Server unter .ch/.li** gehören dem Halter des übergeordneten Domain-Namens und werden von seinem Registrar verwaltet. Sie werden zusammen mit dem Domain-Namen transferiert.

**Ungültige interne Name-Server unter .ch/.li** werden von SWITCH verwaltet, solange der übergeordnete Domain-Name nicht registriert ist. Das Registrieren eines ungültigen internen Name-Servers wird zugelassen. Ein ungültiger interner Name-Server darf nur dem übergeordneten Domain-Namen zugeordnet werden, womit er zu einem gültigen internen Nameserver wird.

Wird ein Domain-Name (`yourname.ch`) gelöscht, bleiben die untergeordneten Name-Server (`ns1.yourname.ch`) weiter bestehen. Wird der Domain-Name später wieder neu registriert, gehören diese Name-Server automatisch dem Halter des neuen Domain-Namens.

**Externe Name-Server** wie zum Beispiel `ns1.yourname.com` können registriert werden und gehören der Registry.

host Attribut	mandatory	Update möglich	Bemerkung
name	Yes	No	
roid	Yes	No	Hnnnnnnnn-SWITCH, supplied by Registry
status	Yes	No	
clID	Yes	No	Von Registry gesetzt
crID			“NOT SUPPORTED“
crDate		No	Von Registry gesetzt
upID			Wird weggelassen.
upDate		No	Wird weggelassen.
trDate		No	Wird weggelassen.

## 4.4.2 host:check

Mit `host:check` können Name-Server auf Ihre Existenz überprüft werden. Der Server antwortet mit `registriert` / `nicht registriert`. Massenabfragen sind nur im Rahmen von Referenz (06) zugelassen.



- Mit `host:check` können maximal 10 Hosts gleichzeitig abgefragt werden, ansonsten antwortet der Server mit dem Fehlercode 2308 („Data management policy violation“).

### Beispiel host check Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:iETF:params:xml:ns:epp-1.0">
  <command>
    <check>
      <host:check
        xmlns:host="urn:iETF:params:xml:ns:host-1.0">
        <host:name>merapi.switch.ch</host:name>
        <host:name>ns2.example.com</host:name>
      </host:check>
    </check>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

### Beispiel host check Response

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <host:chkData xmlns:host="urn:ietf:params:xml:ns:host-1.0">
        <host:cd>
          <host:name avail="0">merapi.switch.ch</host:name>
          <host:reason>In use</host:reason>
        </host:cd>
        <host:cd>
          <host:name avail="1">ns2.example.com</host:name>
        </host:cd>
      </host:chkData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>20071207.92693.285743</svTRID>
    </trID>
  </response>
</epp>
```

### 4.4.3 host:info

Mit `host:info` werden Informationen zu Name-Servern abgefragt. Die Felder `<host:clID>`, `<host:crID>` und `<host:crDate>` werden nur bei den vom Registrar verwalteten Name-Servern angezeigt.

### Beispiel host info Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <info>
      <host:info
        xmlns:host="urn:ietf:params:xml:ns:host-1.0">
        <host:name>merapi.switch.ch</host:name>
      </host:info>
    </info>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

### Beispiel host check Response

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <host:infData xmlns:host="urn:ietf:params:xml:ns:host-1.0">
        <host:name>merapi.switch.ch</host:name>
        <host:roid>H5734-SWITCH</host:roid>
        <host:status s="ok"/>
        <host:status s="linked"/>
        <host:addr ip="v4">130.59.211.10
        </host:addr>
        <host:addr ip="v6">2001:620:0:0:0:0:5</host:addr>
        <host:clID>SWITCH-REGISTRAR</host:clID>
        <host:crID>NOT SUPPORTED</host:crID>
        <host:crDate>1999-11-24T14:48:50+01:00</host:crDate>
      </host:infData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>20071207.92693.285744</svTRID>
    </trID>
  </response>
</epp>
```

#### 4.4.4 *host:create*

Mit *host:create* werden Name-Server registriert. Es gelten die unter Ziff. 4.4.1 oben aufgeführten Berechtigungen. Externe und ungültige interne Name-Server können von jedermann erstellt werden. Gültige interne Name-Server können nur vom Registrar erstellt werden, der den übergeordneten Domain-Namen verwaltet. Maximal 20 IP's per Host und max. 20 Hosts per Domain-Name zulässig.

##### Beispiel *host create* Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <host:create
        xmlns:host="urn:ietf:params:xml:ns:host-1.0">
        <host:name>ns1.yourname.ch</host:name>
        <host:addr ip="v4">192.0.2.2</host:addr>
        <host:addr ip="v4">192.0.2.29</host:addr>
        <host:addr ip="v6">1080:0:0:0:8:800:200C:417A</host:addr>
      </host:create>
    </create>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

##### Beispiel *host create* Response

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <host:creData xmlns:host="urn:ietf:params:xml:ns:host-1.0">
        <host:name>ns1.yourname.ch</host:name>
        <host:crDate>2007-12-07T14:40:04+01:00</host:crDate>
      </host:creData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>20071207.92694.285749</svTRID>
    </trID>
  </response>
</epp>
```

#### 4.4.5 *host:transfer*

Dieser Command ist in Referenz (03) nicht vorgesehen und wird mit dem Fehlercode 2000 abgelehnt. Gültige interne Name-Server gehören dem Halter des entsprechenden Domain-Namens. Die Name-Server werden zusammen mit dem Domain-Namen transferiert.

#### 4.4.6 *host:delete*

Mit *host:delete* werden Name-Server gelöscht. Name-Server dürfen nur gelöscht werden, wenn sie im Registrierungssystem nicht mehr referenziert werden (wenn kein Domain-Name mehr auf den Name-Server verweist).

## Beispiel host delete Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <delete>
      <host:delete
        xmlns:host="urn:ietf:params:xml:ns:host-1.0">
        <host:name>ns1.yourname.ch</host:name>
      </host:delete>
    </delete>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

## Beispiel host delete Response

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>20071210.94223.292958</svTRID>
    </trID>
  </response>
</epp>
```

## 4.4.7 *host:update*

### Beispiel host update Command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <update>
      <host:update>
        xmlns:host="urn:ietf:params:xml:ns:host-1.0">
          <host:name>ns1.yourns.ch</host:name>
          <host:add>
            <host:addr ip="v4">192.0.2.22</host:addr>
          </host:add>
          <host:rem>
            <host:addr ip="v6">1080:0:0:0:8:800:200C:417A</host:addr>
          </host:rem>
        </host:update>
      </update>
      <clTRID>ABC-12345</clTRID>
    </command>
  </epp>
```

### Beispiel host update Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>54321-XYZ</svTRID>
    </trID>
  </response>
</epp>
```

Der Name eines Name-Servers kann über die EPP-Schnittstelle nicht geändert werden.

## 5 Testing

### 5.1 Test System

SWITCH kann zu Testzwecken den Zugang auf den EPP-Testserver (**epp-test.switch.ch**) freischalten. Die Verfügbarkeit des EPP-Testservers ist geringer als jene des produktiven EPP-Servers. In der Regel ist auf dem EPP-Testserver dieselbe Softwareversion wie auf dem produktiven EPP-Server installiert. Vor neuen Software-Releases wird auf dem EPP-Testserver die neue Version in der Regel 3 Tage vor der Installation auf dem produktiven EPP-Server verfügbar sein. Ausnahmen von dieser Regel sind unter anderem dringende Bug-Fixes, Security-Massnahmen oder grössere technische Neuerungen. Diesfalls fällt die Frist weg oder es wird eine Übergangsfrist definiert.

#### 5.1.1 Voraussetzungen für das Test System

- Für das Test System erhält der Registrar zwei individuelle Client Identifier clID (TEST-XXXXXXXXXX, TEST-XXXXXXXXXX) sowie die Passwörter.
- Diese clIDs können vom Registrar während der ganzen Vertragsdauer genutzt werden
- Um das Testsystem einzurichten, benötigt SWITCH mindestens eine IP-Adresse.
- **Connection**  
Host: epp-test.switch.ch  
Port: 7001

## 5.2 Testparcours

Bevor die EPP-Schnittstelle auf der produktiven Umgebung (epp.nic.ch) für einen Registrar zur Nutzung von SWITCH freigeschaltet wird, muss der unten beschriebene Testablauf erfolgreich bestanden werden. Um den Test ausführen zu können, muss mit SWITCH eine Testperiode vereinbart werden. Es kann nur jeweils eine Registrar zu einem bestimmten Zeitpunkt den Test Parcours absolvieren. Darüber hinaus gelten die Bestimmungen gemäss Referenz (6).

Der Registrar erhält von SWITCH eine separate clID und ein Passwort. Diese Credentials sind **nur** für den Testparcours gültig und können nicht für das oben beschriebene Testsystem genutzt werden.

### 5.2.1 Voraussetzungen für den Testparcours

- Der Registrar hat ausführlich auf dem Test System (**epp-test.switch.ch**) getestet
- Für den Testparcours erhält der Registrar eine clID und ein Passwort.
- Diese clID ist nur für den Testparcours gültig und nicht für das Testsystem.
- Um den Testparcours einzurichten, benötigt SWITCH mindestens eine IP-Adresse.
- **Connection**  
Host: epp-test.switch.ch  
Port: 7001

## 5.2.2 Ablauf Testparcours

Die einzelnen Commands sind in der angegebenen Reihenfolge auszuführen. Nach jedem Command muss die Response des EPP-Servers mit dem erwarteten Ergebnis übereinstimmen.

### Schritt 1 - Login und Change password

Description / Tag	Value	Comment
clID	test-registrar-a	test-account client-identifier
pw	*****	initial password
newPW	*****	selbst gewähltes Passwort, Passwort-Regeln beachten gemäss Referenz (6)

Erwartetes Resultat: "Command completed successfully".

### Schritt 2 - Vorhandene Messages abfragen und bestätigen

Description / Tag	Value	Comment
poll op="req"		Msg_Id extrahieren

Erwartetes Resultat: "Command completed successfully; ack to dequeue".

### Schritt 3 - Transfer domain 1

Description / Tag	Value	Comment
domain:name	test-registrar-a-domain-1.ch	
domain:pw	'my&p\$w#d22.'	Transfercode ohne Single Quotes (Apostroph, Hochkomma)

Erwartetes Resultat: "Command completed successfully".

### Schritt 4 - Messages bestätigen

Description / Tag	Value	Comment
poll op="ack" msgID="nn"		msgID from the poll op="req" extrahieren und bei nn einsetzen

Erwartetes Resultat: "Command completed successfully".

### Schritt 5 - check contact 1

Description / Tag	Value	Comment
contact:id	TEST-CONTACT-1	

Erwartetes Resultat: "Command completed successfully. The contact:id is available."

### Schritt 6 - Create contact 1

Description / Tag	Value	Comment
contact:id	TEST- CONTACT -1	
contact:name	Lastname Firstname	
contact:org	Organisation	
contact:street	Testabteilung	
contact:street	Teststrasse 999	
contact:city	Bern	
contact:pc	3001	
contact:cc	CH	
contact:voice	+41.335555555	
contact:email	test1@yourdomain.ch	

Erwartetes Resultat: "Command completed successfully".

### Schritt 7 - Create nameserver 1

Description / Tag	Value	Comment
host:name	ns3.test-registrar-a-domain-1.ch	
host:addr ip="v4"	240.1.1.1	

Erwartetes Resultat: "Command completed successfully".

### Schritt 8 - Info domain 1

Description / Tag	Value	Comment
domain:name	test-registrar-a-domain-1.ch	

Erwartetes Resultat: "Command completed successfully". Die Response enthält die contact:id vom technischen Kontakt für den nächsten Schritt.

### Schritt 9 - Info Tech-Contact

Description / Tag	Value	Comment
contact:id		Wert vom domain:contact type="tech"

Erwartetes Resultat: "Command completed successfully".

### Schritt 10 - Update domain 1 (Transfer vervollständigen)

Description / Tag	Value	Comment
domain:name	test-registrar-a-domain-1.ch	
domain:add domain:ns	ns3.test-registrar-a-domain-1.ch	
domain:add domain:contact type="tech"	TEST-CONTACT-1, oder mit einem anderen, vorher generierten Kontakt	Zu diesem Zeitpunkt ist kein domain:rem notwendig.
domain:rem domain:ns	ns1.test-registrar-a-domain-1.ch	
domain:chg domain:registrant	TEST-CONTACT-1	

Erwartetes Resultat: "Command completed successfully".

### Schritt 11 - Create contact 2

Description / Tag	Value	Comment
contact:id	TEST-CONTACT-2	
contact:name	Lastname2 Firstname2	
contact:org	Organisation2	
contact:street	Testdivision	
contact:street	Teststreet 999	
contact:city	Bern	
contact:pc	3001	
contact:cc	CH	
contact:voice	+41.335555555	
contact:email	test2@yourdomain.ch	

Erwartetes Resultat: "Command completed successfully".

### Schritt 12 - Delete domain 1

Description / Tag	Value	Comment
domain:name	test-registrar-a-domain-1.ch	

Erwartetes Resultat: "Command completed successfully".

### Schritt 13 - Restore domain 1

Description / Tag	Value	Comment
domain:name	test-registrar-a-domain-1.ch	

Erwartetes Resultat: "Command completed successfully".

## Schritt 14 - Update domain 1

Description / Tag	Value	Comment
domain:name	test-registrar-a-domain-1.ch	
domain:chg domain:registrant	TEST-CONTACT-2	

Erwartetes Resultat: "Command completed successfully".

## Schritt 15 - Delete contact 1

Description / Tag	Value	Comment
contact:id	TEST-CONTACT-1	

Erwartetes Resultat: "Command completed successfully".

## Schritt 16 - Delete nameserver 1

Description / Tag	Value	Comment
host:name	ns1.test-registrar-a-domain-1.ch	

Erwartetes Resultat: "Command completed successfully".

## Schritt 17 - Check domain 2

Description / Tag	Value	Comment
domain:name	test-registrar-a-domain-2.ch	

Erwartetes Resultat: "Command completed successfully". The domainname is available.

## Schritt 18 - Create domain 2

Description / Tag	Value	Comment
domain:name	test-registrar-a-domain-2.ch	
domain:hostObj	ns3.test-registrar-a-domain-1.ch	
domain:registrant	TEST-CONTACT-2	

Erwartetes Resultat: "Command completed successfully".

## Schritt 19 - Create nameserver 2 subordinate of domain 2

Description / Tag	Value	Comment
host:name	ns1.test-registrar-a-domain-2.ch	
host:addr ip="v4"	240.1.1.1	

Erwartetes Resultat: "Command completed successfully".

### Schritt 20 - Update domain 1 add ns

Description / Tag	Value	Comment
domain:name	test-registrar-a-domain-1.ch	
domain:add domain:ns	ns1.test-registrar-a-domain-2.ch	

Erwartetes Resultat: "Command completed successfully".

### Schritt 21 - Update contact 2 (Alle nicht erwähnten Werte sollen unverändert bleiben)

Description / Tag	Value	Comment
contact:id	TEST-CONTACT-2	
contact:chg contact:org		Empty
contact:chg contact:addr contact:street	New Division	
contact:chg contact:addr contact:street	Teststreet 999	Bisheriger Wert

Erwartetes Resultat: "Command completed successfully".

### Schritt 22 - Update domain 2 with authinfo

Description / Tag	Value	Comment
domain:name	test-registrar-a-domain-2.ch	
domain:chg domain:authInfo domain:pw	2BARfoo	

Erwartetes Resultat: "Command completed successfully".

### Schritt 23 - Update domain 1 with DNSSEC data

Die Schritte 22 bis 24 müssen nur durchgeführt werden, wenn der Registrar DNSSEC einsetzen will. Voraussetzung ist, dass man mit der Extension :secDNS eingeloggt ist. Und die Testschritte 1 bis 22 durchgeführt hat.

Description / Tag	Value	Comment
domain:name	test-registrar-a-domain-1.ch	
secDNS:add		
secDNS:dsData		
secDNS:keyTag	12346	
secDNS:alg	13	
secDNS:digestType	2	
secDNS:digest	CAFFEEBABE00D87A014 7EFE9877AB7335206ABFF 6FA730BD6239D65CBAC7 E768	

Erwartetes Resultat: "Command completed successfully".

### Schritt 24 – Überprüfen des updates mit domain:info

Description / Tag	Value	Comment
domain:name	test-registrar-a-domain-1.ch	TEST-REGISTRAR mit erhaltener clID ersetzen.
domain:info		

Erwartetes Resultat: "Command completed successfully".

### Schritt 25 – Austauschen der DNSSEC data

Description / Tag	Value	Comment
domain:name	test-registrar-a-domain-1.ch	
secDNS:rem secDNS:dsData secDNS:keyTag secDNS:alg secDNS:digestType secDNS:digest	12346 13 2 CAFFEEBABE00D87A014 7EFE9877AB7335206ABFF 6FA730BD6239D65CBAC7 E768	
secDNS:add secDNS:dsData secDNS:keyTag secDNS:alg secDNS:digestType <b>secDNS:digest</b>	44475 14 2 AABBCCBABE00D87A014 7EFE9877AB7335206ABFF 6FA730BD6239D65CBAC7 E768	

Erwartetes Resultat: “Command completed successfully”.

### Schritt 26 - Logout

Erwartetes Resultat: successfully completed, ending session.

## 6 Weitergehende Informationen

### 6.1 Web-Interface für EPP-Benutzer

Über das Web-Interface stehen Kontobelege und weitere Informationen zur Verfügung. Domain-Namen können nicht über das Web-Interface verwaltet werden.

Kontakt: **registrar@nic.ch**

## Appendix A: Abkürzungen / Begriffe

Begriff	Erläuterung
AuthInfo	Transfercode, welcher benötigt wird, um einen Domain- zu
ccTLDs	Country-Code-Top-Level Domain-Namen
Contact-ID	Handle bzw. Objektreferenz
Command	Befehl, welcher der EPP-Client senden kann, um auf dem EPP-Server eine bestimmte Aktion auszulösen. Die Aktionen sind einem Objekt (Domain-Name, Name-Server und Kontakt) zugeordnet.
DNSSEC	Domain Name System Security Extension
EPP	Extensible Provisioning Protocol
EPP-Benutzer	Registrar, welcher über die EPP-Schnittstelle mit SWITCH
EPP-Schnittstelle	Schnittstelle, welche auf dem EPP-Protokoll basiert.
externer Name-Server	Name-Server, der nicht zu der vom Registrierungssystem verwalteten ccTLD gehört. Bei SWITCH ist dies ein Name-Server, welcher nicht mit .ch oder .li endet.
interner Name-Server	Name-Server, der zu der vom Registrierungssystem verwalteten ccTLD gehört. Bei SWITCH ist dies ein Name-Server, welcher mit .ch oder .li endet.
gültiger interner NS	Interner Name-Server eines registrierten übergeordneten Domain-
ungültiger interner Name-Server	Interner Name-Server eines nicht registrierten übergeordneten Domains.
Registrant	Endkunde, der einen Domain-Namen besitzt.
Registrar	Ein Internet Service Provider mit von SWITCH unterzeichnetem
Registry	Organisation, welche eine Registrierungsstelle für Second-Level Domain-Namen betreibt.
Transfer	Abtretung der Verwaltungsrechte an einem Domain-Namen an einen anderen Registrar.
TLS	Transport Layer Security; Sicherheitsprotokoll für
Übertragung	Der Domain-Name wird einem anderen Halter, das heisst einer anderen natürlichen oder juristischen Person zugeteilt.