

# Registry system the .fr and French overseas cCTLDS

---

## Technical Integration Guide

December 2024

## Table of contents

<b>1</b>	<b>Preface</b> .....	<b>7</b>
1.1	About this document .....	7
1.2	Target audience .....	7
1.3	Typographic conventions .....	7
<b>2</b>	<b>Main integration principles of the EPP protocol</b> .....	<b>8</b>
2.1	RFCs .....	8
2.2	Case of operations with return code 1000 and server behaviour in case of problems	8
2.3	Integration options for the list of notification messages.....	9
2.4	DNSSEC support.....	9
<b>3</b>	<b>Extension frnic-2.0</b> .....	<b>10</b>
3.1	General description of the frnic extension: .....	10
3.2	Description of elements for creating a contact: .....	10
3.3	Description of elements for a transfer: .....	11
3.5	IDN (IDNA2008) .....	12
	Reference documents .....	12
	Brief background on IDN technology .....	12
	Warning .....	12
	Vocabulary .....	13
<b>4</b>	<b>Interface connection settings</b> .....	<b>15</b>
4.1	EPP test bench .....	15
	Test bench: .....	16
	Production bench: .....	16
4.3	Connection and authentication to EPP server .....	16
4.4	Certificat management .....	17

<b>5</b>	<b>Session management commands</b>	<b>18</b>
5.1	<greeting>	18
	Example of <greeting> that can be sent by the AFNIC EPP server:	19
5.2	<hello> command	19
	Example of a <hello> request sent by the client:	20
5.3	Session management commands	20
5.4	The <login> command	20
	Login without declaration of the EPP Fee extension	20
5.5	Strict authentication	21
	Logout	22
<b>6</b>	<b>Contact management</b>	<b>23</b>
6.1	Description	23
6.2	Contacts settings	23
6.3	contact:create - Creating a contact	24
	Contact create PP type with the minimum information required	25
	PP type contact create with "eligibility" and "reachable status	26
	Contact create for PM type contact with the minimum information required	29
	Example of creating a PM type contact with a SIREN number	30
	Example of creating a PM type contact with a DUNS number	32
	Example of creating a PM type contact with a trademark number	34
	Example of creating a PM type contact with an intra- Community VAT number	35
	Example of creating a PM type contact with a local identifier	37
	Example of creating a PM type contact with a WALDEC (associations only)	38
	Example of creating a PM type of contact with the reporting information to the prefecture and the Official Gazette (associations only)	40
	Specifics of the .fr for the management of postal addresses:	41

6.4	<b>contact:update - Updating a contact .....</b>	<b>42</b>
	<b>Example of contact change with a change in the phone number and the email address.....</b>	<b>42</b>
	<b>Example of a change in contact with a change in the postal address .....</b>	<b>43</b>
	<b>Example of a change in contact to remove the restricted publication .....</b>	<b>44</b>
	<b>Example of a change in contact to apply restricted publication.....</b>	<b>45</b>
6.5	<b>contact:info - Information on a contact .....</b>	<b>45</b>
	<b>Example contact:info on a contact of the "Legal Entity" type .....</b>	<b>46</b>
	<b>Example of server response for an "Individual" type contact .....</b>	<b>48</b>
6.6	<b>Le contact:delete – Deleting the contact .....</b>	<b>50</b>
<b>7</b>	<b>Domain name management .....</b>	<b>51</b>
7.1	<b>Multi-year and expiry date calculation.....</b>	<b>51</b>
	<b>Number of years available.....</b>	<b>51</b>
	<b>Explicit renewal rules .....</b>	<b>51</b>
7.2	<b>Grace periods .....</b>	<b>52</b>
	<b>General description .....</b>	<b>52</b>
	<b>List, duration and description.....</b>	<b>52</b>
	<b>Cancellation of a grace period .....</b>	<b>52</b>
7.3	<b>domain:check .....</b>	<b>54</b>
	<b>Example of domain:check command.....</b>	<b>54</b>
7.4	<b>domain:create .....</b>	<b>55</b>
	<b>Example of the creation of a domain name.....</b>	<b>56</b>
	<b>Example of the creation of a domain name with hosts.....</b>	<b>57</b>
7.5	<b>domain:update .....</b>	<b>58</b>
	<b>Example of a domain name update with the addition of authoritative host.....</b>	<b>59</b>
	<b>Example of a domain name update with a change in holder and technical contact.....</b>	<b>60</b>
7.6	<b>domain:info .....</b>	<b>60</b>

<b>Example of the domain:info command for a domain name belonging to the registrar's portfolio.....</b>	<b>61</b>
<b>Example of the domain:info command for a domain name not belonging to the registrar's portfolio with the auth info .....</b>	<b>62</b>
<b>7.7 domain:transfer.....</b>	<b>63</b>
<b>Make a transfer .....</b>	<b>63</b>
<b>Example of a transfer operation command .....</b>	<b>64</b>
<b>Approve or reject a TRANSFER (only for the outgoing registrar) .....</b>	<b>65</b>
<b>Example of the rejection of a transfer operation .....</b>	<b>66</b>
<b>Cancel the TRANSFER (only for the incoming registrar) .....</b>	<b>67</b>
<b>7.8 domain:recover .....</b>	<b>68</b>
<b>Example of a forced changed of registrant .....</b>	<b>69</b>
<b>7.9 domain:renew - RENEW .....</b>	<b>70</b>
<b>Example of an explicit renew operation.....</b>	<b>70</b>
<b>7.10 domain:delete - DELETE.....</b>	<b>71</b>
<b>Example of deleting a domain: .....</b>	<b>71</b>
<b>7.11 rgp:restore - RESTORE.....</b>	<b>72</b>
<b>Example of a restore operation .....</b>	<b>73</b>
<b>8 Host management .....</b>	<b>74</b>
<b>8.1 Description .....</b>	<b>74</b>
<b>8.2 Definitions:.....</b>	<b>74</b>
<b>8.3 host:check .....</b>	<b>74</b>
<b>Example of a check command on 2 host objects.....</b>	<b>75</b>
<b>8.4 host:create .....</b>	<b>76</b>
<b>Example of the creation of a host object .....</b>	<b>76</b>
<b>8.5 host:update .....</b>	<b>77</b>
<b>Example of update of a host object .....</b>	<b>77</b>

8.6 host:info .....	78
Example of a host object info request .....	78
8.7 host:delete .....	79
Example of deletion of a host object.....	79
<b>9 Notifications .....</b>	<b>80</b>
9.2 Management of the notification queue .....	80
9.3 Notifications .....	80
Example of a notification following a requested TRANSFER operation (outgoing registrar).....	81
Example of a notification following a TRANSFER operation completed after approval by the outgoing registrar (incoming registrar).....	85
Example of a notification following a TRANSFER operation completed after no response from the outgoing registrar (8 days after the start of the transfer) (outgoing registrar).....	87
Example of a notification following a TRANSFER completed after no response from the outgoing registrar (8 days after the start of the transfer) (incoming registrar) .....	88
Example of a notification following a TRANSFER operation cancelled by the incoming registrar (outgoing registrar) .....	90
Example of notification following a completed RECOVER operation (outgoing registrar).....	94
Example of notifications following a RESTORE operation.....	95
<b>10 Return codes and error messages.....</b>	<b>96</b>
10.1 Return codes.....	96
10.2 Error messages .....	98

# 1 Preface

## 1.1 About this document

This integration guide compiles all the information needed to integrate the application interface for managing Afnic domain.

This interface provides three access methods:

- **The web interface**
- **EPP** (Extensible Provisioning Protocol): a standard protocol for exchanges between registries and registrars.
- **REST API** (APIs guide of the .fr registration system)

All operations available in EPP are also available on the web interface, except change of EPP password.

Regarding EPP, AFNIC complies with the standard described in the RFCs (see § RFCs). This document describes the points specific to the protocol integrated by AFNIC.

## 1.2 Target audience

This technical document is intended for developers wishing to:

- Obtain a detailed description of the interface
- Find examples that facilitate their integration

This document does not give details of the procedures. For further information on procedures, please refer to **Procedures Guide** and **The Naming Charter**.

## 1.3 Typographic conventions

Throughout the document, the following conventions are used:

Between < > the xml tags describing the EPP frames.

In a blue frame, examples of EPP frames.

The prefixes of EPP frames are represented as follows:

- secDNS **for** urn:ietf:params:xml:ns:secDNS-1.1,
- domain **for** urn:ietf:params:xml:ns:domain-1.0,
- contact **for** urn:ietf:params:xml:ns:contact-1.0,
- host **for** urn:ietf:params:xml:ns:host-1.0,
- rgp **for** urn:ietf:params:xml:ns:rgp-1.0.

## 2 Main integration principles of the EPP protocol

### 2.1 RFCs

As a reminder, here are the **RFCs** on which our EPP implementation is based, and which must be read:

- **RFC 3375** - Generic Registry-Registrar Protocol Requirements : <http://www.ietf.org/rfc/rfc3375.txt>
- **RFC 5730** - Extensible Provisioning Protocol (EPP) : <https://www.ietf.org/rfc/rfc5730.txt>
- **RFC 5731** - Domain Name Mapping : <https://www.ietf.org/rfc/rfc5731.txt>
- **RFC 5732** - Host Mapping : <https://tools.ietf.org/html/rfc5732>
- **RFC 5733** - Contact Mapping : <https://www.ietf.org/rfc/rfc5733.txt>
- **RFC 5734** - EPP over TCP : <http://www.ietf.org/rfc/rfc5734.txt>
- **RFC 3915** - Domain Registry Grace Period Mapping : <https://www.ietf.org/rfc/rfc3915.txt>
- **RFC 5910** - Domain Name System (DNS) Security Extensions Mapping : <https://www.ietf.org/rfc/rfc5910.txt>
- **RFC 8748** - Registry Fee Extension for the Extensible Provisioning Protocol (EPP): <https://tools.ietf.org/html/rfc8748#section-5.1>

Apart from the EPP standard as described in the RFCs, AFNIC has laid down a number of integration principles that you need to know before embarking on the development of an EPP client.

Here are the extensions that can be used optionally:

- The extension for DNSSEC management: **secDNS EPP version 1.1** as defined in RFC 5910.

### 2.2 Case of operations with return code 1000 and server behaviour in case of problems

One precaution is necessary when developing clients connecting to our EPP server. At several times in the rest of the document we mention operations returning a return code **1000**. This is the expected behaviour under normal operating conditions of the registration chain.

In the case of a blocking problem, the server reacts more radically and no operation of the "transform" type on domain names can be taken into account. An error message "command failed" (code **2400/2500**) is then returned for a new command.



## 2.3 Integration options for the list of notification messages

In the case of any response from the server, we have chosen to specify the number of messages in the queue (unless there is no message, in which case, this information will not be provided).

**RFC 5730** does not require this information in the case of responses to **<poll>** command, whereas it is optional for other types of commands. In practice this means that when a message is to be notified to a registrar, the latter is notified by the presence of the **<msgQ>** element in the responses to all the commands sent to the main EPP server (the EPP server for the .FR Rush service does not have this information).

It is strongly recommended to consult these messages as and when they arrive. Indeed, in the list of follow-up messages involving technical change operations, there may well be **TRANSFER** requests that may require a response.

## 2.4 DNSSEC support

The EPP server manages the **secDNS-1.1** extension as described in **RFC 5910**, excluding any other version.

The implementation specifics are as follows:

- The server only supports the “DS data interface” (**<secDNS:dsData>**), section 1 of **RFC 5910**, without any information on the associated key (no **<secDNS:keyData>**) element; the inclusion of information on the key will generate a **2102** error.
- A domain name can have a maximum of 6 associated DS records: the number of **<secDNS:dsData>** elements present in the **<secDNS:add>** section during an **UPDATE** operation is therefore limited so that the final status of the domain name has no more than 6 DS records.
- The maxSigLife element is not supported; its inclusion in the client request will generate an error.
- The “urgent” attribute is not supported; its inclusion in the client request with value 1 will generate a **2102** error.
- During a **TRANSFER** operation, the Afnic extension part **frnic** must include a keepDS flag which is a Boolean:
  - If it is 1, the current DS records of the domain name are retained after the transfer if they are already present.
  - If it is 0, if the transfer is successful, all the existing DS records will be deleted.

## 3 Extension frnic-2.0

### 3.1 General description of the frnic extension:

Prefix used: **frnic**

Various operations require the use of the **frnic** extension. We will briefly describe its usefulness (descriptions of elements and examples of commands in more detail in the sections on the operations).

The **frnic** extension is required for the following:

- **contact:create:** creating a contact
- **contact :info :** contact information
- **contact :update :** update a contact
- **domain:transfer:** transferring a domain name
- **domain:check:** Information on the availability of the domain name
- **frnic:recover:** recover

### 3.2 Description of elements for creating a contact:

- Physical Person:

Element	name	Number	of	occurrences
<frnic:individuallInfos>		0-1		
<frnic:firstName>		1		

- Legal entity (PM) :

Element name	Number of occurrences
<frnic:legalEntityInfos>	1
<frnic:legalStatus s="">	1
<frnic:siren>	0-1
<frnic:VAT>	0-1
<frnic:trademark>	0-1

<frnic:asso>	0-1
<frnic:waldec>	0-1
<frnic:decl>	1 si <frnic:waldec> is absent
<frnic:publannounce="" page="">	1 si <frnic:waldec> is absent
<frnic:DUNS>	0-1
<frnic:local>	0-1

### 3.3 Description of elements for a transfer:

Element name	Number of occurrences
<frnic:domain keepDS="0">	1
<frnic:contact type="admin">	1
<frnic:contact type="tech">	1-3

### 3.4 Description of elements for a recover (frnic :recover):

Element name	Number of occurrences
<frnic:domain keepDS="0">	1
<frnic:name>	1
<frnic:authInfo>	1
<frnic:contact type="registrant">	1
<frnic:contact type="admin">	1
<frnic:contact type="tech">	1-3

The **frnic** extension can also be used optionally in the contact update to change the attributes related to the qualification process.

It is also used in response by the EPP server for **domain:check** and certain EPP notifications.

## 3.5 IDN (IDNA2008)

### Reference documents

The implementation of IDNs at Afnic is based on the IDN 2008 standard (**IDNA2008**), as per the following reference documents:

Definitions and protocol:

- RFC 5890 (08/2010 23 pages) : *Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework*
- RFC 5891 (08/2010 17 pages) : *Internationalized Domain Names in Applications (IDNA): Protocol*
- RFC 5892 (08/2010 70 pages) : *The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)*
- RFC 5894 (08/2010 43 pages) : *Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale*

Punycode encoding standard:

- RFC 3492 (03/2003 35 pages) : *Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)*

### Brief background on IDN technology

Initially, the DNS protocol was not defined to be restricted to a set of characters. It was its use and other limitations of the "time" (the protocol is 30 years old) that led to the definition of syntax rules as we know them today.

The goal of the **IDNA2008** standard is to reconcile human needs and technical constraints by allowing the use of all scripts in domain names.

All these scripts and the characters they are made up of are defined and grouped together in a standard called Unicode. Since the syntax rules for domain names impose the use of only the letters of the Latin alphabet ("a" to "z"), the numbers, the hyphen, and the dot to separate labels, a mechanism for putting Unicode domain names into canonical form and encoding them was developed in order to create names compatible with these rules.

While in applications like web browsers, Unicode names will appear, their DNS resolution will be done using the encoded form (this is normally transparent to the user who should not have to manipulate this form of domain name).

### Warning

Although the impact may seem limited, it is important to note that Afnic implements the **IDNA2008** standard, which differs slightly from the **IDNA2003** standard.

With regard to the treatment of characters supported, the German eszett (ß) is encoded, not transformed into "ss" as was the case in the previous version of the IDN standard.

In addition, the canonicalisation step (nameprep) has been deleted, which will have some impact on the use of our interfaces.

Each Afnic application is now free to apply its own rules on the matter. Besides the fact that Unicode domain names must be in Normal Form C, we have chosen to allow the input of uppercase characters. However, it is their lowercase character equivalents that will actually be taken into account by the system.

Note: the eszett is only accepted in its lowercase version.

Recommendation: it is preferable to enter lowercase characters to anticipate any changes in our IDN policy.

Example: the domain name "Thé-ou-Café.fr" is not legal according to the **IDNA2008** standard. However, we will accept it once it has been normalised as "thé-ou-café.fr".

## Vocabulary

- **Unicode**: Standard enabling any character in any form of writing to be encoded in a unique fashion ([Unicode on wikipedia](#)).
- **UTF-8**: One of the encoding formats used to encode Unicode characters.
- **ISO-8859-15**: One of the ISO 8-bit encoding standards of the Latin alphabet. Also known as Latin9.
- **LatinX**: Other names of certain ISO standards. Unlike Latin1, Latin9 includes the ligation "e in o".
- **LDH**: "LETTER-DIGIT-HYPHEN" the only ASCII characters authorised for the composition of a label in a domain name.
- **ASCII**: "American Standard Code for Information Interchange", the oldest computer standard for encoding characters. Strictly speaking 7-bit, it can only encode 128 characters.
- **ACE**: "ASCII Compatible Encoding" is the encoded version of a domain name in its LDH form (xn-caf-dma in Punycode, i.e. its "A-label form").
- **IDN**: "Internationalized Domain Name" containing characters other than ASCII characters alone.
- **Canonicalisation**: The canonical formation of a string of characters. For example, in Latin, putting a string of characters in their lowercase form is one of the operations that can be involved in a canonicalisation process.
- **Normal Form C**: Normal form requiring that the characters be (pre)composed. A character corresponds to a unique code point. This excludes characters obtained by using diacritical marks combined with base characters.
- **Code point**: Single number associated with a character.
- **Glyph**: Graphical representation of a character.
- **NAMEPREP**: Defines the version in canonical form of a Unicode domain name (was part of **IDNA2003**, no longer exists in **IDNA2008**).

- **Punycode:** Reversible and unique algorithm, used to transform a canonicalised IDN into its ACE form.

## 4 Interface connection settings

This chapter describes how the EPP server works, how to connect to it and what operations are available on the different objects (domain, contact, host).

### 4.1 EPP test bench

#### Sandbox (.tld : .fr et .re uniquement)

Sandbox server

EPP Server :

- **epp-sandbox.nic.tld**
- Port : 700
- Access authenticated by X.509 certificate submitted in advance and compliant with the registry requirements
- Number of simultaneous connections: 4
- Accounts available: 2
- Timeout : 10 minutes

#### Production (.tld : .fr, .pm, .re, .tf, .wf, .yt)

Production servers

EPP servers :

- **epp.nic.tld**
- Port : 700
- Access authenticated by X.509 certificate submitted in advance and compliant with the registry requirements
- Number of simultaneous connections: 3
- Accounts available: 1
- timeout : 10 minutes

## 4.2 Extranet

### Test bench:

<https://extranet-sandbox.nic.tld/> (tld : .fr et .re only)

- Access authenticated with login/password specific to the Extranet
- Account(s) available: 4

### Production bench:

- **[https://extranet.nic.tld](https://extranet.nic.tld/)** (.tld : .fr, .pm, .re, .tf, .wf, .yt)
- Access authenticated with login/password specific to the Extranet
- Account(s) available: 2

## 4.3 Connection and authentication to EPP server

The EPP connection is made via the standard port and the TLS protocol (**TLS1.0 and TLS1.1 are not allowed**), by presenting the X.509 client certificate installed prior, on our server.

Once you are connected, you will receive the greeting indicating what EPP extensions to use/declare when authenticating using the "login" command.

A check is made on what you declare during the login to see if it corresponds to what we accept.

Here is the list of mandatory extensions:

- **domain-1.0, contact-1.0, host-1.0, rgp-1.0, frnic-2.0.**

Here is the list of optional extensions:

- **secDNS-1.1** (si vous utilisez DNSSEC), **fee-1.0**

Omissions (in the required extensions) or additions (e.g. an extension that is not supported by the server) will cause an error and authentication failure.



## 4.4 Certificat management

In order to connect to our registry system via the EPP interface, each registrar must upload a maximum of 2 certificates per extension from their Extranet account, in the "Technical" section.

Criteria required to install an EPP certificate:

- The certificate file provided must be in **PEM format**;
- The certificate must have a title that identifies the owner. **Afnic wishes to see the name of the registrar in the subject** for consistency of authentication information;
- **The size of the RSA public key must be 2048 bits, 3072 bits or 4096 bits.** The ECDSA public key size must be 256 bits.
- The signature algorithm must be at least SHA-256;
- The certificate must be valid in terms of date (installation date between start date and end date);
- **The validity period of a self-signed certificate is a maximum of 2 years**;
- The maximum validity period is 1 year if it is not a self-signed certificate;
- **The private key must not be present in the PEM file of the certificate**;
- The certificate must have the **SSL client Role = YES**;
- The certificate must have the **Client SSL Role CA = YES if it's a self-signed certificate**;
- The complete certification chain must be provided in the PEM file, if it is not a self-signed certificate.

## 5 Session management commands

### 5.1 <greeting>

The **<greeting>** is not a command that the client can send to the EPP server but the welcome banner that the EPP server will send when the connection is made. It is also the reply that will be sent in response to a **<hello>** command (the command is discussed in the next section).

Why dwell on this banner if it is not a command? Simply because the information it provides is important and necessary, among other things, for the **<login>** command.

Although the **<greeting>** reproduced below is given as an example, and details of what it may contain can be found in the **RFC 5730**, you have to be particularly attentive to at least two elements of information, namely:

- The versions of the supported protocols (**<version>** element),
- And the languages supported (**<lang>** element).

Only one of these values will be accepted when the session starts with the **<login>** command.

## Example of <greeting> that can be sent by the AFNIC EPP server:

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <greeting>
    <svID>afnic</svID>
    <svDate>2021-12-06T20:29:24.305022Z</svDate>
    <svcMenu>
      <version>1.0</version>
      <lang>en-US</lang>
      <objURI>urn:ietf:params:xml:ns:host-1.0</objURI>
      <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>
      <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>
      <svcExtension>
        <extURI>urn:ietf:params:xml:ns:rgp-1.0</extURI>
        <extURI>urn:ietf:params:xml:ns:secDNS-1.1</extURI>
        <extURI>urn:ietf:params:xml:ns:epp:fee-1.0</extURI>
        <extURI>urn:ietf:params:xml:ns:epp:frnic-2.0</extURI>
      </svcExtension>
    </svcMenu>
    <dcP>
      <access>
        <all/>
      </access>
      <statement>
        <purpose>
          <admin/>
          <prov/>
        </purpose>
        <recipient>
          <ours/>
          <public/>
        </recipient>
        <retention>
          <stated/>
        </retention>
      </statement>
    </dcP>
  </greeting>
</epp>

```

## 5.2 <hello> command

Although it is not an EPP command in itself, this command is particularly important and useful because it will allow an EPP client to verify that the connection to the server has been properly established. Indeed, once a connection is established with the server, it is possible at any time to send this command to which the server will respond by sending the EPP welcome banner (the **<greeting>**), even if the (**<login>**) authentication phase has not yet been completed.

To the extent that the time-out mechanisms should be enabled to close the "inactive" sessions, it is quite possible to make a "heartbeat" by regularly executing this command to keep seldom used sessions open (of course, the frequency of this "heartbeat" should remain reasonable, taking into account the "time-out" and rate-limiting parameters eventually set up). For example, this command could very well be executed every 2 minutes to keep a connection open and ensure that the server is still listening, which is an acceptable frequency.

## Example of a <hello> request sent by the client

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <hello>test</hello>
</epp>
```

## 5.3 Session management commands

The EPP protocol provides two commands to establish (<login>) and end a session with the server (<logout>). Once the session is established, it will only terminate:

- On client request (<logout>)
- Or if the server, for internal reasons, has to close it ("time-out" during an idle session, technical problems, etc.)
- If the client interrupts the TCP connection (if the interruption occurs during the normal course of use of the client, it is strongly recommended to perform a <logout> before cutting the TCP connection).

Since the number of simultaneous sessions can be limited, their management must be rigorous.

## 5.4 The <login> command

The maximum duration of your EPP session, once connected, is 24 hours. After this time, it will be necessary to perform a "Logout" operation followed by a new "Login" command in order to access the EPP server again to continue performing "operations".

Without any activity following a "Login" (maintaining the connection via a "hello" for example), your session will expire after 10 minutes.

When connecting to the server, it sends a <greeting> banner to the client, thereby indicating that it is ready to receive a session start command. This command requires knowing the EPP identifier generated by AFNIC and the password associated with it. If they are properly input and the number of currently established sessions has not reached the maximum number allowed, the session must normally be established.

## Login without declaration of the EPP Fee extension

Example of request to send:

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <login>
      <clID>xxxxxxxx</clID>
      <pw>xxxxxxxx</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>
        <svcExtension>
          <extURI>urn:ietf:params:xml:ns:rgp-1.0</extURI>
          <extURI>urn:ietf:params:xml:ns:secDNS-1.1</extURI>
          <extURI>urn:ietf:params:xml:ns:launch-1.0</extURI>
          <extURI>http://www.afnic.fr/xml/epp/frnic-2.0</extURI>
        </svcExtension>
      </svcs>
    </login>
    <clTRID>xxxxxxx</clTRID>
  </command>
</epp>

```

Server response:

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>xxxxxxx</clTRID>
      <svTRID>EPP-5e356870-1629-45ec-88ba-6b69f73c5359</svTRID>
    </trID>
  </response>
</epp>

```

## 5.5 Strict authentication

For any command after the login, a strict check is made to ensure the EPP extensions (XML namespaces) used or defined have been declared prior by the client during the login.

If a new extension appears in a command, this command will be rejected.

This means that you must at least explicitly announce:

- The **frnic-2.0** extension for operations on contacts and certain operations on domain names such as **transfer**, **recover**, etc.
- The **rgp-0** extension in order to restore a domain name and display grace period information via a **domain :info**.
- And possibly the **secDNS-1** extension if you want to manage DNSSEC.

Furthermore, a strict check is made to ensure that the EPP extensions chosen by the client at the time of authentication are among the EPP extensions announced by the server. The presence of any other "exotic" extension will result in a failed authentication, as will the absence of any mandatory extension.

### The <logout> command

As we have already indicated, a client wishing to manage EPP sessions must send a session end command **<logout>** (and, ideally, wait for the response from the server) before switching off the TCP connection with the server.

Although the server is able to detect "wild" disconnections from EPP clients, this type of disconnection may not release the limited resources allocated to each registrar as quickly as they want.

To be absolutely clear, if, for example, we only allow N concurrent sessions per registrar on the EPP server, and they are all used at a given time, disconnecting a client without a **<logout>** phase could have the effect of not taking this disconnection immediately into account.

## Logout

Example of request to send:

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <logout/>
    <clTRID>123456</clTRID>
  </command>
</epp>
```

Server response:

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1500">
      <msg>Command completed successfully; ending session</msg>
    </result>
    <trID>
      <clTRID>xxxxxx</clTRID>
      <svTRID>EPP-bb2ac96d-497d-43bb-ba21-d3d7cbc82d87</svTRID>
    </trID>
  </response>
</epp>
```

## 6 Contact management

### 6.1 Description

The contact is an object linked to a domain name. It is defined at registry level by a nic-handle (or roid) in the form of a sequence of letters, numbers and the suffix "-FRNIC".

The ID of the contact is the nic-handle prefix, i.e. the front part of "-FRNIC".

The operations allowed on contact objects are as follows:

- **CREATE**
- **UPDATE**
- **INFO**
- **DELETE**

### 6.2 Contacts settings

Minimum/maximum size of the name: 1 to 128 characters

Minimum size/maximum of the first name (case of individual (PP) contact type):  
1 to 64 characters

Minimum/maximum size of the organisation: 1 to 128 characters

Minimum/maximum size of the street: 3 fields from 1 to 255 characters

Minimum/maximum size of the city: 1 to 64 characters

Minimum/maximum size of the state/region (optional): 1 to 255 characters

Minimum/maximum size of the postal code: 1 to 16 characters

Minimum/maximum size of the country code: 2 characters only

Maximum size of the telephone number: 17 characters (including + and .)

Maximum size of the fax number (optional): 17 characters (including + and .)

Minimum/maximum size of the email address: 6 to 254 characters (64 before @,  
191 after @)

## 6.3 contact:create - Creating a contact

The contact:create operation helps create contacts that will serve as registrant, administrative contact and technical contact for your domain names.

On .fr, contacts are of two types:

- Individual (PP)
- Legal entity (PM)

We will have to use the EPP frnic extension to differentiate between the creation of a PP and a PM.

Attention: For a PP contact, during a contact:create, restricted distribution is "activated" by default in order to respect the anonymization of the data.

Here are the elements of the extension used for "Individuals":

Element name	Number of occurrences
<frnic:firstName>	1
<frnic:firstName>	First Name of the contact (the <contact:name> element will be the family name).



## Contact create PP type with the minimum information required

Example of request to send:

```
<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>XXX000</contact:id>
        <contact:postalInfo type="loc">
          <contact:name>Dupont</contact:name>
          <contact:addr>
            <contact:street>1 Rue des fleurs</contact:street>
            <contact:city>Paris</contact:city>
            <contact:pc>75000</contact:pc>
            <contact:cc>FR</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice>+33.1234567890</contact:voice>
        <contact:email>test@test.fr</contact:email>
        <contact:authInfo>
          <contact:pw>Afn-12345678</contact:pw>
        </contact:authInfo>
      </contact:create>
    </create>
    <extension>
      <frnic:ext xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-
2.0">
        <frnic:create>
          <frnic:contact>
            <frnic:firstName>Michel</frnic:firstName>
          </frnic:contact>
        </frnic:create>
      </frnic:ext>
    </extension>
    <clTRID>TEST-AUTO-1638781382531-navozepuri</clTRID>
  </command>
</epp>
```

Server response :

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <contact:creData>
        <contact:id>CTC6948</contact:id>
        <contact:crDate>2021-12-
06T09:03:08.708867Z</contact:crDate>
      </contact:creData>
    </resData>
    <trID>
      <clTRID>TEST-AUTO-1638781382531-navozepuri</clTRID>
      <svTRID>EPP-621a932b-cb0c-4051-a9e2-d839953372e3</svTRID>
    </trID>
  </response>
</epp>
```

## PP type contact create with "eligibility" and "reachable status

Example of request to send:

```
<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>XXX000</contact:id>
        <contact:postalInfo type="loc">
          <contact:name>Dupont</contact:name>
          <contact:addr>
            <contact:street>1 rue des fleurs</contact:street>
            <contact:city>Paris</contact:city>
            <contact:pc>75000</contact:pc>
            <contact:cc>FR</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice>+33.1234567890</contact:voice>
        <contact:email>test@test.fr</contact:email>
        <contact:authInfo>
          <contact:pw>Afn-123456</contact:pw>
        </contact:authInfo>
      </contact:create>
    </create>
    <extension>
      <frnic:ext xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-2.0">
        <frnic:create>
          <frnic:contact>
            <frnic:individualInfos>
              <frnic:idStatus>ok</frnic:idStatus>
            </frnic:individualInfos>
            <frnic:firstName>Michel</frnic:firstName>
            <frnic:reachable media="email">1</frnic:reachable>
          </frnic:contact>
        </frnic:create>
      </frnic:ext>
    </extension>
    <clTRID>TEST-AUTO-1638783583615-jagepotubi</clTRID>
  </command>
</epp>
```

Server response:

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <contact:creData>
        <contact:id>CTC6959</contact:id>
        <contact:crDate>2021-12-
06T09:39:48.560771Z</contact:crDate>
      </contact:creData>
    </resData>
    <trID>
      <clTRID>TEST-AUTO-1638783583615-jagepotubi</clTRID>
      <svTRID>EPP-aa90d2b6-43c9-454a-9817-655457b53f7e</svTRID>
    </trID>
  </response>
</epp>

```

Here are the elements of the extension needed for “Legal entities”:

Element name	Number of occurrences
<frnic:legalStatus s="">	1
<frnic:siren>	0-1
<frnic:VAT>	0-1
<frnic:trademark>	0-1
<frnic:asso>	0-1
<frnic:waldec>	0-1
<frnic:decl>	1 si <frnic:waldec> is absent
<frnic:publ announce="" page="">	1 si <frnic:waldec> is absent
<frnic:DUNS>	0-1
<frnic:local>	0-1

<frnic:legalEntityInfos>: This element is mandatory to create a “Legal Entity” type contact and must include at least the sub-element <frnic:legalStatus s=""> described below.

List of sub-elements of <frnic:legalEntityInfos>:

- <frnic:legalStatus s="">: This element is used to indicate through the attribute "s", the business name of the entity to be identified (“company”, “partnership”, “other”). The element is empty except in the case where the attribute "s" is equivalent to “other”.

## TECHNICAL SPECIFICATIONS OF THE .FR AND FRENCH OVERSEAS CCTLDS REGISTRY SYSTEM

- <frnic:siren>: contains the SIREN number.
- <frnic:VAT>: contains the intra-Community VAT number.
- <frnic:trademark>: contains the trademark number.
- <frnic:asso>: comprises one or more sub-elements to identify the association:
- <frnic:waldec>: contains the WALDEC number,
- <frnic:decl>: contains the date of declaration to the prefecture,
- <frnic:publ announce="" page=""> contains the date of publication in the official gazette (the "announce" attribute specifies the number of the announcement, the "page" attribute the page number of this announcement).
- <frnic:DUNS>: contains the DUNS number,
- <frnic:local>: contains a local identifier corresponding to none of those described above.

## Contact create for PM type contact with the minimum information required

Example of request to send:

```
<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>XXXXXXX</contact:id>
        <contact:postalInfo type="loc">
          <contact:name>SARL DUPONT</contact:name>
          <contact:addr>
            <contact:street>1 Rue des coquelicots</contact:street>
            <contact:city>Paris</contact:city>
            <contact:pc>75000</contact:pc>
            <contact:cc>FR</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice>+33.1234567890</contact:voice>
        <contact:email>test@test.fr</contact:email>
        <contact:authInfo>
          <contact:pw>Afn-123456</contact:pw>
        </contact:authInfo>
      </contact:create>
    </create>
    <extension>
      <frnic:ext xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-2.0">
        <frnic:create>
          <frnic:contact>
            <frnic:legalEntityInfos>
              <frnic:legalStatus s="company"/>
            </frnic:legalEntityInfos>
          </frnic:contact>
        </frnic:create>
      </frnic:ext>
    </extension>
    <clTRID>TEST-AUTO-1638783895739-lamonokuzu</clTRID>
  </command>
</epp>
```

Server response:

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <contact:creData>
        <contact:id>CTC6961</contact:id>
        <contact:crDate>2021-12-06T09:45:01.43406Z</contact:crDate>
      </contact:creData>
    </resData>
    <trID>
      <clTRID>TEST-AUTO-1638783895739-lamonokuzu</clTRID>
      <svTRID>EPP-6e3d3167-b539-484f-87e5-7330d23535b0</svTRID>
    </trID>
  </response>
</epp>
```

## Example of creating a PM type contact with a SIREN number

Example of request to send:

```

<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>XXXX0000</contact:id>
        <contact:postalInfo type="loc">
          <contact:name>SARL DUPONT SIREN</contact:name>
          <contact:addr>
            <contact:street>1 Rue des Sirenes</contact:street>
            <contact:city>Paris</contact:city>
            <contact:pc>75000</contact:pc>
            <contact:cc>FR</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice>+33.1234567890</contact:voice>
        <contact:email>test@test.fr</contact:email>
        <contact:authInfo>
          <contact:pw>Afn-123456</contact:pw>
        </contact:authInfo>
      </contact:create>
    </create>
    <extension>
      <frnic:ext xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-2.0">
        <frnic:create>
          <frnic:contact>
            <frnic:legalEntityInfos>
              <frnic:legalStatus s="company"/>
              <frnic:siren>123456789</frnic:siren>
            </frnic:legalEntityInfos>
          </frnic:contact>
        </frnic:create>
      </frnic:ext>
    </extension>
    <clTRID>TEST-AUTO-1638784532830-vixogowezi</clTRID>
  </command>
</epp>

```

Server response:

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <contact:creData>
        <contact:id>CTC6970</contact:id>
        <contact:crDate>2021-12-06T09:55:36.74962Z</contact:crDate>
      </contact:creData>
    </resData>
    <trID>
      <clTRID>TEST-AUTO-1638784532830-vixogowezi</clTRID>
      <svTRID>EPP-be226b2f-e458-49e0-a69b-2645028f0f27</svTRID>
    </trID>
  </response>
</epp>
```

## Example of creating a PM type contact with a DUNS number

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <contact:creData>
        <contact:id>CTC6983</contact:id>
        <contact:crDate>2021-12-
06T10:25:43.463078Z</contact:crDate>
      </contact:creData>
    </resData>
    <trID>
      <clTRID>TEST-AUTO-1638786340760-ruvinucuwa</clTRID>
      <svTRID>EPP-63d50edc-0c23-4ecd-8e03-32d1628390be</svTRID>
    </trID>
  </response>
</epp>

<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>XXXX</contact:id>
        <contact:postalInfo type="loc">
          <contact:name>Dupont Duns</contact:name>
          <contact:addr>
            <contact:street>1 Rue des Fleurs</contact:street>
            <contact:city>Paris</contact:city>
            <contact:pc>75000</contact:pc>
            <contact:cc>FR</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice>+33.12345678910</contact:voice>
        <contact:email>test@test.fr</contact:email>
        <contact:authInfo>
          <contact:pw>Afn-123456</contact:pw>
        </contact:authInfo>
      </contact:create>
    </create>
    <extension>
      <frnic:ext xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-
2.0">
        <frnic:create>
          <frnic:contact>
            <frnic:legalEntityInfos>
              <frnic:legalStatus s="company"/>
              <frnic:DUNS>123456789</frnic:DUNS>
            </frnic:legalEntityInfos>
          </frnic:contact>
        </frnic:create>
      </frnic:ext>
    </extension>
    <clTRID>TEST-AUTO-1638786340760-ruvinucuwa</clTRID>
  </command>
</epp>

```



**Server response :**

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <contact:creData>
        <contact:id>CTC6983</contact:id>
        <contact:crDate>2021-12-
06T10:25:43.463078Z</contact:crDate>
      </contact:creData>
    </resData>
    <trID>
      <clTRID>TEST-AUTO-1638786340760-ruvinucuwa</clTRID>
      <svTRID>EPP-63d50edc-0c23-4ecd-8e03-32d1628390be</svTRID>
    </trID>
  </response>
</epp>
```

## Example of creating a PM type contact with a trademark number

```

<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>XXXX0000</contact:id>
        <contact:postalInfo type="loc">
          <contact:name>Dupont Trademark</contact:name>
          <contact:addr>
            <contact:street>1 Rue des Fleurs</contact:street>
            <contact:city>Paris</contact:city>
            <contact:pc>75000</contact:pc>
            <contact:cc>FR</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice>+33.12345669870</contact:voice>
        <contact:email>test@test.fr</contact:email>
        <contact:authInfo>
          <contact:pw>Afn-123456</contact:pw>
        </contact:authInfo>
      </contact:create>
    </create>
    <extension>
      <frnic:ext xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-2.0">
        <frnic:create>
          <frnic:contact>
            <frnic:legalEntityInfos>
              <frnic:legalStatus s="company"/>
              <frnic:trademark>123456</frnic:trademark>
            </frnic:legalEntityInfos>
          </frnic:contact>
        </frnic:create>
      </frnic:ext>
    </extension>
    <clTRID>TEST-AUTO-1638786496078-zoxajifuko</clTRID>
  </command>
</epp>

```

### Server response:

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <contact:creData>
        <contact:id>CTC6984</contact:id>
        <contact:crDate>2021-12-06T10:28:18.641199Z</contact:crDate>
      </contact:creData>
    </resData>
    <trID>
      <clTRID>TEST-AUTO-1638786496078-zoxajifuko</clTRID>
      <svTRID>EPP-db284bc7-4108-4933-85bd-36405828cbe8</svTRID>
    </trID>
  </response>
</epp>

```

## Example of creating a PM type contact with an intra-Community VAT number

```

<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>XXXX0000</contact:id>
        <contact:postalInfo type="loc">
          <contact:name>Dupont VAT</contact:name>
          <contact:addr>
            <contact:street>1 Rue des Fleurs</contact:street>
            <contact:city>Paris</contact:city>
            <contact:pc>75000</contact:pc>
            <contact:cc>FR</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice>+33.1236549870</contact:voice>
        <contact:email>test@test.fr</contact:email>
        <contact:authInfo>
          <contact:pw>Afn-123456</contact:pw>
        </contact:authInfo>
      </contact:create>
    </create>
    <extension>
      <frnic:ext xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-
2.0">
        <frnic:create>
          <frnic:contact>
            <frnic:legalEntityInfos>
              <frnic:legalStatus s="company"/>
              <frnic:VAT>FR123456</frnic:VAT>
            </frnic:legalEntityInfos>
          </frnic:contact>
        </frnic:create>
      </frnic:ext>
    </extension>
    <clTRID>TEST-AUTO-1638786613785-vecimabalo</clTRID>
  </command>
</epp>

```

Server response:

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <contact:creData>
        <contact:id>CTC6985</contact:id>
        <contact:crDate>2021-12-06T10:30:15.997057Z</contact:crDate>
      </contact:creData>
    </resData>
    <trID>
      <clTRID>TEST-AUTO-1638786613785-vecimabalo</clTRID>
      <svTRID>EPP-ad37afbc-0e13-4775-b133-be97c09021fa</svTRID>
    </trID>
  </response>
</epp>
```

## Example of creating a PM type contact with a local identifier

```

<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>XXXX0000</contact:id>
        <contact:postalInfo type="loc">
          <contact:name>Dupont Local Id</contact:name>
          <contact:addr>
            <contact:street>1 Rue des fleurs</contact:street>
            <contact:city>Paris</contact:city>
            <contact:pc>75000</contact:pc>
            <contact:cc>FR</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice>+33.12345669870</contact:voice>
        <contact:email>test@test.fr</contact:email>
        <contact:authInfo>
          <contact:pw>Afn-123456</contact:pw>
        </contact:authInfo>
      </contact:create>
    </create>
    <extension>
      <frnic:ext xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-
2.0">
        <frnic:create>
          <frnic:contact>
            <frnic:legalEntityInfos>
              <frnic:legalStatus s="company"/>
              <frnic:local>654987</frnic:local>
            </frnic:legalEntityInfos>
          </frnic:contact>
        </frnic:create>
      </frnic:ext>
    </extension>
    <clTRID>TEST-AUTO-1638786845778-viberumozi</clTRID>
  </command>
</epp>

```

### Server response

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <contact:creData>
        <contact:id>CTC6989</contact:id>
        <contact:crDate>2021-12-
06T10:34:34.552189Z</contact:crDate>
      </contact:creData>
    </resData>
    <trID>
      <clTRID>TEST-AUTO-1638786845778-viberumozi</clTRID>
      <svTRID>EPP-662487c2-ef77-468d-ae0c-427ddacb6f39</svTRID>
    </trID>
  </response>
</epp>

```

## Example of creating a PM type contact with a WALDEC (associations only)

```

<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>XXXX0000</contact:id>
        <contact:postalInfo type="loc">
          <contact:name>Association Dupont</contact:name>
          <contact:addr>
            <contact:street>1 Rue des fleurs</contact:street>
            <contact:city>Paris</contact:city>
            <contact:pc>75000</contact:pc>
            <contact:cc>FR</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice>+33.1234567890</contact:voice>
        <contact:email>test@test.fr</contact:email>
        <contact:authInfo>
          <contact:pw>Afn-123456</contact:pw>
        </contact:authInfo>
      </contact:create>
    </create>
    <extension>
      <frnic:ext xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-
2.0">
        <frnic:create>
          <frnic:contact>
            <frnic:legalEntityInfos>
              <frnic:legalStatus s="association"/>
              <frnic:asso>
                <frnic:waldec>w123456</frnic:waldec>
              </frnic:asso>
            </frnic:legalEntityInfos>
          </frnic:contact>
        </frnic:create>
      </frnic:ext>
    </extension>
    <clTRID>TEST-AUTO-1638787115349-ritijalero</clTRID>
  </command>
</epp>

```

Server response :

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <contact:creData>
        <contact:id>CTC6992</contact:id>
        <contact:crDate>2021-12-
06T10:38:38.883354Z</contact:crDate>
      </contact:creData>
    </resData>
    <trID>
      <clTRID>TEST-AUTO-1638787115349-ritijalero</clTRID>
      <svTRID>EPP-f7fb8fe1-f69a-4810-98a3-cbb7c22f7c90</svTRID>
    </trID>
  </response>
</epp>
```

## Example of creating a PM type of contact with the reporting information to the prefecture and the Official Gazette (associations only)

```

<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>XXXX0000</contact:id>
        <contact:postalInfo type="loc">
          <contact:name>Dupont JO</contact:name>
          <contact:addr>
            <contact:street>1 Rue des Fleurs</contact:street>
            <contact:city>Paris</contact:city>
            <contact:pc>75000</contact:pc>
            <contact:cc>FR</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice>+33.1234567890</contact:voice>
        <contact:email>test@test.fr</contact:email>
        <contact:authInfo>
          <contact:pw>Afn-123456</contact:pw>
        </contact:authInfo>
      </contact:create>
    </create>
    <extension>
      <frnic:ext xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-
2.0">
        <frnic:create>
          <frnic:contact>
            <frnic:legalEntityInfos>
              <frnic:legalStatus s="association"/>
              <frnic:asso>
                <frnic:decl>2011-05-02</frnic:decl>
                <frnic:publ announce="123456"
page="15">2011-05-07</frnic:publ>
              </frnic:asso>
            </frnic:legalEntityInfos>
          </frnic:contact>
        </frnic:create>
      </frnic:ext>
    </extension>
    <clTRID>TEST-AUTO-1638788196227-vovuwaxixe</clTRID>
  </command>
</epp>

```



Server response :

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <contact:creData>
        <contact:id>CTC6997</contact:id>
        <contact:crDate>2021-12-
06T10:57:05.194086Z</contact:crDate>
      </contact:creData>
    </resData>
    <trID>
      <clTRID>TEST-AUTO-1638788196227-vovuwaxixe</clTRID>
      <svTRID>EPP-d9a8980a-1eda-412d-9425-bf3cc8236a64</svTRID>
    </trID>
  </response>
</epp>

```

### Specifics of the .fr for the management of postal addresses:

Contrary to what is indicated in **RFC 5731**:

- Only one element of the **<contact:postalInfo>** type can be provided
- Only the "loc" type for postal addresses is accepted.

Contrary to what is indicated in RFC 5733:

- The **<contact:id>** element, although mandatory, is not taken into account by our server. This implies that, contrary to the EPP standard, the registrar cannot choose the identifier for the contact whose creation is requested. AFNIC allocates contact identifiers using its own algorithms. Of course, if the creation is successful, the identifier is indicated in the server response.
- The **<contact:authInfo>** element, although mandatory, is not taken into account because it is not used. It is not possible to associate a password per contact object, but like **<contact:id>**, we have chosen to keep it in the request sent to ensure simpler compatibility with existing client codes.
- The **<contact:disclose>**, element which is optional in the mapping contact, is not processed either. So it should not be present, otherwise the command will return an error message.

## 6.4 contact:update - Updating a contact

The contact:update operation is used to update certain attributes of a contact.

Only the registrar to which this contact is attached can request a change in it. The authentication mechanism via **<contact:authInfo>** has not been set up to manage contacts.

The information contained in the **<frnic:individualInfos>** and **<frnic:legalEntityInfos>** elements cannot be changed. The surname and first name of the contacts cannot be changed.

### Example of contact change with a change in the phone number and the email address

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
<command>
<update>
<contact:update xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
<contact:id>CTC6948</contact:id>
<contact:chg>
<contact:voice>+33.7894561230</contact:voice>
<contact:email>test2@test.fr</contact:email>
</contact:chg>
</contact:update>
</update>
<clTRID>TEST-AUTO-1638788549956-semepajeni</clTRID>
</command>
</epp>
```

#### Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
<response>
<result code="1000">
<msg>Command completed successfully</msg>
</result>
<trID>
<clTRID>TEST-AUTO-1638788549956-semepajeni</clTRID>
<svTRID>EPP-04499bef-b628-4391-9326-8a9722192662</svTRID>
</trID>
</response>
</epp>
```

## Example of a change in contact with a change in the postal address

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <update>
      <contact:update xmlns:contact="urn:ietf:params:xml:ns:contact-
1.0">
        <contact:id>CTC6948</contact:id>
        <contact:chg>
          <contact:postalInfo type="loc">
            <contact:addr>
              <contact:street>2 Rue des
fleurs</contact:street>
              <contact:city>Paris</contact:city>
              <contact:pc>75000</contact:pc>
              <contact:cc>FR</contact:cc>
            </contact:addr>
          </contact:postalInfo>
        </contact:chg>
      </contact:update>
    </update>
    <clTRID>TEST-AUTO-1638789026985-tokubaluko</clTRID>
  </command>
</epp>

```

### Server response

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>TEST-AUTO-1638789026985-tokubaluko</clTRID>
      <svTRID>EPP-53aaed2c-66a3-4780-afff-87678feaa367</svTRID>
    </trID>
  </response>
</epp>

```

## Example of a change in contact to remove the restricted publication

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <update>
      <contact:update xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
        <contact:id>CTC6948</contact:id>
        <contact:chg>
          </contact:chg>
        </contact:update>
      </update>
      <extension>
        <frnic:ext xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-2.0">
          <frnic:update>
            <frnic:contact>
              <frnic:restrictedPublication>0</frnic:restrictedPublication>
            </frnic:contact>
          </frnic:update>
        </frnic:ext>
      </extension>
      <clTRID>123456</clTRID>
    </command>
  </epp>
```

### Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>123456</clTRID>
      <svTRID>EPP-7d31768b-beb0-48de-8249-3b2a3ae67f8a</svTRID>
    </trID>
  </response>
</epp>
```

## Example of a change in contact to apply restricted publication

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <update>
      <contact:update xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
        <contact:id>CTC6948</contact:id>
        <contact:chg>
        </contact:chg>
      </contact:update>
    </update>
    <extension>
      <frnic:ext xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-2.0">
        <frnic:update>
          <frnic:contact>
            <frnic:restrictedPublication>1</frnic:restrictedPublication>
          </frnic:contact>
        </frnic:update>
      </frnic:ext>
    </extension>
    <clTRID>123456</clTRID>
  </command>
</epp>
```

### Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>123456</clTRID>
      <svTRID>EPP-b3eab780-3b77-438a-867f-078d4f7a4eeb</svTRID>
    </trID>
  </response>
</epp>
```

## 6.5 contact:info - Information on a contact

The contact:info operation provides detailed information on the contact polled.

### Specifics of the .fr for the contact:info command:

Given the particular management of these objects, a number of elements are not present or have a meaning different from that described in **RFC 5733** in the response sent by the server. Here is the list:

- <contact:crID> (adapted),
- <contact:crDate> (adapted),
- <contact:upID> (deleted),
- <contact:trDate> (deleted),
- <contact:authInfo> (deleted),
- <contact:disclose> (deleted).

In addition, the frnic extension is needed to take into account the identification data.

The value of the **<contact:crID>** element is that of the registrar with whom it is currently referenced.

The value of the **<contact:crDate>** element is systematically returned but remains questionable due to the history of AFNIC contacts.

Another limitation with respect to **RFC 5733** is that only the registrar related to this contact object can request information about it.

## Example contact:info on a contact of the “Legal Entity” type

```
<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>CTC40905</contact:id>
      </contact:info>
    </info>
    <clTRID>TEST-AUTO-1638877225848-cirekizesi</clTRID>
  </command>
</epp>
```

Server response :

```

<response>
  <result code="1000">
    <msg>Command completed successfully</msg>
  </result>
  <resData>
    <contact:infData>
      <contact:id>CTC40905</contact:id>
      <contact:roid>CTC40905-AFNIC</contact:roid>
      <contact:status s="ok"/>
      <contact:postalInfo type="loc">
        <contact:name>SARL TEST</contact:name>
        <contact:addr>
          <contact:street>1 rue des fleurs</contact:street>
          <contact:city>Paris</contact:city>
          <contact:pc>75000</contact:pc>
          <contact:cc>FR</contact:cc>
        </contact:addr>
      </contact:postalInfo>
      <contact:voice>+33.1234567890</contact:voice>
      <contact:email>test@test.fr</contact:email>
      <contact:clID>IANA9996</contact:clID>
      <contact:crID>IANA9996</contact:crID>
      <contact:crDate>2021-12-07T11:41:50.832441Z</contact:crDate>
    </contact:infData>
  </resData>
  <extension>
    <frnic:ext>
      <frnic:resData>
        <frnic:infData>
          <frnic:contact>
            <frnic:legalEntityInfos>
              <frnic:idStatus source="registrar" when="2021-12-07T11:41:50.832843Z">no</frnic:idStatus>
              <frnic:legalStatus s="company"/>
            </frnic:legalEntityInfos>
            <frnic:obsoleted>0</frnic:obsoleted>
          </frnic:contact>
        </frnic:infData>
      </frnic:resData>
    </frnic:ext>
  </extension>
  <trID>
    <svTRID>EPP-a66a4ab9-5bba-40e4-886e-e6b32367d94f</svTRID>
  </trID>
</response>
</epp>

```

## Example of server response for an “Individual” type contact

```
<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>CTC7629</contact:id>
      </contact:info>
    </info>
    <clTRID>TEST-AUTO-1638877225848-cirekizesi</clTRID>
  </command>
</epp>
```



```

<response>
  <result code="1000">
    <msg>Command completed successfully</msg>
  </result>
  <resData>
    <contact:infData>
      <contact:id>CTC40906</contact:id>
      <contact:roid>CTC40906-AFNIC</contact:roid>
      <contact:status s="ok"/>
      <contact:postalInfo type="loc">
        <contact:name>SARL TEST</contact:name>
        <contact:addr>
          <contact:street>1 rue des fleurs</contact:street>
          <contact:city>Paris</contact:city>
          <contact:pc>75000</contact:pc>
          <contact:cc>FR</contact:cc>
        </contact:addr>
      </contact:postalInfo>
      <contact:voice>+33.1234567890</contact:voice>
      <contact:email>test@test.fr</contact:email>
      <contact:clID>IANA9996</contact:clID>
      <contact:crID>IANA9996</contact:crID>
      <contact:crDate>2021-12-07T11:48:30.401432Z</contact:crDate>
    </contact:infData>
  </resData>
  <extension>
    <frnic:ext>
      <frnic:resData>
        <frnic:infData>
          <frnic:contact>
            <frnic:restrictedPublication>true</frnic:restrictedPublication>
            <frnic:individualInfos>
              <frnic:idStatus source="registrar" when="2021-12-07T11:48:30.401459Z">no</frnic:idStatus>
            </frnic:individualInfos>
            <frnic:firstName>Marie</frnic:firstName>
            <frnic:obsoleted>0</frnic:obsoleted>
          </frnic:contact>
        </frnic:infData>
      </frnic:resData>
    </frnic:ext>
  </extension>
  <trID>
    <svTRID>EPP-c67df771-358b-4580-abec-78314d85efa6</svTRID>
  </trID>
</response>
</epp>

```

## 6.6 Le contact:delete – Deleting the contact

The contact operation: delete allows you to delete a contact that is not linked to any domain name.

Example of request:

```
<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>CTC6959</contact:id>
      </contact:delete>
    </delete>
    <clTRID>123456</clTRID>
  </command>
</epp>
```

Server response:

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>123456</clTRID>
      <svTRID>EPP-5bacc663-e0d0-4fb1-9f59-a6ddfbab0b15</svTRID>
    </trID>
  </response>
</epp>
```

## 7 Domain name management

### 7.1 Multi-year and expiry date calculation

#### Number of years available

Domain names can be registered and renewed for a period of 1 to 10 years.

#### Explicit renewal rules

The validity period of a .fr domain name can be from 1 to 10 years. This period may be renewed by a **RENEW** operation.

##### **Functioning of explicit RENEW:**

The **RENEW** operation renews the domain name over a maximum period of 10 years from the expiry date of the domain name.

Example **conforming** to the renewal rule for a maximum period of 10 years:

- Initial expiry date: 13/12/2016 at 12:34,
- Date of renew operation: 13/12/2016 at 12:54,
- Number of additional years requested: 10

The new expiry date requested should be 13/12/2026 at 12:34.

Example **non-conforming** to the renewal rule for a maximum period of 10 years:

- Initial expiry date: 14/12/2016
- Date of renew operation: 15/11/2015
- Number of additional years requested: 9

The **RENEW** command should renew the name until 14/12/2025. By calculating the number of years between the RENEW operation date and the new expiry date, the result is greater than 10 years. The renewal is therefore denied by the registration chain. The expiration date applied will be calculated by the system not to exceed 10 years.

Special cases of TRANSFER, RECOVER and RESTORE operations:

If a **TRANSFER**, **RECOVER** or **RESTORE** operation involves a domain name for which the validity period is over 9 years at the time of the operation, the operation will not add 1 year but will change the expiry date so that it corresponds to 10 maximum years of registration.

The **TRANSFER**, **RECOVER** and **RESTORE** operations therefore are not blocked in this specific case.

## 7.2 Grace periods

### General description

Following:

- There is a 5-day grace period for **CREATE, TRANSFER and RENEW** operations,
- **DELETE** operations have a 30-day grace period,
- **AUTORENEW** operation have a grace period (between 30 and 60 days).

If the domain name is deleted during the grace period, the operation is charged and a credit note will be issued.

If the charged operation is an explicit or automatic transfer or renewal, the expiry date is reset to its value before the operation in question.

### List, duration and description

List of grace periods activated for the .fr and the French Overseas Territories (as per **RFC 3915**):

Grace period	Duration (days)	Description
addPeriod	5	After a CREATE operation
transferPeriod	5	After a TRANSFER operation
renewPeriod	5	After a RENEW operation
redemptionPeriod	30	After a DELETE operation

### Cancellation of a grace period

When a domain name is within the grace period and it is subject to a new fee-paying operation, the grace period corresponding to the initial operation is cancelled. If the new fee-paying operation entitles the holder to a grace period, it is activated at the end of this new operation.

Specific case:

A **DELETE** operation on a domain name during a grace period following a **CREATE "addPeriod"** operation does not trigger a redemption period **"redemptionPeriod"**. The domain name immediately deleted and available again for registration.

The possible cases are summarised in the following table:

**TECHNICAL SPECIFICATIONS OF THE .FR AND FRENCH OVERSEAS CCTLDS REGISTRY SYSTEM**

Initial operation / grace period activated	Operation performed during the grace period	Cancellation of the grace period (Yes/No)	New grace period activated	Billing of initial operation	Credit note (Yes/No)
CREATE / addPeriod	TRANSFER	Yes	transferPeriod	Yes	No
CREATE / addPeriod	RECOVER	Yes	-	Yes	No
CREATE / addPeriod	RENEW	Yes	renewPeriod	Yes	No
CREATE / addPeriod	DELETE	-	-	Yes	Yes
CREATE / addPeriod	UPDATE	No	-	Yes	No
RENEW / renewPeriod	TRANSFER	Yes	transferPeriod	Yes	No
RENEW / renewPeriod	RECOVER	Yes	-	Yes	No
RENEW / renewPeriod	RENEW	Yes	renewPeriod	Yes	No
RENEW / renewPeriod	DELETE	-	redemptionPeriod	Yes	Yes
RENEW / renewPeriod	UPDATE	No	-	Yes	No
TRANSFER / transferPeriod	TRANSFER	Yes	transferPeriod	Yes	No
TRANSFER / transferPeriod	RECOVER	Yes	-	Yes	No
TRANSFER / transferPeriod	RENEW	Yes	renewPeriod	Yes	No
TRANSFER / transferPeriod	DELETE	-	redemptionPeriod	Yes	Yes
TRANSFER / transferPeriod	UPDATE	No	-	Yes	No
DELETE / redemptionPeriod	RESTORE	-	-	No	No

Initial grace period activated	Operation performed during the grace period	Cancellation of the grace period (Yes/No)	New grace period activated	Billing of initial operation	Credit note (Yes/No)
AUTORENEW / autorenewPeriod	DELETE	Yes	redemptionPeriod	No	No

## 7.3 domain:check

The domain:check operation is used to verify the availability of a domain name and determine the reasons for its unavailability.

The availability in the domain:check is immediate availability.

If you are told that the domain name is available, you can immediately register it with a domain:create command.

Unavailability does not mean that the domain cannot be registered, but it is sometimes necessary to go through the preliminary stages (see the Procedures Guide).

Availability is indicated via a boolean (0 or 1) in the response to the domain:check:

- `<domain:name avail="0">domaine-non-disponible.fr</domain:name>` means that the domain is not available.
- `<domain:name avail="1">domaine-disponible.fr</domain:name>` means that the domain is available.

This response, in the case of an unavailable domain name, contains the reason for the unavailability: domain:reason.

Here are the messages in the domain:reason element in response to the domain:check:

### Example of domain:check command

```

<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>mondomainetest3.fr</domain:name>
      </domain:check>
    </check>

    <clTRID>TEST-AUTO-1638886529376-sabelepore</clTRID>

  </command>
</epp>

```

## Server response

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-2.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <msgQ count="5" id="f859898e-baab-420c-b8e1-985714fed12c"/>
    <resData>
      <domain:chkData>
        <domain:cd>
          <domain:name
avail="1">mondomainetest3.fr</domain:name>
        </domain:cd>
      </domain:chkData>
    </resData>
    <extension>
      <frnic:ext>
        <frnic:resData>
          <frnic:chkData>
            <frnic:domain>
              <frnic:cd>
                <frnic:name forbidden="0"
reserved="0">mondomainetest3.fr</frnic:name>
              </frnic:cd>
            </frnic:domain>
          </frnic:chkData>
        </frnic:resData>
      </frnic:ext>
    </extension>
    <trID>
      <clTRID>TEST-AUTO-1638886529376-sabelepore</clTRID>
      <svTRID>EPP-62541ce9-2cc5-4a41-8eb5-45dbed51ab84</svTRID>
    </trID>
  </response>
</epp>

```

## 7.4 domain:create

The domain:create operation is used to register a domain name. It consists of the following elements:

- The domain name,
- The holder contact,
- The administrative contact,
- 1 to 3 technical contacts,
- 0 to 8 authoritative hosts,
- 1 to 10 years of registration,
- The authInfo,
- With the secDNS-1.1 extension, it can also include DS records.

## Example of the creation of a domain name

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <create>
      <domain:create xmlns:domain="urn:ietf:params:xml:ns:domain-
1.0" xsi:schemaLocation="urn:ietf:params:xml:ns:domain-
1.0.xsd">
        <domain:name>mondomainetest.fr</domain:name>
        <domain:period unit="y">1</domain:period>
        <domain:registrant>CTC6948</domain:registrant>
        <domain:contact type="admin">CTC6948</domain:contact>
        <domain:contact type="tech">CTC6961</domain:contact>
        <domain:authInfo>
          <domain:pw>Test;123456789</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <clTRID>123456 </clTRID>
  </command>
</epp>
```

## Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:creData>
        <domain:name>mondomainetest.fr</domain:name>
        <domain:crDate>2021-12-06T12:46:06.017789Z</domain:crDate>
        <domain:exDate>2022-12-06T12:46:05.98754Z</domain:exDate>
      </domain:creData>
    </resData>
    <trID>
      <clTRID>123456 </clTRID>
      <svTRID>EPP-c8b5448b-d1db-449e-8cd8-cfc1bac49c2a</svTRID>
    </trID>
  </response>
</epp>
```



## Example of the creation of a domain name with hosts

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <create>
      <domain:create xmlns:domain="urn:ietf:params:xml:ns:domain-
1.0" xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-
1.0.xsd">
        <domain:name>mondomainetest2.fr</domain:name>
        <domain:period unit="y">1</domain:period>
        <domain:ns>
          <domain:hostObj>ns1.mondomainetest.paris</domain:hostObj>
          <domain:hostObj>ns1.mondomainetest.fr</domain:hostObj>
          </domain:ns>
          <domain:registrant>CTC6948</domain:registrant>
          <domain:contact type="admin">CTC6970</domain:contact>
          <domain:contact type="tech">CTC6970</domain:contact>
          <domain:authInfo>
            <domain:pw>Test;123456789</domain:pw>
          </domain:authInfo>
        </domain:create>
      </create>
      <clTRID>123456 </clTRID>
    </command>
  </epp>

```

## Server response

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:creData>
        <domain:name>mondomainetest2.fr</domain:name>
        <domain:crDate>2021-12-06T13:57:23.940904Z</domain:crDate>
        <domain:exDate>2022-12-06T13:57:23.894113Z</domain:exDate>
      </domain:creData>
    </resData>
    <trID>
      <clTRID>123456 </clTRID>
      <svTRID>EPP-f974ef1a-39d5-4984-9172-300c34c381c2</svTRID>
    </trID>
  </response>
</epp>

```

## 7.5 domain:update

The domain:update operation is used to update the attributes of the domain name:

- Holder contact
- Administrative contact
- Technical contact(s)
- Authoritative host(s)
- DS record(s)
- authInfo
- EPP client status.

The command may include a single attribute or all the attributes. It is therefore possible to change all of your domain name attributes at the same time.

Depending on the attributes to change, here are the structures to use:

- For administrative and technical contact(s), for authoritative hosts and client status:  
Use **<domain:add>** and **<domain:rem>** (if an attribute must be replaced or removed) to indicate the new and old attributes. The order is important and you must therefore put **<domain:add>** before **<domain:rem>**, see examples below.
- For DS records:  
Use the secDNS-1.1 extension which you must have declared during login. Use the elements **<secDNS:add>** and **<secDNS:rem>** (if records must be added or removed).
- For the holder contact and authInfo:  
Use **<domain:chg>** by then providing only the final attribute(s) required.

## Example of a domain name update with the addition of authoritative host

```
<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>mondomainetest.fr</domain:name>
        <domain:add>
          <domain:ns>
            <domain:hostObj>ns1.mondomainetest.paris</domain:hostObj>
          </domain:ns>
        </domain:add>
      </domain:update>
    </update>
    <clTRID>123456</clTRID>
  </command>
</epp>
```

### Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <extension/>
    <trID>
      <clTRID>123456</clTRID>
      <svTRID>EPP-2b9457b3-0d2a-4fb5-b655-6ce7acaa08b2</svTRID>
    </trID>
  </response>
</epp>
```

## Example of a domain name update with a change in holder and technical contact

```
<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>mondomainetest.fr</domain:name>
        <domain:add>
          <domain:contact type="tech">CTC6985</domain:contact>
        </domain:add>
        <domain:rem>
          <domain:contact type="tech">CTC6961</domain:contact>
        </domain:rem>
        <domain:chg>
          <domain:registrant>CTC6970</domain:registrant>
        </domain:chg>
      </domain:update>
    </update>
    <clTRID>123456</clTRID>
  </command>
</epp>
```

### Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <extension/>
    <trID>
      <clTRID>123456</clTRID>
      <svTRID>EPP-8bb71e28-0097-488c-8d8b-812e9d2694f4</svTRID>
    </trID>
  </response>
</epp>
```

## 7.6 domain:info

The domain:info operation retrieves information about a domain name.

If you are not the registrar in charge of the domain name in question, you must provide the authInfo to perform the domain:info information operation.

The response that the server returns does not contain all the elements described in **RFC 5731**:

- The first notable difference is the **<domain:roid>** element: Although we have unique identifiers for domain names in our database, they do not quite meet the "specifications" defined in RFC. A "roid" should be created for each creation of object in the database; a domain name once created, deleted and re-created should, logically, be assigned different "roids" for each create operation.

At AFNIC, a unique ID is associated with a domain name when it is inserted for the first time into the database. It follows it, even if it is deleted in the meantime (it is never re-assigned).

To this unique ID we concatenate the "-FRNIC" suffix, as we do for all contact objects.

- The status of a domain name can be specified either in the **<resData>** part of the response or in the extensions. However, unlike the RFC, this information is not optional. A domain name that does not have a specific status necessarily has the **<domain:status s="ok"/>** element present in the **<resData>** part of the response. The particular statuses are prohibitions (\*Prohibited\*, \*Hold), pending operations (pending\*) and the absence of host (inactive). Similarly, the absence of this element necessarily implies that information on the status of the domain name is in the **<extension>** part of the response.
- The elements **<domain:crID>** (the ID of the registrar that created the domain name the first time), **<domain:upID>** (the ID of the registrar that last updated the domain name) and **<domain:trDate>** (the date of the last completed transfer) are not present.

### Example of the domain:info command for a domain name belonging to the registrar's portfolio

```
<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>mondomainetest.fr</domain:name>
      </domain:info>
    </info>
    <clTRID>123456 </clTRID>
  </command>
</epp>
```

Server response :

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <msgQ count="79" id="8593de55-de7b-41a3-9d0a-c2ef2626ff48"/>
    <resData>
      <domain:infData>
        <domain:name>mondomainetest.fr</domain:name>
        <domain:roid>DOM1213-FRNIC</domain:roid>
        <domain:status s="inactive"/>
        <domain:registrant>CTC6948</domain:registrant>
        <domain:contact type="admin">CTC6948</domain:contact>
        <domain:contact type="tech">CTC6961</domain:contact>
        <domain:host>ns1.mondomainetest.fr</domain:host>
        <domain:clID>IANA9998</domain:clID>
        <domain:crID>IANA9998</domain:crID>
        <domain:crDate>2021-12-06T12:46:06.017789Z</domain:crDate>
        <domain:upID>IANA9998</domain:upID>
        <domain:upDate>2021-12-06T12:52:15.239325Z</domain:upDate>
        <domain:exDate>2022-12-06T12:46:05.98754Z</domain:exDate>
        <domain:authInfo>
          <domain:pw>Test;123456789</domain:pw>
        </domain:authInfo>
      </domain:infData>
    </resData>
    <extension>
      <secDNS:infData/>
    </extension>
    <trID>
      <clTRID>123456 </clTRID>
      <svTRID>EPP-617c39a7-5d8e-46d8-b035-aafdea590163</svTRID>
    </trID>
  </response>
</epp>

```

**Example of the domain:info command for a domain name not belonging to the registrar’s portfolio with the auth info**

```

<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>tondomainetest.fr</domain:name>
      </domain:info>
    </info>
    <clTRID>123456 </clTRID>
  </command>
</epp>

```

## Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2201">
      <msg>Authorization error</msg>
      <extValue>
        <reason>The registrar is not allowed to do this
operation</reason>
      </extValue>
    </result>
    <trID>
      <clTRID>123456 </clTRID>
      <svTRID>EPP-ef508c74-93bf-416c-b0e2-a80db14c38e6</svTRID>
    </trID>
  </response>
</epp>
```

## 7.7 domain:transfer

The domain:transfer operation transfers a domain name to your portfolio.

### Make a transfer

Specifics of the .fr for the domain:transfer command:

- The **<domain:pw>** element (sub-element of **<domain:authInfo>**) cannot be used with the "roid" attribute to indicate that the **authInfo** provided is linked to the holder or a contact associated with the domain name instead of the domain name itself. For the .fr, the **authInfo** can be linked only to the domain name.
- During the **TRANSFER**, the holder contact data is cloned in the incoming registrar (unless a "contact" object with exactly the same information already exists with this registrar). Be careful: the cloning occurs after the **transfer** operation has been completed.
- In RFC 5731, section 3.2.4, the **TRANSFER** command may include the optional **<domain:period>** element. Given that the transfer operation adds no more than one year of registration to the domain name, our EPP server will ignore the **<domain:period>** element provided and will not return an error message. Since the change is validated only at the end of the operation, the new expiry date is applied to the domain name at that time. As a result, the **<domain:exDate>** element is absent from the server response.
- The frnic extension is necessary to:
  - associate with the transferred domain name the technical and administrative contacts related to the registrar that carried out the **transfer**,
  - specify whether any DNSSEC configuration present (DS records) should be kept in case the **transfer** is successful.

Here are the specific elements found in the XML query sent by the EPP client:

Element name	Number of occurrences
<domain:pw>	1
<frnic:domain keepDS="0">	1
<frnic:contact type="admin">	1
<frnic:contact type="tech">	1-3

The keepDS element is an XML boolean and must be a value of either: 0, 1, true, false.

Its presence is mandatory.

The domain name has "pendingTransfer" status enabled.

## Example of a transfer operation command

```

<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>auto-bar-211206-batarang-jaune-de-la-
terre.fr</domain:name>
        <domain:period unit="y">1</domain:period>
        <domain:authInfo>
          <domain:pw>Test123456789</domain:pw>
        </domain:authInfo>
      </domain:transfer>
    </transfer>
    <extension>
      <frnic:ext xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-
2.0">
        <frnic:transfer>
          <frnic:domain keepDS="1">
            <frnic:contact
type="admin">CTC6970</frnic:contact>
            <frnic:contact type="tech">CTC6970</frnic:contact>
          </frnic:domain>
        </frnic:transfer>
      </frnic:ext>
    </extension>
    <clTRID>123456</clTRID>
  </command>
</epp>

```



## Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1001">
      <msg>Command completed successfully; action pending</msg>
    </result>
    <resData>
      <domain:trnData>
        <domain:name>auto-bar-211206-batarang-jaune-de-la-
terre.fr</domain:name>
        <domain:trStatus>pending</domain:trStatus>
        <domain:reID>IANA9998</domain:reID>
        <domain:reDate>2021-12-06T20:49:00.206057Z</domain:reDate>
        <domain:acID>IANA9996</domain:acID>
        <domain:acDate>2021-12-06T20:51:00.205653Z</domain:acDate>
        <domain:exDate>2024-12-06T12:19:27.801242Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>123456</clTRID>
      <svTRID>EPP-6527dc9e-7560-45eb-86d1-62c752acdb8f</svTRID>
    </trID>
  </response>
</epp>
```

## Approve or reject a TRANSFER (only for the outgoing registrar)

When the transfer is initiated by a registrar, the current domain name registrar may approve the **TRANSFER** or reject it:

- Approving the **TRANSFER** will immediately validate the operation.
- Rejecting the **TRANSFER** will extend the operation deadline to 22 days (from the date of the operation).

For more details, please refer to the *Procedures Guide*.

## Example of the approval of a transfer operation

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <transfer op="approve">
      <domain:transfer xmlns:domain="urn:ietf:params:xml:ns:domain-
1.0">
        <domain:name>auto-bar-211207-oreiller-casse-des-
collines.fr</domain:name>
        <domain:authInfo>
          <domain:pw>Test123456789</domain:pw>
        </domain:authInfo>
      </domain:transfer>
    </transfer>
    <clTRID>$clTRID </clTRID>
  </command>
</epp>
```

Server response :

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <msgQ count="116" id="5b973451-2df4-4968-909c-5f08cbf37c58"/>
    <resData>
      <domain:trnData>
        <domain:name>auto-bar-211207-oreiller-casse-des-
collines.fr</domain:name>
        <domain:trStatus>clientApproved</domain:trStatus>
        <domain:reID>IANA9996</domain:reID>
        <domain:reDate>2021-12-07T09:59:46.014745Z</domain:reDate>
        <domain:acID>IANA9995</domain:acID>
        <domain:acDate>2021-12-07T10:00:33.572793Z</domain:acDate>
        <domain:exDate>2023-12-07T09:54:31.63089Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>$clTRID </clTRID>
      <svTRID>EPP-4eec3fa6-b453-458e-b318-296d3613be1b</svTRID>
    </trID>
  </response>
</epp>

```

Example of the rejection of a transfer operation

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <transfer op="reject">
      <domain:transfer xmlns:domain="urn:ietf:params:xml:ns:domain-
1.0">
        <domain:name>auto-bar-211206-batarang-jaune-de-la-
terre.fr</domain:name>
        <domain:authInfo>
          <domain:pw>Test123456789</domain:pw>
        </domain:authInfo>
      </domain:transfer>
    </transfer>
    <clTRID>123456</clTRID>
  </command>
</epp>

```

## Server response

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <msgQ count="80" id="8593de55-de7b-41a3-9d0a-c2ef2626ff48"/>
    <resData>
      <domain:trnData>
        <domain:name>auto-bar-211206-batarang-jaune-de-la-
terre.fr</domain:name>
        <domain:trStatus>clientRejected</domain:trStatus>
        <domain:reID>IANA9998</domain:reID>
        <domain:reDate>2021-12-06T20:49:00.206057Z</domain:reDate>
        <domain:acID>IANA9996</domain:acID>
        <domain:acDate>2021-12-06T20:54:00.206057Z</domain:acDate>
        <domain:exDate>2024-12-06T12:19:27.801242Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>123456</clTRID>
      <svTRID>EPP-2a0dc06e-8096-4a35-8794-d2f0fadbbd05</svTRID>
    </trID>
  </response>
</epp>

```

## Cancel the TRANSFER (only for the incoming registrar)

It is possible, while the transfer is in progress, to cancel it with the domain:transfer {cancel} command.

### Example of the cancellation of a transfer operation

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <transfer op="cancel">
      <domain:transfer xmlns:domain="urn:ietf:params:xml:ns:domain-
1.0">
        <domain:name>auto-bar-211207-ancien-oreiller-unique-du-
desespoir.fr</domain:name>
        <domain:authInfo>
          <domain:pw>Test123456789</domain:pw>
        </domain:authInfo>
      </domain:transfer>
    </transfer>
    <clTRID>123456 </clTRID>
  </command>
</epp>

```

## Server response

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <msgQ count="4" id="c9aecaca-ec3e-415a-9409-33063ea9f3ea"/>
    <resData>
      <domain:trnData>
        <domain:name>auto-bar-211207-ancien-oreiller-unique-du-
desespoir.fr</domain:name>
        <domain:trStatus>clientCancelled</domain:trStatus>
        <domain:reID>IANA9998</domain:reID>
        <domain:reDate>2021-12-07T10:12:44.694646Z</domain:reDate>
        <domain:acID>IANA9996</domain:acID>
        <domain:acDate>2021-12-07T10:13:37.293912Z</domain:acDate>
        <domain:exDate>2023-12-07T08:18:55.7485Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>123456 </clTRID>
      <svTRID>EPP-5ca1968c-89d6-4647-afe4-a1abad2e1498</svTRID>
    </trID>
  </response>
</epp>

```

## 7.8 domain:recover

The recovery operation allows for a "forced" change of registrant and can also allow for a change of registrar in the same operation.

This operation is used for certain procedures.

For more details, please refer to the Procedures Guide.

This operation requires the provision of :

- The inseparable triplet "domain name / holder contact id / authorization code" as well as the new administrative and technical contact(s).
- The <frnic:keepDS> element (DNSSEC configuration)

## Example of a forced changed of registrant

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <extension>
    <frnic:ext xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-2.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
      <frnic:command>
        <frnic:recover op="request">
          <frnic:domain keepDS="1">
            <frnic:name>mondomaine.fr</frnic:name>
            <frnic:authInfo>
              <domain:pw>codederecover</domain:pw>
            </frnic:authInfo>
            <frnic:registrant>CTC64742</frnic:registrant>
            <frnic:contact
type="admin">CTC64742</frnic:contact>
              <frnic:contact
type="tech">CTC64742</frnic:contact>
            </frnic:domain>
          </frnic:recover>
          <frnic:clTRID>TEST-AUTO-1664371433735-
lakutakunu</frnic:clTRID>
        </frnic:command>
      </frnic:ext>
    </extension>
  </epp>
```

### Server response :

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-2.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <msgQ count="2" id="20aa32b8-a8c0-419c-80a7-ccac69445e51"/>
    <extension>
      <frnic:ext>
        <frnic:resData>
          <frnic:recData>
            <frnic:domain>
              <frnic:name>mondomaine.fr</frnic:name>
              <frnic:reID>IANA9995</frnic:reID>
              <frnic:reDate>2022-09-
28T13:23:53.876602Z</frnic:reDate>
              <frnic:reHldID>CTC64742</frnic:reHldID>
              <frnic:acID>xxxx</frnic:acID>
              <frnic:acHldID>CTC64741</frnic:acHldID>
            </frnic:domain>
          </frnic:recData>
        </frnic:resData>
      </frnic:ext>
    </extension>
    <trID>
      <clTRID>TEST-AUTO-1664371433735-lakutakunu</clTRID>
      <svTRID>EPP-a794cb24-0dad-4e26-a364-9f476aa3c279</svTRID>
    </trID>
  </response>
</epp>
```

## 7.9 domain:renew - RENEW

The domain:renew operation allows you to renew a domain name for 1 to 10 years of the registration.

See section ==> Rule on maximum registration period.

The command requires the provision of:

- The domain name,
- The requisite number of additional years of registration,
- The current expiry date.

Note that we only accept the additional registration time in years.

### Example of an explicit renew operation

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <renew>
      <domain:renew xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>mondomainetest.fr</domain:name>
        <domain:curExpDate>2022-12-06</domain:curExpDate>
        <domain:period unit="y">1</domain:period>
      </domain:renew>
    </renew>
    <clTRID>123456</clTRID>
  </command>
</epp>
```

### Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:renData>
        <domain:name>mondomainetest.fr</domain:name>
        <domain:exDate>2023-12-06T12:46:05.98754Z</domain:exDate>
      </domain:renData>
    </resData>
    <trID>
      <clTRID>123456</clTRID>
      <svTRID>EPP-d6f4f4f0-90ba-49e5-80b5-a12ce22e4017</svTRID>
    </trID>
  </response>
</epp>
```

## 7.10 domain:delete - DELETE

The domain:delete operation is used to delete a domain name. The latter then enters a redemption period for 30 days.

A domain:delete operation during an "addPeriod" does not trigger a redemption period; the domain is completely deleted and available again for registration.

For more details, please refer to the Procedures Guide.

The domain name has "pendingTransfer" status enabled. This status can be viewed using the domain:info command.

The RGP extension also indicates that the domain nm is in redemption period: "redemptionPeriod".

A notification is generated in your notification queue when the domain name is actually deleted after 30 days of redemption.

### Example of deleting a domain:

```
<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>mondomainetest2.fr</domain:name>
      </domain:delete>
    </delete>
    <clTRID>123456</clTRID>
  </command>
</epp>
```

The server response after deletion of a domain that is not in addPeriod

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1001">
      <msg>Command completed successfully; action pending</msg>
    </result>
    <trID>
      <clTRID>123456</clTRID>
      <svTRID>EPP-9999f59e-77dd-42fa-9772-12c2b91ff827</svTRID>
    </trID>
  </response>
</epp>
```

The server response after deletion of a domain that is in addPeriod

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>123456</clTRID>
      <svTRID>EPP-e37ac9d1-c24f-4763-a5d3-59209b2b5226</svTRID>
    </trID>
  </response>
</epp>
```

### **Recommendation :**

Before performing a domain:delete command, we recommend that you perform a domain:info command to identify the list of hosts that depend on this domain name.

In fact, if the domain name to be deleted is used in the name of the host object and this host object is used by other domain names, your domain:delete command will not be successful.

You must first rename the host(s) using the domain name that you want to delete.

## **7.11 rgp:restore - RESTORE**

The rgp:restore operation as part of a domain:update restores a domain name that is in a redemption period (following a deletion).

### **Specifics of the .fr:**

Unlike the restoration in two phases that is practised for certain TLDs, for the .fr, the restoration is immediate and it is not necessary to send a report.

A notification is generated in your notification queue when the domain name has been restored to indicate that deletion has been aborted.

Attention: the rgp:restore operation as part of a domain:update does not allow any other parameter of the domain name to be modified. Any command containing another modified parameter will result in the failure of the command.

For example, we cannot restore and change the name servers in the same update command.



## Example of a restore operation

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <update>
      <domain:update xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
        <domain:name>mondomaine.fr</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>
    <clTRID>123456 </clTRID>
  </command>
</epp>
```

## Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <msgQ count="4" id="bc3621c8-1f47-4401-b632-50a174a04c0e"/>
    <trID>
      <clTRID>123456 </clTRID>
      <svTRID>EPP-65b23a44-93e6-4afa-875b-06e44b68dca6</svTRID>
    </trID>
  </response>
</epp>
```

## 8 Host management

### 8.1 Description

The EPP server is compliant with **RFC 5732** for all syntactic constraints, attributes, and the number of occurrences of each attribute.

The operations allowed on host objects are as follows:

- **CHECK,**
- **CREATE,**
- **UPDATE,**
- **INFO,**
- **DELETE.**

**TRANSFER** and **RENEW** operations are not supported.

### 8.2 Definitions:

Sponsor of a host object: the sponsor of a host object is a registrar that has created the host and that has the ability to perform the **UPDATE and DELETE** operations on this host object.

Specific case: In the case of domain names configured with glue hosts. If the domain is transferred, the glue host is also transferred. The incoming registrar becomes the sponsor of the host.

Internal host: A host that is part of the interface zone to which you are connected:

- Example: A host called ns.example.fr is internal to the .fr zone.
- Counter example : A host called ns.example.re is external to the .fr zone

External host : it's a host that is not part of the zone of the interface on which you are connected:

- Example: A host called ns.example.com is external to the zones .fr, .re, .yt, .tf, .pm, .wf.
- Example : A host called ns.example.re is external to the .fr zone

### 8.3 host:check

The **<host:check>** command is used to know the availability of a host. It is useful in particular before initiating a host object create command.

Availability is indicated via a boolean (0 or 1) in the response to the host:check.

This response, in the case of an unavailable host, contains the reason for the unavailability: (<host:reason>In use</host:reason>).

This command accepts up to seven hosts as input.

The command consists of only one element:

- The name of the host object: **<host:name>**.

## Example of a check command on 2 host objects

```
<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>ns1.mondomainetest.paris</host:name>
        <host:name>ns2.mondomainetest.fr</host:name>
      </host:check>
    </check>
    <clTRID>123456 </clTRID>
  </command>
</epp>
```

## Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:host="urn:ietf:params:xml:ns:host-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <host:chkData>
        <host:cd>
          <host:name
            avail="0">ns1.mondomainetest.paris</host:name>
          <host:reason>In use</host:reason>
        </host:cd>
        <host:cd>
          <host:name avail="1">ns2.mondomainetest.fr</host:name>
        </host:cd>
      </host:chkData>
    </resData>
    <trID>
      <clTRID>123456 </clTRID>
      <svTRID>EPP-dd7ea546-04e8-489f-b32b-867fc39417a0</svTRID>
    </trID>
  </response>
</epp>
```

In case of an error, the server returns the error code and the reason for the error.

## 8.4 host:create

A **<host:create>** command for a host object is made up of 2 elements:

- **<host:name>** The name of the host object. To create an **Internal host**, the domain name must necessarily exist.
- **<host:addr ip="...">** 0 to 8 IP addresses associated with the host object. To create an External Host, the **<host:create>** command must not contain any IP addresses.

### Example of the creation of a host object

```
<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.mondomainetest.paris</host:name>
      </host:create>
    </create>
    <clTRID>123456 </clTRID>
  </command>
</epp>
```

### Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:host="urn:ietf:params:xml:ns:host-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <host:creData>
        <host:name>ns1.mondomainetest.paris</host:name>
        <host:crDate>2021-12-06T12:50:44.426617Z</host:crDate>
      </host:creData>
    </resData>
    <trID>
      <clTRID>123456 </clTRID>
      <svTRID>EPP-9ddf1511-1105-4f17-85a6-4270a5f180c0</svTRID>
    </trID>
  </response>
</epp>
```

In case of an error, the server returns the error code and the reason for the error.

## 8.5 host:update

The **<host:update>** command is used to update a host object. This operation is subject to conditions.

A host object can only be updated:

- by the Sponsor of the object (**<host:clID>** attribute of the object),
- in the case of an external host object (excluding .fr, etc.), if it is not used by a domain name of another registrar.

Only the IP addresses and the host name can be modified.

In the case of an external host, it is not possible to add IP addresses.

It is not possible to delete all the IP addresses of a glue host.

The command consists of only one mandatory element:

- **<host:name>** The name of the host object.

And 3 optional elements:

- **<host:add>** one or more **<host:addr>** to be added,
- **<host:rem>** one or more **<host:addr>** to be deleted,
- **<host:chg>** new **<host:name>** of this host object.

### Example of update of a host object

```
<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.mondomainetest.fr</host:name>
        <host:add>
          <host:addr
ip="v6">2001:67c:eaf4:894e:9b1f:f916:15cb:8e4a</host:addr>
          </host:addr>
        </host:add>
      </host:update>
    </update>
    <clTRID>123456 </clTRID>
  </command>
</epp>
```

### Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>123456 </clTRID>
      <svTRID>EPP-c40345f8-9c28-4f23-a622-a70f86b554b1</svTRID>
    </trID>
  </response>
</epp>
```

In case of an error, the server returns the error code and the reason for the error.

## 8.6 host:info

The **<host:info>** command provides all the information associated with a host object. The command consists of a single element:

- The name of the host object.

### Example of a host object info request

```
<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>ns1.mondomainetest.fr</host:name>
      </host:info>
    </info>
    <clTRID>123456 </clTRID>
  </command>
</epp>
```

### Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:host="urn:ietf:params:xml:ns:host-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <host:infData>
        <host:name>ns1.mondomainetest.fr</host:name>
        <host:roid>HOST340-FRNIC</host:roid>
        <host:status s="linked"/>
        <host:status s="ok"/>
        <host:addr ip="v4">127.10.10.30</host:addr>
        <host:clID>IANA9998</host:clID>
        <host:crID>IANA9998</host:crID>
        <host:crDate>2021-12-06T12:53:54.905997Z</host:crDate>
      </host:infData>
    </resData>
    <trID>
      <clTRID>123456 </clTRID>
      <svTRID>EPP-bcf6b752-9b8f-42af-bca0-40bb81d65dfc</svTRID>
    </trID>
  </response>
</epp>
```

In case of an error, the server returns the error code and the reason for the error.

A domain:info command will allow you to identify the list of hosts that depend on this domain name.

## 8.7 host:delete

The **<host:delete>** command is used to delete a host object. This operation is subject to conditions:

- A host object can only be deleted by the object's sponsor (**<host:clID>** attribute of the object)
- A host object cannot be deleted if it is associated with a domain object.

The command consists of only one element:

- The name of the host object.

### Example of deletion of a host object

```
<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.mondomainetest1.paris</host:name>
      </host:delete>
    </delete>
    <clTRID>123456</clTRID>
  </command>
</epp>
```

### Server response

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>123456</clTRID>
      <svTRID>EPP-c601282b-b7eb-4472-bee6-5197be140a35</svTRID>
    </trID>
  </response>
</epp>
```

In case of an error, the server returns the error code and the reason for the error.

## 9 Notifications

This notification queue displays notifications from the most recent to the oldest. We recommend that you empty this notification list as you go along.

The number of notifications received is displayed in all orders.

- There are two steps in viewing the notifications, req and ack
- The number of notifications present is systematically mentioned in all the responses of any command.

### 9.2 Management of the notification queue

#### Retrieve the first message from the queue

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <poll op="req"/>
    <clTRID>123456</clTRID>
  </command>
</epp>
```

#### Acknowledge receipt of the first message in the queue

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <msgQ count="78" id="5cce0d87-c659-4f65-89f2-55e61501b464"/>
    <trID>
      <clTRID>123456 </clTRID>
      <svTRID>EPP-cc80c7a2-c69c-448e-b640-c026fdd63469</svTRID>
    </trID>
  </response>
</epp>
```

### 9.3 Notifications

In this chapter you will find examples of notifications for :

TRANSFER requested,

TRANSFER finished,

TRANSFER cancelled,

RECOVER finished.



## Example of a notification following a requested TRANSFER operation (outgoing registrar)

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="2" id="f003d854-5c84-4ae3-a2b2-0559d3696307">
      <qDate>2022-09-26T12:24:05.597287Z</qDate>
      <msg>Transfer requested.</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name>mondomainetest.fr</domain:name>
        <domain:trStatus>pending</domain:trStatus>
        <domain:reID>xxxx</domain:reID>
        <domain:reDate>2022-09-26T12:24:05.585631Z</domain:reDate>
        <domain:acID>xxxx</domain:acID>
        <domain:acDate>2022-09-26T12:26:05.58532Z</domain:acDate>
        <domain:exDate>2024-09-26T12:24:04.424345Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>$clTRID </clTRID>
      <svTRID>EPP-57504f96-6ca8-47b3-ba52-3a51e6f89f86</svTRID>
    </trID>
  </response>
</epp>

```

## Example of notification following a requested TRANSFER operation (incoming registrar)

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="2" id="dace0e00-cac1-4ccb-8d3c-96de4abc9af8">
      <qDate>2022-09-26T12:20:55.487218Z</qDate>
      <msg>Transfer approved.</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name>mondomainetest.fr</domain:name>
        <domain:trStatus>clientApproved</domain:trStatus>
        <domain:reID>xxxx</domain:reID>
        <domain:reDate>2022-09-26T12:20:55.121861Z</domain:reDate>
        <domain:acID>xxxx</domain:acID>
        <domain:acDate>2022-09-26T12:20:55.483991Z</domain:acDate>
        <domain:exDate>2024-09-26T12:20:54.211253Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>$clTRID </clTRID>
      <svTRID>EPP-d1cd8e2b-a604-4fe4-996c-0d328305e4de</svTRID>
    </trID>
  </response>
</epp>

```

## Example of notification following an approved TRANSFER operation (incoming registrar)

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="2" id="dace0e00-cac1-4ccb-8d3c-96de4abc9af8">
      <qDate>2022-09-26T12:20:55.487218Z</qDate>
      <msg>Transfer approved.</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name>mondomainetest.fr</domain:name>
        <domain:trStatus>clientApproved</domain:trStatus>
        <domain:reID>xxxx</domain:reID>
        <domain:reDate>2022-09-26T12:20:55.121861Z</domain:reDate>
        <domain:acID>xxxx</domain:acID>
        <domain:acDate>2022-09-26T12:20:55.483991Z</domain:acDate>
        <domain:exDate>2024-09-26T12:20:54.211253Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>$clTRID </clTRID>
      <svTRID>EPP-dlcd8e2b-a604-4fe4-996c-0d328305e4de</svTRID>
    </trID>
  </response>
</epp>

```

## Example of notification following a TRANSFER operation completed after approval by the outgoing registrar (outgoing registrar)

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="4884876e-1679-4f10-9c64-b3f14b800694">
      <qDate>2022-09-26T15:05:33.90624Z</qDate>
      <msg>Transfer completed.</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name>mondomainetest.fr</domain:name>
        <domain:trStatus>clientApproved</domain:trStatus>
        <domain:reID>xxxx</domain:reID>
        <domain:reDate>2022-09-26T15:05:33.494613Z</domain:reDate>
        <domain:acID>xxxx</domain:acID>
        <domain:acDate>2022-09-26T15:05:33.906147Z</domain:acDate>
        <domain:exDate>2024-09-26T15:05:32.686214Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>$clTRID </clTRID>
      <svTRID>EPP-dbeb370a-77e0-4c4b-ba7d-1df6e8a729e1</svTRID>
    </trID>
  </response>
</epp>

```

## Example of a notification following a TRANSFER operation completed after approval by the outgoing registrar (incoming registrar)

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="5fd8fc23-1381-4d7f-b8e9-5b6732e6668a">
      <qDate>2022-09-26T12:20:55.506324Z</qDate>
      <msg>Transfer completed.</msg>
    </msgQ>
    <resData>
      <domain:panData>
        <domain:name paResult="1">mondomainetest.fr</domain:name>
        <domain:paTRID>
          <clTRID>${clTRID} </clTRID>
          <svTRID>EPP-a263fb9f-e6b2-43b9-b9f3-670ebd47d3d0</svTRID>
        </domain:paTRID>
        <domain:paDate>2022-09-26T12:20:55.121861Z</domain:paDate>
      </domain:panData>
    </resData>
    <trID>
      <clTRID>${clTRID} </clTRID>
      <svTRID>EPP-6bab00b7-2127-4ff4-b34f-5ccd80629954</svTRID>
    </trID>
  </response>
</epp>

```

## TECHNICAL SPECIFICATIONS OF THE .FR AND FRENCH OVERSEAS CCTLDS REGISTRY SYSTEM

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="4884876e-1679-4f10-9c64-b3f14b800694">
      <qDate>2022-09-26T15:05:33.90624Z</qDate>
      <msg>Transfer completed.</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name> mondomainetest.fr</domain:name>
        <domain:trStatus>clientApproved</domain:trStatus>
        <domain:reID>xxxx</domain:reID>
        <domain:reDate>2022-09-26T15:05:33.494613Z</domain:reDate>
        <domain:acID>xxxx</domain:acID>
        <domain:acDate>2022-09-26T15:05:33.906147Z</domain:acDate>
        <domain:exDate>2024-09-26T15:05:32.686214Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>$clTRID </clTRID>
      <svTRID>EPP-dbeb370a-77e0-4c4b-ba7d-1df6e8a729e1</svTRID>
    </trID>
  </response>
</epp>
```

## Example of a notification following a TRANSFER operation completed after no response from the outgoing registrar (8 days after the start of the transfer) (outgoing registrar)

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="6a3cbebd-fd4f-4ee5-9c07-55c83935bf74">
      <qDate>2022-09-26T13:58:48.52489Z</qDate>
      <msg>Transfer completed.</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name>mondomainetest.fr </domain:name>
        <domain:trStatus>serverApproved</domain:trStatus>
        <domain:reID>xxxx</domain:reID>
        <domain:reDate>2022-09-
26T13:56:41.212855Z</domain:reDate>
        <domain:acID>xxxx</domain:acID>
        <domain:acDate>2022-09-
26T13:58:48.470437Z</domain:acDate>
        <domain:exDate>2024-09-
26T13:56:39.936357Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>$clTRID </clTRID>
      <svTRID>EPP-a2bd03da-8a4b-43cd-98e0-e5487279fa1c</svTRID>
    </trID>
  </response>
</epp>

```

## Example of a notification following a TRANSFER completed after no response from the outgoing registrar (8 days after the start of the transfer) (incoming registrar)

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="2" id="db7ad0a1-bcbb-4f2b-9a63-30aaaa8f2c55">
      <qDate>2022-09-26T13:58:48.539662Z</qDate>
      <msg>Transfer completed.</msg>
    </msgQ>
    <resData>
      <domain:panData>
        <domain:name>
paResult="1">mondomainetest.fr</domain:name>
          <domain:paTRID>
            <clTRID>TEST-AUTO-1664200601055-daduvoxoco</clTRID>
            <svTRID>EPP-60ddb3c-dc57-4723-b9b7-
d3963040e55e</svTRID>
          </domain:paTRID>
          <domain:paDate>2022-09-
26T13:56:41.212855Z</domain:paDate>
        </domain:panData>
      </resData>
      <trID>
        <clTRID>${clTRID} </clTRID>
        <svTRID>EPP-16fde3b2-feed-4db1-a781-62d604bbfae7</svTRID>
      </trID>
    </response>
  </epp>

```



## TECHNICAL SPECIFICATIONS OF THE .FR AND FRENCH OVERSEAS CCTLDS REGISTRY SYSTEM

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="cb34f58a-8029-42a0-8e13-c21acc62e3d6">
      <qDate>2022-09-26T13:58:48.543832Z</qDate>
      <msg>Transfer completed.</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name> mondomainetest.fr</domain:name>
        <domain:trStatus>serverApproved</domain:trStatus>
        <domain:reID>xxxx</domain:reID>
        <domain:reDate>2022-09-
26T13:56:41.212855Z</domain:reDate>
        <domain:acID>xxxx</domain:acID>
        <domain:acDate>2022-09-
26T13:58:48.470437Z</domain:acDate>
        <domain:exDate>2024-09-
26T13:56:39.936357Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>$clTRID </clTRID>
      <svTRID>EPP-2ba86f6a-fed2-4586-8251-1a05bfa712c7</svTRID>
    </trID>
  </response>
</epp>
```

## Example of a notification following a TRANSFER operation cancelled by the incoming registrar (outgoing registrar)

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="cb34f58a-8029-42a0-8e13-c21acc62e3d6">
      <qDate>2022-09-26T13:58:48.543832Z</qDate>
      <msg>Transfer completed.</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name>mondomainetest.fr</domain:name>
        <domain:trStatus>serverApproved</domain:trStatus>
        <domain:reID>xxxx</domain:reID>
        <domain:reDate>2022-09-26T13:56:41.212855Z</domain:reDate>
        <domain:acID>xxxx</domain:acID>
        <domain:acDate>2022-09-26T13:58:48.470437Z</domain:acDate>
        <domain:exDate>2024-09-26T13:56:39.936357Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>$clTRID </clTRID>
      <svTRID>EPP-2ba86f6a-fed2-4586-8251-1a05bfa712c7</svTRID>
    </trID>
  </response>
</epp>

```

## TECHNICAL SPECIFICATIONS OF THE .FR AND FRENCH OVERSEAS CCTLDS REGISTRY SYSTEM

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="cb34f58a-8029-42a0-8e13-c21acc62e3d6">
      <qDate>2022-09-26T13:58:48.543832Z</qDate>
      <msg>Transfer completed.</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name> mondomainetest.fr</domain:name>
        <domain:trStatus>serverApproved</domain:trStatus>
        <domain:reID>xxxx</domain:reID>
        <domain:reDate>2022-09-
26T13:56:41.212855Z</domain:reDate>
        <domain:acID>xxxx</domain:acID>
        <domain:acDate>2022-09-
26T13:58:48.470437Z</domain:acDate>
        <domain:exDate>2024-09-
26T13:56:39.936357Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>$clTRID </clTRID>
      <svTRID>EPP-2ba86f6a-fed2-4586-8251-1a05bfa712c7</svTRID>
    </trID>
  </response>
</epp>
```

## Example of a notification following a TRANSFER operation cancelled by the incoming registrar)

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="2ff26ea8-6ec8-47c3-9dc6-c6566294f6b3">
      <qDate>2022-09-26T12:24:05.914567Z</qDate>
      <msg>Transfer aborted.</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name>mondomainetest.fr</domain:name>
        <domain:trStatus>clientCancelled</domain:trStatus>
        <domain:reID>xxxx</domain:reID>
        <domain:reDate>2022-09-
26T12:24:05.585631Z</domain:reDate>
        <domain:acID>xxxx</domain:acID>
        <domain:acDate>2022-09-
26T12:24:05.883857Z</domain:acDate>
        <domain:exDate>2024-09-
26T12:24:04.424345Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>$clTRID </clTRID>
      <svTRID>EPP-45b359db-5fa4-4087-a140-fb5028958ab5</svTRID>
    </trID>
  </response>
</epp>

```

## TECHNICAL SPECIFICATIONS OF THE .FR AND FRENCH OVERSEAS CCTLDS REGISTRY SYSTEM

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="2ff26ea8-6ec8-47c3-9dc6-c6566294f6b3">
      <qDate>2022-09-26T12:24:05.914567Z</qDate>
      <msg>Transfer aborted.</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name> mondomainetest.fr</domain:name>
        <domain:trStatus>clientCancelled</domain:trStatus>
        <domain:reID>xxxx</domain:reID>
        <domain:reDate>2022-09-
26T12:24:05.585631Z</domain:reDate>
        <domain:acID>xxxx</domain:acID>
        <domain:acDate>2022-09-
26T12:24:05.883857Z</domain:acDate>
        <domain:exDate>2024-09-
26T12:24:04.424345Z</domain:exDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>$clTRID </clTRID>
      <svTRID>EPP-45b359db-5fa4-4087-a140-fb5028958ab5</svTRID>
    </trID>
  </response>
</epp>
```

## Example of notification following a completed RECOVER operation (outgoing registrar)

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:frnic="http://www.afnic.fr/xml/epp/frnic-2.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="bd89ec85-cb17-4d75-a286-ce10eeef637">
      <qDate>2022-09-20T13:22:59.397404Z</qDate>
      <msg>Recover completed.</msg>
    </msgQ>
    <extension>
      <frnic:ext>
        <frnic:resData>
          <frnic:recData>
            <frnic:domain>
              <frnic:name>mondomainetest.fr</frnic:name>
              <frnic:reID>xxxx</frnic:reID>
              <frnic:reDate>2022-09-
20T13:22:59.397305Z</frnic:reDate>
              <frnic:acID>xxxx</frnic:acID>
            </frnic:domain>
          </frnic:recData>
        </frnic:resData>
      </frnic:ext>
    </extension>
    <trID>
      <clTRID>$clTRID </clTRID>
      <svTRID>EPP-616e938c-fea8-48a8-aeac-6350aa6deeea</svTRID>
    </trID>
  </response>
</epp>

```

## Example of notifications following a RESTORE operation

```

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="2" id="889e305f-81ca-485f-bd1d-6c400889cd8a">
      <qDate>2022-09-20T13:37:19.954275Z</qDate>
      <msg>Deletion aborted. Domain successfully restored.</msg>
    </msgQ>
    <resData>
      <domain:panData>
        <domain:name paResult="1">mondomainetest.fr</domain:name>
        <domain:paTRID>
          <clTRID>TEST-AUTO-1663681039737-tanigajiri </clTRID>
          <svTRID>EPP-693f7732-2522-4ead-a3e0-d0ef7061b7f2</svTRID>
        </domain:paTRID>
        <domain:paDate>2022-09-20T13:37:19.9541Z</domain:paDate>
      </domain:panData>
    </resData>
    <trID>
      <clTRID>TEST-AUTO-1663681040752-rizusezeji </clTRID>
      <svTRID>EPP-94aa6860-d5c4-4558-a7cd-f5530ff83286</svTRID>
    </trID>
  </response>
</epp>

```

# 10 Return codes and error messages

## 10.1 Return codes

The return codes (element `<result code="xyzz">`) respond to a logic and are coded in 4 digits.

The **RFC 5730** describes the literal values that are associated with these codes and returned in the `<msg>` element. If the server has to be localised, a corresponding translation will be proposed; the meaning of titles will however be kept as close as possible.

The "1000 series" corresponds to the codes returned when the operation requested by the client has been taken into account:

- **1000**: This is the normal return code for a command that has been normally and completely executed and is not addressed by any other return code for this series.
- **1001**: This code indicates that the command has been taken into account but that its complete execution has been delayed. The final result will be known later on and will be sent in a message placed in the notification queue of the registrar(s) concerned by this operation. The number of commands for which this code is systematically returned is limited, but it is also returned if, for unusual reasons, the server needs to delay the execution of a command (only for restore) that would normally return a **1000** code.
- **1300**: This result code is reserved for the `<poll>` command (in query mode) and indicates that there are no pending messages.
- **1301**: This result code is also reserved for the `<poll>` command and indicates that there is a message in the server's response that is ready to be deleted from the message queue.
- **1500**: This return code will be used to respond to a successful `<logout>` request.

The "2000 series" corresponds to the codes returned when a problem has been encountered and the command could not be taken into account normally.

- **2000**: Code returned when the command sent is unknown.
- **2001**: Code returned when a syntax error has been encountered.
- **2002**: Code returned when the command received is syntactically correct but cannot be interpreted because it is out of context.
- Example: a logout command received while the client has not yet completed the login phase.
- **2003**: Code returned when a required parameter is missing in a command.
- Example: a `<transfer op="query">` command for which the `<authInfo>` element is missing.
- **2004**: Code returned when the value of an element is outside the range specified by the EPP protocol.



- Example: trying to create the domain name "-trop-de-tirets- **-et-d-espaces-.fr**" will return this type of error.
- **2101**: This code is returned when the server receives a valid EPP command that however is not implemented in our integration.
- Example: a **<contact:transfer>** command.
- **2102**: This code is returned when the client sends a command that is implemented by our server but contains an option that is not implemented.
- **2103**: This code is returned if the client sends a command containing an extension that is unknown to the server.
- **2106**: Code returned when **<domain:transfer>** command is executed on a domain name which is not eligible for transfer. For example, if the request is from the registrar that already manages this domain name.
- **2200**: Code returned when validating the login/password while logging in to the EPP server.
- **2201**: Code returned when the registrar attempts to execute a command for which it does not have the rights.
- Example: an outgoing registrar that tries to send a "cancel" command on a domain name after a **TRANSFER** operation.
- **2202**: Code returned when the registrar wishing to execute a command could have done so if it had provided the correct authorisations. This code is typically used for a domain name **TRANSFER** request when the password provided (**<authInfo>**) is invalid.
- **2300**: Code returned when a domain name **TRANSFER** command is sent and the domain in question is already pending **TRANSFER**.
- **2304**: Code returned when the status of an object is not compatible with the command sent to the server.
- Example: a domain name update command sent when the domain has the "deleted" status and is in the redemption period.
- **2305**: Code returned when the execution of a command cannot be completed due to dependencies on other objects in the database prevent it.
- Example: a domain name deletion request when the zone still contains name servers used for other domain names administered by AFNIC.
- **2306**: Code returned when the server receives a syntactically valid EPP command element value that however does not comply with an AFNIC-specific rule. In most cases, the error returned indicates the problematic element as well as the rule that was not complied with. Sometimes the error message is not present, this code being used as a default value.
- Example: when an individual type of contact with civil status data (a potential holder of a domain name) is used as a technical contact in a command.
- **2400**: Code returned when an internal problem is preventing the command from being completed. This code indicates that our registration chain is

encountering problems. It is important to warn the AFNIC support service when this event occurs.

- **2500**: Code returned in a case similar to code 2400. But in this case the server decides to close the session. It is important to warn the AFNIC support service when this event occurs.
- **2501**: This code is returned if the **<login>** phase cannot be completed.
- **2502**: This code is returned if the number of sessions per registrar is limited and if the limit has been reached when a new **<login>** command is sent.

## 10.2 Error messages

Unlike the content of the **<msg>** element, whose final list can be found in **RFC 4930**, this list is liable to change. Some cases can be refined while others may disappear, depending on changes in registration policies, for example. For now, only an English version exists for these messages.