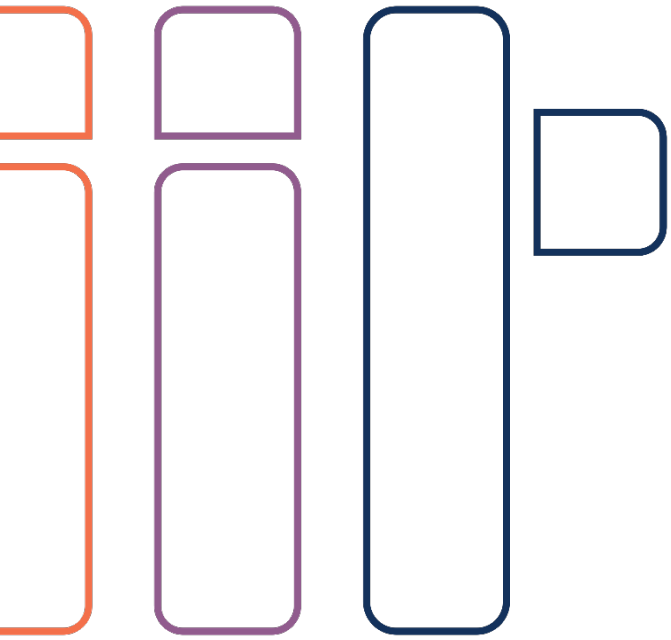


TECHNICAL REPORT

IIT TR-02/2023

Design and implementation of an EPP server for ccTLD .it

M. Loffredo, L. Luconi Trombacchi, M. Martinelli

ISTITUTO DI INFORMATICA E TELEMATICA

Consiglio Nazionale delle Ricerche

Design and implementation of an EPP server for ccTLD .it

Loffredo M., Luconi Trombacchi L., Martinelli M.
IIT - CNR

Table of contents

Abstract

EPP (Extensible Provisioning Protocol) is a synchronous XML-based client-server protocol for the provisioning and management of objects related to domain names. The operations on domain names are carried out by organizations, called Registrars, that have an active contract with the Registry of a TLD (Top Level Domain). This document details the HW/SW design and implementation of the country code TLD .it (henceforth ccTLD.it) EPP server, and the operations allowed by the server for registering and maintaining .it domain names.

1 Introduction

The domain name registration system of the .it Registry (henceforth also "Registry of the ccTLD .it") has been designed to allow Registrars to register and maintain domain names in real-time.
The system implements EPP to comply with internationally accepted standards.
The .it implementation provides secure connections for the management of the objects related to the registration and maintenance of domain names, namely:

- "domain" objects containing information on the Registrant (i.e. the assignee of the domain name), on technical and administrative contacts, and on the authoritative nameservers associated with the domain name itself;
- "contact" objects containing general information on the contacts referenced in "domain" objects i.e. the Registrant, the technical and administrative contacts.

The implementation of the various procedures has been designed in order to comply as closely as possible with the EPP standard as defined in the RFCs[1]. The standard foresees the possibility to define extensions to the protocol so as to try to meet special needs, but these extensions only have a local scope and, with a few exceptions, the main ccTLDs and gTLDs that now implement synchronous systems have sought to limit them to the maximum and to adopt the universally recognized and accepted standard.

The contents of this document are structured as in the following:

- Section 2 provides a brief description of SLDs edu.it and gov.it.
- Section 3 introduces some terms and definitions whose knowledge is required for the full comprehension of the proposed arguments.
- The general HW/SW architecture of the EPP server is described in Section 4.
- The charsets accepted by the EPP server and the statuses considered for domains and contacts, the URLs of the servers that can be accessed by the Registrars in the live environment and the limits that have been set on those servers are described in Section 5.
- Section 6 provides an EPP overview and a description of how commands (i.e. requests and

[1] RFC 5730 - Extensible Provisioning Protocol (EPP)
RFC 5731 - Extensible Provisioning Protocol Domain Name Mapping
RFC 5732 - Extensible Provisioning Protocol Host Mapping
RFC 5733 - Extensible Provisioning Protocol Contact Mapping
RFC 3375 - Generic Registry-Registrar Protocol Requirements
RFC 3735 - Guidelines for Extending the Extensible Provisioning Protocol
RFC 3915 - Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol
RFC 5910 - Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol

responses) are arranged.
- The commands for managing EPP work sessions are described in Section 7.
- The commands for registering and maintaining domain names and contacts associated with them are described in Section 8. To facilitate the reading of the EPP commands sent by the Registrars, the required fields are specified in **bold**, while the fields that are extensions to the standard protocol are in *italics*.
- Section 9 briefly presents the operations on domains defined by the .it Registry that are executed out of the EPP server's scope.
- The commands for querying the EPP objects (i.e. domains, contacts and polling queue) are included in Section 10.
- Section 11 describes how.it Registry has implemented the DNSSEC protocol with a particular focus on how DNSSEC information is delivered through EPP.
- All other detailed information pertaining the ccTLD .it EPP server implementation is shown in Section 12.

2 SLD edu.it and gov.it

In addition to the ccTLD .it domains, the EPP server allows Registrars to request operations on the SLDs edu.it and gov.it.
The SLD edu.it is reserved to only Italian public schools that are uniquely identified by a mechanographic code.
The registration of a domain name in the SLD gov.it is allowed only to the Central State Public Administrations and to the National Social Security and Welfare Institutions, indicated in the list of public administrations identified pursuant to Article 1, paragraph 3, of Law no. 196 of 31 December 2009 and subsequent amendments. This list (List S13) is managed by ISTAT and published annually in the Official Gazette.
The request for the assignment of a domain name in the SLD gov.it can only be made by those Public Administrations present in the aforementioned list and registered in the Index of Public Administrations (IPA) managed by AgID (Agency for Digital Italy).
The registration procedure of a gov.it domain name is preceded by a validation phase of the requesting Public Administration. This phase, which shall be carried out by a duly delegated person, is aimed at verifying that the Public Administration possesses the requisites, as per par. 1 above, to be able to register a domain name in the SLD gov.it.
The Validation Procedure generates a Validation Code that shall be indicated in all requests for registration of gov.it domain names by the same administration.
The Validation Code expires after 1 year.

3 Terms and definitions

The following table contains the definition of the terms used in this document.

Term	Definition
A Record	Acronym of Address resource record, which shows the correspondence between a name and its associated IPv4 address.

Accreditation Test	Verification of Registrar's technical ability to operate with the synchronous system.
ASCII	Acronym of American Standard Code for Information Interchange, it is the 7 bit character-encoding commonly used in computers.
Atomic commands	Commands for one specified action. EPP commands are atomic - there are no successes or partial failures even if the related request may not terminate.
Authinfo	Authorization password used by the Registrant to request specific operations.
Auto renew period	The grace period following the automatic renewal of a domain name at its expiration date.
Authoritative Nameserver	A Nameserver which owns the data for a particular zone of the name tree.
Billing	The transactions costs that will be listed on the invoice sent to the Registrars. The invoice contains all the transactions carried out by the Registrars with respect to a particular payment. The billing may not necessarily take place at the same time as the charge.
Bulk Transfer	Transfer, between two Registrars, of a considerable number of domain names.
Cardinality	Minimum or maximum value of definition options for a field.
ccTLD	Acronym of country code Top Level Domain, it univocally identifies a nation on the basis of ISO-3166 encoding (e.g. Italy = "it").
Change	Operation that allows the information associated with a domain name or a contact registered in the DBAN to be changed. This operation can be carried out by Registrars, Registrants and by the Registry.
Client	Computer that accesses to the resources supplied by another computer (server) on a local network or on the Internet.
CNAME (Record)	Acronym of Canonical Name, it is a record that defines an alternative name with which the same machine can be identified. These resource records are used to create the so called "alias".
Command	One or more specific words in the operating system or in the management menu of programs that are typed via the keyboard or activated using a mouse, and that execute a particular operation.
Database of Assigned Names (DBAN)	Database maintained by the .it Registry, where all the data regarding assigned domain names in the ccTLD .it are managed. In this document, DBAN, Registry Database and Database of the Registry are equivalent.
DBAN	Acronym of Database of Assigned Names (see the above definition).
DS Record	Acronym of Delegation Signer (DS), it is a resource record used by DNSSEC for implementing the "chain-of-trust" between a parent zone and a child zone. A zone manager generates a "digest" of the public key (DNSKEY record) associated with the digitally signed domain name and transmits it to the parent zone manager who associates it with the delegation of that domain name through a DS record.

Debit	Withdrawal from Registrar's credit of the cost of transactions that are supposed to be invoiced, including any VAT. The cost of each transaction/operation is debited immediately so that the Registrar's credit is always up to date.
Default	Pre-defined value if user gives no specific value.
Digest	The hash generated from the public key.
DNS	Acronym of Domain Name System (RFC1035), it is the system used to convert domain names into IP addresses and vice versa.
DNSKEY Record	DNSSEC Resource record containing a public key.
DNSSEC	Acronym of Domain Name System SECurity extensions, it is the protocol (RFC4033) which uses public/private keys for cryptography to ensure that the information is coming from an authoritative nameserver, and has not been altered during its transmission through the network. DNSSEC enables: - DNS servers to sign their own resource records (RR) with a private key; - DNS resolvers to verify the information through its associated public key.
DNS Delegation	Through entering records in the respective files of the zone, allows the activation of a domain name on the Internet.
Domain name	Association between a public IP address and a string of characters to guarantee the consistency of the associations between IP addresses and domain names. The conversion of domain names into IP addresses and vice versa is guaranteed by the DNS. A domain name is made up of several parts.
Drop Time	Process that provides for the cancellation of the domain names that are in pendingDelete/pendingDelete status, at fixed times.
DUPn	Format of the contact ID used in the duplication of a contact following a Domain Transfer operation.
EPP	Acronym of Extensible Provisioning Protocol, it is the synchronous client-server protocol based on XML. In the implementation of the .it Registry, it offers secure connections for managing objects linked to the registration and maintenance of domain names.
Expire	Field that shows, for invoicing, the expiry date of a domain name registered in the DBAN. It is automatically updated by the system at the end of the maintenance period of a domain name (one year).
Extension	Sequence of alphanumerical characters that specifies a command.
First come first served	Chronological order of arrival that determine the order in which requests are processed.
Glue record	IP address of a nameserver necessary for the correct functioning of the resolution process of domain names.
Grace period/Auto renew period	The 15 (fifteen) days immediately following the expiry of the domain name.

gTLD	Acronym of generic Top Level Domain, it is the univocal tag for the suffix of a tree of Internet domain names, of a generic type: the generic TLDs or gTLDs are made up of 3 or more characters, and can be subdivided into two kinds: “sponsored” TLDs (sTLDs) and “unsponsored” TLDs (uTLDs).
Host/Nameserver/Name server	Server that translates a network address in textual format into the corresponding numerical address. It is also known as DNS (Domain Name System). The nameserver can be subordinate or not subordinate to the associated domain name. For example, the nameserver ns.example.it is subordinate to the domain name example.it. In this document “host”, “name server” and “nameserver” are equivalent and they are used to identify a generic nameserver.
HTTPS	Secure HTTP protocol for access to web server.
ICANN	Acronym of Internet Corporation for Assigned Names and Numbers, it is the not for profit organization responsible for Internet Protocol addresses, the protocol identifiers, managing Top-Level Domains, generic domains (gTLD) and the interntional code (ccTLD), as well as the root server. ICANN safeguards the operating stability of the Internet, promotes competition, wides the representation of the global community of the Internet and develops policies via participatory and consensual processes (http://www.icann.org).
ID	Acronym of IDentifier, an alphanumeric code that univocally identifies a contact (“registrant”, “admin” or “tech”) within the DBAN.
IDN (Internationalised Domain Name)	Domain name containing non-ASCII characters, such as letters with accent, which belong to the charset Latin-1 Supplement, Latin Extended-A, Latin Extended-B, Greek, Greek Extended and Cyrillic. The non-ASCII characters indicated above enable the registration of the IDNs in the 24 official languages of the European Union.
IETF	Acronym of Internet Engineering Task Force, it is the international forum concerned with the development and the promotion of standard Internet (https://www.ietf.org/).
Internet	Network of calculators around the world that interconnect thousands of national and international networks that use TCP/IP protocol, thus allowing the exchange of information.
IP	Acronym of Internet Protocol, it was created to interconnect heterogeneous networks by technology, performance and management. The current version is also called IPv4 to distinguish it from the more recent IPv6 that was conceived so as to manage better the increasing number of computers connected to the Internet.
IP address	Acronym of Internet Protocol address, it is a numerical sequence that univocally identifies a machine that is connected to the Internet, either permanently or occasionally.
ISO 3166-1	Standard that provides codes for country names.
KSK	Acronym of Key Signing Key, it is a public/private key pair. The KSK private key is used to generate a digital signature for the ZSK (Zone Signing Key). The KSK public key is stored in the DNS to be used to authenticate the ZSK.

Login	Authentication procedure via username and password. In EPP it corresponds to a specific command to begin a work session.
Maintenance	Automatic renewal of a domain name registered in the DBAN.
Multistatus	Combination of more than one status associated with a domain name or contact.
MX Record	Acronym of Mail eXchange, it is the resource record that indicates which are the servers that manage the email for a certain domain name.
NS Record	Acronym of Name Server, it is the resource record that indicates which are the authoritative nameservers for a certain domain name.
Name server/Nameserver/Host	Server that translates a network address in textual format into the corresponding numerical address. It is also known as DNS (Domain Name System). The nameserver can be subordinate or not subordinate to the associated domain name. For example, the nameserver ns.example.it is subordinate to the domain name example.it. In this document "host", "name server" and "nameserver" are equivalent and they are used to identify a generic nameserver.
Object	A set of data that identifies an element (Domain, Registrant, Contact, Registrar) inside the DBAN.
Parsing	Subdivision of the instructions of a program into their various components so that they can be interpreted by the compiler and transformed into executable commands.
Polling queue	The queue of all the messages that the client receives from the server. By querying their polling queue, Registrars can see some messages related to domain names (actions started, currently under way, or concluded on a given domain name), authentication and credit level.
Public Adiministration	A public administration allowed to register a gov.it domain..
Query	Queries to the nameserver.
RAIN-NG	Acronym of Registrar Advanced INterface Next Generation, it is the portal provided by .it Registry for Registrars only.
Reason for error	Reasons for error used by the Registry's synchronous server.
Record	Data structure, logically connected, that contains a set of fields that can be identified by a number or a name.
Redemption period	The 30 (thirty) days after the request for cancellation of a domain name by the Registrar.
Registration Form	The form provided by Registrars to end users to collect registration data.
Resource Records (RRs)	Records containing information about a certain DNS zone. There are several types of resource records such as SOA, NS, MX, etc..
Referring	For contact objects, it indicates the correspondence between them and the ID contacts present in other objects in the DBAN.
REG tag	Registrar tag.
Registrant	Person or organization that requests the registration of a domain name or who has already been assigned one.

Registrar	Organizations that carry out registrations of domain names on behalf of themselves or the Registrants, according to the "Rules for assignment and management of domain names under the ccTLD .it". In order to become Registrar, an organization must pass an accreditation procedure arranged by the Registry and must have the proper technical infrastructure.
Registration	Entering of a new domain name or contact into the DBAN.
Registry	Organization responsible for assigning domain names, managing the registries and the primary nameservers for a TLD. It is delegated to this task directly by ICANN. In this document, .it Registry and Registry of the ccTLD .it are equivalent.
Registry Database	Database maintained by the .it Registry, where all data relating to domain names assigned in the ccTLD .it are stored and managed.
RFC	Acronym of Request For Comments, it is the document that gives specifics regarding new research, innovation and methodologies in computer science and the Internet.
RR	Acronym of Resource Record, it contains information about a certain DNS zone. There are several types of resource records such as SOA, NS, MX, etc..
RRset	Acronym of Resource Record set, a set of resource records (RR) of the same type.
RRSIG Record	Acronym of Resource Record Signature, it is the resource record defined by the DNSSEC containing a cryptographic signature for a set of resource records of the same type (RRset).
Server	Computer in a network that sends files to other computers in the network and that executes applications on their behalf.
SLD	In the DNS hierarchy, a second-level domain (SLD) is a domain that is directly below a top-level domain (TLD).
SOA Record	Acronym of Start Of Authority, it is the resource record that defines the machine on which the primary nameserver is active for the domain name and some "working parameters" of the secondary nameservers.
SSL	Acronym of Secure Sockets Layer, it is the cryptographic protocol that allows secure communication between two points in the network.
Stateful	In application protocols, such as EPP, it refers to the status of the communication session.
Status	A status characterizes the current operational condition of an object and its possible future transitions.
sTLD	Acronym of sponsored Top Level Domain - the gTLDs managed by a sponsor that represents the community and which proves to have an affinity with it. The organization to which is delegated specific responsibilities regarding the management of a Registry in a sTLD, for example in policy formuation regarding the operations of the TLD. An sTLD has a Charter approved by ICANN that defines its purpose and how the TLD must be managed.
Subordinate nameserver	A nameserver is defined subordinate to a given domain name if it belongs to the zone of the domain name itself. For example, the nameserver ns.example.it is subordinate to the domain name example.it.

Tag	Also used to name the code that marks the beginning and end of the entity in the said languages, e.g. HTML, SGML and XML.
TLD	Acronym of Top Level Domain, it is the Univocal identifier of the suffix of a tree of Internet domain names, immediately under the root, and thus also known as “First Level Domains”.
Update	Update operation.
URI	Acronym of Uniform Resource Identifier, it is a string that univocally identifies a generic resource e.g. web address, document, image, file, service, email address. An URL is a URI, more commonly known as a web address.
URL	Acronym of Universal Resource Locator, it is the address for a web page in alphabetic format. The URL is transformed into an IP address by the DNS.
UTC	Acronym of Universal Time Coordinated, it is the reference time zone from which all the other time zones in the world are calculated. It coincides with the GMT (Greenwich Mean Time) with less infinitesimals.
uTLD	Acronym of unsponsored Top Level Domain, gTLDs that are not sponsored, e.g. “.com” or “.info”. They work directly following policies established by the global internet community and more specifically by ICANN.
XML	Acronym of eXtensible Markup Language, it is the meta language for creating markup languages for exchanging data between websites and applications that may be based on different systems. A markup language uses particular markers (tags) to indicate the function of the various parts of the code (e.g. <tag attributes>content</tag>).
XML schema	The only language describing the content of an XML file that has reached the official (1.1) validation of the W3C.
Zone of the ccTLD .it	DNS master file of the ccTLD .it in which all the active delegations in the ccTLD .it are inserted.
ZSK	Acronym of Zone Signing Key, it is a public/private key pair. The ZSK private key is used to generate a digital signature, known as RRSIG, for each of the resource record sets (RRset) in a zone. The ZSK public key is stored in the DNS (DNSKEY record) to authenticate an RRSIG.
Validation Code	The code generated at the end of the Validation Procedure for a gov.it Registrant..
Validation Procedure	The validation procedure of the requesting Public Administration that precedes the registration of gov.it domain name.
Verification Procedure	The verification procedure settled by the Registry to check the accuracy of Registrants’data.

4 Design and implementation of HW/SW architecture

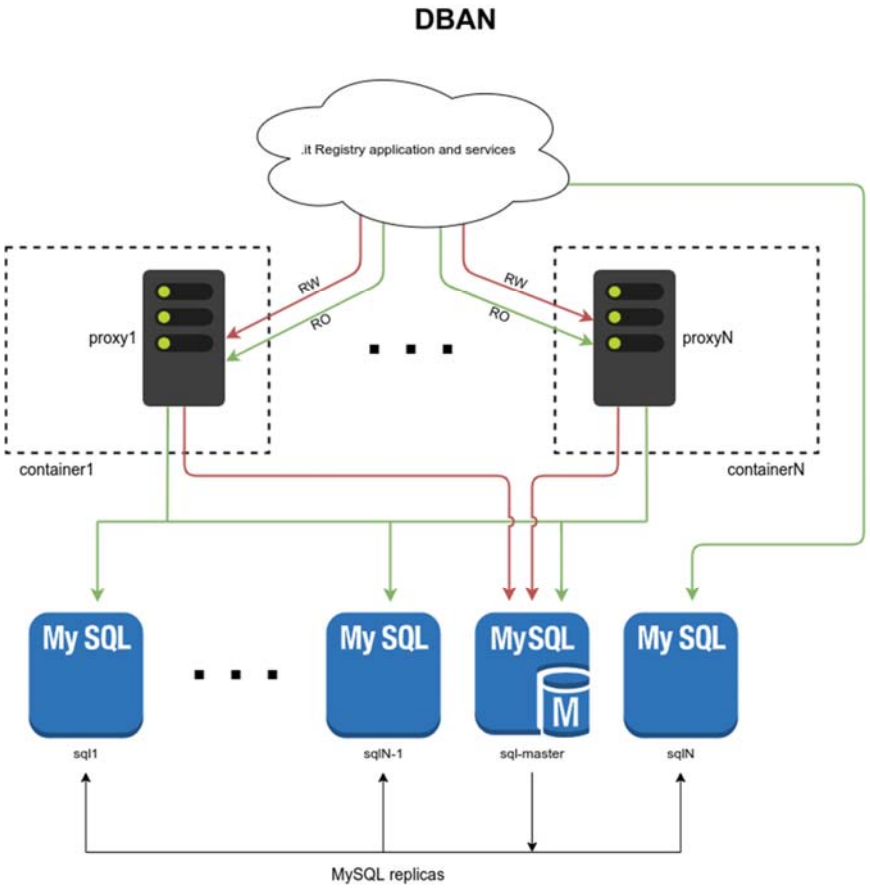
4.1 HTTPS as transport protocol

Unlike most of the EPP servers' implementations that make use of TLS as transport protocol, the ccTLD .it opted for using HTTPS. Clients issue the EPP requests by using the HTTP POST method and the EPP responses are returned as an HTTP response body. Basically, the ccTLD .it EPP server is a RESTful web service.
The reasons supporting that decision were the following:

- being the EPP session not strictly bound to the connection as explained in Section 7.1, an EPP command can be served by any machine of a server pool that is able to access the session information, not necessarily by the machine that has previously served the Login command;
- as a consequence of the above statement, the load of a server can be efficiently balanced among all the pool machines and they can be stopped due to maintenance operations without breaking down the live EPP sessions.

4.2 DBAN implementation

Since the DBAN must store all the information about domains, contacts and Registrars and hence plays a fundamental role for the full efficiency of the ccTLD .it, in order to ensure high degrees of performance and resiliency, it has been implemented through a redundant and distributed database.
The database is made up of a MySQL cluster including several nodes: one acting as master and the other ones as slaves. The master node is the only one accepting Read-Write requests, the other ones are Read-Only nodes.
Any access is mediated by a pool of SQL proxies including several MaxScale nodes working as an active-passive cluster. Proxies are in charge of routing the traffic towards the underlying MySQL cluster nodes depending on the query type (i.e. read or write), the instance load and the address of the incoming request.
A complete backup of the DBAN is performed daily while an incremental backup takes place every hour. The daily backup is replied on an external datacenter to comply with the disaster recovery plan.
The architecture of DBAN is shown in the following figure.



Other information is stored on No-SQL databases, such as:

- the configuration parameters of services and applications are maintained by an Apache ZooKeeper cluster;
- the information of the live EPP sessions as well as the Registrars' counts of EPP commands (i.e. Domain Check and Domain Create) subject to limitations are stored on a Redis Sentinel system.

Both the two systems are replicated over a pool of several servers to guarantee the automatic failover.

4.3 EPP server building and deployment

The EPP server has been developed by the ccTLD .it as a J2EE application deployed on a WildFly application server. Each WildFly instance (see the next paragraph) is automatically installed and run inside a Docker container, installed in turn on a virtual machine. The physical servers are supported by the Linux operating system.

As it happens for any application and service of the ccTLD .it, all the source code has been developed in-house by leveraging the functionalities provided by well-founded open-source Java projects like Hibernate, which provides a framework for mapping an object-oriented domain model to a relational database, and Apache Maven, used for building and managing of any Java-based projects, and many others.

4.4 Achieving fault-tolerance and scalability

The EPP server has not been designed as a standalone service but rather as a group of servers operating to achieve the maximum degree of fault-tolerance and scalability.
The HW/SW architecture is described here in the following.

Firstly, the EPP server has been logically split in two distinct services:

- the URL epp.nic.it is used for registering brand new domain names and requesting all other maintenance operations on domains and related contacts;
- the URL epp-deleted.nic.it is used only for registering domain names deleted by less than 7 days.

Each logical EPP server is made up of a load balancer distributing the requests to a pool of real EPP servers. New servers can be easily added to each pool when the response time becomes high.
The load balancer is implemented by a pool of NGINX proxies: one acting as a primary server and the others acting as backup servers taking over as primary automatically if the current primary fails.
NGINX is a web server that allows for load balancing of HTTP traffic across web or application server groups, with several algorithms and advanced features.
Servers' efficiency is monitored through Nagios. Nagios is an open-source computer-software application useful to monitor systems, networks and infrastructure and to provide related alerting services.
The server code is continually verified by an ad-hoc application that implements more than 1,300 developer-side tests for Java source code through the Junit framework. Apache JMeter has instead been used to check the server behavior and measure its performance simulating the overload of requests coming in some specific time ranges (e.g. Drop Time – Section 8.6.2).

4.5 Facing pending operations

The EPP server communicates asynchronously with other applications and services of the ccTLD .it to implement operations which require the execution of additional extra EPP operations. This is done via the most popular open source, multi-protocol, Java-based message broker namely Apache ActiveMQ. As an example of such interaction, it is worth mentioning the validation process of a DNS/DNSSEC configuration. It takes place as in the following:

- each time a request for registering a domain name is submitted by a Registrar to the EPP server, the domain is passed into the inactive/*dnsHold* status and a request to check the DNS/DNSSEC configuration is issued on an ActiveMQ queue whose listener is a DNS configuration checker. The same goes for any already registered domain name whose DNS/DNSSEC configuration needs to be validated with the only difference that the target status is pendingUpdate instead of inactive/*dnsHold*;
- the checker validates the configuration and issues the result on another ActiveMQ queue whose listener is a service in charge of:
  - passing the domain into the ok status and notifying the Registrar that the validation process has ended successfully
  - or, keeping the domain in the inactive/*dnsHold* (or pendingUpdate) and notifying the Registrar of the possible error (notifications occur by inserting a message in the EPP polling queue of the Registrar).
- another service periodically requests the checker to verify again the DNS/DNSSEC configuration

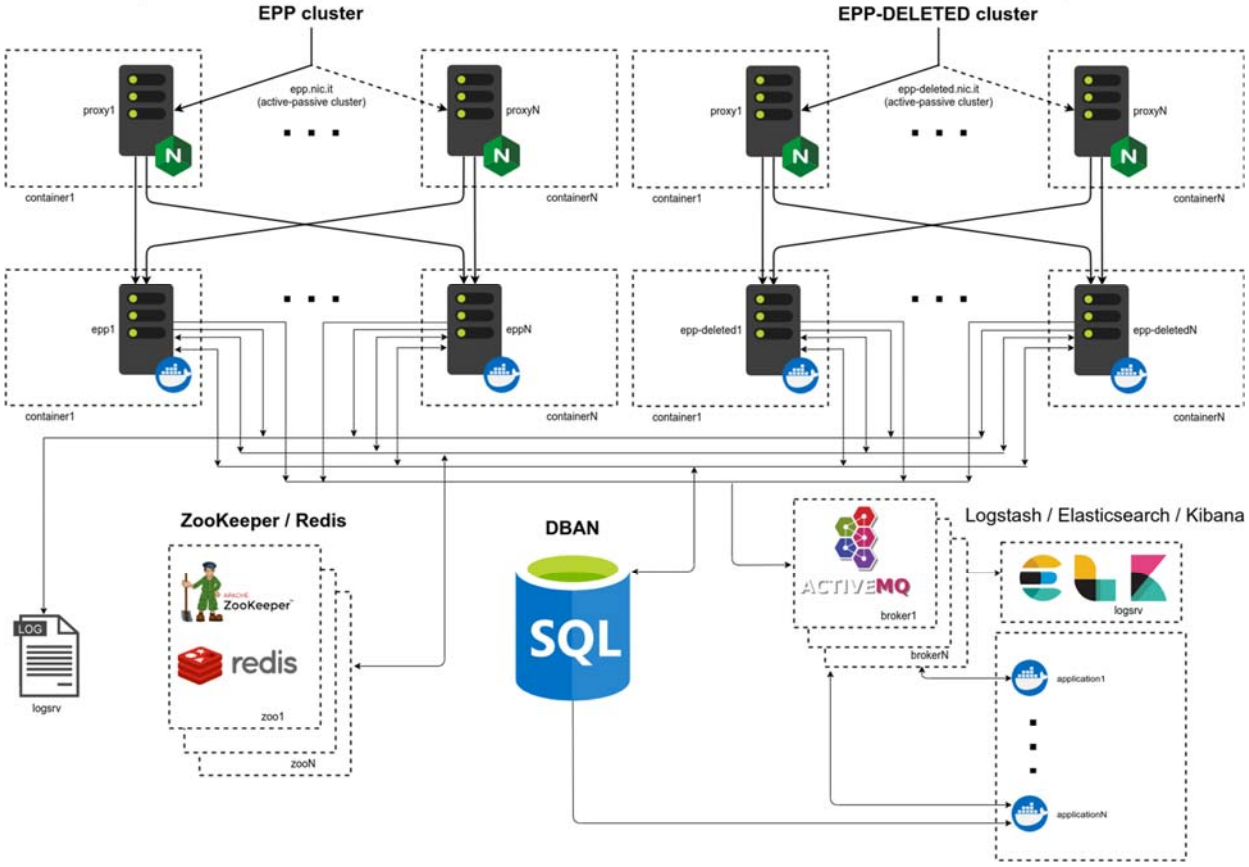
whose validation is still pending.

Even in this case, a pool of brokers is paired together in a master-slave configuration so that if a master fails then one of the slaves takes over thus avoiding the loss of important data.

4.6 Logging

Logs about request and response of the EPP server instances are centralized on a single server. Some information about EPP commands as well as the requests and responses in raw format are issued via Apache ActiveMQ to an ELK stack (i.e. Elastic Search, Logstash, Kibana) to make the users able to easily consult the logs. ElasticSearch is a search and analytics engine. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms them, and then sends them to a "stash" like ElasticSearch. Kibana lets users to visualize data with charts and graphs in ElasticSearch.

The complete HW/SW architecture of the EPP server, including the DBAN and the other services contributing to manage the domain names of the ccTLD .it, is shown in the following figure.



5 Assumptions

5.1 Characters accepted during the process of registration of a .it domain name

The characters accepted during the process of registration of a .it domain name are indicated below:

- ASCII: digits (0-9), letters (a-z) and hyphen (-);
- non-ASCII: all characters belonging to the charset Latin-1 Supplement (see "Appendix C"), Latin Extended-A (see "Appendix D"), Latin Extended-B (see "Appendix E"), Greek (see "Appendix F"), Greek Extended (see "Appendix G") and Cyrillic (see "Appendix H").

The non-ASCII characters indicated above enable the registration of the IDNs in the 24 official languages of the European Union.

5.1.1 Homoglyphs

In order to avoid any problems associated with the use of homoglyphs/homographs (e.g. the ASCII character "c" and the "c" in Cyrillic, although apparently the same, have different Unicode encoding - U +0063 and U +0441, respectively), the .it Registry does not accept requests for domain names that contain characters belonging to different charset (Latin, Greek, Cyrillic), clearly excluding the suffix ".it" and the domain names belonging to the geographical tree.
For more clarity, below some examples of accepted or not accepted domain names are indicated:

Domain name	**Acceptance**
αβγ.it	Yes (all characters belong to the Greek charset)
фдӈ̌.it	No (the first two characters belong to the Cyrillic charset while the third belongs to the Greek charset)
фдf.it	No (the first two characters belong to the Cyrillic charset while the third to the Latin charset)
ӈ̌ββӈ̌.it	No (the third character belongs to the Latin charset, while the others belong to the Greek charset)
αβγ.ge.it	Yes (ge.it belongs to the geographical tree)
αβγ.gε.it	No (gε.it does not belong to the domain names of the geographical tree)
фдф.viareggio.lu.it	Yes (viareggio.lu.it belongs to the geographical tree)

5.1.2 "Remapped" characters

A single non-ASCII character can also represent some combination of characters (ASCII and non-ASCII). For example, in the Greek alphabet, the combination of the characters "ǐ" and "ι" can also be represented by a single character "ΐ".
In order to avoid any problems arising from this situation, every combination of characters present in a domain name referable to a single character, is replaced by the equivalent single character.
Both, the combination of characters and the single characters with which they are replaced, and the Unicode encodings are indicated in the table below.

Combination of characters	Unicode Encoding combination of characters	Remapped character	Unicode Encoding Remapped character
ὰι	U+1F00 U+03B9	ᾀ	U+1F80
άι	U+1F01 U+03B9	ᾁ	U+1F81
ᾶι	U+1F02 U+03B9	ᾂ	U+1F82
ᾶι	U+1F03 U+03B9	ᾃ	U+1F83
ᾰι	U+1F04 U+03B9	ᾄ	U+1F84
ᾱι	U+1F05 U+03B9	ᾅ	U+1F85
ᾲι	U+1F06 U+03B9	ᾆ	U+1F86
ᾳι	U+1F07 U+03B9	ᾇ	U+1F87
ὴι	U+1F20 U+03B9	ᾐ	U+1F90
ήι	U+1F21 U+03B9	ᾑ	U+1F91
ῆι	U+1F22 U+03B9	ᾒ	U+1F92
ῆι	U+1F23 U+03B9	ᾓ	U+1F93
ῄι	U+1F24 U+03B9	ᾔ	U+1F94
῅ι	U+1F25 U+03B9	ᾕ	U+1F95
ῂι	U+1F26 U+03B9	ᾖ	U+1F96
ῃι	U+1F27 U+03B9	ᾗ	U+1F97
ὼι	U+1F60 U+03B9	ᾠ	U+1FA0
ώι	U+1F61 U+03B9	ᾡ	U+1FA1
ῶι	U+1F62 U+03B9	ᾢ	U+1FA2
ῶι	U+1F63 U+03B9	ᾣ	U+1FA3
ῲι	U+1F64 U+03B9	ᾤ	U+1FA4
ῳι	U+1F65 U+03B9	ᾥ	U+1FA5
ῷι	U+1F66 U+03B9	ᾦ	U+1FA6
ῷι	U+1F67 U+03B9	ᾧ	U+1FA7
ὰι	U+1F70 U+03B9	ᾲ	U+1FB2
Αι	U+03B1 U+03B9	ᾳ	U+1FB3
Άι	U+03AC U+03B9	ᾴ	U+1FB4
ᾶι	U+1FB6 U+03B9	ᾷ	U+1FB7
ὴι	U+1F74 U+03B9	ῂ	U+1FC2
Ηι	U+03B7 U+03B9	ῃ	U+1FC3
Ήι	U+03AE U+03B9	ῄ	U+1FC4
ῆι	U+1FC6 U+03B9	ῇ	U+1FC7
ὼι	U+1F7C U+03B9	ῲ	U+1FF2
Ωι	U+03C9 U+03B9	ῳ	U+1FF3
Ώι	U+03CE U+03B9	ῴ	U+1FF4
ῶι	U+1FF6 U+03B9	ῷ	U+1FF7

For example, if you wish to register the domain name ἀιἀιἀιἀι.it, as the combination ἀ (U+1F00) and ι (U+03B9) is replaced by the single equivalent character ᾀ (U+1F180); the domain name that is actually possible to request and register becomes ᾀᾀᾀᾀ.it.
Obviously, the constraint on the minimum length of a domain name is applied after the replacement of the combination of characters (if occurred). Therefore, a registration request for the domain ἀιἀι.it will be rejected by the system, since after the replacement of the remapped characters; the domain name becomes ᾀᾀ.it and no longer meets the requirement of a minimum length of 3 characters.
If a Registrar wishes to request the registration of a domain name containing any of the combination of characters indicated above, the server will register the domain name containing the equivalent single character in replacement of the combination of characters. Then the server will notify the Registrar of the successful replacement, both in the response of the Domain Create command and in the polling queue by means of a dedicated message.
For the implementation of this new feature, a new extension called "remappedIdnData" has been defined. This contains the IDN domain name requested by the Registrar and the one actually recorded as a result of the transformation. This extension is defined in the schema file extdom-2.0.

5.2 Statuses

A status characterizes the current operational condition of an object and its possible future transitions.
In the ccTLD .it registration system there is a subdivision between the actual statuses of a domain name and the additional constraints that are applicable to the statuses themselves, which may affect the acceptance and processing of commands. In the registration system of .it domain names the concept of "multi-status" of a domain name is highlighted and used. This means that, at every moment of the life cycle of a domain name, it cannot be associated with a single status but to a combination of statuses that determine both its current situation and the constraints imposed by the Registrar or the Registry. These constraints impose the operations a domain name can be subject to.
The same considerations also apply to contact objects although the transitions to which contacts are subject to in the ccTLD .it are much simpler than those defined for domain names.

5.2.1 Domain name statuses

Status	**Description**	**DNS**
ok	Domain name registered, active and available for any operation.	Delegated
inactive/*dnsHold*	Domain name registered but inactive and available for any operation. This status is attributed to domain names for which the periodic checks of the DNS configuration, performed by the Registry, is not positive.	Not delegated
pendingUpdate	Domain name for which a change of the authoritative name servers or of the Delegation Signer (DS) records has been requested and a positive DNS configuration check is pending. The domain name remains in that status for a maximum of 5 (five) days. If the new DNS configuration is not positively validated by the Registry within this term, it is abandoned. The domain name returns to the previous status.	Delegated

inactive/clientHold	Domain name for which the Registrar has suspended operations and inhibited any transaction, following filing of legal proceedings regarding the use and/or assignment of the domain name. The only operation permitted is the removal of "clientHold" by the Registrar.	Not delegated
inactive/serverHold	Domain name for which the Registry recognizes the need to make the domain name inactive; the data in the DBAN associated with domain name remain unchanged. Domain name on which the .it Registry has concluded negatively a Verification Procedure as well as the .it Registry has received either an order sent by the authorities and notified in accordance with the law or the request of the Registrant whose use of the domain name has been judicially challenged.	Not delegated
pendingDelete/ redemptionPeriod	Registered domain name for which the Registrar has asked the deletion on behalf of the Registrant. The only operation permitted is the recovery of the domain name by the Registrar within 30 (thirty) days of the date of transfer to that status.	Not delegated
pendingTransfer	Domain name for which a "Change of Registrar" is in progress. The "old" Registrar may reject the "Change of Registrar" only if a notice regarding the domain name has been duly served by the relevant authorities. The domain name remains in this status for a maximum of 1 (one) day. If the "Change of Registrar" is not explicitly cancelled by the new Registrar or rejected by the old one within this term, it will be automatically approved by the Registry.	Depends on the statuses with which it is associated
pendingTransfer/*bulk*	Domain name for which a Bulk Transfer operation is underway.	Depends on the statuses with which it is associated
autoRenewPeriod	Identifies the period of 15 (fifteen) days immediately after expiry of the domain name.	Depends on the statuses with which it is associated
clientDeleteProhibited	Constraint imposed by the Registrar to prevent deletion of a domain name. The Registrar may not raise this limit to prevent the Registrant from requesting deletion of a domain name, unless valid grounds are provided.	Depends on the statuses with which it is associated
clientUpdateProhibited	Constraint set by the Registrar to prevent a domain name from being changed. The Registrar may not set this constraint to prevent	Depends on the statuses with which it is

	the Registrant from requesting a change of domain name, unless valid grounds are provided.	associated
clientTransferProhibited	Constraint set by the Registrar to prevent the transfer of a domain name to another Registrar. If the Registrar, based on a decision of the administrative, judicial or police authorities, notified in the proper form, or a decision relating to the use and/or assignment of domain names maintained by them, is about to put the domain name into "clientTransferProhibited" or, simultaneously, into "clientDeleteProhibited/ clientUpdateProhibited/ clientTransferProhibited", then said Registrar is required to notify the Registry of the decision in question, together with a copy for verification purposes, however without this preventing the Registrar from arranging for what may be needed. The Registrar may only reject the "Change of Registrar" if a notice, for said domain name, has been duly served by the competent authorities.	Depends on the statuses with which it is associated
clientDeleteProhibited/ clientUpdateProhibited/ clientTransferProhibited	Constraints set by the Registrar to prohibit any transaction to the domain name, following the filing of legal proceedings on the domain name regarding its use and/or assignment. If the Registrar, based on a decision of the administrative, judicial or police authorities, notified in the proper form, or a decision relating to the use and/or assignment of domain names maintained by them, is about to put the domain name into "clientTransferProhibited" or, simultaneously, into "clientDeleteProhibited/ clientUpdateProhibited/ clientTransferProhibited", then said Registrar is required to notify the Registry of the decision in question, together with a copy for verification purposes, however without this preventing the Registrar from arranging for what may be needed. The only operation permitted is the removal of the "client" constraints by the Registrar.	Depends on the statuses with which it is associated
serverDeleteProhibited	Constraint set by the Registry to prevent deletion of a domain name.	Depends on the statuses with which it is associated
serverUpdateProhibited	Constraint set by the Registry to prevent change of a domain name.	Depends on the statuses with which it is associated
serverTransferProhibited	Constraint set by the Registry to prevent the transfer of a domain name to another Registrar.	Depends on the statuses with which it is

		associated
serverDeleteProhibited/ serverUpdateProhibited/ serverTransferProhibited	Constraints set by the Registry to prohibit any transaction to the domain name.	Depends on the statuses with which it is associated
pendingDelete/pendingDelete	Identifies the period preceding final deletion of the domain name from the Registry's DBAN, which must take place within the Drop Time period, established and publicly notified on the web site of the Registry. Domain names are also placed in this status when they are deleted by either the Registrar or the Registry as a result of the Verification Procedure.	Not delegated
challenged	Domain name contested by a third party, not available for the "Change of Registrant". It identifies all domain names for which a challenge procedure is active.	Depends on the statuses with which it is associated
inactive/*revoked*	Domain name revoked by the Registry and not immediately available for free assignment.	Not delegated
inactive/*toBeReassigned*	Domain name for which a reassignment or challenge procedure has ended successfully. The domain name can be registered within 30 (thirty) days by the opposing party.	Not delegated
ok/*noRegistrar* inactive/*dnsHold*/*noRegistrar*	Domain name for which the Registrar has no active contract with the Registry. The domain name retains this status until the date in the expire field. The only operations permitted are: "Change of Registrar" (together with a "Change of Registrant", if required) by the Registrant and recovery of the domain name by the same Registrar in case it has a new contract with the Registry.	Depends on the statuses with which it is associated
inactive/*noRegistrar*	Domain name for which the Registrar no longer has an active contract with the Registry or for which a "Change of Registrar" operation gave a negative result beyond the autoRenewPeriod. The domain name has reached the date in the "expire" field and retains this status for 60 (sixty) days. The only operations permitted are: "Change of Registrar" (together with "Change of Registrant", if required) by the Registrant and recovery of the domain name by the same Registrar in case it has a contract with the Registry.	Not delegated
inactive/*notRenewed*	Domain name that has reached the expire date and has not been renewed automatically due to insufficient credit on the part of the Registrar. Domain names in this status are recovered	Not delegated

	automatically as soon as the Registrar's credit allows. The domain name retains this status for 30 (thirty) days. The only operations permitted are: automatic recovery by the system as soon as the Registrar's credit allows and "Change of Registrar" (together with "Change of Registrant", if required) by the Registrant.	
reserved	Non-registered domain name reserved for a particular Registrant.	Not delegated
unassignable	Domain name that cannot be assigned to any registrants.	Not delegated
deleted	Domain name for which a deletion operation has ended. The domain name is freely available for assignment.	Not delegated

5.2.2 Contact statuses

Status	**Description**
ok	Contact that has been registered in the DBAN but it has not been referenced by a domain name yet. There are no constraints on the contact.
linked	Contact that has been registered in the DBAN and is referenced by at least one domain name. There are no constraints on the contact apart from the implicit one, which cannot be deleted as it is currently referenced by at least one domain name.
clientDeleteProhibited	Constraint set by the Registrar to prevent the deletion of a contact.
clientUpdateProhibited	Constraint set by the Registrar to prevent the update of a contact.
serverDeleteProhibited	Constraint set by the Registry to prevent the deletion of a contact.
serverUpdateProhibited	Constraint set by the Registry to prevent the update of a contact or its association with a domain name.

5.3 Server URLs

The commands described in the following of this document should be submitted to the "epp.nic.it" Registry server except for the registration of domain names that have been cancelled for less than 7 (seven) days. In this case the request must be sent to the "epp-deleted.nic.it" server (Section 8.1.3).

5.4 Server limits

A maximum of 5 static IP addresses is allowed to access the above-mentioned servers. The maximum number of sessions that can be open at the same time is 5, irrespectively of the number of clients used. The table in Section 12.8shows the maximum number of commands to verify whether a particular

domain name is registered in the DBNA that a Registrar can send in one day to the "epp.nic.it" and "epp-deleted.nic.it" servers. This table also indicates the maximum number of daily requests for the registration of domain names that have been cancelled within less than 7 (seven) days that Registrars can send to the "epp-deleted.nic.it" server.

6 EPP (Extensible Provisioning Protocol)

EPP is a stateful XML application protocol where the information exchanged between client and server includes the concept of status, and both systems keep track of the status of the communication session. The protocol can be supported by different transport protocols (e.g. TCP, HTTP).
Initially the clients identify themselves to the server, using secure connections, and are authenticated, and then exchange with the server information on the services implemented and the objects managed. The clients then start the sessions based on a "request-response" communication pattern.
All the EPP commands are atomic (there is no partial success or failure even though the effect of a transaction might not be completed by the conclusion of the corresponding request).
The basic elements of the protocol are:

- Identification of the services offered by the server
- Commands
- Responses
- Extensions to the protocol

The EPP uses XML namespaces to provide an extensible management paradigm of objects, and for identifying XML Schemas required to parse and validate the XML content both of the base protocol and of any extensions.

6.1 XML Schemas supported by Registry's EPP server

All definitions of commands or objects used in the EPP implementation by the Registry's server are included in XML Schemas.
Since EPP is extensible, all extensions to the standard are, in turn, described in other XML Schemas.
It is necessary therefore that the client of the server supports the following XML Schemas:

- standard XML Schema of EPP:
  - *epp-1.0.xsd*: EPP schema v1.0 (RFC5730)
  - *eppcom-1.0.xsd*: EPP shared structures schema v1.0 (RFC5730)
  - *domain-1.0.xsd*: EPP domain provisioning schema v1.0 (RFC5731)
  - *contact-1.0.xsd*: EPP contact provisioning schema v1.0 (RFC5733)
  - *host-1.0.xsd*: EPP host provisioning schema v1.0 (RFC5732). This schema is recognized but not supported.
- XML Schema that cover IETF extensions:
  - *rgp-1.0.xsd*: EPP domain name extension schema v1.0 for Registry grace period processing (RFC3915)
  - *secDNS-1.1.xsd*: EPP Domain Name System (DNS) Security extension schema v.1.1 (RFC5910)
- XML Schema that cover extensions defined by the Registry:
  - *extepp-2.0.xsd*: IT-NIC EPP extension schema v2.0.

- *extcon-1.0.xsd*: IT-NIC EPP contact extension schema v1.0
- *extdom-2.0.xsd:* IT-NIC EPP domain extension schema v2.0
- *extsecDNS-1.0.xsd*: IT-NIC EPP DNSSEC extension schema v1.0
- *extgovcon-1.0.xsd*: IT-NIC EPP contact extension schema v1.0 for SLD gov.it
- *extgovdom-1.0.xsd*: IT-NIC EPP domain extension schema v1.0 for SLD gov.it

6.2 EPP Commands

The EPP commands are processed by the server in the order they are received from the client.
For each request issued by the client, the server sends an immediate response that confirms the receipt and processing of the request.
There are three categories of EPP commands that the Registrar (the client) may submit to the Registry (the server):

- commands for managing the session (login, logout, hello);
- commands for querying the server to obtain information on domain names and registered contacts as well as the presence of messages in the Registrar's polling queue (i.e. the queue of messages that the Registrar receives from the Registry);
- commands for registering and maintaining domain names and contacts associated with them.

In cases where the request must continue offline, in addition to sending the response, the server will notify the client that the command was received and processed, but that the requested action is not yet complete. Subsequently, the server notifies the client that the offline processing is complete.
Request and response examples will be shown for each command in the following of this document.

6.3 Structure of EPP requests and responses

Each request issued by the client contains the following elements:

- An initial standard header

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

- An element that may be of two types:
  - `<hello>`: to send a hello;
  - `<command>`: to send the any other request. This element, together with other elements of the command, also contains the following:
    - An optional element `<extension>` which can be used for the extensions defined by the server to the requests of the commands
    - An optional element `<clTRID>` (client transaction identifier) that can be used by the client to logically identify a transaction. It is an alphanumeric string with a minimum length 3 and maximum of 64 characters (e.g. `<clTRID>ABC-12345</clTRID>`)
- A final standard element `</epp>`

Each response from the server contains the following elements:

- An initial standard header

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp
```

```
xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
```

- A number of different elements depending on the request received:
  - in response to an <hello> command:
    - a <greeting> element
  - in response to any other command:
    - one or more <result> elements that document the success or failure of command execution. If the command was processed successfully, only one <result> element must be returned. Conversely, in the event of failure, multiple <result> elements may be returned to document failure conditions. Each <result> element contains the following attribute and child elements:
      - an attribute code that describes the success or failure of the command;
      - a <msg> element with a human-readable description of the response code in the language specified by the optional lang attribute;
      - zero or more <value> elements that identify a client-provided element (including XML tag and value) or other information that caused a server error condition;
      - zero or more <extValue> elements that can be used to provide additional diagnostic information, containing in turn:
        - a <value> element that identifies a client-provided element (including XML tag and value) that caused a server error condition;
        - a <reason> element with a human-readable description of the reason for the error in the language specified by the optional lang attribute;
    - an optional <msgQ> element that describes the messages in the polling queue of the Registrar. If the polling queue is empty, <msgQ> must not be present. If the queue is not empty, it must be present in response to a Poll Req and may be present in commands other than Poll Req. Each <msgQ> element contains the following:
      - a count attribute that describes the number of messages that exist in queue;
      - an id attribute used to uniquely identify the message at the head of the queue;
      - a <qDate> element, in the response to the Poll Req command, which contains the date when the message was enqueued;
      - a <msg> element, in the response to the Poll Req command, which contains a human-readable description of the message in the language specified by the optional lang attribute.
    - An optional <resData> element that contains child elements specific to the command and associated object.
    - An optional <extension> element that may be used for server-defined response extensions.
    - A <trID> (transaction identifier) element containing the transaction identifier assigned by the server to the command for which the response is being returned.
    - A final standard </epp> element.

    - Two new elements were defined in the namespace extepp-2.0 in order to obtain a more structured error message in comparison with the previous version:
      - the <wrongValue> element within the <value> element is structured as follows:
        - <element>: containing the name of the missing tag or with a wrong value;
        - <namespace>: containing the namespace of the missing tag or with a wrong value;
        - <value>: containing the wrong value;
      - the <reasonCode> element within the <value> element of the <extValue> element that contains the reason code of the implementation of the .it Registry.

6.3.1 Example of a response with message in queue

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <msgQ id="227" count="1">
      <qDate>2013-02-21T14:20:00+01:00</qDate>
      <msg lang="en">DNS check ended successfully</msg>
    </msgQ>
    <trID>
      <svTRID>f1a98ba5-08f9-4ea8-8e6c-e798c6af3cad</svTRID>
    </trID>
  </response>
</epp>
```

7 Commands for managing the work sessions

Before starting a working session with the EPP server, the Registrar must have asked the Registry for the following information:

- EPP server address;
- username for login (<clID>);
- password to be used in the login (<pw>).

The Registrar must notify, through RAIN-NG, the Registry of the physical address of the machines that will host the Registrar's EPP clients.
A normal dialog between a client and the Registry's EPP server has the following sequence of actions:

- The client connects to the server via a secure SSL connection via HTTPS
- The server responds by identifying itself and presenting the commands and extensions that it supports
- The client logs in specifying name and password,
- The client periodically queries its polling queue to check and collect any messages from the server
- The client sends commands to the server, which responds immediately
- The client ends the session

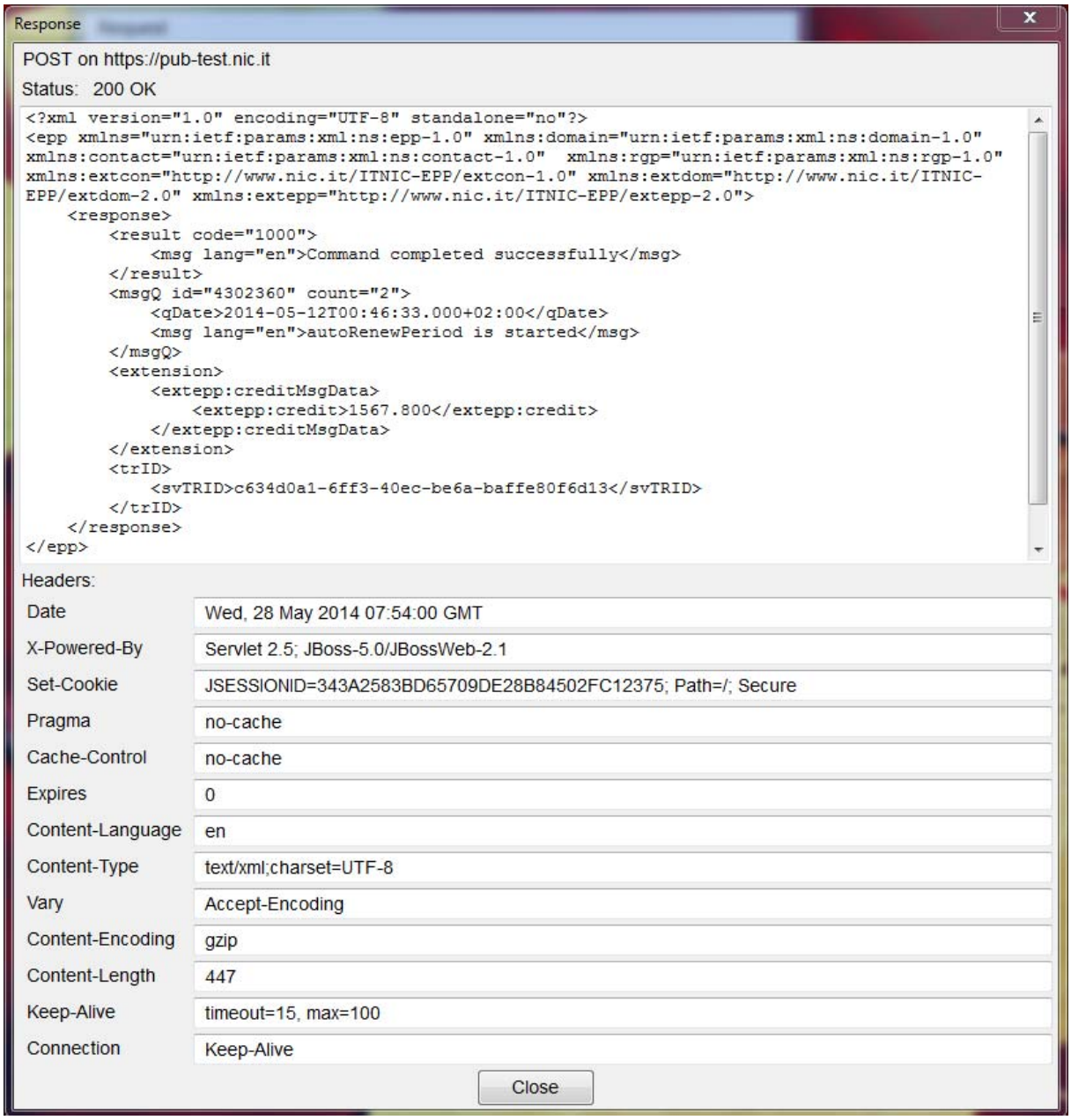
Consequently, in addition to commands for querying and managing contacts (Section 10.1) and domains (Section 10.2) and the polling queue (Sections 0and 12.4), there are also commands for managing the connection.

EPP provides three commands to manage a work session:

- Login
- Logout
- Hello

7.1 EPP session

EPP is an applicative protocol with session (stateful). In the implementation of ccTLD .it, it is based upon the HTTPS protocol which viceversa is without session (stateless).
An EPP session starts with a successful Login command (Section 7.2) and ends with a successful Logout command (Section 7.3).
To enable clients to submit EPP commands within a session, the server generates and inserts in the cookie of the Login response a session identifier (JSESSIONID).
Therefore, the EPP clients should be able to manage the session identifier returned in the Login response for the following EPP commands till Logout.
The Hello command (Section 7.4) is the only one that can be sent out of the scope of an EPP session. If the registrar wants to periodically submit the Hello command to keep a session alive, it should use a session identifier generated previously for the session otherwise the session will expire due to timeout expiration.
The EPP server of ccTLD .it does not accept Login or Hello (if it is submitted out of the scope of an opened session) that contain a session identifier already set.
In the following, it is reported an example of a Login response with a session identifier created by the server and identified by the JSESSIONID parameter within the cookie.
The example has been realized by using the tool Poster of Mozilla FireFox.



7.2 Login

The Login request is used by the client to start a working session with the EPP server. The following table displays the fields in the request for Login:

Field	Description	XML Tag	XML Attribute Tag	Cardinality	Length	Notes
Registrar's ID	Unambiguous Registrar identifier	clID		1	1-16	Alphanumeric value given by Registry to Registrar

Password	Registrar's authentication password	pw		1	6-16	Alphanumeric value defined by Registrar
New Password	New authentication password of Registrar	newPW		0-1	6-16	Alphanumeric value defined by Registrar to change his own password
Server version	Currently active server version	version		1		Current server version is "1.0"
Language	Language chosen for server to client messages	lang		1		Allowed values: en (default), it
Objects namespace URIs	URIs of the namespaces of standard EPP that represent the objects that will be dealt with in the session	objURI		2		Allowed namespaces: contact-1.0 domain-1.0
Extension namespace URIs	URIs of the namespaces of EPP extensions	extURI		4		Allowed namespaces: extepp-2.0 extcon-1.0 extdom-2.0 rgp-1.0 secDNS-1.1 extsecDNS-1.0 extgovcon-1.0 extgovdom-1.0

The Login response contains the Registrar's available credit. This extension is not available for the version of the EPP server used for the accreditation test.

7.2.1 Examples of Login requests from a "non DNSSEC accredited" Registrar

Example 1
Example of Login request from a "non DNSSEC accredited" Registrar:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <login>
      <clID>DEMO-REGISTRAR</clID>
      <pw>14nov07</pw>
      <options>
        <version>1.0</version>
        <lang>en</lang>
      </options>
      <svcs>
        <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>
        <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>
        <svcExtension>
          <extURI>http://www.nic.it/ITNIC-EPP/extepp-2.0</extURI>
```

```
          <extURI>http://www.nic.it/ITNIC-EPP/extcon-1.0</extURI>
          <extURI>http://www.nic.it/ITNIC-EPP/extdom-2.0</extURI>
          <extURI>urn:ietf:params:xml:ns:rgp-1.0</extURI>
        </svcExtension>
      </svcs>
    </login>
  </command>
</epp>
```

Example 2
Example of Login request with password change from a “non DNSSEC accredited” Registrar:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <login>
      <clID>DEMO-REGISTRAR</clID>
      <pw>14nov07</pw>
      <newPW>25xyz$64</newPW>
      <options>
        <version>1.0</version>
        <lang>en</lang>
      </options>
      <svcs>
        <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>
        <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>
        <svcExtension>
          <extURI>http://www.nic.it/ITNIC-EPP/extepp-2.0</extURI>
          <extURI>http://www.nic.it/ITNIC-EPP/extcon-1.0</extURI>
          <extURI>http://www.nic.it/ITNIC-EPP/extdom-2.0</extURI>
          <extURI>urn:ietf:params:xml:ns:rgp-1.0</extURI>
        </svcExtension>
      </svcs>
    </login>
  </command>
</epp>
```

Example 13
Example of Login request from a Registrar opening an EPP session to make operations on both gov.it contacts and domains.:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <login>
      <clID>DEMO-REGISTRAR</clID>
      <pw>14nov07</pw>
      <options>
        <version>1.0</version>
        <lang>en</lang>
      </options>
      <svcs>
        <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>
        <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>
        <svcExtension>
          <extURI>http://www.nic.it/ITNIC-EPP/extepp-2.0</extURI>
          <extURI>http://www.nic.it/ITNIC-EPP/extcon-1.0</extURI>
          <extURI>http://www.nic.it/ITNIC-EPP/extdom-2.0</extURI>
          <extURI>urn:ietf:params:xml:ns:rgp-1.0</extURI>
          <extURI>http://www.nic.it/ITNIC-EPP/extgovcon-1.0</extURI>
```

```
          <extURI>http://www.nic.it/ITNIC-EPP/extgovdom-1.0</extURI>
        </svcExtension>
      </svcs>
    </login>
  </command>
</epp>
```

7.2.2 Example of Login request from a "DNSSEC accredited" Registrar

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <login>
      <clID>DEMO-REG</clID>
      <pw>14nov07</pw>
      <options>
       <version>1.0</version>
       <lang>en</lang>
      </options>
      <svcs>
       <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>
       <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>
       <svcExtension>
         <extURI>http://www.nic.it/ITNIC-EPP/extepp-2.0</extURI>
         <extURI>http://www.nic.it/ITNIC-EPP/extcon-1.0</extURI>
         <extURI>http://www.nic.it/ITNIC-EPP/extdom-2.0</extURI>
         <extURI>urn:ietf:params:xml:ns:rgp-1.0</extURI>
         <extURI>urn:ietf:params:xml:ns:secDNS-1.1</extURI>
         <extURI>http://www.nic.it/ITNIC-EPP/extsecDNS-1.0</extURI>
       </svcExtension>
      </svcs>
    </login>
  </command>
</epp>
```

7.2.3 Examples of Login responses to requests from a "non DNSSEC accredited" Registrar

Example 1
Response to a Login request with charge of operations not activated:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <trID>
      <svTRID>cfec00d5-5a14-4ee0-b3ca-4e2339bc6119</svTRID>
    </trID>
  </response>
 </epp>
```

Example 2
Response to a Login request with charge of operations activated:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <extension>
      <extepp:creditMsgData>
        <extepp:credit>48739.112</extepp:credit>
      </extepp:creditMsgData>
    </extension>
    <trID>
      <svTRID>898b1da3-e544-4c47-9251-6f621d4ae37a</svTRID>
    </trID>
   </response>
  </epp>
```

7.2.4 Example of Login response to a request from a "DNSSEC accredited" Registrar

In the case that the Login request is sent by a "DNSSEC accredited" Registrar who, in the request, has indicated the two namespaces related to the DNSSEC extensions, the response contains the following header:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp
 xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extsecDNS="http://www.nic.it/ITNIC-EPP/extsecDNS-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0">
```

7.2.5 Example of Login response to a request from a Registrar who opened a session to make operations on both gov.it contacts and domains

In the case that the Login request is sent by a Registrar who, in the request, has indicated the two namespaces related to the gov.it extensions, the response contains the following header. The example below regards a Login request submitted by a "non DNSSEC accredited" Registrar.

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

```
 xmlns:rgp=”urn:ietf:params:xml:ns:rgp-1.0”
 xmlns:extcon=”http://www.nic.it/ITNIC-EPP/extcon-1.0”
 xmlns:extdom=”http://www.nic.it/ITNIC-EPP/extdom-2.0”
 xmlns:extepp=”http://www.nic.it/ITNIC-EPP/extepp-2.0”
 xmlns:extcon=”http://www.nic.it/ITNIC-EPP/extgovcon-1.0”
 xmlns:extdom=”http://www.nic.it/ITNIC-EPP/extgovdom-1.0”>
```

7.3 Logout

The Logout request is used by client to end a work session with the EPP server. The server, upon expiry of the timeout, may close a session opened with a client after persistent inactivity.

7.3.1 Logout request

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
 <command>
   <logout/>
 </command>
</epp>
```

7.4 Hello

The Hello request is used for two different purposes and can be sent:

- before a Login to query an EPP server on the services implemented and the objects manipulated by the EPP commands;
- during a working session to keep the session active and prevent the client from being disconnected due to timeout.

The Hello request provides a single empty `<hello>`.
The server responds to a hello request with a `<greeting>` which contains all the information needed to start a working session:

- name of the server;
- current date of the server;
- current version of the server;
- languages;
- URI of the namespace of the EPP objects that can be manipulated;
- URI of the namespace of any extensions;
- a section, expressed by `<dcp>` (data collection policy), which lists the security policies implemented by the server for access and data management.

7.4.1 Hello request

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

```
</epp>
```

7.4.2 Greeting response of Registry's EPP server

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <greeting>
    <svID>NIC-IT EPP Registry</svID>
    <svDate>2013-02-22</svDate>
    <svcMenu>
      <version>1.0</version>
      <lang>en</lang>
      <lang>it</lang>
      <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>
      <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>
      <svcExtension>
        <extURI>http://www.nic.it/ITNIC-EPP/extepp-2.0</extURI>
        <extURI>http://www.nic.it/ITNIC-EPP/extcon-1.0</extURI>
        <extURI>http://www.nic.it/ITNIC-EPP/extdom-2.0</extURI>
        <extURI>urn:ietf:params:xml:ns:rgp-1.0</extURI>
        <extURI>http://www.nic.it/ITNIC-EPP/extgovcon-1.0</extURI>
        <extURI>http://www.nic.it/ITNIC-EPP/extgovdom-1.0</extURI>
      </svcExtension>
    </svcMenu>
    <dcp>
      <access>
        <all ns9:type="ns10:string"
xmlns:ns9="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns10="http://www.w3.org/2001/XMLSchema"/>
      </access>
      <statement>
        <purpose>
          <admin ns9:type="ns10:string"
      xmlns:ns9="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:ns10="http://www.w3.org/2001/XMLSchema"/>
          <prov ns9:type="ns10:string"
      xmlns:ns9="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:ns10="http://www.w3.org/2001/XMLSchema"/>
        </purpose>
        <recipient>
          <ours>
          </ours>
          <public ns9:type="ns10:string"
xmlns:ns9="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns10="http://www.w3.org/2001/XMLSchema"/>
        </recipient>
        <retention>
          <stated ns9:type="ns10:string"
xmlns:ns9="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns10="http://www.w3.org/2001/XMLSchema"/>
        </retention>
      </statement>
    </dcp>
  </greeting>
```

```
</epp>
```

8 Operations allowed on domain names

There are two categories of operations allowed on the domain names:

- operations carried out by the Registrar on its own behalf, on behalf of the Registrant, or at the request of a competent Authority;
- operations carried out by the Registry or at the request of a competent Authority.

The main operations provided in the registration system of the .it Registry are as follows:

- registration and renewal;
- change of the Registrant;
- change of the Registrar (with or without a simultaneous change of the Registrant);
- deletion of a contact or a domain name;
- recovery of a domain name;
- revocation of a domain name (at the request of a competent Authority or by the Registry).

In addition to the above is a "Bulk Transfer", that is the transfer of a considerable number of domain names between two Registrars.

8.1 Registering new domain names

Requests for registration are made via EPP by the Registrar for its own account or for the Registrant.
The assignment of domain names in the ccTLD .it is on a "first come first served" basis.
The date and time of registration of a domain name coincide with the date and time of entry in the DBAN of a request that is syntactically and semantically correct.
In order to register new domain names, the Registrar must first register all the contacts referenced in the new domain names (if not yet present in the Registry Database), that is to say the Registrant ("registrant"), the administrative contact ("admin"), and the technician contacts ("tech"). The Registrar can then proceed with the registration of new domain names using the previously registered contacts.
The registrations are made through the use of EPP Create command. Depending on the type of object to register, it is referred to as Contact Create for contacts, or Domain Create for domain names.
The creation of the hosts associated with a domain name, as with changes to them, takes place at the same time as the Domain Create and Domain Update commands. The host object is seen as a property of the domain object and thus the transactions on the host are not implemented.

8.1.1 Contact Create

The registration of a new contact is made by using Contact Create command.

8.1.1.1 Contact object fields required

The Contact Create request needs the following information:

- **ID of the contact** (with a specific format described below)
- **Just one PostalInfo** organized as follows:
  - **Name**
  - Organization (required if the Registrant is not a natural person)
  - **Address** organized as follows:
    - **Street/Square 1**
    - Street/Square 2
    - Street/Square 3
    - **City**
    - **Province**
    - **Postcode (postal code)**
    - **Country (country code)**
- Phone
  - Phone extension
- Fax
  - Fax extension
- Email
- Contact AuthInfo
- **ConsentForPublishing**
- Registrant Data (only required for Registrants)
  - *Nationality (nationalityCode)*
  - *EntityType*
  - *RegCode*
  - *SchoolCode (only required for edu.it Registrants)*
- Gov.it Registrant Data (*only required for gov.it Registrants*)
  - *IpaCode*
  - *UoCode*

The ID of the contact is an alphanumeric code that uniquely identifies a contact ("registrant", "admin" and "tech") within the Registry Database.
The characters accepted are: letters (a-z A-Z), hyphen (-) and digits (0-9).
The AuthContact Info, i.e. the password for the authorization of the request for specific transactions, is ignored by the server. However, since it is a required field, it must always be filled, if necessary just with an empty string value (Section 8.1.1.3).
The section relating to Registrant data is required only if the contact to register is the same as the Registrant of the domain names.
If the section concerning the Registrant is not filled in, the contactID to be registered can only be referenced as a technical or administrative contact of the domain name. If, however, the section relating to the Registrant is filled in, the same contactID can be used to reference both the Registrant of domain names and the administrative contact and/or technical contact.

The following table shows the contact object fields and their correspondence with the XML tag of the request:

Field	Description	XML Tag	XML Tag Attribute	Cardinality	Length	Value
Contact ID	Unambiguous identifier of the contact (technical, administrative or registrant)	contact:id		1	1-16	Alphanumeric value given by Registrar. Characters accepted are: letters (a-z A-Z), hyphen (-) and numbers (0-9).
PostalInfo		contact:postalInfo (only type="loc")		1		
Name	First and family name of the contact	contact:name		1	2-255	
Organization	Identifies the organization which the contact belongs to	contact:org		0-1	2-255	Mandatory only if the Registrant is not a natural person (EntityType <> 1). It must be equal to the Name value if the Registrant is a natural person (EntityType = 1). If Nationality = IT, it must contain at least one alphabetical character. If Nationality <> IT, it must contain at least one alphanumeric character.
Address	Postal address: street, city, province, post code, country	contact:addr		1		
Street/Square	Name of street or square plus house/office number; of the registered office / residence of the contact	contact:street		1-3	1-128	A max. of 3 street fields can be filled.
City	Name of the city of the registered office / residence of the contact	contact:city		1	1-128	

Province	Initials of the province or name of the foreign country relating to the registered office / residence of the contact	contact:sp		1	1-128	If Country = IT, the province must contain the two letters corresponding to an Italian province.
Post code	Postcode of the city relating to the registered office / residence of the contact	contact:pc		1	1-16	Post code.
Country	Acronym of the country relating to the registered office / residence of the contact	contact:cc		1	2	The ISO 3166-1 code of the country must be given (e.g. IT, FR). This coincides with the value of the Nationality field if the Registrant is not a natural person (EntityType <> 1).
Telephone	Telephone number of the contact	contact:voice		1		ISO international format must be used (e.g. +39.050315000).
Telephone extension number	Extension number	contact:voice	x	0-1	1-10	A value with a maximum of ten digits must be inserted.
Fax	Fax number of the contact	contact:fax		0-1		ISO international format must be used (e.g. +39.050315000).
Fax extension number	Extension Number	contact:fax	x	0-1	1-10	Max. 10 (ten) digits can be inserted.

Email	Contact's email address	contact:email		1		Use the RFC2822 and following format (e.g.: user@domain.it). If the email address contains, on the right of the "@" character, non-ASCII characters (Section 5.1), it must be indicated in Punycode format (e.g.: "postmaster@xn--citt-3na.it" and not "postmaster@città.it").
AuthInfo	Identifies the authorization password for specific operations on the contact	contact:authInfo		1		Ignored by server. Can be present and can be empty.
Disclose		contact:disclose		0-1		Ignored by server.
ConsentForPublishing	Boolean value to allow the publication of contact's personal data	extcon:consentForPublishing		1	1	Values allowed: true/1 to allow false/0 to deny In case of "registrant" contacts other than natural persons, individual firms and professional persons, the value of this field must always be true/1.
Registrant's Data		extcon:registrant		0-1		

Nationality	Identifies contact's nationality	extcon:nationalityCode		1	2	Must be one of the ISO 3166-1 codes corresponding to the nationality of the Registrant (e.g.: IT, FR). If the Registrant is not a natural person (EntityType <> 1) it must be equal to the Nation value.
EntityType	Numeric value to identify Registrant typology	extcon:entityType		1	1	Allowed values: 1. Italian and foreign natural persons 2. Companies/one man companies 3. Individual firms, freelance workers/ professionals 4. non-profit organizations 5. public organizations 6. other subjects 7. foreigners who match 2-6.
RegCode	Domain name registrant's Tax-code	extcon:regCode		1	1-36	If EntityType = 1: if the Registrant is an Italian natural person, it contains his/her tax code (“Codice Fiscale”); for foreigners it can contain a document number. If EntityType = 7: it contains the VAT number. In all the other cases, it must be equal to VAT number or the numeric tax code.
SchoolCode	Edu.it Domain name registrant's school-code	extcon:schoolCode		0-1		The mechanographic code of a school

Gov.it Registrant's Data		extgovcon:govRegistrant		0-1		
IpaCode	Gov.it Domain name registrant's IPA-code	extgovcon:ipaCode		1		The IPA code of the Public Administration
UoCode	Gov.it Domain name registrant's UO-code	extgovcon:uoCode		0-1		The UO code of the Organizational Unit belonging to the Public Administration identified by the previous ipaCode. This information is optional and shall be indicated if the Registrant is a UO.

8.1.1.2 Validation steps for Contact Create request

The system verifies that the request to Contact is compatible with:

- the constraints present in the XML Schema epp-1.0.xsd, eppcom-1.0.xsd, contact-1.0.xsd, extcon-1.0.xsd (Section 6.1);
  - the constraints present in the XML Schema extgovcon-1.0.xsd, if the request is for creating a gov.it Registrant;
- the following additional restrictions:
  - the ID of the person must not begin with the prefix "DUP" (used to identify duplicated contacts);
  - the ID of the contact must contain letters (a-z A-Z), hyphen (-) and/or digits (0-9);
  - the contact with the specified ID must not be present in the Registry Database;
  - the request must contain all the required fields;
  - it must comply with the cardinality and length of different fields;
  - the telephone and fax numbers must be in ISO format (e.g.: +39.0503139811). The international code (e.g.: +39) indicated in the telephone and fax numbers must be valid. Any extension numbers (telephone and fax) in the "x" extension of the "voice" and "fax" fields can only have a maximum of 10 digits;
  - the value of the Email field must be in the format defined by RFC5322. Moreover, if on the right of the "@" character there are non-ASCII characters (Section 5.1), it must be indicated in Punycode format (e.g.: "postmaster@xn--citt-3na.it" and not "postmaster@città.it");
  - the values of the Nation and Nationality fields must follow ISO 3166-1 (e.g.: IT, FR);
  - if the Registrant is other than a natural person (EntityType <> 1):
    - the Nation and Nationality fields must match and must correspond to the ISO 3166-1 code of a country of the European Economic Area (EEA), the Vatican City State, the Republic of San Marino, the Swiss Confederation or the United Kingdom;
  - if the Registrant is other than natural person, individual firm and professional person (EntityType <> 1 and EntityType <> 3):
    - the value of the ConsentForPublishing field must always be true/1;

- if the Registrant is a natural person (EntityType = 1):
  - the Nation and Nationality fields may differ but at least one of them must correspond to the ISO 3166-1 code of a country of the European Economic Area (EEA), the Vatican City State, the Republic of San Marino, the Swiss Confederation or the United Kingdom;
  - the Name and Organization (if specified) fields must be the same. Where the Organization is not specified, it will be forced by the system to the value of the Name field;
- if the Nation = IT, the Province field must contain the tag of two letters corresponding to a province in Italy;
- if Nationality = IT:
  - the Organization field must contain at least one alphabetical character;
  - if EntityType = 1, RegCode must have the format of a tax code;
  - if EntityType <> 1, RegCode must be of the form of a VAT number (11 digits) or tax identification number;
- if Nationality <> IT:
  - the Organization field must contain at least one alphanumeric character;
  - if the Registrant is an entity other than a natural person, the field EntityType must contain the value 7.
- As for the schoolCode field, the EPP server verifies that the mechanographic code:
  - is valid;
  - corresponds to an existing and active educational institution when registering the domain name;
  - is associated with the educational institution corresponding to the numeric tax code;
  - the EntityType field shall the value '2' (companies), '4' (no-profit organizations) or '5' (public organizations).
- As for the gov.it Registrant information, the EPP server verifies that:
  - the Registrar added the extgovcon-1.0 namesapce to the Login request;
  - the Organization field shall coincide with the name of the Public Administration, presented to the user at the end of the Validation Procedure. If this is not the case, the system replaces the name indicated in the EPP Create Contact with the name presented to the user at the end of the Validation Procedure and returns the assigned name as the Contact Create tresponse extension;
  - it is recommended that the Address field coincides with the postal address of the Public Administration, presented to the user at the end of the Validation Procedure;
  - the email shall take the PEC value presented to the user at the end of the Validation Procedure;
  - the Nationality field shall take the value 'it' (or 'IT');
  - the EntityType:field it shall the value '5' (public organizations);
  - the RegCode field shall coincide with the tax code of the Public Administration, or of the UO (if the Registrant is a UO and has it), presented to the user at the end of the Validation Procedure;
  - the IpaCode field shall be associated to the Public Administration identified by the RegCode field;
  - the UoCode shall belong to the Public Administration identified by the previous ipaCode..

8.1.1.3 Examples of a Contact Create request

Example 1
Contact Create command for registering an administrative or technical contact. In the request for Contact Create below, the section on data from the Registrant is missing:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
  <epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
    <command>
      <create>
```

```
        <contact:create
          xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
          xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0
          contact-1.0.xsd">
          <contact:id>mr0001</contact:id>
          <contact:postalInfo type="loc">
            <contact:name>Mario Rossi</contact:name>
            <contact:addr>
              <contact:street>Via Moruzzi 1</contact:street>
              <contact:city>Pisa</contact:city>
              <contact:sp>PI</contact:sp>
              <contact:pc>56124</contact:pc>
              <contact:cc>IT</contact:cc>
            </contact:addr>
          </contact:postalInfo>
          <contact:voice x="2111">+39.050315</contact:voice>
          <contact:fax>+39.0503152593</contact:fax>
          <contact:email>mario.rossi@example.it</contact:email>
          <contact:authInfo>
            <contact:pw></contact:pw>
          </contact:authInfo>
        </contact:create>
      </create>
       <extension>
         <extcon:create
           xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
            xsi:schemaLocation="http://www.nic.it/ITNIC-EPP/extcon-1.0 extcon-
1.0.xsd">
          <extcon:consentForPublishing>true</extcon:consentForPublishing>
          </extcon:create>
       </extension>
      <clTRID>ABC-12345</clTRID>
    </command>
  </epp>
```

Example 2

Contact Create for the registration of a Registrant contact. The section relating to the Registrant data is present in the command.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
  <epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
    <command>
      <create>
        <contact:create
          xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
          xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0
          contact-1.0.xsd">
          <contact:id>mr0001</contact:id>
          <contact:postalInfo type="loc">
            <contact:name>Mario Rossi</contact:name>
            <contact:org>Mario Rossi</contact:org>
            <contact:addr>
              <contact:street>Via Moruzzi 1</contact:street>
              <contact:city>Pisa</contact:city>
              <contact:sp>PI</contact:sp>
              <contact:pc>56124</contact:pc>
              <contact:cc>IT</contact:cc>
            </contact:addr>
```

```
          </contact:postalInfo>
          <contact:voice x="2111">+39.050315</contact:voice>
          <contact:fax>+39.0503152593</contact:fax>
          <contact:email>mario.rossi@example.it</contact:email>
          <contact:authInfo>
            <contact:pw></contact:pw>
          </contact:authInfo>
        </contact:create>
      </create>
       <extension>
         <extcon:create
           xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
           xsi:schemaLocation="http://www.nic.it/ITNIC-EPP/extcon-1.0 extcon-
1.0.xsd">
          <extcon:consentForPublishing>true</extcon:consentForPublishing>
           <extcon:registrant>
             <extcon:nationalityCode>IT</extcon:nationalityCode>
             <extcon:entityType>1</extcon:entityType>
             <extcon:regCode>RSSMRA64C14G702Q</extcon:regCode>
           </extcon:registrant>
         </extcon:create>
       </extension>
      <clTRID>ABC-12345</clTRID>
    </command>
  </epp>
```

8.1.1.4 Edu.it Registrants Specifics

The following is an example of an extension in the EPP Create Contact request for a registrant of an edu.it domain name:

4

```
  <extcon:registrant>
    <extcon:nationalityCode>IT</extcon:nationalityCode>
    <extcon:entityType>5</extcon:entityType>
    <extcon:regCode>80231570583</extcon:regCode>
    <extcon:schoolCode>RMIC8BK005</extcon:schoolCode>
  </extcon:registrant>
```

A similar extension is given both in the response to an EPP Info Contact request and in the response to a Domain Info request submitted by a Registrar to view data relating to a gov.it domain name and its Registrant.

8.1.1.5 Gov.it Registrants Specifics

The following are two examples of an extension in the EPP Create Contact request for a registrant of an gov.it domain name:

Example 1
Only the ipaCode was specified during the Validation Procedure:

```
  <extension>
    <extcon:create>
      <extcon:consentForPublishing>true</extcon:consentForPublishing>
      <extcon:registrant>
        <extcon:nationalityCode>IT</extcon:nationalityCode>
        <extcon:entityType>5</extcon:entityType>
        <extcon:regCode>80054330586</extcon:regCode>
```

```
      </extcon:registrant>
    </extcon:create>
    <extgovcon:create>
      <extgovcon:govRegistrant>
        <extgovcon:ipaCode>cnr</extgovcon:ipaCode>
      </extgovcon:govRegistrant>
    </extgovcon:create>
  </extension>
```

Example 2
Both ipaCode and uoCode have been specified during validation:

```
<extension>
  <extcon:create>
    <extcon:consentForPublishing>true</extcon:consentForPublishing>
    <extcon:registrant>
      <extcon:nationalityCode>IT</extcon:nationalityCode>
      <extcon:entityType>5</extcon:entityType>
      <extcon:regCode>80054330586</extcon:regCode>
    </extcon:registrant>
  </extcon:create>
  <extgovcon:create>
    <extgovcon:govRegistrant>
      <extgovcon:ipaCode>cnr</extgovcon:ipaCode>
      <extgovcon:uoCode>WD4UO9</extgovcon:uoCode>
    </extgovcon:govRegistrant>
  </extgovcon:create>
</extension>
```

A similar extension is given both in the response to an EPP Info Contact request and in the response to a Domain Info request submitted by a Registrar to view data relating to a gov.it domain name and its Registrant.

Below is an example of an extension of the response to the EPP Create Contact request, for which the system has replaced the name indicated in the EPP Create Contact with the name presented to the user at the end of the Validation Procedure:

```
<extension>
  <extgovcon:creData>
    <extgovcon:govOrgChanged>orgPA</extgovcon:govOrgChanged>
  </extgovcon:creData>
</extension>
```

8.1.1.6 Examples of responses to a Contact Create request

Example 1
Response to a successful Contact Create:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp
  xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
  xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
  xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
  xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
  xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
  xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
  xmlns="urn:ietf:params:xml:ns:epp-1.0">
```

```
  <response>
   <result code="1000">
     <msg lang="en">Command completed successfully</msg>
   </result>
  <resData>
    <contact:creData>
      <contact:id>MR0001</contact:id>
      <contact:crDate>2013-04-16T11:43:32+02:00</contact:crDate>
   </contact:creData>
   </resData>
  <trID>
    <clTRID>ABC-12345</clTRID>
    <svTRID>e0638c11-d2bc-47bb-ac05-d44fb19606af</svTRID>
  </trID>
 </response>
</epp>
```

Example 2
Response to a failed Contact Create. The error is in the tax-code of the Registrant indicated in RegCode of the request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp
  xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
  xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
  xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
  xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
  xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
  xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
  xmlns="urn:ietf:params:xml:ns:epp-1.0">
 <response>
   <result code="2004">
     <msg lang="en">Parameter value range error</msg>
     <value>
       <extepp:wrongValue>
         <extepp:element>regCode</extepp:element>
         <extepp:namespace>http://www.nic.it/ITNIC-EPP/extcon-
1.0</extepp:namespace>
         <extepp:value>LFFMRA64M22H999P</extepp:value>
       </extepp:wrongValue>
     </value>
     <extValue>
       <value>
         <extepp:reasonCode>8027</extepp:reasonCode>
       </value>
       <reason lang="en">Registrant: invalid reg code</reason>
     </extValue>
   </result>
   <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>72a0a5ea-92a2-4086-ad58-222fa653bc55</svTRID>
   </trID>
  </response>
</epp>
```

8.1.1.7 Effects of the Contact Create request

If the Contact Create command submitted by the Registrar is executed successfully and passes the validation steps described in Section 8.1.1.2, a contact is registered in the Registry Database and the

following fields are set:

- if the contact is a "registrant" and is a natural person (i.e., EntityType = 1), the Organization field, if empty, is forced to the value of the Name field;
- registration data (coinciding with the date and time of insertion of the contact into the Database);
- current client ID;
- client ID that carried out the registration;
- the contact goes into ok.

8.1.2 Domain Create

New domain names are registered by using the Domain Create command.

8.1.2.1 Domain object fields required

The Domain Create command needs the following information:

- Domain name
- Period of validity of the domain name (ignored by the server - default 1 year)
- List of host items associated with the domain name organized as follows:
  - **host name**
  - for hosts subordinate to the domain name:
    - an IPv4 address and, if any, an IPv6 address.
- **Registrant contact** (i.e. the ID of the contact associated with the Registrant)
- **Admin contact** (i.e. the ID of the administrative contact)
- **Tech contact** (i.e. the ID of the technical contact)
- **AuthInfo of the domain name**
- Validation Code of the gov.it domain name registrant (*only required for gov.it domains*)

If the Registrant is a natural person (EntityType = 1), the Registrant and administrative contact (admin) must be the same. These fields will therefore contain the same contactID associated with a contact already registered in the Registry Database, including the extension of the Registrant.
If the Registrant contact indicated in the Domain Create request does not contain all the requested fields, the registration operation fails. This situation can occur, for example, in case of Registrant contacts created in the old "asynchronous" registration system and migrated into the new "synchronous" registration system.

The following table shows the fields of the domain object and the related correspondence with the XML tag of the request:

Field	Description	XML Tag	XML Tag Attribute	Cardinality	Value
Domain name	Domain name to be registered	domain:name		1	Bear in mind the following limitations: - second level domain name minimum length is 3 characters - maximum length for every domain name part is 63 characters. Total length cannot be greater than 255 characters - characters accepted: ASCII and non-ASCII characters indicated in Section 5.1 - each domain name part cannot begin or end with the "-" character - the domain name must not begin with the string "xn--", which is reserved for the IDN encoding of a domain name. In case of IDN domain names, the domain name must be indicated in native format and not in Punycode (e.g.: "città.it" and not "xn--citt-3na.it")
Period	Domain name validity period	domain:period		0-1	Ignored by server. Default value is a year
Time unit		domain:period	unit="y|m"	0-1	
Domain name associated hosts	Domain name associated hosts list as host name and ip address couples	domain:ns		1	
Host properties		domain:hostAttr		2-6	Number of associated hosts for a domain must be between 2 and 6

Host name	Associated host name	domain:hostName		1	If the host name contains non-ASCII characters (Section 5.1) it must be indicated in Punycode format (e.g.: “ns.xn--citt-3na.it” and not “ns.città.it”)
IP address	Host's IP address	domain:hostAddr		0-2	Only required for the hosts subordinate to the domain name and it is needed to generate the “glue records”. It is possible to indicate up to a maximum of two IP addresses for nameserver where at least one and not more than one, must be an IPv4
IP address type	Specify IPv4 or IPv6 for every IP address	domain:hostAddr	ip	0-1 (default “v4”)	Default type is “IPv4”
Registrant	Identifies the person or the organization requesting a domain name registration or has already one assigned	domain:registrant		1	Must contain the contactID associated with the Registrant, already registered in the Registry's database by the Registrar
Admin contact	Identifies the domain name admin contact	domain:contact	type=“admin”	1	Must contain the contactID associated with the admin contact, already registered in the Registry's database by the Registrar. If the Registrant is a natural person, admin and Registrant fields must be the same
Technical Contact	Identifies the domain name technical contact	domain:contact	type=“tech”	1-6	Must contain the contactID associated with the technical contact, already registered in the Registry's database by the Registrar
Domain name AuthInfo	Identifies the authorization password for specific operation on the domain name	domain:authInfo		1	Alphanumeric value given by the Registrant to the domain name Registrant. Its length is between 8 to 32 characters.

Gov.it domain name registration		extgovdom:create		0-1	
Gov.it Validation Code		extgovdom:validationCode		1	A gov.it Validation Code

Section 0 contains some examples of Domain Create requests for the registration of a domain name by a "non DNSSEC accredited" Registrar.

Section **Errore. L'origine riferimento non è stata trovata.**instead, contains examples of Domain Create requests for the registration of a digitally signed domain name by a "DNSSEC accredited" Registrar.

8.1.2.2 Validation steps for Domain Create Request

The system verifies that the request for Domain Create is compatible with:

- the constraints present in the XML Schema epp-1.0.xsd, eppcom-1.0.xsd, domain-1.0.xsd, host-1.0.xsd (Section 6.1);
  - the constraints present in the XML Schema extgovdom-1.0.xsd, if the request is for creating a gov.it domain name;
- the following additional restrictions:
  - the Registrar that has sent the request must not be suspended;
  - the domain name has the ".it" region suffix;
  - the domain name cannot be reserved, geographical or non-assignable as specified in the "Rules";
  - the domain name applied for must not be present in the Registry Database;
  - the domain name requested must meet the following requirements:
    - minimum length of 3 characters for second-level domain names;
    - maximum length of 63 characters for each component of the domain name. The length must not exceed 255 characters;
    - characters accepted: ASCII and non-ASCII characters indicated in Section **Errore. L'origine riferimento non è stata trovata.**;
    - each component cannot begin or end with a hyphen (-);
    - must not contain the string "xn--" in the first four characters, which is reserved for the IDN encoding of a domain name. In case of IDN domain names, the domain name must be indicated in native format and not in Punycode (e.g.: "città.it" and not "xn--citt-3na.it");
  - the request must contain all the required fields;
  - it must comply with the cardinality of different fields;
  - AuthInfo must have a minimum length of 8 characters and maximum of 32 characters;
  - the ID of the contacts referenced in the domain name to register (registrant, admin and tech) must already be present in the Registry Database;
  - the Registrant with the specified ID must be present in the Registry Database and created as a Registrant contact (i.e. the fields filled for the data section of the Registrant);
  - the contact list must not contain the same two contacts with the same role;
  - the number of administrative and technical contacts must comply with the table in Section 12.8;
  - if the domain name is requested by a natural person, the Registrant and Admin fields must be the same, that is the Admin field must contain the same contactID as the one indicated in the

Registrant field;

- if the domain name is requested by a subject other than natural person, the ConsentForPublishing field of the Registrant with the specified ID must be set to true / 1;
- the number of hosts to be associated with the domain name must comply with the table in Section 12.8;
- the host list cannot contain two hosts with the same IP address or with the same name;
- the number of the IP addresses of the host/s subordinate to the domain name must comply with what stated in the table in Section 12.8;
- for each host subordinate to the domain name, it is possible to specify up to a maximum of two IP addresses for nameserver where one and not more than one is of IPv4 type (glue record);
- the hosts containing non-ASCII characters (Section 5.1) must be indicated in Punycode format (e.g.: "ns.xn--citt-3na.it" and not "ns.città.it");
- if the domain name is an edu.it domain name, the registrant referenced in the edu.it domain name must contain the schoolCode field;
- if the domain name is a gov.it domain name, the EPP server checks that:
  - the Registrar added the extgovdom-1.0 namesapce to the Login request;
  - the ValidationCode field is present;
  - the ValidationCode field has not expired;
  - the Registrant contact, referenced in the request, contains the values of the RegCode, IpaCode, and possibly UoCode fields, corresponding to the Validation Code obtained during the Validation Procedure.

8.1.2.3 Examples of a Domain Create request

Example 1
Domain Create command to register a domain name ("example.it") maintained by subordinate hosts ("ns1.example.it" and "ns2.example.it"):

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  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>example.it</domain:name>
       <domain: period unit="y">1</domain:period>
       <domain:ns>
         <domain:hostAttr>
           <domain:hostName>ns1.example.it</domain:hostName>
           <domain:hostAddr ip="v4">193.205.245.70</domain:hostAddr>
         </domain: hostAttr>
         <domain: hostAttr>
          <domain:hostName>ns2.example.it</domain:hostName>
          <domain:hostAddr ip="v4">193.205.245.77</domain:hostAddr>
         </domain:hostAttr>
       </domain:ns>
       <domain:registrant>mr0001</domain:registrant>
       <domain:contact type="admin">cl8013</domain:contact>
       <domain:contact type="tech">mb8015</domain:contact>
       <domain:authInfo>
```

```
        <domain:pw>22fooBAR</domain:pw>
       </domain:authInfo>
     </domain:create>
   </create>
   <clTRID>ABC-12345</clTRID>
 </command>
</epp>
```

The IP addresses of the hosts “ns1.example.it” and “ns2.example.it” are mandatory because they are subordinate with respect to the domain name “example.it” to register.

Example 2
Domain Create for the registration of a domain name (“paperino.it”) managed by hosts that are not subordinate (“ns1.example.it” e “ns.dominio.org”):

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   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>paperino.it</domain:name>
        <domain:period unit="y">1</domain:period>
          <domain:ns>
            <domain:hostAttr>
              <domain:hostName>ns1.example.it</domain:hostName>
            </domain:hostAttr>
            <domain:hostAttr>
              <domain:hostName>ns.dominio.org</domain:hostName>
            </domain:hostAttr>
          </domain:ns>
        <domain:registrant>mm-001</domain:registrant>
        <domain:contact type="admin">mm-001</domain:contact>
        <domain:contact type="tech">mb-001</domain:contact>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

The IP addresses of the hosts “ns1.example.it” e “ns.dominio.org” are not inserted into the request because they are not subordinate with respect to the domain name “paperino.it” to register.

Example 3
Domain Create command to register a domain name (“example.it”) maintained by subordinate hosts (“ns1.example.it” and “ns2.example.it”). The nameserver ns1.esempio.it is associated with both IPv4 and IPv6 addresses:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  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>example.it</domain:name>
        <domain: period unit="y">1</domain:period>
        <domain:ns>
          <domain:hostAttr>
            <domain:hostName>ns1.example.it</domain:hostName>
            <domain:hostAddr ip="v4">192.12.192.5</domain:hostAddr>
            <domain:hostAddr ip="v6">2a00:d40:1:1::5</domain:hostAddr>
          </domain: hostAttr>
          <domain: hostAttr>
            <domain:hostName>ns2.example.it</domain:hostName>
            <domain:hostAddr ip="v4">193.205.245.77</domain:hostAddr>
          </domain:hostAttr>
        </domain:ns>
        <domain:registrant>mr0001</domain:registrant>
        <domain:contact type="admin">cl8013</domain:contact>
        <domain:contact type="tech">mb8015</domain:contact>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

The examples shown above correspond to Scenario 1 in Section 8.1.4.

8.1.2.4 Gov.it Domains Specifics

An example of an extension in an EPP Create Domain request for a gov.it domain name is shown below:

```
  <extension>
    <extgovdom:create>
      <extgovdom:validationCode>validationCode</extgovdom:validationCode>
    </extgovdom:create>
  </extension>
```

8.1.2.5 Examples of responses to a Domain Create request

Example 1
Response to a successful Domain Create:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
```

```
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:creData>
        <domain:name>esempio.it</domain:name>
        <domain:crDate>2013-04-16T11:43:32+02:00</domain:crDate>
        <domain:exDate>2014-04-16T23:59:59+02:00</domain:exDate>
      </domain:creData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>be47652c-5c3f-4e77-b41d-1104df945cc4</svTRID>
    </trID>
  </response>
 </epp>
```

The domain name specified in the Domain Create request is registered in the Registry's Database and is put in inactive/*dnsHold* status.

Example 2
Response to a failed Domain Create. The error is in the fact that in the request for regisration of a domain name a contact not created as a registrant is shown as a Registrant.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2308">
      <msg lang="en">Data management policy violation</msg>
      <value>
        <extepp:wrongValue>
          <extepp:element>registrant</extepp:element>
           <extepp:namespace>urn:ietf:params:xml:ns:domain-1.0</extepp:namespace>
          <extepp:value>CL-007</extepp:value>
        </extepp:wrongValue>
      </value>
      <extValue>
        <value>
          <extepp:reasonCode>8030</extepp:reasonCode>
        </value>
        <reason lang="en">Contact is not a registrant</reason>
      </extValue>
    </result>
    <trID>
       <clTRID>ABC-12345</clTRID>
       <svTRID>5579b9db-d1ad-46a2-850f-381ea1fb1551</svTRID>
    </trID>
  </response>
</epp>
```

Example 3
Response to a successful Domain Create following the registration of a domain name with "*remapped*" characters (Section 5.1.2).

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"  xmlns:extcon="http://www.nic.it/ITNIC-
EPP/extcon-1.0" xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <msgQ id="296945" count="21">
      <qDate>2014-05-14T10:00:47.000+02:00</qDate>
      <msg lang="en">dnsHold is started</msg>
    </msgQ>
    <resData>
      <domain:creData>
        <domain:name>ᾳᾳᾳᾳ.it</domain:name>
        <domain:crDate>2014-05-15T16:13:02.092+02:00</domain:crDate>
        <domain:exDate>2015-05-15T23:59:59.999+02:00</domain:exDate>
      </domain:creData>
    </resData>
    <extension>
      <extdom:remappedIdnData>
        <extdom:idnRequested>άιάιάιάι.it</extdom:idnRequested>
        <extdom:idnCreated>ᾳᾳᾳᾳ.it</extdom:idnCreated>
      </extdom:remappedIdnData>
    </extension>
    <trID>
      <clTRID>RTRT-00017</clTRID>
      <svTRID>25a61bec-0c95-4c8c-b842-6c10fb594074</svTRID>
    </trID>
  </response>
</epp>
```

The response includes the extension “extdom:remappedIdnData” that contains the IDN domain name requested in the Domain Create command (“extdom:idnRequested”) and the domain name actually recorded as a result of transformation (“extdom:idnCreated”).

8.1.2.6 Effects of the Domain Create request

If the Domain Create command submitted by the Registrar is executed successfully and passes the validation steps described in Section 8.1.2.2, a domain is registered in the Registry Database and the following fields are set:

- date of registration (coinciding with the date and time for entering the domain name in the Database);
- current client ID;
- client ID that carried out the registration;
- expiry date of the domain name;
- the domain name goes into inactive/*dnsHold* status;
- the domain name is debited to the Registrar and is immediately available for invoicing.

The contacts specified that were not referenced go into ok/linked status.
The Registry, upon registration of the domain names in the Database, emails the Registrant a summary

of the data in the Database for the registered domain name.

If the Registrant is a natural person (EntityType = 1), individual firm or professional person (EntityType = 3) the email format is the following:

Subject: 10300 - New registration of the domain name <name of the domain>

We inform you that on <registration date> the domain name <name of the domain> has been registered through the Registrar < Registrar>.

The outcome of the registration is:

Registrant:
Address:
Country:
Nationality:
Phone:
Fax:
RegCode:
Email:
EntityType:

The domain name <name of the domain> has been put in <status> status.

The Registrant, has made the following choice concerning consent:

- having read the information regarding personal data treatment for registration: YES
- consent for personal data treatment for diffusion and accessibility via the Internet: <YES/NO (value of consentForPublishing) >

and has made the following declarations and has accepted the following clauses:

- to be citizen or resident in one of the countries of the European Economic Area (EEA), the Vatican City State, the Republic of San Marino, the Swiss Confederation or the United Kingdom;
- to be aware of and to accept that the registration and management of the domain name are subject to the "Rules of assignment and management of domain names in the ccTLD .it" and the "Regulations for the Resolution of disputes in the ccTLD .it and subsequent modifications;
- to have right of use and/or legal availability of the registered domain name requested and not to prejudice, with this registration request, the rights of third parties;
- to be aware that in order to fulfill personal data on the database of assigned domain names, and for their possible diffusion and accessibility on Internet, it is necessary to give express consent checking the relevant boxes on the basis of the information below. On the Registry website (http://www.nic.it) the document "The policy of the .it Registry about the Whois database" is available;
- to be aware of and to accept that in case of erroneous or false declaration in the present request, the Registry will proceed to the immediate revocation of the domain name, reserving the right to take out further legal action. In this case the revocation cannot give rise in any way whatsoever to requests for damages to the Registry;

- to release the Registry from any responsibility deriving from assignment or use of the domain name on the part of the requesting subject;
- to accept Italian jurisdiction and the laws of the Italian State Ordinance.

We inform you that the Registrar mentioned above is responsible for personal data treatment and that the CNR, through the Institute of Informatics and Telematics, is the holder.
As specified in the Registration Form, the data will be released to third parties for the activation of opposition and the defense of rights as well as the fulfillment of obligations of law or regulation.
Should you need further information, please contact the Registrar indicated in the registration and whose data are available on the website of the Registry http://www.nic.it.

Best regards,

Registro .it
Istituto di Informatica e Telematica
CNR - AREA DELLA RICERCA
Via Giuseppe Moruzzi, 1 - I-56124 PISA
Tel: +39 050 3139811
Email: hostmaster@nic.it

If the Registrant is other than natural person, individual firm and professional person (EntityType <> 1 and EntityType <> 3) the email format is the following:

Subject: 10307 - New registration of the domain name <name of the domain>

We inform you that on <registration date> the domain name <name of the domain> has been registered through the Registrar <Registrar>.

The outcome of the registration is:

Registrant:
Address:
Country:
Phone:
Fax:
RegCode:
Email:
EntityType:

The domain name <name of the domain> has been put in <status> status.

The Registrant, has made the following choice concerning consent:

- having read the information regarding personal data treatment for registration: YES

and has made the following declarations and has accepted the following clauses:

- to have the registered office based in one of the countries of the European Economic Area (EEA), the Vatican City State, the Republic of San Marino, the Swiss Confederation or the United Kingdom;

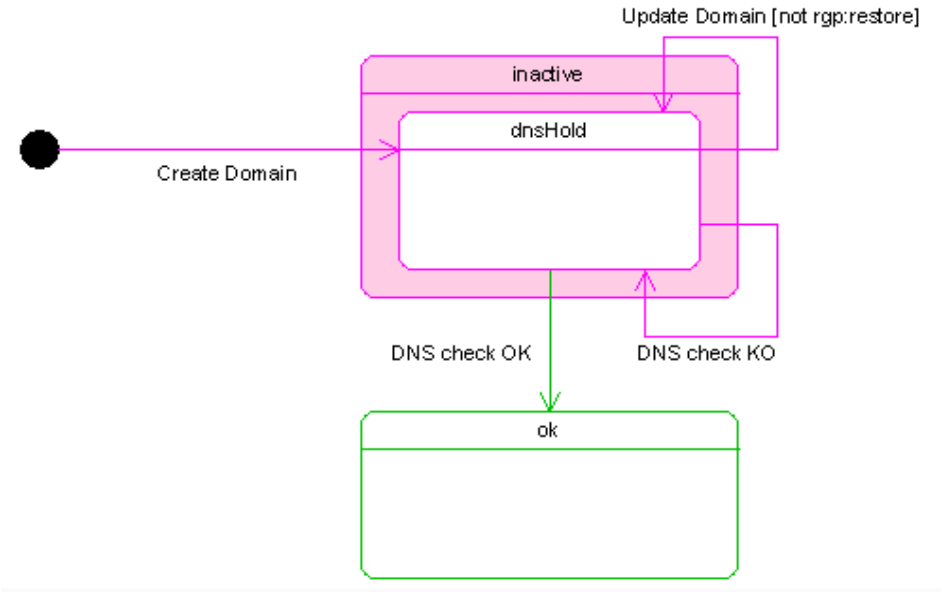
- to be aware of and to accept that the registration and management of the domain name are subject to the "Rules of assignment and management of domain names in the ccTLD .it" and the "Regulations for the Resolution of disputes in the ccTLD .it and subsequent modifications;
- to have right of use and/or legal availability of the registered domain name requested and not to prejudice, with this registration request, the rights of third parties;
- to be aware of and to accept that in case of erroneous or false declaration in the present request, the Registry will proceed to the immediate revocation of the domain name, reserving the right to take out further legal action. In this case the revocation cannot give rise in any way whatsoever to requests for damages to the Registry;
- to release the Registry from any responsibility deriving from assignment or use of the domain name on the part of the requesting subject;
- to accept Italian jurisdiction and the laws of the Italian State Ordinance.

We inform you that the Registrar mentioned above is responsible for personal data treatment and that the CNR, through the Institute of Informatics and Telematics, is the holder.
As specified in the Registration Form, the data will be released to third parties for the activation of opposition and the defense of rights as well as the fulfillment of obligations of law or regulation.
Should you need further information, please contact the Registrar indicated in the registration and whose data are available on the website of the Registry http://www.nic.it.

Best regards,

Registro .it
Istituto di Informatica e Telematica
CNR - AREA DELLA RICERCA
Via Giuseppe Moruzzi, 1 - I-56124 PISA
Tel: +39 050 3139811
Email: hostmaster@nic.it

The DNS configuration for the list of hosts mentioned in Domain Create command is checked in a non-simultaneous transaction. If the host configuration check fails, the EPP server of the Registry inserts in the polling queue of the Registrar, a notification report with all the checks carried out and their results. The domain name remains in inactive/*dnsHold* status if its DNS configuration is not correct. The server will periodically check whether the DNS configuration for the domain name in inactive/*dnsHold* is positive. When this occurs, the server will put in the Registrar polling queue a message that DNS has been successful and the domain name itself will go into ok status. At the same time, the Registry emails the Registrant the same communication above mentioned.
A domain name in inactive/*dnsHold*, can be subjected to any operation.
The following diagram shows the various steps leading to the registration of a new domain name:



8.1.2.7 Checking the functionality of the nameservers

The verification phase of the configuration of the nameservers that must be associated with the domain name takes place after the registration of the new domain name in the Registry Database or after the change of the hosts (Section 0).
The procedure for the control of nameservers analyzes the hosts that must be associated with domain names registered in the Registry Database that are either in inactive/*dnsHold* status or in pendingUpdate and executes the appropriate query (i.e. queries to the nameserver) to verify that it is actually operative. In particular:

- there must be at least 2 (two) authoritative nameservers for the domain name, and they must correspond exactly to those found in the registration request of the domain name or in the hosts change request;
- the IPv4 and, if indicated, the IPv6 address of the hosts subordinate to the domain name, must correspond to those actually associated with them in the DNS;
- the IPv4 and, if indicated, the IPv6 address of the hosts subordinate to the domain name, must answer in an authoritative way;
- the domain name cannot be associated with a CNAME record;
- the name of the nameserver specified in the SOA record for the domain name cannot be a CNAME;
- the names of the authoritative nameservers for the domain name cannot be CNAMEs;
- if there is an MX registration it cannot be associated with a CNAME;
- if, during the checking procedure, at least one nameserver returns the following responses:
  - Not responding
  - Not reachable
  - Not running
  - Non-existent domain
  - Host not found

- Server failure
- Query failed

the procedure returns an error;

- all hosts in the registration must be authoritative for the domain name registered.

The list of the checks carried out by the dns validator of the Registry is the following:

- *NameserversResolvableTest*: it verifies what are the nameservers that are resolvable
- *NSQueryAnswerTest*: it verifies what are the nameservers for which the query has given an answer to the NS query. The answer must be authoritative and contains a NOERROR return code
- *IPSOATest*: if a nameserver has more than one IP address, it carries out a query of SOA type for all the indicated addresses verifying whether they respond in an authoritative way or not, and that the serial numbers match
- *NameserverReturnCodeTest*: it verifies that the header of the response of the queries carried out for the nameservers to be validated, contains a NOERROR return code
- *AATest*: it verifies whether the nameservers that must be validated, are authoritative for the domain name or not
- *NSCountTest*: it verifies that the number of NS records complies with the table in Section 12.8
- *NSCompareTest*: it verifies the correspondence between the nameservers listed in the registration request or in the hosts change request with those inserted in the NS records of the zone file
- *CNAMEHostTest*: it verifies that the nameservers inserted in the SOA, NS and where indicated MX records, are not a CNAME
- *IPCompareTest*: it verifies the correspondence between the IP addresses, if indicated, of the namservers to be validated and those actually associated with them by means of A records
- *MXQueryAnswerTest*: it verifies what are the nameservers for which the query has given an answer to the MX query. The answer must be authoritative and contains a NOERROR return code
- *MXCompareTest*: it verifies that the same mailservers are indicated in the MX records of the zone file of the nameservers to be validated
- *MXRecordIsPresentTest*: it verifies the presence of the MX records in the zone file of the nameservers to be validated. MX records are not compulsory
- *SOAMasterCompareTest*: it verifies that the same hostname is present in the SOA record of the zone file of the nameservers to be validated
- *SOAQueryAnswerTest*: it verifies what are the nameservers for which the query has given an answer to the SOA query. The answer must be authoritative and contains a NOERROR return code.

In case of digitally signed domain names, further checks, described in Section 11.5of this document, are activated.

The DNS configurations checks are activated in the following way:

- if the domain name is in inactive/*dnsHold* status:
  - immediately after the Domain Create request;
  - immediately after every subsequent Domain Update request to change the DNS configuration;

- every 30 minutes for the first 30 days following the registration of the domain name;
- once a day after the 30 days following the registration of the domain name;

- if the domain name is in pendingUpdate status:
  - immediately after every Domain Update request to change the DNS configuration;
  - every 30 minutes until the expiration of the period of pendingUpdate (5 days).

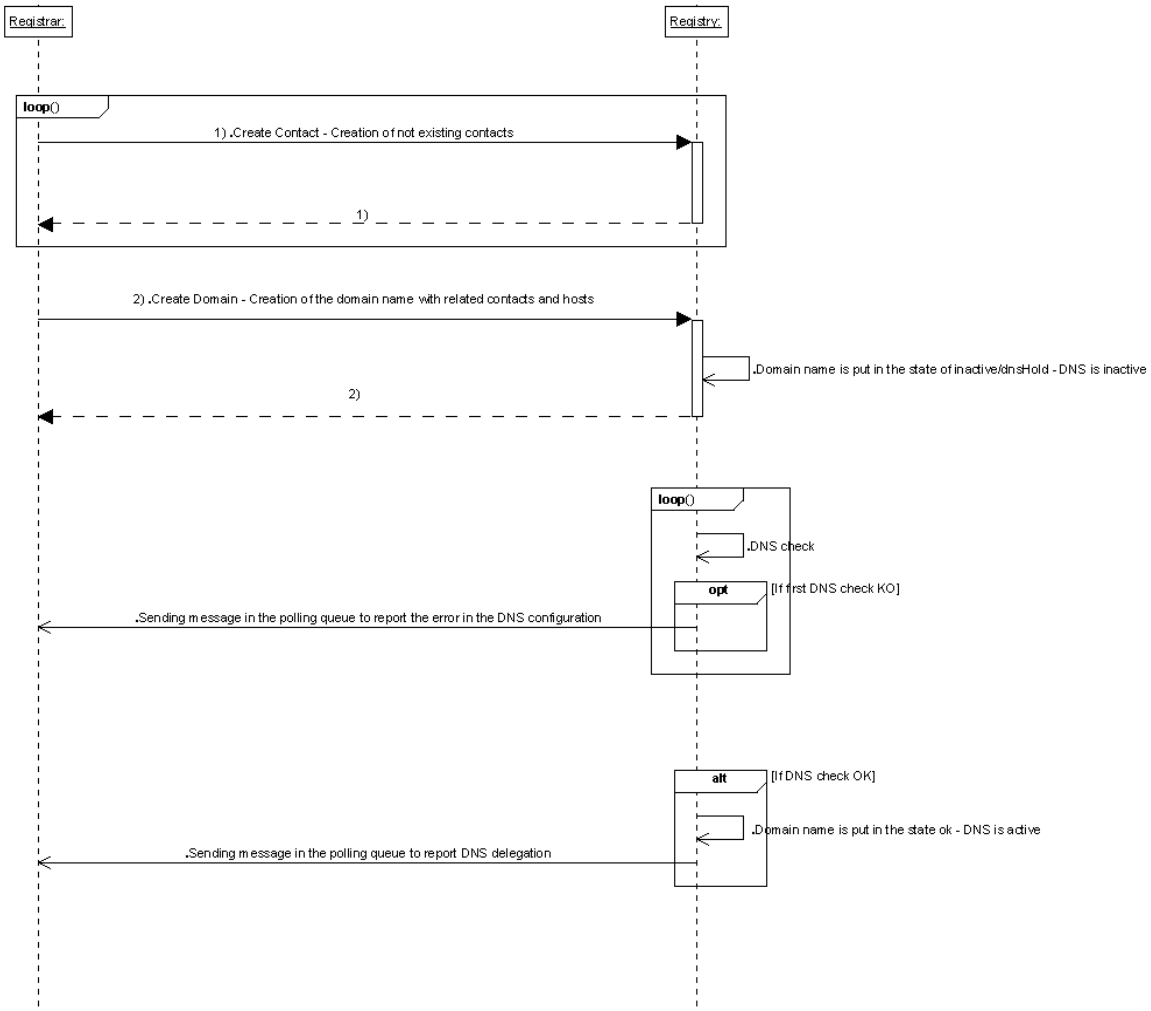
8.1.3 Request for a domain name subject to a cancellation in the previous 7 (seven) days

To register a domain name that is subject to a cancellation in the previous 7 (seven) days, the Registrar must submit the Domain Create command (Section **Errore. L'origine riferimento non è stata trovata.**) to the "epp-deleted.nic.it" server.
Requests sent to the "epp.nic.it" server will therefore be rejected.
The table in Section 12.8of this document shows the maximum number of daily requests for the registration of domain names that have been cancelled within less than 7 (seven) days that Registrars can send to the "epp-deleted.nic.it" server.

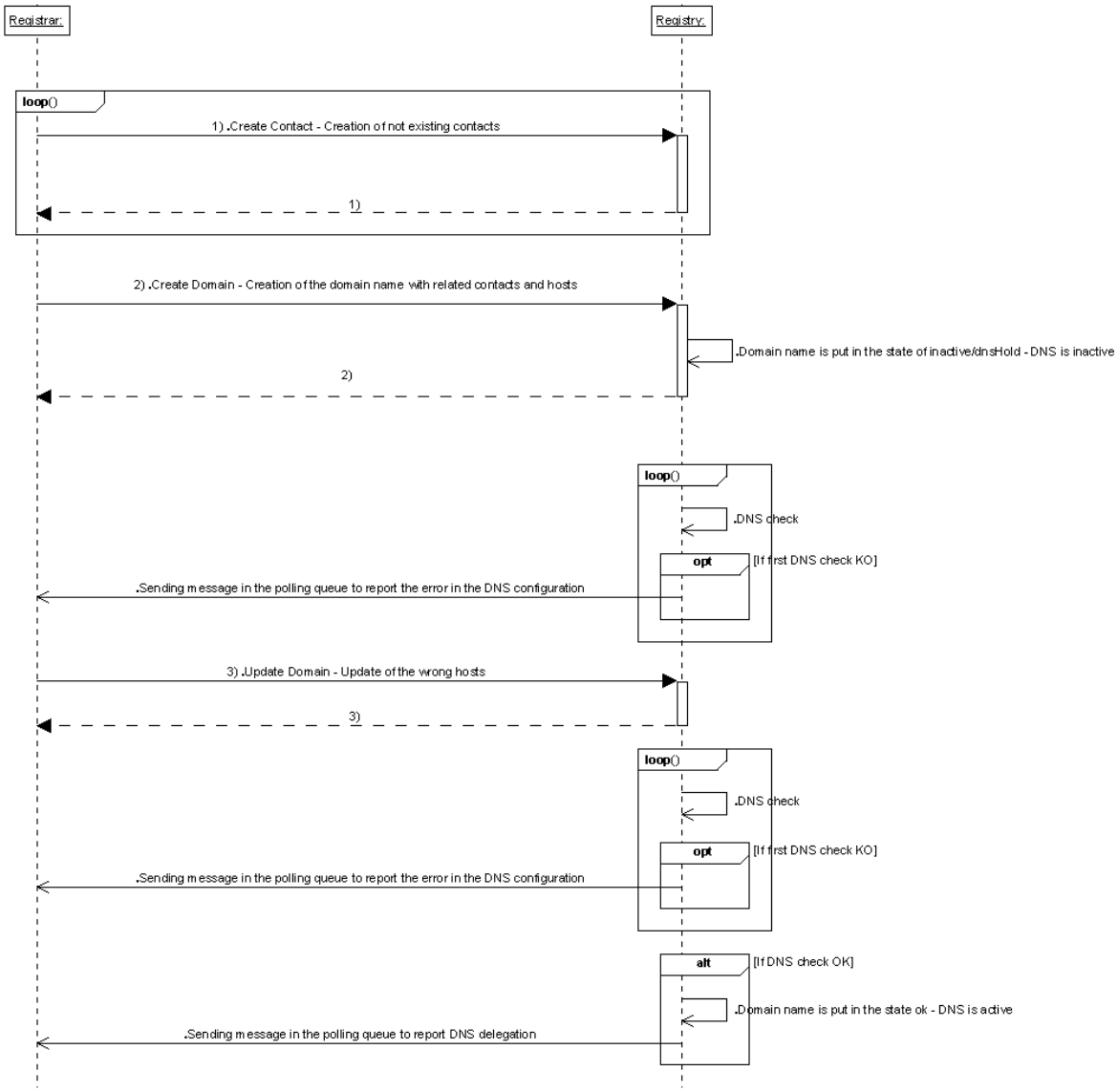
8.1.4 Examples of domain names registration

Here are two possible scenarios to complete the registration of a domain name:

- in **scenario 1** a domain name is registered via a sequence of Contact Create and Domain Create commands. The configuration of the nameserver is checked after Domain Create:


Registrar:
Registry:
loop()
1) .Create Contact - Creation of not existing contacts
1)
2) .Create Domain - Creation of the domain name with related contacts and hosts
.Domain name is put in the state of inactive/dnsHold - DNS is inactive
2)
loop()
.DNS check
opt
[If first DNS check KO]
.Sending message in the polling queue to report the error in the DNS configuration
alt
[If DNS check OK]
.Domain name is put in the state ok - DNS is active
.Sending message in the polling queue to report DNS delegation

- in **scenario 2** a domain name is registered with a list of hosts that is wrong, and is then changed by the Registrar through a Domain Update. The configuration of the nameserver is checked after Domain Update.



8.2 Simple change

Simple changes are carried out using the EPP Update command, the Registrar submits a request to Contact Update or Domain Update, depending on the object to update.
With a simple change, the Registrar can only update certain fields of a contact or domain in the Registry Database. The Registrar may change the authoritative nameservers and the AuthInfo (i.e. the password for the authorization of the request for specific transactions) of a domain name, its administrative and technical contacts, and information associated with them.
Transactions classified as “simple change” are not charged to the Registrar and can be requested by a suspended Registrar as well.

8.2.1 Simple change to registered contact

Simple changes regarding a registered contact in the Registry Database are done by using Contact Update. This command allows the Registrar to perform the following steps:

- Addition or removal of a status
- Addition or change of the following fields:
  - PostalInfo organized as follows:
    - Name
    - Organization
    - Address structure:
    - Street/Square 1
    - Street/Square 2
    - Street/Square 3
    - City
    - Province
    - ZIP code (postcode)
    - Country
  - Phone
  - Fax
  - Email
  - ConsentForPublishing (in case of Registrants other than natural persons, the field cannot be set to false / 0)
  - Registrant data (if the contact is not already a Registrant)
    - Nationality (nationalityCode)
    - EntityType
    - RegCode

To identify the contact on which to perform the requested transaction, the Registrar uses the ID of the contact, which is mandatory.
The policies adopted by the Registry do not allow data regarding the Registrant extension to be changed (either individually or all together) once they have been set. The Registrar has two ways to set the information of its Registrant:

- when registering the contact with a Contact Create. In this way the contact is registered as a potential Registrant of one or more domain names;
- with a change after the registration, via a Contact Update. In this way the contact, who was initially registered as a technical contact (tech) and/or administrative (admin), can then be associated as a Registrant of one or more domain names.

8.2.1.1 Validation steps for the simple change of a registered contact

The system verifies that the request to Contact Update is compatible with:

- the constraints present in the XML Schema epp-1.0.xsd, eppcom-1.0.xsd, contact-1.0.xsd, extcon-1.0.xsd (Section 6.1);
- the following additional restrictions:

- the contact with the ID specified must be present in the Registry Database;
- restrictions must apply on the values and the minimum and maximum cardinality of the fields given for Contact Create;
- the current status of the contact must not be serverUpdateProhibited or clientUpdateProhibited;
- statuses that can be added must only be those that begin with the prefix “client”;
- the list of statuses by adding or removing statuses must not contain duplicates;
- a status already associated with the contact cannot be added;
- a status not associated with the contact cannot be removed;
- the new value of the Email field must be in the format defined by RFC 5822 and subsequent modifications. Moreover, if on the right of “@” character there are non-ASCII characters (Section 5.1), it must be specified in Punycode format (e.g.: “postmaster@xn--citt-3na.it” and not “postmaster@città.it”);
- if the contact is a Registrant (fields of Registrant data section filled):
  - the Name field is changeable only if the Registrant is other than a natural person (i.e., EntityType <> 1);
  - the Country field is changeable only if the Registrant is a natural person (i.e., EntityType = 1);
  - the Organization field, if initially empty, must be completed if the Registrant is a natural person (i.e., EntityType = 1) the value of the field organization must be the same as the field Name;
  - the Organization field, once set, is no longer changeable;
  - the Nation, Nationality, EntityType and RegCode fields, if empty, must be filled in;
  - the Nationality, EntityType and RegCode fields, once set, are no longer changeable (individually or all together);
- if the contact is a Registrant other than a natural person, individual firm and professional person (EntityType <> 1 and EntityType <> 3), the ConsentForPublishing field must always have the value true/1.

8.2.1.2 Examples of a Contact Update request

Example 1
Contact Update for changing phone number and email address, and for the addition of the clientDeleteProhibited to prevent the cancellation of the contact:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 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"
       xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0 contact-1.0.xsd">
        <contact:id>mr0001</contact:id>
        <contact:add>
          <contact:status s="clientDeleteProhibited"/>
        </contact:add>
        <contact:chg>
          <contact:voice>+39.05863152111</contact:voice>
          <contact:email>info@example.it</contact:email>
        </contact:chg>
      </contact:update>
    </update>
    <clTRID>ABC-12345</clTRID>
```

```
  </command>
</epp>
```

Example 2
Contact Update to change the data relating to consent for publication of personal data (in case of Registrants other than natural persons, the ConsentForPublishing field cannot be set to false / 0):

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 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"
       xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0 contact-1.0.xsd">
        <contact:id>mm001</contact:id>
       <contact:chg>
        </contact:chg>
      </contact:update>
    </update>
    <extension>
      <extcon:update
       xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
       xsi:schemaLocation="http://www.nic.it/ITNIC-EPP/extcon-1.0 extcon-1.0.xsd">
        <extcon:consentForPublishing>false</extcon:consentForPublishing>
      </extcon:update>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

8.2.1.3 Examples of responses to a Contact Update request

Example 1
Response to a successful Contact Update:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>31bc0bcb-527b-459f-a7d7-92594f8e9cde</svTRID>
    </trID>
  </response>
</epp>
```

Example 2
Response to a failed Contact Update. The error is due to the fact that the contact cannot be changed

because it is in clientUpdateProhibited. The only change permitted is the removal of the said constraint.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2304">
      <msg lang="en">Object status prohibits operation</msg>
      <extValue>
        <value>
          <extepp:reasonCode>8008</extepp:reasonCode>
        </value>
        <reason lang="en">Contact has status clientUpdateProhibited</reason>
      </extValue>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>a01bcd54-2142-42be-b446-9114add5d966</svTRID>
    </trID>
  </response>
</epp>
```

8.2.1.4 Effects of Contact Update

If the Contact Update requested by the Registrar is executed successfully and passes the validation steps described in Section 0, the contact in the Registry Database is changed as requested. The following contact fields are also updated:

- the Organization field, if empty, is forced to the value of the Name field when the request was a change to set the fields in the Registrant data section and the Registrant is a natural person (i.e., EntityType = 1);
- date of last change;
- the client ID that performed the last change;
- the status of the contact remains unchanged unless the addition or removal of the member of the contact is requested.

8.2.2 Simple change of a registered domain name

The simple change of a domain name registered in the Registry Database is carried out through the use of Domain Update. This command allows the Registrar of the domain name to do the following:

- add and/or remove host (complete IP addresses for subordinate host);
- change of the contact-type admin;
- add and/or remove contact-type tech;
- add and/or remove statuses;
- change of the AuthInfo associated with the domain name.

The Domain Update command is also used for the modification or the removal of the *Delegation Signer* (DS) records associated with a digitally signed domain name by a "DNSSEC accredited" Registrar (Section 11.3.3).

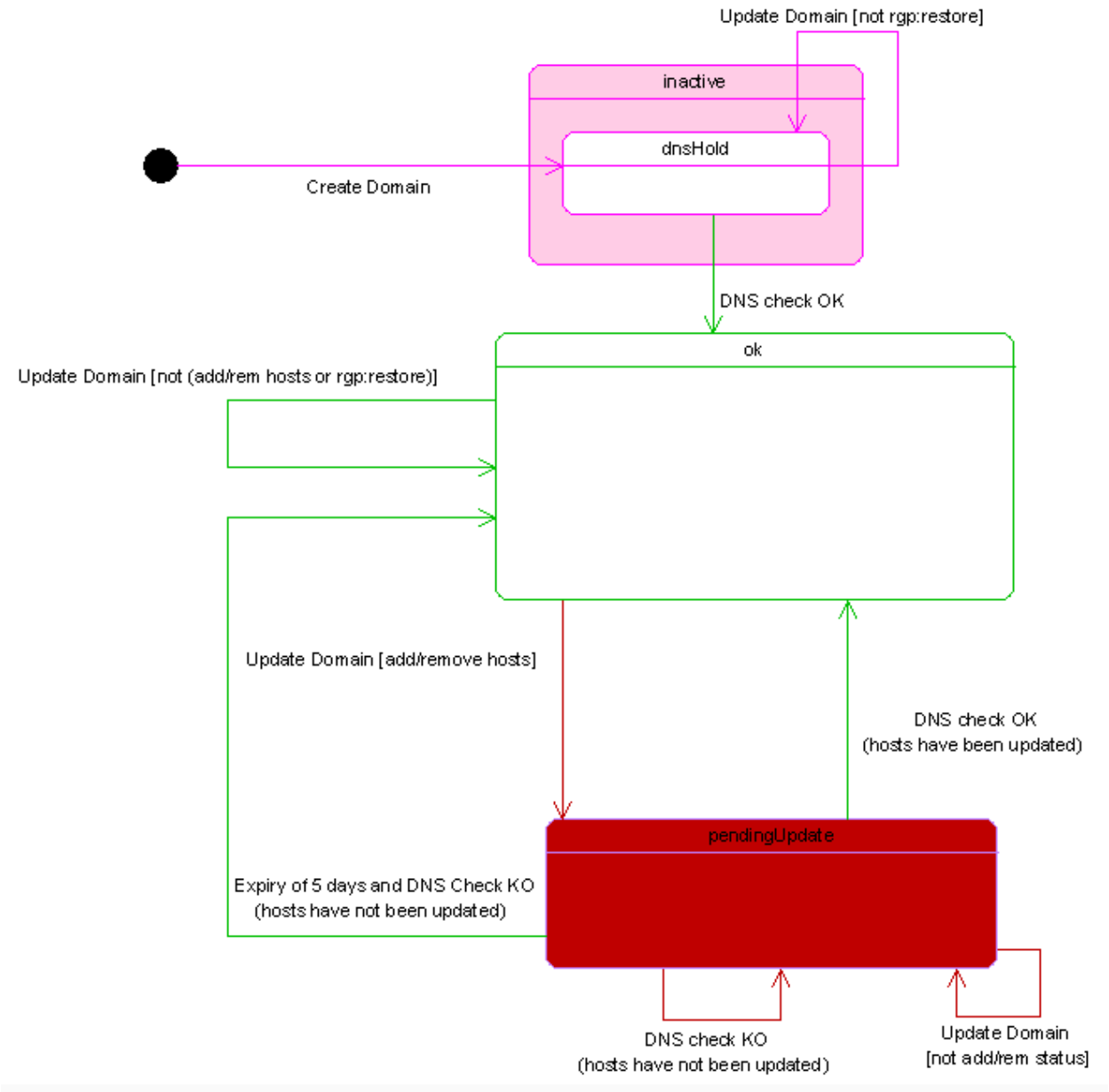
To identify the domain name on which to perform the requested transaction, the Registrar uses the Name field that is mandatory.
If the transaction is required to change the AuthInfo associated with the domain name, the Registrar must notify the Registrant about the new value.
In compliance with the policies adopted by the Registry, the Registrar cannot send a Domain Update command that contains simultaneously more than one of the following operations:

- change of the Registrant;
- change of hosts or of DS records associated with the domain name;
- change of the status;
- recovery of a deleted domain name (Section 0).

However, it is possible to make a change that affects the tech/admin contacts or the AuthInfo at the same time as one of the changes listed above.
In all the changes of hosts associated with the domain name requiring the removal of a host, it is enough to simply return the name to be removed (in section <rem> of the command) without any IP address associated with it.
In all the changes of hosts associated with the domain name requiring the addition of a host, it is enough to simply return the name to be added (in section <add> of the command), together with its possible IP addresses for a subordinate host to the domain name.
The change of the IP addresses of a subordinate host, however, always takes place as the addition of the host with the new IP addresses (in section <add> of the command) and the simultaneous removal of the host for which the change in IP addresses has been requested (in section <rem> of the command) without reporting the IP addresses currently associated with it and currently in the Registry Database (see example 3 in Section 0).
In order to change the hosts associated with a domain name that is in inactive/*dnsHold* (or inactive/*dnsHold*/autoRenewPeriod) status, since the domain name does not have a validated configuration yet, the Registrar has to send a Domain Update command containing only the <add> section with the entire new DNS configuration (see example 4 in Section 0).
In order to change the hosts associated with a domain name that is, on the contrary, in ok (or ok/autoRenewPeriod) or pendingUpdate (or pendingUpdate/autoRenewPeriod) status, since the domain name has a configuration that is already validated, the Registrar has to send a Domain Update command in which the <add> section contains the hosts to add as to the current validated DNS configuration of the domain name (with the IP addresses for the host subordinate to the domain name). The <rem> section, instead, will contain the hosts that must be removed from the validated DNS configuration (without specifying the IP addresses of the subordinate hosts to the domain name - see example 5 in Section 0).
If the Registrant contact (registrant) of the domain name for which a Domain Update has been requested does not contain all the mandatory fields, the change fails, apart from the change of status. This situation may occur, for example, for registrant contacts created in the old "asynchronous" registration system and migrated to the new "synchronous" registration system.
In the event that the Domain Update does not require changes to hosts or requires changes to other fields as well as the host, validation steps are executed on these fields. If the validation steps are unsuccessful, the Registrar receives a negative response and the change is rejected (see examples 5 and 6 in Section

0), so the domain name does not change its status.
If the validation steps have been successful and the changes do not affect the host, the Registrar gets a response that the transaction has been completed successfully, and the changes are reflected immediately in the Registry Database (see examples 1, 2 and 3 in Section 80). The domain name only changes its status if a request has been made to change the status of the domain name itself.
If the validation steps have been successful but the changes affect the host as well, the Registrar gets a response that the transaction was successful but not completely finished (see example 4 in Section 0). The domain name then goes into pendingUpdate for a maximum of 5 (five) days. In this period, the server performs the validation of the DNS (Section 8.1.2.7).
If the DNS check is positive, the changes to the host associated with the domain name appear in the Database and the server will insert into the polling queue a message indicating the successful completion of the change request.
If the DNS check is negative, the server will insert into the polling queue a message that notifies the Registrar of the error found in the new DNS configuration proposal, and the changes to the host associated with the domain name requests are not reported in the Database.
On expiry of the 5 (five) days, the server will insert into the polling queue a message that notifies the Registrar of the expiry of the pendingUpdate period. Thus the change of hosts is not made to the Database and the domain name is put back into its previous status with the old configuration.
During pendingUpdate, the Registrar may change hosts, Registrant and the contacts associated with the domain name and this will mean that the information regarding the old amendment will be replaced with new information but the period of 5 (five) days pendingUpdate status will not begin again.
During the pendingUpdate phase, the Registrar may interrupt the change of hosts operation and reset immediately the domain name status prior to the above mentioned operation without, therefore, waiting for the 5 (five) days provided for pendingUpdate status. In this case, the Registrar will have to send a Domain Update command in which the <add> section contains all the hosts of the last validated DNS configuration accompanied by the IP addresses but only for the hosts subordinate to the domain name. The <rem>, section on the contrary, will contain all the hosts of the last validated DNS configuration without the IP addresses of the hosts subordinated to the domain name (see example 6 in Section 0).

The following diagram shows the various stages that make up the procedure to change the registration of a domain name:



8.2.2.1 Validation steps for the simple change of a registered domain name

The system verifies that the request for Domain Update is compatible with:

- the constraints present in the XML Schema epp-1.0.xsd, eppcom-1.0.xsd, domain-1.0.xsd, host-1.0.xsd (Section 6.1);
- the following additional restrictions:
  - the domain name for which the transaction is requested must be present in the Registry Database;
  - restrictions must apply on the values and the minimum and maximum cardinality of the fields given controlling Domain Create;
  - the domain name for which the transaction was requested must not be in one of the following statuses: pendingTransfer, pendingTransfer/*bulk*, pendingDelete/pendingDelete,

serverUpdateProhibited, inactive/serverHold, inactive/*revoked*, inactive/*toBeReassigned*, ok/*noRegistrar*, inactive/*dnsHold*/*noRegistrar*, inactive/*noRegistrar*;

- if the domain name for which the transaction was requested is in clientUpdateProhibited or inactive/clientHold, the Registrar can only remove that status;
- if the domain name for which the transaction was requested is in inactive/*dnsHold*, the Registrar may submit a new DNS configuration inserting only (in the <add> section of the command) the whole list of hosts to be validated. The <rem> section therefore will not have to be inserted;
- if the domain name for which the transaction was requested is in pendingUpdate, the Registrar cannot add or remove any constraint;
- if the domain name for which the transaction was requested is in pendingDelete/redemptionPeriod, the Registrar may only send either a Domain Update with ext = restore request (Section 0) or a Domain Update request for the addition or the removal of clientTransferProhibited and/or clientUpdatedProhibited constraints;
- the list of statuses by adding or removing statuses cannot contain duplicates;
- a status already associated with the domain name cannot be added;
- a status not associated with the domain name cannot be removed;
- contacts to add or remove with the IDs specified must be in the Registry Database;
- contacts to add or remove the domain name must have been registered by the same Registrar that submits the request for the change of the domain name;
- the contact list to add or remove must not contain the same two contacts with the same role;
- a contact already associated with the domain name cannot be added with the same role;
- the number of admin and technical contacts must comply with the table in Section 12.8;
- a contact that is not associated with the domain name cannot be removed;
- the host list must not contain two hosts with the same IP addresses or with the same name;
- the number of hosts to associate with the domain name must comply with the table in Section 12.8;
- a host already associated with the domain name cannot be added;
- a host that is not associated with the domain name cannot be removed;
- if the Registrant is a natural person (EntityType = 1), the administrative contact cannot be changed;
- the new AuthInfo, if specified in the request, must differ from that stored for the domain name for which the transaction was requested;
- the new AuthInfo, if specified in the request, must have a minimum length of 8 characters and maximum of 32 characters;
- more than one of the following cannot be carried out at the same time:
  - change of hosts associated with the domain name
  - change of the Registrant
  - change of the status
  - recovery of a domain name (Section 0)

8.2.2.2 Examples of Domain Update requests

Example 1
Domain Update command for the change of hosts associated with the domain name, the addition of a technical contact and the clientDeleteProhibited status to prevent the cancellation of the domain name itself:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 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>example.it</domain:name>
        <domain:add>
          <domain:ns>
            <domain:hostAttr>
              <domain:hostName>ns3.example.it</domain:hostName>
              <domain:hostAddr ip="v4">193.205.245.7</domain:hostAddr>
            </domain:hostAttr>
          </domain:ns>
          <domain:contact type="tech">mak21</domain:contact>
          <domain:status s="clientDeleteProhibited">
          </domain:status>
        </domain:add>
        <domain:rem>
          <domain:ns>
            <domain:hostAttr>
              <domain:hostName>ns1.example.it</domain:hostName>
            </domain:hostAttr>
          </domain:ns>
        </domain:rem>
      </domain:update>
    </update>
    <clTRID>ABC-12345</clTRID>
   </command>
 </epp>
```

Example 2

Domain Update to replace one of the nameservers associated with the domain name (ns1.example.it) with a new one (ns4.example.it):

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 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>example.it</domain:name>
        <domain:add>
          <domain:ns>
            <domain:hostAttr>
              <domain:hostName>ns4.example.it</domain:hostName>
              <domain:hostAddr ip="v4">193.205.245.9</domain:hostAddr>
            </domain:hostAttr>
          </domain:ns>
        </domain:add>
        <domain:rem>
          <domain:ns>
            <domain:hostAttr>
              <domain:hostName>ns1.example.it</domain:hostName>
```

```
            </domain:hostAttr>
          </domain:ns>
        </domain:rem>
      </domain:update>
    </update>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example 3

Domain Update for the change of the IP address of one of the authoritative nameservers (ns3.example.it) associated with the domain name. This type of transaction is implemented by adding the nameserver for which the IP address change has been requested (ns3.example.it) by inserting also its new IP address as well as the name of the nameserver (Section 0) and the removal of the same nameserver (ns3.example.it) whose IP address must be changed:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 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>example.it</domain:name>
        <domain:add>
          <domain:ns>
            <domain:hostAttr>
              <domain:hostName>ns3.example.it</domain:hostName>
              <domain:hostAddr ip="v4">193.205.245.8</domain:hostAddr>
            </domain:hostAttr>
          </domain:ns>
        </domain:add>
        <domain:rem>
          <domain:ns>
            <domain:hostAttr>
              <domain:hostName>ns3.example.it</domain:hostName>
            </domain:hostAttr>
          </domain:ns>
        </domain:rem>
      </domain:update>
    </update>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example 4

Change of hosts of a domain name in inactive/*dnsHold* status (Section 0): it is assumed that the last configuration submitted to the DNS validation is ns1.example.com and ns2.example.com and the Registrar wants to change it in ns1.example.com and ns3.example.com. In this case the Registrar has to send a Domain Update command containing only the <add> section with the entire list of nameservers that must be associated with the domain name, as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 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>example.it</domain:name>
        <domain:add>
         <domain:ns>
           <domain:hostAttr>
             <domain:hostName>ns1.example.com</domain:hostName>
           </domain:hostAttr>
           <domain:hostAttr>
             <domain:hostName>ns3.example.com</domain:hostName>
           </domain:hostAttr>
         </domain:ns>
        </domain:add>
      </domain:update>
    </update>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

If the new configuration contains host that are subordinate to the domain name, it is necessary to specify the IP address, for each of them, as in the example below:

```
              <domain:hostAttr>
                  <domain:hostName>ns3.example.it</domain:hostName>
                  <domain:hostAddr ip="v4">193.205.245.8</domain:hostAddr>
             </domain:hostAttr>
```

Example 5

Change of hosts of a domain name in pendingUpdate or ok status (Section 0): it is assumed that the last validated DNS configuration, that is the one that is currently in the Registry Database, is ns1.example.com and ns2.example.com and the Registrar wants to replace the nameserver ns1.example.com with the subordinate nameserver to the domain name ns1.example.it. The Domain Update command that the Registrar will have to send is the following:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 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>example.it</domain:name>
        <domain:add>
         <domain:ns>
           <domain:hostAttr>
             <domain:hostName>ns1.example.it</domain:hostName>
             <domain:hostAddr ip="v4">193.205.245.6</domain:hostAddr>
           </domain:hostAttr>
         </domain:ns>
        </domain:add>
       <domain:rem>
         <domain:ns>
           <domain:hostAttr>
             <domain:hostName>ns1.example.com</domain:hostName>
```

```
          </domain:hostAttr>
        </domain:ns>
       </domain:rem>
     </domain:update>
   </update>
   <clTRID>ABC-12345</clTRID>
 </command>
</epp>
```

Example 6

Change of hosts of a domain name in pendingUpdate status in order to put the domain name again in ok status and with the old configuration (Section 0): it is assumed that the last validated DNS configuration, that is the one that is currently in the Registry Database, is ns1.example.com and ns2.example.com and a validation process of a new DNS configuration whose result is negative is being carried out. In order to put again the domain name in ok status with the current DNS configuration without waiting for the 5 (five) days provided for pendingUpdate status, the Domain Update command to be sent is the following:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 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>example.it</domain:name>
        <domain:add>
         <domain:ns>
           <domain:hostAttr>
             <domain:hostName>ns1.example.com</domain:hostName>
           </domain:hostAttr>
           <domain:hostAttr>
             <domain:hostName>ns2.example.com</domain:hostName>
           </domain:hostAttr>
         </domain:ns>
        </domain:add>
       <domain:rem>
         <domain:ns>
           <domain:hostAttr>
             <domain:hostName>ns1.example.com</domain:hostName>
           </domain:hostAttr>
           <domain:hostAttr>
             <domain:hostName>ns2.example.com</domain:hostName>
           </domain:hostAttr>
         </domain:ns>
        </domain:rem>
      </domain:update>
    </update>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

If the new configuration contains hosts subordinate to the domain name, it is necessary, for each of them, to specify the IP address but only in the <add> section, as in the example below showed:

```
           <domain:hostAttr>
             <domain:hostName>ns3.example.it</domain:hostName>
             <domain:hostAddr ip="v4">193.205.245.8</domain:hostAddr>
```

```
            </domain:hostAttr>
```

8.2.2.3 Examples of responses to a Domain Update request

Example 1
Response to successful Domain Update:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>8b51e1e8-5178-442b-a55e-6bc8990493ea</svTRID>
    </trID>
  </response>
</epp>
```

Example 2
Response to successful Domain Update for the addition of the clientUpdateProhibited status:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <extension>
      <extdom:chgStatusMsgData >
      <extdom:name>esempio.it</extdom:name>
    <extdom:targetStatus>
     <domain:status lang="en" s="clientUpdateProhibited"/>
  </extdom:targetStatus>
  </extdom:chgStatusMsgData>
  </extension>
 <trID>
 <clTRID>ABC-12345</clTRID>
 <svTRID>7f356d29-2b83-4aed-b627-e341672ef6a9</svTRID>
 </trID>
  </response>
</epp>
```

Example 3
Response to a successful Domain Update for the addition of the clientHold status:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <extension>
      <extdom:chgStatusMsgData>
        <extdom:name>esempio.it</extdom:name>
        <extdom:targetStatus>
          <domain:status lang="en" s="inactive"/>
          <domain:status lang="en" s="clientHold"/>
        </extdom:targetStatus>
      </extdom:chgStatusMsgData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>e248879c-9107-4bdc-81ef7a95365e22c7</svTRID>
    </trID>
  </response>
</epp>
```

Example 4

Response to a successful Domain Update for the change of the hosts associated with the domain "example.it". The domain name goes into pendingUpdate while waiting for DNS validation.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <extension>
      <extdom:chgStatusMsgData>
        <extdom:name>example.it</extdom:name>
        <extdom:targetStatus>
          <domain:status lang="en" s="pendingUpdate"/>
        </extdom:targetStatus>
      </extdom:chgStatusMsgData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>b15d5ec6-779a-4619-925ae7d10df167d4</svTRID>
    </trID>
  </response>
</epp>
```

Example 5
Response to a failed Domain Update. The error is due to the fact that the domain name cannot be changed because it is in cleintUpdateProhibited status. The only permitted change is the removal of the said constraint.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2304">
      <msg lang="en">Object status prohibits operation</msg>
      <extValue>
        <value>
          <extepp:reasonCode>9026</extepp:reasonCode>
        </value>
        <reason lang="en">Domain has status clientUpdateProhibited</reason>
      </extValue>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>5d138ab4-92db-43a0-a6d7-3ccedf017b7d</svTRID>
    </trID>
  </response>
</epp>
```

Example 6
Response to a failed Domain Update. The error is due to the fact that the tech contact indicated in the request is not in the Registry Database.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2004">
      <msg lang="en">Parameter value range error</msg>
      <value>
        <extepp:wrongValue>
           <extepp:element>contact</extepp:element>
           <extepp:namespace>urn:ietf:params:xml:ns:domain-1.0</extepp:namespace>
           <extepp:value>TECH25</extepp:value>
        </extepp:wrongValue>
      </value>
      <extValue>
        <value>
          <extepp:reasonCode>9003</extepp:reasonCode>
        </value>
        <reason lang="en">Contact does not exist</reason>
      </extValue>
    </result>
```

```
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>0c2c30ad-3e70-47f8-927f-9d10d72d6755</svTRID>
    </trID>
  </response>
</epp>
```

8.2.2.4 Effects of Domain Update for simple change

If the Domain Update command submitted by the Registrar is executed successfully and passes the validation steps described in Section 8.2.2.1, the registration of the domain name in the Registry Database is changed in accordance with the changes requested. The following fields of the domain are also updated:

- date of last change;
- the client ID that performed the last change;
- the status of the domain name remains unchanged unless the addition or removal of the statuses of the domain name is requested.

8.3 Change of Registrant

The change of Registrant allows the Registrar to change the assignee of the domain name at the request of the new Registrant. To achieve this type of transaction, the old Registrant has to inform the new Registrant regarding the value of the AuthInfo associated with the domain name. At the same time as the change of the Registrant, the Registrar will provide new AuthInfo which is different from that currently associated with the domain name.
The change in the Registrant is made by using Domain Update.

8.3.1 Validation steps for the change of Registrant

The system verifies that the request for Domain Update is compatible with:

- the constraints present in the XML Schema epp-1.0.xsd, eppcom-1.0.xsd, domain-1.0.xsd (Section 6.1);
  - the constraints present in the XML Schema extgovdom-1.0.xsd, if the request is for creating a gov.it domain name;
- the following additional restrictions:
  - the Registrar that has sent the request must not be suspended;
  - the domain name for which the transaction was requested must be present in the Registry Database;
  - the domain name for which the transaction was requested must not be in the status pendingTransfer, pendingTransfer/*bulk*, pendingDelete/pendingDelete, pendingDelete/redemptionPeriod, inactive/clientHold, inactive/serverHold, clientUpdateProhibited, serverUpdateProhibited, inactive/*revoked*, inactive/*toBeReassigned*. If the domain name is in the status ok/*noRegistrar*, inactive/*dnsHold*/*noRegistrar* or inactive/*noRegistrar* the change of Registrant is allowed only if it is associated with the change of Registrar (Section 8.5);
  - the new Registrant with the specified ID must be a contact in the Registry Database with all the compulsory data and the section related to the Registrant data;

- if the new Registrant with the specified ID is a subject other than a natural person individual firm and professional firm (EntityType <> 1 and EntityType <> 3), the ConsentForPublishing field of the new Registrant must be set to true / 1;
- the new AuthInfo specified in the request must differ from the one associated with the domain name for which the transaction was requested;
- the new AuthInfo specified in the request must have a minimum length of 8 characters and maximum of 32 characters.
- if the domain name is a gov.it domain name, the EPP server checks that:
  - the Registrar added the extgovdom-1.0 namesapce to the Login request;
  - the ValidationCode field is present;
  - the ValidationCode field has not expired;
  - the new Registrant contact, referenced in the request, contains the values of the RegCode, IpaCode, and possibly UoCode fields, corresponding to the Validation Code obtained during the Validation Procedure.

8.3.2 Example Domain Update request for the change of Registrant

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 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>example.it</domain:name>
       <domain:chg>
          <domain:registrant>mr002</domain:registrant>
          <domain:authInfo>
            <domain:pw>NEW2BARfoo</domain:pw>
          </domain:authInfo>
        </domain:chg>
      </domain:update>
    </update>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

8.3.3 Gov.it Domains Specifics

For gov.it domains, the validation code must be indicated as an extension in the EPP Update Domain request.

```
  <extension>
    <extgovdom:update>
      <extgovdom:validationCode>validationCode</extgovdom:validationCode>
    </extgovdom:update>
  </extension>
```

8.3.4 Effects of the Domain Update for the change of Registrant

If the Domain Update command for changing the Registrant requested by the Registrar is executed successfully and passes the validation steps described in Section 0, the domain object is changed and it

is associated with the new Registrant and the new AuthInfo.
If the new Registrant is a natural person (EntityType = 1) and no new administrative contact is specified in the request, the system automatically changes it by entering the ID associated with the new Registrant.
The Registry sends an email to the new and the old Registrant in the following formats:

Mail for the new Registrant (in case of natural person - EntityType = 1 and in case of individual firm and professional person - EntityType = 3):

Subject: 10300 - New registration of the domain name <name of the domain>

We inform you that on <registration date> the domain name <name of the domain> has been registered through the Registrar < Registrar>.

The outcome of the registration is:

Registrant:
Address:
Country:
Nationality:
Phone:
Fax:
RegCode:
Email:
EntityType:

The domain name <name of the domain> has been put in <status> status.

The Registrant, has made the following choice concerning consent:

- having read the information regarding personal data treatment for the registration: YES
- consent for personal data treatment for diffusion and accessibility via the Internet: <YES/NO (value of consentForPublishing)>

and has made the following declarations and has accepted the following clauses:

- to be citizen or resident in a country of the European Economic Area (EEA), the Vatican City State, the Republic of San Marino, the Swiss Confederation or the United Kingdom;
- to be aware of and to accept that the registration and management of the domain name are subject to the “Rules of assignment and management of domain names in the ccTLD .it” and the “Regulations for the Resolution of Disputes in the ccTLD .it” and subsequent modifications;
- to have right of use and/or legal availability of the registered domain name requested and not to prejudice, with this registration request, the rights of third parties;
- to be aware that in order to fulfill personal data on the database of assigned names, and for their possible diffusion and accessibility on Internet, it is necessary to give express consent checking the relevant boxes on the basis of the information below. On the Registry website (http://www.nic.it) the document “The policy of the .it Registry about the Whois database” is available;
- to be aware of and to accept that in the case of erroneous or false declaration in the present request, the Registry will proceed to the immediate revocation of the domain name, reserving the right to take out further legal action. In this case the revocation cannot give rise in any way whatsoever to

requests for damages to the Registry;

- to release the Registry from any responsibility deriving from assignment and use of the domain name on the part of the requesting subject;
- to accept Italian jurisdiction and the laws of the Italian State Ordinance.

We inform you that the Registrar mentioned above is responsible for personal data treatment and that the CNR, through the Institute of Informatics and Telematics, is the holder.
As specified in the Registration Form, the data will be released to third parties for the activation of opposition and for the defense of rights as well as the fulfillment of obligations of law or regulation.
Should you need further information, please contact the Registrar indicated in the registration and whose data are also available on the website of the Registry http://www.nic.it.

Best regards,

Registro .it
Istituto di Informatica e Telematica
CNR - AREA DELLA RICERCA
Via Giuseppe Moruzzi, 1 - I-56124 PISA
Tel: +39 050 3139811
Email: hostmaster@nic.it

Mail for the new Registrant (in case of subjects other than natural person, individual firm and professional person - EntityType <> 1 and EntityType <> 3):

Subject: 10307 - New registration of the domain name <name of the domain>

We inform you that on <registration date> the domain name <name of the domain> has been registered through the Registrar <Registrar>.

The outcome of the registration is:

Registrant:
Address:
Country:
Phone:
Fax:
RegCode:
Email:
EntityType:

The domain name <name of the domain> has been put in <status> status.

The Registrant, has made the following choice concerning consent:

- having read the information regarding personal data treatment for the registration: YES

and has made the following declarations and has accepted the following clauses:

- to have the registered office based in a country of the European Economic Area (EEA), the Vatican City State, the Republic of San Marino, the Swiss Confederation or the United Kingdom;

- to be aware of and to accept that the registration and management of a domain name are subject to the “Rules of assignment and management of domain names in the ccTLD .it” and the “Regulations for the Resolution of Disputes in the ccTLD .it” and subsequent modifications;
- to have right of use and/or legal availability of the registered domain name requested and not to prejudice, with this registration request, the rights of third parties;
- to be aware of and to accept that in the case of erroneous or false declaration in the present request, the Registry will proceed to the immediate revocation of the domain name, reserving the right to take out further legal action. In this case the revocation cannot give rise in any way whatsoever to requests for damages to the Registry;
- to release the Registry from any responsibility deriving from assignment and use of the domain name on the part of the requesting subject;
- to accept Italian jurisdiction and the laws of the Italian State Ordinance.

We inform you that the Registrar mentioned above is responsible for personal data treatment and that the CNR, through the Institute of Informatics and Telematics, is the holder.
As specified in the Registration Form, the data will be released to third parties for the activation of opposition and for the defense of rights as well as the fulfillment of obligations of law or regulation.
Should you need further information, please contact the Registrar indicated in the registration and whose data are also available on the website of the Registry http://www.nic.it.

Best regards,

Registro .it
Istituto di Informatica e Telematica
CNR - AREA DELLA RICERCA
Via Giuseppe Moruzzi, 1 - I-56124 PISA
Tel: +39 050 3139811
Email: hostmaster@nic.it

Mail to the former Registrant:

Subject: 10301 - Modification of the Registrant for the domain name <name of the domain>

We inform you that an operation of modification of the Registrant for the domain name <name of the domain> was made on <date of the operation>. Therefore <former Registrant> is no longer the Registrant of the domain name in object.

Best regards,

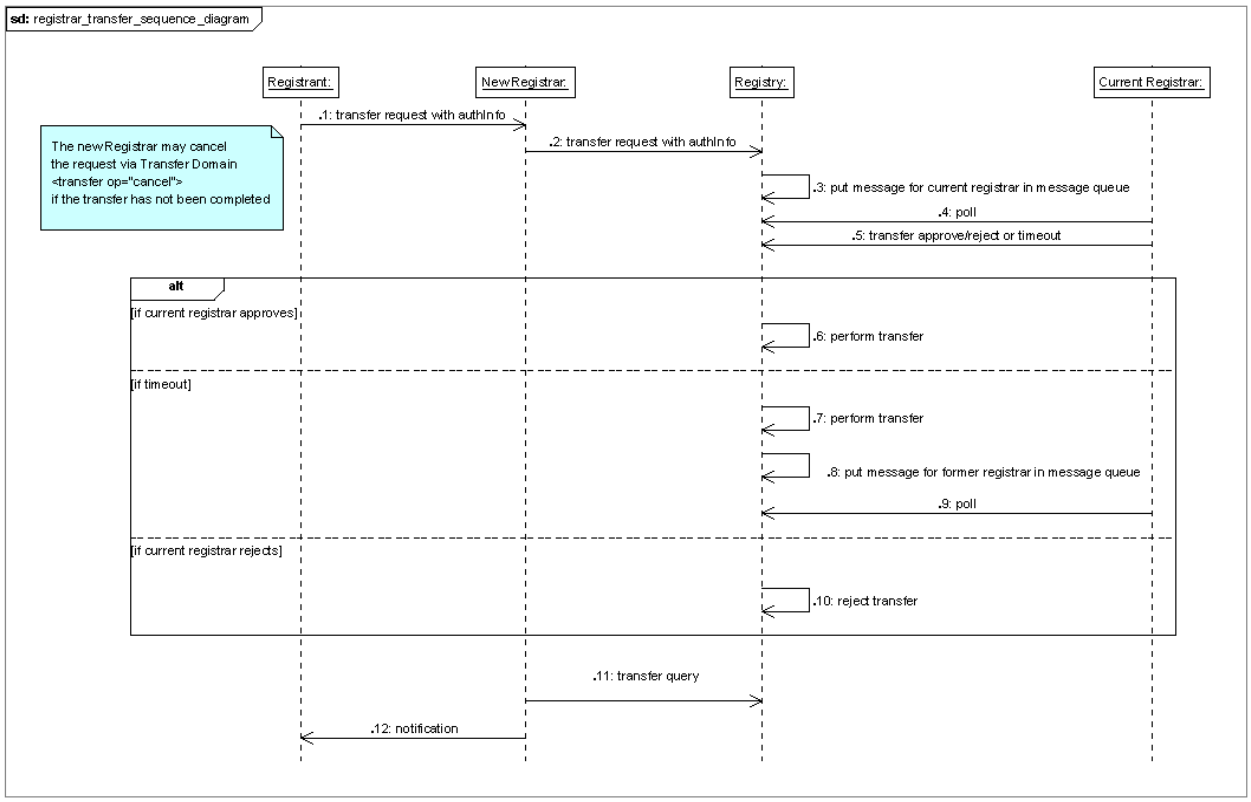
Registro .it
Istituto di Informatica e Telematica
CNR - AREA DELLA RICERCA
Via Giuseppe Moruzzi, 1 - I-56124 PISA
Tel: +39 050 3139811
Email: hostmaster@nic.it

8.4 Change of Registrar

The change of Registrar means that a domain name can be transferred from the current Registrar to another Registrar.
The request to change the Registrar is made by the new Registrar, on behalf of the Registrant, using Domain Transfer with the attribute op = “request”.
To achieve this, the Registrant must first notify the new Registrar regarding the AuthInfo currently associated with the domain name, so that the new Registrar can insert it in the request to change the Registrar (Section 0).
If Domain Transfer with op = “request” is successfully performed by the new Registrar and passes the validation steps described in section 8.4.1, the domain name goes into pendingTransfer (pendingTransfer/autoRenewPeriod if the domain name was in ok/autoRenewPeriod or inactive/*dnsHold*/autoRenewPeriod status).
The domain name remains in this status up to a maximum of 1 (one) day during which the new Registrar can cancel the transaction by sending a Domain Transfer command with the attribute op = “cancel” (Section 0).
The old Registrar, however, may take the following actions:

- accept the request to change the Registrar by sending Domain Transfer with the attribute op = “approve” (Section 0);
- reject the request to change the Registrar by sending Domain Transfer with the attribute op = “reject” (Section 0).

The commands for the deletion, approval and/or non-approval of transfer by the new and the old Registrar may be submitted without the AuthInfo for as long as the domain name is in pendingTransfer.
At the end of the pendingTransfer period, requests that have not been explicitly cancelled, rejected or approved are approved automatically by the system.
The server notifies both the old and the new Registrar of each step generated by the change of Registrar via a message in the polling queue.
Any information relating to a transfer in progress or completed may also be viewed by both the Registrars using the command Domain Transfer with the attribute op = “query”. This command will be addressed in detail in Section 0.
Once the domain name is in pendingTransfer status, the change of the Registrar is charged to the new Registrar, but not immediately included in the transactions to be invoiced. The transaction will be invoiced at the end of the change of Registrar, once the transfer has been accepted either by the old Registrar or automatically by the system. In the other cases, the transaction will be returned to the new Registrar and will thus not be billed.



8.4.1 Validation steps for the modification of the Registrar

The system verifies that the request for Domain Transfer with op = “request” is compatible with:

- the constraints present in the XML Schema epp-1.0.xsd, eppcom-1.0.xsd, domain-1.0.xsd, RGP-1.0.xsd, extdom-2.0.xsd (Section 6.1);
- the following additional restrictions:
  - the Registrar that has sent the request must not be suspended;
  - the domain name for which the transaction is requested must be present in the registry Database;
  - the domain name for which the transaction is requested must not be in pendingTransfer, pendingTransfer/*bulk*, pendingUpdate, pendingDelete/pendingDelete, inactive/clientHold, inactive/serverHold, pendingDelete/redemptionPeriod, inactive/*revoked*, clientTransferProhibited, serverTransferProhibited, inactive/*toBeReassigned*;
  - the AuthInfo specified in the request must match the AuthInfo associated with the domain name in the registry Database.

8.4.2 Example of a Domain Transfer request with op = “request”

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <transfer op="request">
      <domain:transfer
       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>example.it</domain:name>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:transfer>
    </transfer>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

8.4.3 Example of a response to a request for Domain Transfer with op = “request”

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1001">
      <msg lang="en">Command completed successfully; action pending</msg>
    </result>
    <resData>
      <domain:trnData>
        <domain:name>esempio.it</domain:name>
        <domain:trStatus>pending</domain:trStatus>
        <domain:reID>NEW-REGISTRAR</domain:reID>
        <domain:reDate>2013-02-25T07:40:00+01:00</domain:reDate>
        <domain:acID>DEMO-REGISTRAR</domain:acID>
        <domain:acDate>2013-02-25T23:59:59+01:00</domain:acDate>
       </domain:trnData>
    </resData>
    <trID>
       <clTRID>ABC-12345</clTRID>
       <svTRID>4abdc58b-88c6-410d-9a64-c476231ccffb</svTRID>
    </trID>
  </response>
</epp>
```

8.4.4 Example of a request for Domain Transfer with op = “cancel”

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <transfer op="cancel">
      <domain:transfer
       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>example.it</domain:name>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:transfer>
    </transfer>
    <clTRID>ABC-12345</clTRID>
```

```
  </command>
</epp>
```

8.4.5 Example of a response to a request for Domain Transfer with op = “cancel”

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1001">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:trnData>
       <domain:name>example.it</domain:name>
         <domain:trStatus>clientCancelled</domain:trStatus>
         <domain:reID>NEW-REGISTRAR</domain:reID>
         <domain:reDate>2013-02-25T07:40:00+01:00</domain:reDate>
         <domain:acID>DEMO-REGISTRAR</domain:acID>
         <domain:acDate>2013-02-25T15:00:59+01:00</domain:acDate>
       </domain:trnData>
      </resData>
      <trID>
        <clTRID>ABC-12345</clTRID>
        <svTRID>c1a27773-0527-4fc3-8f4f-00a4134d37bb</svTRID>
      </trID>
   </response>
</epp>
```

8.4.6 Example of a request for Domain Transfer with op = “approve”

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <transfer op="approve">
      <domain:transfer
       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>example.it</domain:name>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:transfer>
    </transfer>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

8.4.7 Example of a response to a request for Domain Transfer with op = “approve”

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1001">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:trnData>
        <domain:name>example.it</domain:name>
        <domain:trStatus>clientApproved</domain:trStatus>
        <domain:reID>NEW-REGISTRAR</domain:reID>
        <domain:reDate>2013-02-25T07:40:00+01:00</domain:reDate>
        <domain:acID>DEMO-REGISTRAR</domain:acID>
        <domain:acDate>2013-02-25T11:15:00+01:00</domain:acDate>
      </domain:trnData>
    </resData>
    <trID>
     <clTRID>ABC-12345</clTRID>
     <svTRID>83829902-9aca-49d6-869d-b131232e80f9</svTRID>
    </trID>
  </response>
</epp>
```

8.4.8 Example of a request for Domain Transfer with op = “reject”

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <transfer op="reject">
      <domain:transfer
       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>example.it</domain:name>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:transfer>
    </transfer>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

8.4.9 Example of a response to a request for Domain Transfer with op = “reject”

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
```

```
  <response>
    <result code="1001">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:trnData>
        <domain:name>example.it</domain:name>
        <domain:trStatus>clientRejected</domain:trStatus>
        <domain:reID>NEW-REGISTRAR</domain:reID>
        <domain:reDate>2013-02-25T07:40:00+01:00</domain:reDate>
        <domain:acID>DEMO-REGISTRAR</domain:acID>
        <domain:acDate>2013-02-25T12:25:00+01:00</domain:acDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>600e7e8f-8c0d-4391-b225-2cea22b525b3</svTRID>
    </trID>
  </response>
</epp>
```

8.4.10 Effects of a Domain Transfer

If the change of Registrar is successful, i.e. endorsed by the old Registrar or automatically by the system:

- the “registrant” and “admin” contacts referenced in the registration of the domain name are automatically copied by the system and they are assigned a new ID in the format DUPn where:
- “DUP” is a string of characters that is always present;
- “n” is a random number (9 digits);
- the technical contact of the domain name is updated with the same contactID (in the format DUPn) of the Registrant referenced in the registration of the domain name;
- the expiry date of the domain name is updated;
- the domain name is put in ok or inactive/*dnsHold* status;
- the transaction is billed to the new Registrar.

The new Registrar may use the contacts with ID in the format DUPn or register their own contacts to be associated with the domain name (via the Contact Create + Domain Update, in Sections **Errore. L'origine riferimento non è stata trovata.** and 0, respectively). It is recommended that they use their contact-IDs by creating new ones, if not yet registered in the Registry Database. If the new Registrar wishes to replace the DUPn IDs, they will have to first register their IDs (if not yet present in the Database) using Contact Create (Section **Errore. L'origine riferimento non è stata trovata.**). Subsequently, using Domain Update, the Registrar will update the registration of the domain name in order to replace the various DUPn IDs with the IDs that they have just registered (Section 0).
It is the new Registrar’s responsibility to update the value of the AuthInfo and, if necessary, the hosts/Delegation Signer (DS) records associated with the domain name (also through a Domain Update, Section 0).
If the new Registrar has opted for the replacement of the DUPn associated with the Registrant with their own ID and also wishes to change the host associated with the domain name, they must carry out two Domain Update transactions:

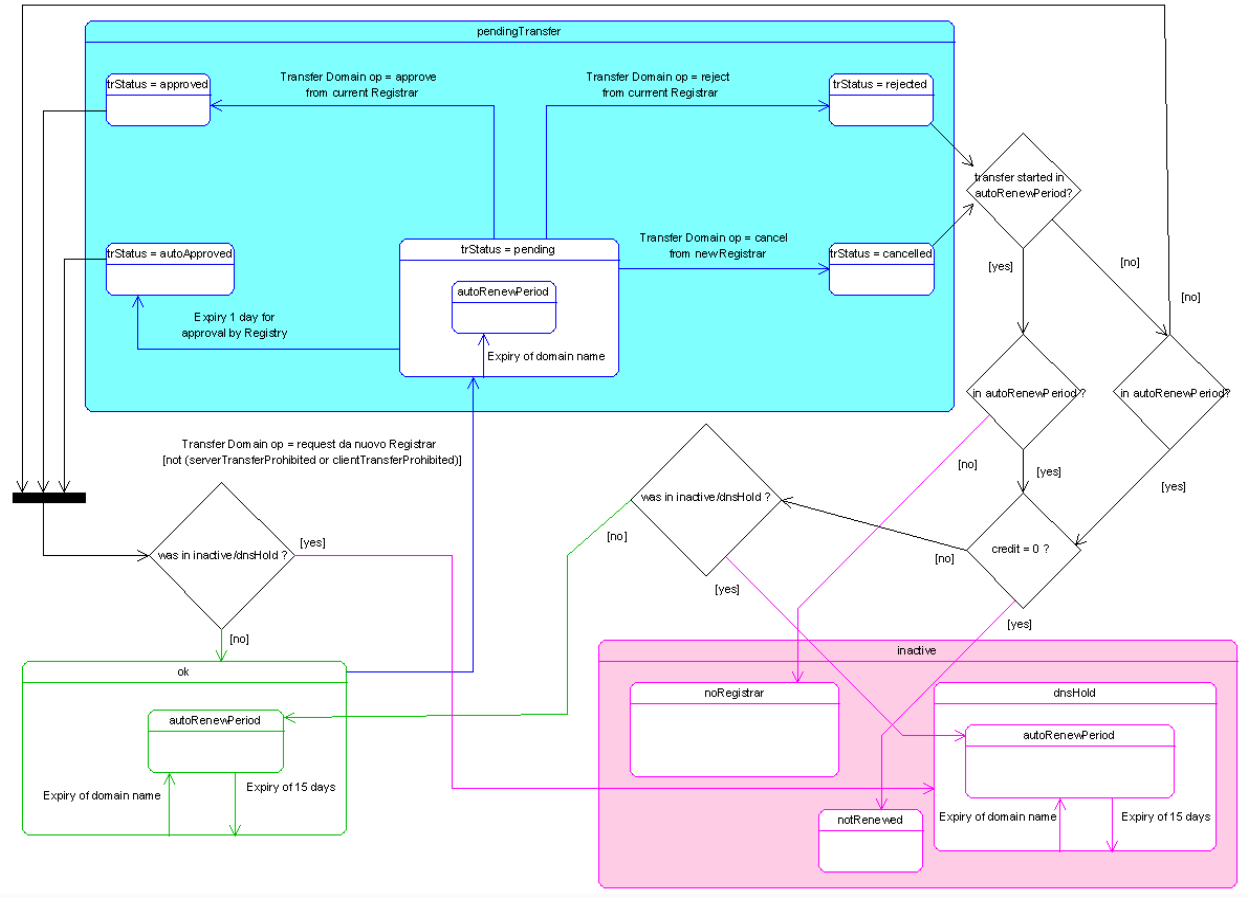
- the first to change the hosts associated with the domain name and, if necessary, replace the IDs of

the technical and administrative contacts;

- the second to change the Registrant by replacing the DUPn with the new ID at the same time as the change of AuthInfo associated with the domain name.

The order in which these Domain Update requests are submitted to the server can also be inverted.
On the other hand, if the new Registrar has decided to keep the duplicate Registrant (DUPn), they can change the host name associated with the domain with just one Domain Update (Section 0), replace the AuthInfo and, if necessary, the IDs of technical and administrative contacts.
Any DUPn duplicate contacts that are no longer referenced in any domain name in the Registry Database, may be cancelled by the Registrar using Contact Delete (Section 0).
If the change of the Registrar is not successful, i.e. is rejected by the old Registrar or cancelled by the new Registrar:

- the cost is re-debited to the new Registrar and therefore will not be invoiced;
- the domain name is placed in a status that can vary depending on whether the transaction started or not in the autoRenewPeriod and whether the old Registrar has enough credit or not, so:
  - if the transaction started in autoRenewPeriod status and the auto renew period has not yet expired, the domain name goes into one of the following statuses:
    - ok/autoRenewPeriod or inactive/*dnsHold*/autoRenewPeriod if the old Registrar has sufficient credit;
    - inactive/*notRenewed* if the old Registrar has insufficient credit;
  - if the transaction started in autoRenewPeriod status and the auto renew period has expired, the domain name goes into inactive/*noRegistrar*;
  - if the transaction did not start in autoRenewPeriod status and the auto renew period in the meanwhile has expired, the domain name goes into one of the following statuses:
    - ok/autoRenewPeriod or inactive/*dnsHold*/autoRenewPeriod if the old Registrar has sufficient credit;
    - inactive/*notRenewed* if the old Registrar has insufficient credit;
  - if the transaction did not start in autoRenewPeriod, and there has been no expiry, the domain name goes back to ok or inactive/*dnsHold* status.



8.5 Change of Registrar with the simultaneous change in the Registrant

The change of Registrar and change of Registrant of a domain name registered in the Registry Database may be performed simultaneously using a Domain Transfer-Trade. The command is similar to that used for the change of the Registrar (Domain Transfer with op = “request”, Section 0) with an additional extension (<extdom:trade>) that allows the new Registrar, at the request of the new Registrant, to change the Registrant and consequently the value of the AuthInfo associated with the domain name itself. To do this type of transaction, the old Registrant has to inform the new Registrant of the value of the AuthInfo currently associated with the domain name, which the new Registrar will then have to insert in the request. At the same time as the change of the Registrant with the change of the Registrar, the new Registrar must provide new AuthInfo (Section 0) that must be notified to the new Registrant.
The new Registrar is only charged for the cost relating to the change of the Registrar.
Regarding the interaction of the new and old Registrar with the EPP server and the possibilities that the server has for notifying both the steps of the status of the transfer, what was said for the change of the Registrar is valid here too (Section 8.4).

8.5.1 Validation steps for the change of the Registrar with a simultaneous change of the Registrant

The system verifies that the Domain Transfer-Trade request is compatible with:

- restrictions on the request for Domain Transfer with op = “request” for the change of the Registrar (Section 8.4.1);

- restrictions on the Domain Update request for the change of Registrant (Section 0).

8.5.2 Example of a request for Domain Transfer-Trade

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <transfer op="request">
      <domain:transfer
       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>example.it</domain:name>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:transfer>
    </transfer>
    <extension>
      <extdom:trade
       xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
       xsi:schemaLocation="http://www.nic.it/ITNIC-EPP/extdom-2.0 extdom-2.0.xsd">
        <extdom:transferTrade>
          <extdom:newRegistrant>ml001</extdom:newRegistrant>
          <extdom:newAuthInfo>
            <extdom:pw>NEW2fooBAR</extdom:pw>
          </extdom:newAuthInfo>
        </extdom:transferTrade>
      </extdom:trade>
    </extension>
  <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

8.5.3 Gov.it Domains Specifics

For gov.it domains, the validation code must be indicated as an extension in the EPP Transfer Domain request.

```
  <extension>
    <extdom:trade>
      <extdom:transferTrade>
        <extdom:newRegistrant>ml001</extdom:newRegistrant>
        <extdom:newAuthInfo>
          <extdom:pw>NEW2fooBAR</extdom:pw>
        </extdom:newAuthInfo>
      </extdom:transferTrade>
    </extdom:trade>
    <extgovdom:trade>
      <extgovdom:validationCode>validationCode</extgovdom:validationCode>
    </extgovdom:trade>
  </extension>
```

8.5.4 Examples of responses to a request for Domain Transfer-Trade

Example 1

Response to a successful Domain Transfer-Trade:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1001">
      <msg lang="en">Command completed successfully; action pending</msg>
    </result>
    <resData>
      <domain:trnData>
        <domain:name>esempio.it</domain:name>
        <domain:trStatus>pending</domain:trStatus>
        <domain:reID>NEW-REGISTRAR</domain:reID>
        <domain:reDate>2009-02-16T22:30:14+01:00</domain:reDate>
        <domain:acID>DEMO-REGISTRAR</domain:acID>
        <domain:acDate>2009-02-16T23:59:59+01:00</domain:acDate>
      </domain:trnData>
    </resData>
    <extension>
      <extdom:trade>
        <extdom:transferTrade>
          <extdom:newRegistrant>ml001</extdom:newRegistrant>
          <extdom:newAuthInfo>
            <extdom:pw>NEW2fooBAR</extdom:pw>
          </extdom:newAuthInfo>
        </extdom:transferTrade>
      </extdom:trade>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>f656389e-e858-4740-9480-3b8a06b20be9</svTRID>
    </trID>
  </response>
</epp>
```

Example 2
Response to a failed Domain Transfer-Trade. The error is due to wrong AuthInfo in the request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2202">
      <msg lang="en">Invalid authorization information</msg>
      <value>
        <extepp:wrongValue>
          <extepp:element>name</extepp:element>
          <extepp:namespace>urn:ietf:params:xml:ns:domain-1.0</extepp:namespace>
          <extepp:value>unknown.it</extepp:value>
```

```
        </extepp:wrongValue>
      </value>
      <extValue>
        <value>
          <extepp:reasonCode>9085</extepp:reasonCode>
        </value>
        <reason lang="en">Invalid domain authorization information or domain does
not exist</reason>
      </extValue>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>4162bf80-2de2-42bc-937a-72255b0cd83c</svTRID>
    </trID>
  </response>
</epp>
```

Example 3
Response to a failed Domain Transfer-Trade. The error is due to the fact that the domain is not in a status allowed by the transaction requested:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2304">
     <msg lang="en">Object status prohibits operation</msg>
      <extValue>
        <value>
          <extepp:reasonCode>9022</extepp:reasonCode>
        </value>
        <reason lang="en">Domain has status clientTransferProhibited</reason>
      </extValue>
    </result>
    <trID>
     <clTRID>ABC-12345</clTRID>
     <svTRID>feb2a1bf-7635-4c80-9f0c-8bda707e396c</svTRID>
    </trID>
   </response>
  </epp>
```

8.5.5 Effects of Domain Transfer-Trade

The impact of the Transfer-Trade is the sum of those of a Domain Transfer for the change of the Registrar (Section 0) with those of a Domain Update for the change of the Registrant (Section 0).
If the change of the Registrar change is successful, i.e. is approved by old Registrar or automatically by the system:

- the domain object is changed by associating with it both the new Registrant and the new AuthInfo;
- to the “admin” and “tech” contacts are assigned the same contactID of the new Registrant of the domain name indicated in the Domain Transfer-Trade command;
- the expiry date of the domain name is updated;

- the domain name goes into ok or inactive/*dnsHold* status;
- the transaction is billed to the new Registrar;
- the system sends an email to both the new and the old Registrant in the format specified in Section 0.

If the change of the Registrar along with the simultaneous change of Registrant is completed successfully, the new Registrar can, if necessary, update the hosts/Delegation Signer (DS) records associated with the domain name and the technical and/or administrative contact with just one Domain Update transaction (Section 0). Please note that the administrative contact can be changed only in the event the new Registrant is different from a natural person (EntityType <> 1).
If the change of Registrar is not successful, i.e. is rejected by the old Registrar or cancelled by the new Registrar:

- the change of the Registrant is not made, with the result that the Registrant and the AuthInfo currently associated with the domain name remain unchanged;
- the cost is re-credited to the new Registrar and is not invoiced;
- the status of the domain name changes according to the procedures described in Section 0.

8.6 Procedures for deletion and recovery

Deletes are carried out using the EPP Delete command: the Registrar must submit a Contact Delete or Domain Delete request, depending on the object to be deleted. A restore can be made following a cancellation, but only for domain objects.

8.6.1 Deletion of a registered contact

The deletion of a "registrant", "admin" or "tech" contact registered in the Registry Database is carried out by using Contact Delete.
This can only be requested by the Registrar and can be executed only if the contacts for which the cancellation request are not referenced in any domain object in the Registry Database and thus are not in ok/*linked* status.
In any case, the Registry will automatically delete all the contacts in the Database that have not been referenced in the registration of any domain name for more than 60 (sixty) days.

8.6.1.1 Contact Delete

Contacts are deleted using Contact Delete. To identify the contact on which to perform the requested transaction, the Registrar must use the ID of the contact.

8.6.1.2 Validation steps for deleting a contact

The system verifies that the request to Contact Delete is compatible with:

- the constraints present in the XML Schema epp-1.0.xsd, eppcom-1.0.xsd, contact-1.0.xsd (Section 6.1);
- the following additional restrictions:
    - the contact with the specified ID must be present in the Registry Database;

- the contact with the specified ID must not be referenced in the registration of a domain name (i.e. the status of the contact for which the cancellation has been requested must be different from ok/linked);
- the contact with the specified ID must not be in one of the following statuses: ok/linked, clientDeleteProhibited, serverDeleteProhibited.

8.6.1.3 Example of request for Contact Delete

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <delete>
      <contact:delete
       xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
       xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0 contact-1.0.xsd">
        <contact:id>cl001</contact:id>
      </contact:delete>
    </delete>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

8.6.1.4 Examples of responses to a request for Contact Delete

Example 1
Response to a successful request for Contact Delete:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>96e857f8-449d-41c4-9591-999d1ee5510e</svTRID>
    </trID>
  </response>
</epp>
```

Example 2
Response to unsuccessful Contact Delete. The error is due to the fact that the contact is referenced in the registration of at least one domain name.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
```

```
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2305">
      <msg lang="en">Object association prohibits operation</msg>
      <value>
        <extepp:wrongValue>
          <extepp:element>id</extepp:element>
          <extepp:namespace>urn:ietf:params:xml:ns:contact-1.0</extepp:namespace>
          <extepp:value>MB8015</extepp:value>
        </extepp:wrongValue>
      </value>
      <extValue>
        <value>
          <extepp:reasonCode>8005</extepp:reasonCode>
        </value>
        <reason lang="en">Contact is associated with domains</reason>
      </extValue>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>a84849d3-ac01-4e4b-b9cd-36948dda02bd</svTRID>
    </trID>
  </response>
</epp>
```

8.6.1.5 Effects of Contact Delete

If the Contact Delete requested by the Registrar is executed successfully and passes the validation steps described in Section 0, the object contact is immediately removed from the list of active contacts and, therefore, cannot be subject to any transaction.

8.6.2 The Drop Time

The Drop Time process enables the cancellation of domain names at scheduled times.

8.6.2.1 The Drop Time in the ccTLD .it

The .it domain names that are in pendingDelete/pendingDelete status, are definitively cancelled from the Database of the Registry according to the Drop Time process.
At 01:00:00 of each day the Registry automatically creates the lists containing the domain names that in the previous day were in pendingDelete/pendingDelete status. These lists, as soon as they are created, are published in the dedicated section of the website of the Registry (http://www.nic.it/droptime). The above mentioned lists, in addition to the list of domain names, contain the date and exact time in which the domain names will be cancelled.
Once cancelled, the domain names are immediately available online for free allocation.
Please note that, in order to register a domain name that has been subject to a cancellation in the previous 7 (seven) days, the Registrar will have to submit the Domain Create request to the server "epp-deleted.nic.it" (Section 8.1.3).

8.6.2.2 Example of cancellation according to Drop Time process

Suppose to have 100 domain names that on May 2 have been put in pendingDelete/pendingDelete status:

- these domain names, at 01:00:00 of May 3, will compose, for example, two lists of domain names in cancellation (list A and list B);
- the two lists are published on the Website of the Registry immediately after their creation;
- the domain names indicated in the first list (list A) are cancelled at 09:00 of May 5;
- the domain names indicated in the second list (list B) are cancelled at 16:00 of May 5.

8.6.3 Deleting a registered domain name

A domain name registered in the Registry Database is deleted using Domain Delete.
The transaction allows the Registrar to cancel a domain name at the request of the Registrant or a competent Authority, that is, the maintenance period having expired, when the juridical obligation on the basis of which the Registrar was obliged to maintain the domain name is terminated, in fact or in law.

8.6.3.1 Validation steps for the deletion of a domain name

The system verifies that the Domain Delete request is compatible with:

- the constraints present in the XML Schema epp-1.0.xsd, eppcom-1.0.xsd, domain-1.0.xsd (Section 6.1);
- the following additional restrictions:
  - the domain name for which the transaction is requested must be present in the Registry Database;
  - the domain name for which the transaction is requested must not be in pendingDelete/pendingDelete, pendingDelete/redemptionPeriod, pendingUpdate, pendingTransfer, pendingTransfer/*bulk*, ok/*noRegistrar*, inactive/*dnsHold*/*noRegistrar*, inactive/*noRegistrar*, inactive/serverHold, inactive/clientHold, clientDeleteProhibited, serverDeleteProhibited, inactive/*revoked*, inactive/*toBeReassigned*.

8.6.3.2 Example of a request for Domain Delete

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <delete>
      <domain:delete
       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>example.it</domain:name>
      </domain:delete>
    </delete>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

8.6.3.3 Examples of responses to a request for Domain Delete

Example 1
Response to a successful Domain Delete:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
  <result code="1000">
    <msg lang="en">Command completed successfully</msg>
   </result>
  <extension>
    <extdom:chgStatusMsgData>
      <extdom:name>esempio.it</extdom:name>
      <extdom:targetStatus>
        <domain:status lang="en" s="pendingDelete"/>
        <rgp:rgpStatus lang="en" s="redemptionPeriod"/>
      </extdom:targetStatus>
    </extdom:chgStatusMsgData>
  </extension>
  <trID>
    <clTRID>ABC-12345</clTRID>
    <svTRID>841a4cc0-de99-4215-9d4f14c0089dbddd</svTRID>
  </trID>
  </response>
</epp>
```

Example 2

Response to a failed Domain Delete. The error is due to the fact that the domain name is in a status that prohibits a delete.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2304">
      <msg lang="en">Object status prohibits operation</msg>
      <extValue>
        <value>
          <extepp:reasonCode>9024</extepp:reasonCode>
        </value>
        <reason lang="en">Domain has status clientDeleteProhibited</reason>
      </extValue>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>0d495519-4bb0-4e6a-810e-858cee002b71</svTRID>
    </trID>
  </response>
</epp>
```

8.6.3.4 Effects of Domain Delete

If the Domain Delete command requested by the Registrar is executed successfully and passes the validation steps described in Section 0, the domain name is not immediately removed from the Database of the Registry.
The domain name goes into pendingDelete/redemptionPeriod for a maximum of 30 (thirty) days.
The domain names that are in pendingDelete/redemptionPeriod status are not delegates in the ccTLD .it zone and can be recovered through the Domain Update command with ext = restore (Section 0).
On the expiry of the pendingDelete/redemptionPeriod, the domain name goes into pendingDelete/pendingDelete status to be finally cancelled according to the Drop Time process described in Section 0.
Once deleted, the domain name is removed from the Database of the Registry and is immediately available online for a new registration.
The domain names that are in pendingDelete/pendingDelete status cannot be subjected to any transaction.

8.6.4 Restoring a deleted domain name

A domain name that has been deleted from the Registry Database after a successful delete can be restored through the Domain Update command with ext=restore.
The transaction allows the Registrar to restore a domain name in pendingDelete/redemptionPeriod or pendingDelete/clientTransferProhibited/redemptionPeriod, at the request of the Registrant or a competent Authority. The domain names which are in ok/*noRegistrar*, inactive/*dnsHold*/*noRegistrar* or inactive/*noRegistrar* status can be restored through the Domain Update command with ext=restore as well.
A domain name that has been deleted from the Registry Database after a successful delete can be restored also by a suspended Registrar.

8.6.4.1 Validation steps for the restoration of a domain name

The system verifies that the request for restoration of a domain name is compatible with:

- the constraints present in the XML Schema epp-1.0.xsd, eppcom-1.0.xsd, domain-1.0.xsd, RGP-1.0.xsd (Section 6.1);
- the following additional restrictions:
  - the domain name for which the transaction is requested must be present in the Registry Database;
  - the domain name for which the transaction is requested must be in one of the following statuses: pendingDelete/redemptionPeriod, pendingDelete/clientTransferProhibited/redemptionPeriod, ok/*noRegistrar*, inactive/*dnsHold*/*noRegistrar* or inactive/*noRegistrar*.

8.6.4.2 Example of Domain Update request with ext = restore

The process of restoring a domain name is implemented as an extension of a normal request for Domain Update, using rgp-1.0.xsd and the interaction modes between client and server as described in the RFC3915. The EPP server of the Registry implements the following grace periods:

- auto renew period: this is the grace period following the automatic renewal of a domain name on the expiry of its validity;
- redemption period: this is the grace period following the receipt of a client request for a Domain Delete. When the name is in this status it can be restored. The effect of the restore is to immediately

recover the domain name that is reported in the status prior to receipt of the request for Domain Delete. Unlike what is specified in the RFC mentioned above, the EPP server of the Registry does not need a report from the client to justify the request to restore a domain name previously subject to a Domain Delete command.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 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>example.it</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>ABC-12345</clTRID>
  </command>
</epp>
```

8.6.4.3 Examples of responses to a request for Domain Update with ext=restore

Example 1
Response to successful Domain Update with ext=restore:

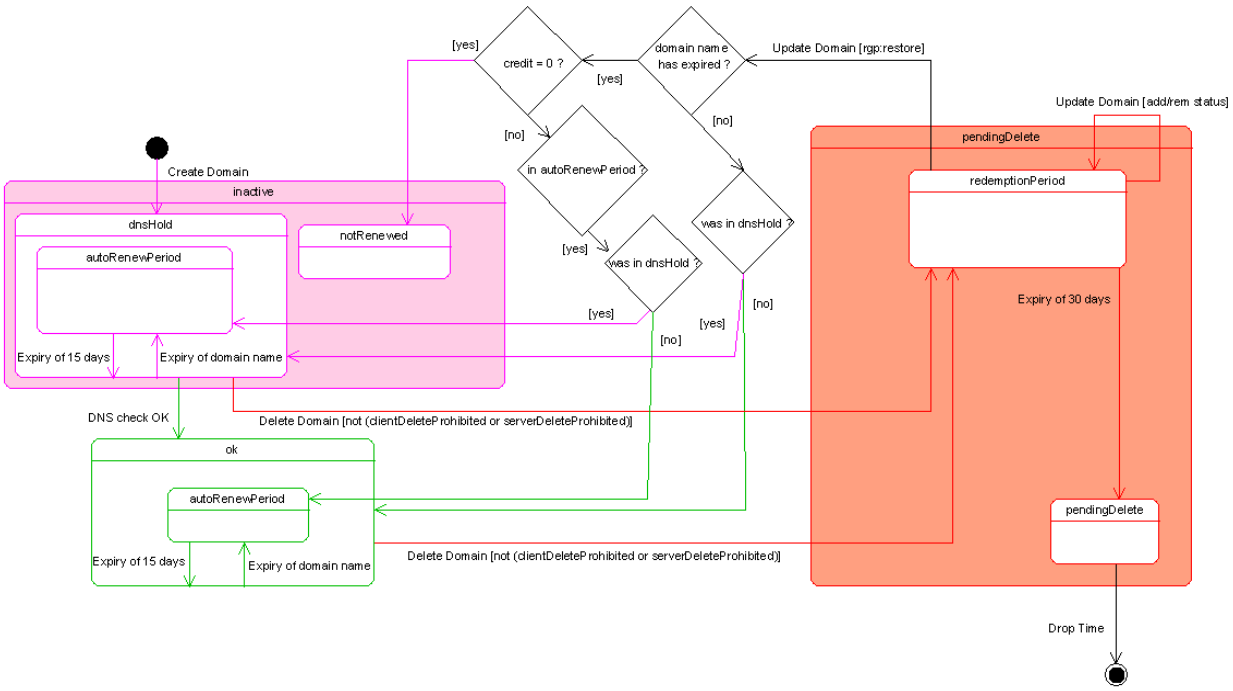
```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <extension>
      <extdom:chgStatusMsgData>
        <extdom:name>esempio.it</extdom:name>
        <extdom:targetStatus>
          <domain:status lang="en" s="ok"/>
        </extdom:targetStatus>
      </extdom:chgStatusMsgData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>08ef9111-a1ee-45ea-ac3d-2459f6cd6fec</svTRID>
    </trID>
  </response>
</epp>
```

Example 2
Response to failed Domain Update con ext=restore. The error is due to the fact that the domain name is a status that prohibits a restore.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2304">
      <msg lang="en">Object status prohibits operation</msg>
      <extValue>
        <value>
          <expepp:reasonCode>9055</extepp:reasonCode>
        </value>
        <reason lang="en">Domain has status ok</reason>
      </extValue>
    </result>
    <trID>
       <clTRID>ABC-12345</clTRID>
       <svTRID>0abb9699-a408-45ba-9241-fb8cbbb8f6e0</svTRID>
    </trID>
  </response>
</epp>
```

8.6.4.4 Effects of Domain Update with ext = restore

If the command Domain Update with ext = restore requested by the Registrar is successful and passes the validation steps described in Section 0, the domain name is immediately restored.
The process of restoring a domain name is charged to the Registrar and is immediately available for invoicing except for the domain names restoration from ok/*noRegistrar*, inactive/*dnsHold*/*noRegistrar* or inactive/*noRegistrar* status.
Following the receipt of Domain Update with ext = restore, the domain name is listed in the status preceding the request for cancellation.
The following diagram shows the various steps in the deletion and possible restoration of a domain name:



9 Asynchronous operations

Not all the operations allowed on domain names of ccTLD .it are synchronous and, hence, performed through the EPP server. Some operations are requested by the Registrar or executed autonomously by the Registry out-of-band.
Although this document aims to present all the operations that can be performed by the Registrars via the EPP server, it's worth taking a short look at the asynchronous operations just to take note of those statuses they introduce which can affect the execution of the synchronous operations.

9.1 Transfer of a large number of domain names through a Bulk Transfer

A Bulk Transfer permits, with one transaction, the transfer of a considerable number of domain names between two Registrars.
A Bulk Transfer can only be requested through an online procedure available on the RAIN-NG portal (https://rain-ng.nic.it) and by sending the Registry a written document signed by both the Registrars involved in the transaction.
Once the Bulk Transfer request has been accepted, an automatic procedure verifies that the domain names on the list are in a status compatible with the Bulk Transfer. The procedure checks that the domain names are contained in the list are in one of the following statuses: ok, ok/autoRenewPeriod, inactive/*dnsHold*, inactive/*dnsHold*/autoRenewPeriod, inactive/*notRenewed*, inactive/*dnsHold*/*challenged*, ok/*challenged*.
The procedure then places the domain names into pendingTransfer/*bulk* status. At the end, the Registry will restore the domain names that are in pendingTransfer/*bulk* to the status before the transaction.
If the Bulk Transfer ends successfully, the expiry date of the domain names is not updated, which therefore remains unchanged.

9.2 Revocation of a registered domain name

The Registry may revoke a domain name.
The domain names revoked go into inactive/*revoked* status and remain in such status for 30 (thirty) days. After this period, the Registry puts these domain names into pendingDelete/pendingDelete status to be finally removed from the DBAN according to the Drop Time process.
In the event that a domain name revoked is the subject of an opposition (and thus is also in *challenged* status), after the above mentioned 30 (thirty) days, the domain name will go, instead, from inactive/*revoked* to inactive/*toBeReassigned* status.

9.3 Change to toBeReassigned status

At the end of a challenge and/or reassignment procedure that entails the assignment of a domain name to the subject that activated the challenge, the Registry passes the domain name into inactive/*toBeReassigned* status. The domain name can be assigned, within thirty days following the domain name's move into inactive/*toBeReassigned* status, only by the subject that made the challenge. Domain names in this status are not active as they are no longer delegated in the zone of the ccTLD .it.

9.4 Renewal of a registered domain name

The validity period of one domain name is one year and is determined by the expiry date indicated in the expire field of the registration.
As the expiry is reached, the domain name is put in autoRenewPeriod, that is the status that identifies the 15 days following the expiry of the domain name.
During the 15 days provided for the auto renew period, the domain name can be subject to all the possible operations of maintenance foreseen by the registration system of the ccTLD .it.
As the expiry date is reached, the renewal fee is immediately charged to the current Registrar, if the domain name is in a status that permits it.
The invoicing of the operation, on the contrary, occurs after the 15 days period provided for the auto renew period. If during such period of time, the domain name is transferred to another Registrar or cancelled, the renewal fee is credited again to the Registrar and the operation is not invoiced. Otherwise, in the absence of one of the above operations of Registrar transfer or cancellation, the domain name is renewed by the Registry for the following 12 months.

9.5 Change to noRegistrar status

With the term noRegistrar the Registry identifies all those domain names managed by a Registrar who no longer has an active contract with the Registry. The Registry must inform the Registrant of each domain name managed by the Registrar that the contract has terminated.
There are three different statuses noRegistrar:

- ok*/noRegistrar* and inactive*/dnsHold/noRegistrar:* all the domain names of a Registrar who no longer has an active contract with the Registry and which have not yet expired. The domain names remain in ok/*noRegistrar* (or inactive/*dnsHold*/*noRegistrar*) status until they expire. Only the domain names that are in ok/*noRegistrar* status are delegated in the ccTLD .it zone;
- inactive*/noRegistrar*: all the domain names that have expired and for which the Registrar no longer has an active contract with the Registry, plus all those domain names for which a change of the Registrar was not successful beyond the expiry of the “grace period”. The domain names remain in inactive/*noRegistrar* status for a maximum period of 60 (sixty) days. After this period of time the

domain names go in pendingDelete/pendingDelete status. The domain names that are in inactive/*noRegistrar* status are not delegated in the zone of the ccTLD .it.

In both cases, the only operations allowed on the domain name are: the change of the Registrar, the change of the Registrar with the simultaneous change of the Registrant, the recovery by the current Registrar in case his contract with the Registry is active.

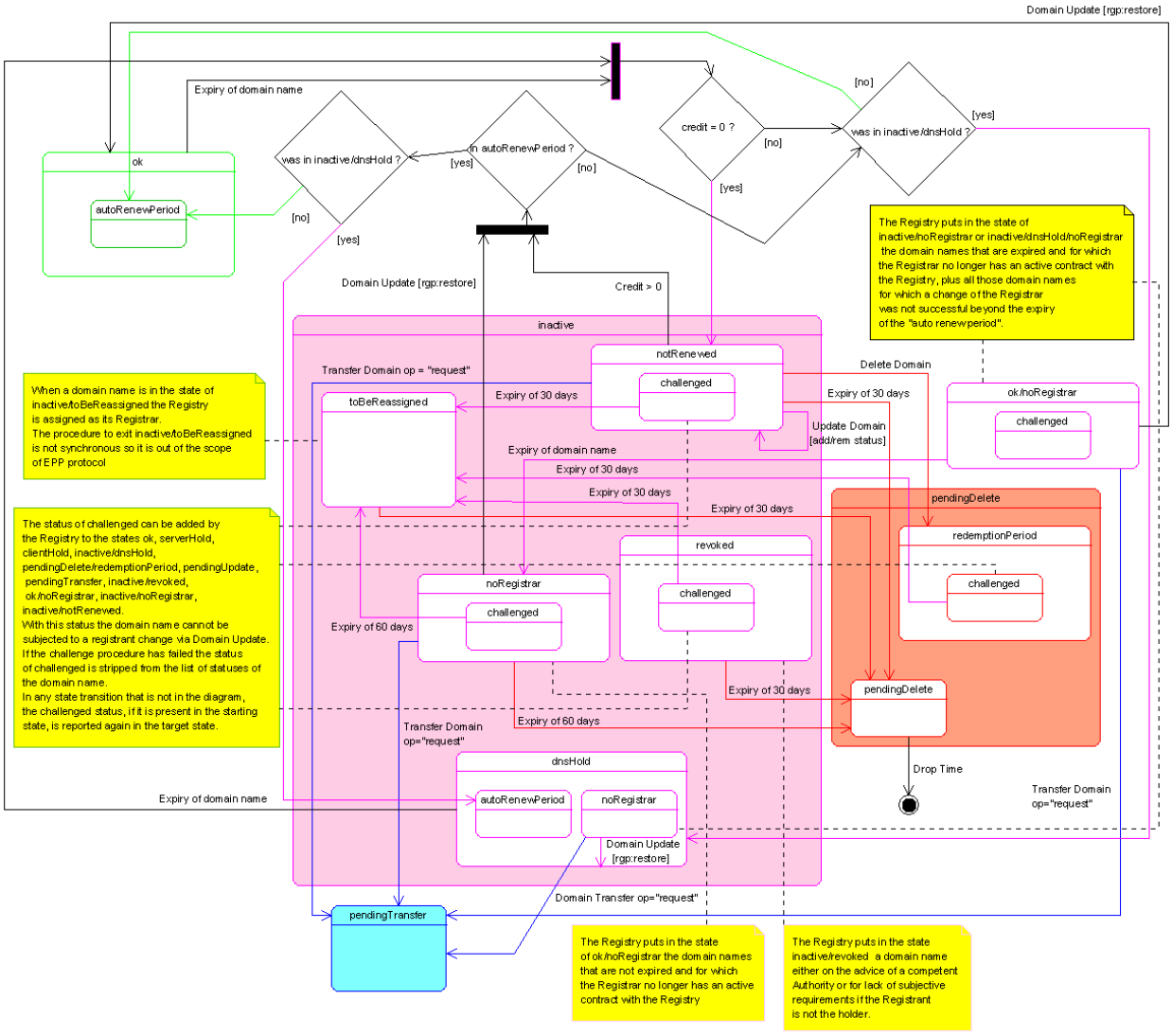
9.6 Change to notRenewed status

The status *notRenewed* identifies all those domain names that were not automatically renewed because of the Registrar's low credit. If, at the expiry of the period of validity of the domain name, the Registrar has not enough credit for the renewal, the domain name goes into inactive/*notRenewed* status.
A domain name can come out of inactive/*notRenewed* status, as a result of:

- sufficient credit being made available for the renewal. The domain name, in this case, goes into:
  - ok or inactive/*dnsHold*, if the auto renew period is exceeded;
  - ok/autoRenewPeriod or inactive/*dnsHold*/autoRenewPeriod, if additional credit was made available before the domain name exceeded the autoRenewPeriod.

  In both cases renewal will be charged, but only billed in the first case;
- expiry of the maximum period, 30 (thirty) days, in inactive/*notRenewed*. The domain name in this case goes into pendingDelete/pendingDelete status;
- request to change the Registrar (with or without the simultaneous change of the Registrant). The domain name in this case goes into pendingTransfer status;
- request for cancellation. The domain name in this case goes into pendingDelete/redemptionPeriod status.

The only operations allowed in this status are: change of the Registrar (with or without the simultaneous change of Registrant), change to the constraints placed by the Registrar on the domain name, and deletion.



10 Commands for querying the server

EPP provides clients with commands to query the server. There is a distinction between:

- commands that query the server about the status of objects in the Registry Database, i.e. Check, Info, and Transfer Query commands:
  - the Check command allows a client to determine whether an object or domain contact is in the Database Registry, and thus whether or not it is available for registration. This command can be sent by any Registrar;
  - the Info command allows a client to obtain information about contact or domain in the Database and to display the current value of the property of the object including the current status, or relations with other objects (e.g. what contacts are associated - for various roles - with a registered domain name). This command can only be sent by the Registrar that manages the object; and for domain objects only, also by the Registrar that holds the AuthInfo;
  - the Transfer Query command allows a client to see the status of a request to change the current

Registrar or the last change of the Registrar (to which an object has been submitted). This command is applicable only for domain objects because in the synchronous implementation of the Registry, contact objects cannot be transferred. This command can only be sent by the Registrar who currently manages the domain name, or by the Registrar who, in the case of a current request to change the Registrar, holds the AuthInfo (typically the Registrar that sent the request, i.e. the new Registrar);

- commands that query the server on the presence of messages in the client's polling queue, i.e. Poll Req and Poll Ack commands (Sections 0and 12.4).

10.1 Queries on contact objects

The format of the Contact Check command provides that the client may send in their request a list of contact IDs (up to a maximum number determined by the policies of the server and corresponding to the MAX_CHECK parameter described in Section 12.8. In its response the server reports, for each identifier in the request, whether it can be used or not in a registration.

10.1.1 Contact Check

10.1.1.1 Example of request to Contact Check

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
<command>
 <check>
  <contact:check xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
     <contact:id>mm001</contact:id>
     <contact:id>mb001</contact:id>
     <contact:id>cl001</contact:id>
     <contact:id>bb001</contact:id>
   </contact:check>
  </check>
  <clTRID>ABC-12345</clTRID>
 </command>
</epp>
```

10.1.1.2 Example of a response to a Contact Check request

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <contact:chkData>
        <contact:cd>
          <contact:id avail="false">MM001</contact:id>
       </contact:cd>
       <contact:cd>
```

```
        <contact:id avail="false">MB001</contact:id>
      </contact:cd>
      <contact:cd>
        <contact:id avail="true">CL001</contact:id>
      </contact:cd>
      <contact:cd>
        <contact:id avail="true">BB001</contact:id>
      </contact:cd>
    </contact:chkData>
  </resData>
  <trID>
    <clTRID>ABC-12345</clTRID>
    <svTRID>fa0d0881-cdba-41c1-8f73-d8e52f69b2cc</svTRID>
  </trID>
 </response>
</epp>
```

10.1.2 Contact Info

The command Contact Info requires that the client can send only one contact identifier for which to request information. The contact object, identified by the ID specified in the command must be present in the Registry Database and must be currently associated with the client who submits the request: if not, the server returns an error.
The server, in its response, returns the information of the object deriving from registration and change commands submitted by the client, including any extensions in addition to those assigned automatically by the system.
The following table shows additional contact object fields and their correspondence with the XML tags:

Field	Description	XML Tag	XML Tag Attribute	Cardinality	Length	Notes
ROID (Repository Object Identifier)	Contact unambiguous identifier in the Registry database	contact:roid				Assigned automatically by the system
Creation date	Contact first registration date and time	contact:crDate				Assigned automatically by the system
Current client ID	Current registrar	contact:clID				Assigned automatically by the system
Client ID who made the creation	Registrar who made the contact registration	contact:crID				Assigned automatically by the system
Client ID who made the modification	Registrar who modified the contact	contact:upID				Assigned automatically by the system and visible only if the contact has been modified
Update date	Last contact modification date and time	contact:upDate				Assigned automatically by the system

Status	Identifies the statuses that the contact is in	contact:status				The default value, assigned by the system at the registration of the contact in the Registry Database, is ok. When the contact is referenced in the domain name registration, it goes into ok/linked

10.1.2.1 Example of Contact Info request

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <info>
      <contact:info
       xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
       xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0 contact-1.0.xsd">
        <contact:id>MISSING001</contact:id>
      </contact:info>
    </info>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

10.1.2.2 Examples of responses for Contact Info request

Example 1
A response to a Contact Info request relating to a contract that has not yet been registered in the Registry Database:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2303">
      <msg lang="en">Object does not exist</msg>
      <value>
        <extepp:wrongValue>
          <extepp:element>id</extepp:element>
          <extepp:namespace>urn:ietf:params:xml:ns:contact-1.0</extepp:namespace>
```

```
          <extepp:value>MISSING001</extepp:value>
        </extepp:wrongValue></value>
        <extValue>
          <value>
            <extepp:reasonCode>9003</extepp:reasonCode>
          </value>
          <reason lang="en">Contact does not exist</reason>
        </extValue>
     </result>
     <trID>
        <clTRID>ABC-12345</clTRID>
        <svTRID>48b7d02d-eeee-4292-8822-4ae089f47be4</svTRID>
     </trID>
   </response>
</epp>
```

Example 2
Response to a Contact Info relating to a contact registered by a different Registrar from the one who submitted the request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2201">
      <msg lang="en">Authorization error</msg>
      <extValue>
        <value>
          <extepp:reasonCode>6001</extepp:reasonCode>
        </value>
        <reason lang="en">Lack of permissions to process command
        </reason>
     </extValue>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>f913c565-e954-4c74-a4c1-754397f5e171</svTRID>
    </trID>
  </response>
</epp>
```

Example 3
Response to a Contact Info regarding a “tech” contact:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
```

```
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <contact:infData>
        <contact:id>TECH001</contact:id>
        <contact:roid>ITNIC-8</contact:roid>
        <contact:status s="ok" lang="en"/>
        <contact:status s="linked" lang="en"/>
        <contact:postalInfo type="loc">
          <contact:name>Mirco Bartolini</contact:name>
          <contact:org>Demo Registrar Srl</contact:org>
            <contact:addr>
              <contact:street>via 4 Novembre,12</contact:street>
              <contact:city>Barga</contact:city>
              <contact:sp>LU</contact:sp>
              <contact:pc>55052</contact:pc>
              <contact:cc>IT</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice x="">+39.0583123456</contact:voice>
        <contact:fax x="">+39.058375124</contact:fax>
        <contact:email>mirco.bartolini@hotmail.it</contact:email>
        <contact:clID>DEMO-REGISTRAR</contact:clID>
        <contact:crID>DEMO-REGISTRAR</contact:crID>
        <contact:crDate>2013-02-19T14:46:35+01:00</contact:crDate>
        <contact:upID>DEMO-REGISTRAR</contact:upID>
        <contact:upDate>2013-03-21T12:35:51+01:00</contact:upDate>
      </contact:infData>
    </resData>
    <extension>
     <extcon:infData>
       <extcon:consentForPublishing>true</extcon:consentForPublishing>
     </extcon:infData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>bd336711-a0d5-443c-812b-cadb2d3431b3</svTRID>
    </trID>
  </response>
</epp>
```

Example 4
Response to a Contact Info regarding a “registrant” contact:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <contact:infData>
        <contact:id>MR0001</contact:id>
        <contact:roid>ITNIC-24</contact:roid>
        <contact:status s="ok" lang="en"/>
```

```
        <contact:status s="linked" lang="en"/>
        <contact:postalInfo type="loc">
          <contact:name>Mario Rossi</contact:name>
          <contact:org>NIC-IT Inc.</contact:org>
          <contact:addr>
            <contact:street>via Moruzzi,1</contact:street>
            <contact:city>Pisa</contact:city>
            <contact:sp>PI</contact:sp>
            <contact:pc>56124</contact:pc>
            <contact:cc>IT</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice x="2111">+39.050315</contact:voice>
        <contact:fax x="">+39.0503152593</contact:fax>
        <contact:email>mario.rossi@esempio.it</contact:email>
        <contact:clID>DEMO-REGISTRAR</contact:clID>
        <contact:crID>DEMO-REGISTRAR</contact:crID>
        <contact:crDate>2013-02-27T11:28:32+01:00</contact:crDate>
        <contact:upID>DEMO-REGISTRAR</contact:upID>
        <contact:upDate>2013-02-29T12:28:22+01:00</contact:upDate>
      </contact:infData>
    </resData>
    <extension>
      <extcon:infData>
        <extcon:consentForPublishing>true</extcon:consentForPublishing>
        <extcon:registrant>
          <extcon:nationalityCode>IT</extcon:nationalityCode>
          <extcon:entityType>1</extcon:entityType>
          <extcon:regCode>RSSMRA64C14G702Q</extcon:regCode>
        </extcon:registrant>
      </extcon:infData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>fccc34cc-b26d-4fa8-be0d-af08eb33c549</svTRID>
    </trID>
  </response>
</epp>
```

10.2 Queries on domain objects

10.2.1 Domain Check

The command Domain Check allows the client to send a list of domain names (up to a maximum number determined by the policies of the server and corresponding to the MAX_CHECK parameter described in Section 12.8. The server, for each domain name listed in the request, indicates whether it can be inserted or not in a registration request and, if not, the reason (e.g. it is already present in the Registry Database or it is reserved).

10.2.1.1 Example of a Domain Check request

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <check>
      <domain:check
```

```
       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>example1.it</domain:name>
        <domain:name>example2.it</domain:name>
        <domain:name>example3.it</domain:name>
      </domain:check>
    </check>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

10.2.1.2 Example of a response to a Domain Check request

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:chkData>
        <domain:cd>
          <domain:name avail="false">esempio1.it</domain:name>
          <domain:reason lang="en">Domain is registered</domain:reason>
        </domain:cd>
        <domain:cd>
          <domain:name avail="false">esempio2.it</domain:name>
          <domain:reason lang="en">Domain is registered</domain:reason>
        </domain:cd>
        <domain:cd>
          <domain:name avail="true">esempio3.it</domain:name>
        </domain:cd>
      </domain:chkData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>20fd2709-5ed6-4091-89a3-826f703e10b9</svTRID>
    </trID>
  </response>
</epp>
```

10.2.2 Domain Info

The Domain Info command allows the client to request information for a single domain name. The object domain associated with the domain name specified must be present in the Registry Database, and the client who submits the application must be the one currently associated with the domain name or another client in possession of the AuthInfo: otherwise the server will send a response error.
The command in question can be sent to the server of the Registry with the “InfContacts” extension. In this case the command must necessarily contain, not only the domain name, but also the AuthInfo currently associated with the domain name, otherwise the server will send an error message.
The use of this extension allows the client to view the information contained in the Database of the Registry for the domain name for which the command was submitted, as well as the data associated with

the contacts referenced in the domain name.
The “infContacts” extension can take on the following values:

- *registrant*: allows the client to obtain information about the Registrant;
- *admin*: allows the client to obtain information about the administrative contact;
- *tech*: allows the client to obtain information about the technical contacts;
- *all*: allows the client to obtain information about all the contacts referenced in the domain name, that is, the “registrant”, “admin” and “tech” contact types.

Paragraphs 0 and 0 contain, respectively, some examples of Domain Info requests without the use of the “infContacts” extension and some examples of response sent by the server to that type of request.
Paragraphs 0 and 0 contain, respectively, some examples of Domain Info requests containing the extension “infContacts” and the responses of the server.

The server, in its response, returns the information of the object deriving from registration and change commands submitted by the client, including any extensions in addition to those assigned automatically by the system.
The response can contain four extensions:

- *extdom:infData* contains the NIC-It statuses, which along with the standards described in the domain-1.0.xsd schema, describe the status of the domain in question. The statuses are described in the extdom-2.0.xsd (tag: extdom:ownStatus) schema.
- *rgp:infData* contains the statuses belonging to the extension of the protocol for the grace period, which along with the standards described in the domain-1.0.xsd schema, describe the status of the domain in question. The statuses are described in the rgp-1.0.xsd (tag rgp:rgpStatus) schema.
- *extdom:infNsToValidateData* contains the most recent DNS configuration of the verified domain:
  - if the domain is in inactive/*dnsHold*, since there is no configuration present that has been verified as correct, the response contains only this information with regard to the name servers associated with the domain.
  - if the domain is in pendingUpdate, the response contains both the most recent configuration that has been verified as being correct (in the domain:ns section) and the most recently verified configuration;
- *extdom:infContactsData* contains the information about the tech, admin e registrant contacts (the same one of the Contact Info response) according to value of op attribute selected in the related request.

The following table shows additional fields of the domain object and the related correspondence with the XML tags:

Field	Description	XML Tag	XML Tag Attribute	Cardinality	Notes
ROID (Repository Object Identifier)	Domain name unambiguous identifier in the Registry database	domain:roid			Assigned automatically by the system
Creation date	Domain name first registration date and time	domain:crDate			Assigned automatically by the system
Current client ID	Current Registrar	domain:clID			Assigned automatically by the system
Client ID who made the creation	Registrar who made the domain name registration	domain:crID			Assigned automatically by the system
Client ID who made the modification	Registrar who modified the domain name registration	domain:upID			Assigned automatically by the system and visible only if the domain name has been modified
Expiry date	Domain name expiry date and time	domain:exDate			Automatically updated by the system at the maintenance period expiration date (a year)
Update date	Domain name last modification date and time	domain:upDate			Assigned automatically by the system
Transfer date	Registrar modification operation end date and time	domain:trDate			Assigned automatically by the system and visible only if the Registrar of the domain name has been modified
Status	Identifies the statuses that the contact is in	domain:status			At the registration of the domain name in the Registry Database, the domain name is put in inactive/*dnsHold* status

Section 0contains examples of Domain Info requests on a digitally signed domain name.

10.2.2.1 Examples of a Domain Info request without extension "infContacts"

Example 1
Domain Info request submitted by the Registrar of a domain (and thus without AuthInfo)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <info>
      <domain:info
       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 hosts="all">example.it</domain:name>
      </domain:info>
    </info>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example 2
Domain Info request submitted by a Registrar who is different from the one in the domain name registration. In this case the Registrar must insert the value of the AuthInfo currently associated with the domain name:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <info>
      <domain:info
       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 hosts="all">example.it</domain:name>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:info>
    </info>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

10.2.2.2 Examples of responses to a Domain Info request without extension "infContacts"

Example 1
Response to a Domain Info request for a domain name not yet registered in the Registry Database:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2202">
      <msg lang="en">Invalid authorization information</msg>
      <value>
        <extepp:wrongValue>
          <extepp:element>name</extepp:element>
```

```
          <extepp:namespace>urn:ietf:params:xml:ns:domain-1.0</extepp:namespace>
          <extepp:value>missing.it</extepp:value>
        </extepp:wrongValue>
      </value>
      <extValue>
        <value>
          <extepp:reasonCode>9085</extepp:reasonCode>
        </value>
        <reason lang="en">Invalid domain authorization information or domain does
not exist</reason>
      </extValue>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>88b45952-b85d-4363-98f3-7917ebc06090</svTRID>
    </trID>
  </response>
</epp>
```

Example 2

Response to a Domain Info request submitted by a Registrar who maintains the domain name or by a different Registrar who holds the AuthInfo:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData>
        <domain:name>esempio.it</domain:name>
        <domain:roid>ITNIC-666</domain:roid>
        <domain:status s="ok" lang="en"/>
        <domain:registrant>REG001</domain:registrant>
        <domain:contact type="admin">REG001</domain:contact>
        <domain:contact type="tech">TECH001</domain:contact>
        <domain:ns>
          <domain:hostAttr>
            <domain:hostName>ns1.esempio.it</domain:hostName>
            <domain:hostAddr ip="v4">192.0.2.1</domain:hostAddr>
          </domain:hostAttr>
          <domain:hostAttr>
           <domain:hostName>ns2.esempio.it</domain:hostName>
           <domain:hostAddr ip="v4">192.0.2.2</domain:hostAddr>
          </domain:hostAttr>
        </domain:ns>
        <domain:host>ns1.esempio.it</domain:host>
        <domain:host>ns2.esempio.it</domain:host>
        <domain:clID>DEMO-REGISTRAR</domain:clID>
        <domain:crID>DEMO-REGISTRAR</domain:crID>
        <domain:crDate>2013-02-21T15:18:12+01:00</domain:crDate>
        <domain:upID>DEMO-REGISTRAR</domain:upID>
        <domain:upDate>2013-03-12T16:21:08+01:00</domain:upDate>
```

```
        <domain:exDate>2014-02-21T23:59:59+01:00</domain:exDate>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:infData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>fd505964-ddac-4b21-b2b6-54423be597d3</svTRID>
    </trID>
  </response>
</epp>
```

Example 3

Response to a Domain Info request submitted by a Registrar at the termination of a change of the Registrar. Note the presence of duplicated contacts and the date of transfer.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData>
        <domain:name>esempio.it</domain:name>
        <domain:roid>ITNIC-666</domain:roid>
        <domain:status s="ok" lang="en"/>
        <domain:registrant>DUP679000001</domain:registrant>
        <domain:contact type="admin">DUP679000001</domain:contact>
        <domain:contact type="tech">DUP142000001</domain:contact>
        <domain:ns>
          <domain:hostAttr>
            <domain:hostName>ns1.esempio.it</domain:hostName>
            <domain:hostAddr ip="v4">192.0.2.1</domain:hostAddr>
          </domain:hostAttr>
          <domain:hostAttr>
            <domain:hostName>ns2.esempio.it</domain:hostName>
            <domain:hostAddr ip="v4">192.0.2.2</domain:hostAddr>
          </domain:hostAttr>
        </domain:ns>
        <domain:host>ns1.esempio.it</domain:host>
        <domain:host>ns2.esempio.it</domain:host>
        <domain:clID>NEW-REGISTRAR</domain:clID>
        <domain:crID>DEMO-REGISTRAR</domain:crID>
        <domain:crDate>2013-02-21T15:18:12+01:00</domain:crDate>
        <domain:upID>NEW-REGISTRAR</domain:upID>
        <domain:upDate>2013-02-25T07:54:50+01:00</domain:upDate>
        <domain:exDate>2014-02-25T23:59:59+01:00</domain:exDate>
        <domain:trDate>2014-02-25T23:59:59+01:00</domain:trDate>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:infData>
```

```
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>b70fa9c1-23cf-4efb-836a-bed359802d95</svTRID>
    </trID>
  </response>
</epp>
```

Example 4
Response to a Domain Info request on a domain name that is in inactive/*dnsHold* status.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData>
        <domain:name>esempio.it</domain:name>
        <domain:roid>ITNIC-40</domain:roid>
        <domain:status s="inactive" lang="en"/>
        <domain:registrant>MM001</domain:registrant>
        <domain:contact type="admin">MM001</domain:contact>
        <domain:contact type="tech">MB001</domain:contact>
        <domain:clID>DEMO-REGISTRAR</domain:clID>
        <domain:crID>DEMO-REGISTRAR</domain:crID>
        <domain:crDate>2013-07-04T13:03:15+02:00</domain:crDate>
        <domain:upID>DEMO-REGISTRAR</domain:upID>
        <domain:upDate>2013-07-07T14:00:10+02:00</domain:upDate>
        <domain:exDate>2014-07-04T23:59:59+02:00</domain:exDate>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:infData>
    </resData>
    <extension>
      <extdom:infData>
        <extdom:ownStatus lang="en" s="dnsHold"/>
      </extdom:infData>
      <extdom:infNsToValidateData>
        <extdom:nsToValidate>
          <domain:hostAttr>
            <domain:hostName>ns1.esempio.it</domain:hostName>
          </domain:hostAttr>
          <domain:hostAttr>
            <domain:hostName>ns2.esempio.it</domain:hostName>
          </domain:hostAttr>
          <domain:hostAttr>
            <domain:hostName>dns.dominio.net</domain:hostName>
          </domain:hostAttr>
          <domain:hostAttr>
            <domain:hostName>dns.dominio.com</domain:hostName>
          </domain:hostAttr>
        </extdom:nsToValidate>
```

```
      </extdom:infNsToValidateData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>38ac1a72-c80b-4b0d-8ba1-654770a1e23c</svTRID>
    </trID>
  </response>
</epp>
```

10.2.2.3 Examples of Domain Info request with extension "infContacts"

Domain Info request submitted by a Registrar to visualize the data relating to the domain name esempio.it and associated Registrant. In this case the Registrar must insert the AuthInfo value currently associated with the domain name:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <info>
      <domain:info
       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 hosts="all">esempio.it</domain:name>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:info>
    </info>
    <extension>
     <extdom:infContacts op="registrant"
       xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
       xsi:schemaLocation="http://www.nic.it/ITNIC-EPP/extdom-2.0
       extdom-2.0.xsd"/>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

In order to visualize the details of all the contacts referenced in the registration of the domain name it is enough to add the value "all" to the extension infContacts, as indicated below:

```
<extension>
  <extdom:infContacts op="all"
   xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
   xsi:schemaLocation="http://www.nic.it/ITNIC-EPP/extdom-2.0
   extdom-2.0.xsd"/>
</extension>
```

Similarly, to visualize the technical or administrative contact, the Registrar must send a Domain Info request where the value of the extension infContacts is "tech" or "admin".

10.2.2.4 Examples of response to a Domain Info request with extension "infContacts"

Example 1
Response to a Domain Info request, relating to the domain name esempio.it, submitted by the Registrar

with the extension infContacts op="registrant". In this case the server, in addition to the information relating to the domain name, also returns the data associated with the “registrant” contact type referenced in it:

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0" xmlns:extcon="http://www.nic.it/ITNIC-
 EPP/extcon-1.0" xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData>
        <domain:name>esempio.it</domain:name>
        <domain:roid>ITNIC-162761</domain:roid>
        <domain:status lang="en" s="ok"/>
        <domain:registrant>MR0001</domain:registrant>
        <domain:contact type="admin">MR0001</domain:contact>
        <domain:contact type="tech">TECH001</domain:contact>
        <domain:contact type="tech">TECH002</domain:contact>
        <domain:ns>
          <domain:hostAttr>
            <domain:hostName>ns1.esempio.it</domain:hostName>
            <domain:hostAddr ip="v4">193.205.245.6</domain:hostAddr>
          </domain:hostAttr>
          <domain:hostAttr>
            <domain:hostName>ns2.esempio.it</domain:hostName>
            <domain:hostAddr ip="v4">193.205.245.7</domain:hostAddr>
          </domain:hostAttr>
        </domain:ns>
        <domain:host>ns1.esempio.it</domain:host>
        <domain:host>ns2.esempio.it</domain:host>
        <domain:clID>DEMO-REG</domain:clID>
        <domain:crID>DEMO-REG</domain:crID>
        <domain:crDate>2013-01-24T16:41:53.000+01:00</domain:crDate>
        <domain:exDate>2014-01-24T16:41:53.000+01:00</domain:exDate>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:infData>
    </resData>
    <extension>
      <extdom:infContactsData>
        <extdom:registrant>
          <extdom:infContact>
            <contact:id>MR0001</contact:id>
            <contact:roid>ITNIC-326982</contact:roid>
            <contact:status lang="en" s="ok"/>
            <contact:status lang="en" s="linked"/>
            <contact:postalInfo type="loc">
              <contact:name>Mario Rossi</contact:name>
              <contact:org>Mario Rossi</contact:org>
              <contact:addr>
                <contact:street>Via Moruzzi, 1</contact:street>
                <contact:city>Pisa</contact:city>
                <contact:sp>PI</contact:sp>
                <contact:pc>56100</contact:pc>
                <contact:cc>IT</contact:cc>
```

```
            </contact:addr>
          </contact:postalInfo>
          <contact:voice x="">+39.050315111</contact:voice>
          <contact:fax x="">+39.050315111</contact:fax>
          <contact:email>mario.rossi@esempio.it</contact:email>
          <contact:clID>DEMO-REG</contact:clID>
          <contact:crID>DEMO-REG</contact:crID>
          <contact:crDate>2013-01-24T16:41:53.000+01:00</contact:crDate>
        </extdom:infContact>
        <extdom:extInfo>
          <extcon:consentForPublishing>true</extcon:consentForPublishing>
          <extcon:registrant>
            <extcon:nationalityCode>IT</extcon:nationalityCode>
            <extcon:entityType>1</extcon:entityType>
            <extcon:regCode>RSSMRA64C14G702Q</extcon:regCode>
          </extcon:registrant>
        </extdom:extInfo>
      </extdom:registrant>
    </extdom:infContactsData>
  </extension>
  <trID>
    <clTRID>ABC-12345</clTRID>
    <svTRID>fc205304-74dc-4dbe-89ff-411f12adca0a</svTRID>
  </trID>
 </response>
</epp>
```

Example 2

Response to a Domain Info request, relating to the domain name esempio.it, submitted by the Registrar with extension infContacts op="tech". In this case the server, in addition to the information relating to the domain name, also returns the data associated with the technical contacts referenced in it:

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0" xmlns:extcon="http://www.nic.it/ITNIC-
 EPP/extcon-1.0" xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData>
        <domain:name>esempio.it</domain:name>
        <domain:roid>ITNIC-162761</domain:roid>
        <domain:status lang="en" s="ok"/>
        <domain:registrant>MR0001</domain:registrant>
        <domain:contact type="admin">MR0001</domain:contact>
        <domain:contact type="tech">TECH001</domain:contact>
        <domain:contact type="tech">TECH002</domain:contact>
        <domain:ns>
          <domain:hostAttr>
            <domain:hostName>ns1.esempio.it</domain:hostName>
           <domain:hostAddr ip="v4">193.205.245.6</domain:hostAddr>
          </domain:hostAttr>
          <domain:hostAttr>
            <domain:hostName>ns2.esempio.it</domain:hostName>
           <domain:hostAddr ip="v4">193.205.245.7</domain:hostAddr>
          </domain:hostAttr>
```

```
    </domain:ns>
    <domain:host>ns1.esempio.it</domain:host>
    <domain:host>ns2.esempio.it</domain:host>
    <domain:clID>DEMO-REG</domain:clID>
    <domain:crID>DEMO-REG</domain:crID>
    <domain:crDate>2013-01-24T16:41:53.000+01:00</domain:crDate>
    <domain:exDate>2014-01-24T16:41:53.000+01:00</domain:exDate>
    <domain:authInfo>
      <domain:pw>22fooBAR</domain:pw>
    </domain:authInfo>
  </domain:infData>
</resData>
<extension>
  <extdom:infContactsData>
    <extdom:contact type="tech">
      <extdom:infContact>
        <contact:id>TECH001</contact:id>
        <contact:roid>ITNIC-326980</contact:roid>
        <contact:status lang="en" s="ok"/>
        <contact:status lang="en" s="linked"/>
        <contact:postalInfo type="loc">
          <contact:name>Mirco Bartolini</contact:name>
          <contact:org>Demo Registrar Srl</contact:org>
          <contact:addr>
            <contact:street>via 4 Novembre, 12</contact:street>
            <contact:city>Barga</contact:city>
            <contact:sp>LU</contact:sp>
            <contact:pc>55052</contact:pc>
            <contact:cc>IT</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice x="">+39.0583123456</contact:voice>
        <contact:fax x="">+39.058375124</contact:fax>
        <contact:email>mirco.bartolini@demoreg.it</contact:email>
        <contact:clID>DEMO-REG</contact:clID>
        <contact:crID>DEMO-REG</contact:crID>
      <contact:crDate>2013-01-24T16:41:53.000+01:00</contact:crDate>
      </extdom:infContact>
      <extdom:extInfo>
        <extcon:consentForPublishing>true</extcon:consentForPublishing>
      </extdom:extInfo>
    </extdom:contact>
    <extdom:contact type="tech">
      <extdom:infContact>
        <contact:id>TECH002</contact:id>
        <contact:roid>ITNIC-326982</contact:roid>
        <contact:status lang="en" s="ok"/>
        <contact:status lang="en" s="linked"/>
        <contact:postalInfo type="loc">
          <contact:name>Andrea Bianchi</contact:name>
          <contact:org>Demo Registrar Srl</contact:org>
          <contact:addr>
            <contact:street>via 4 Novembre, 12</contact:street>
            <contact:city>Barga</contact:city>
            <contact:sp>LU</contact:sp>
            <contact:pc>55052</contact:pc>
            <contact:cc>IT</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice x="">+39.0583123458</contact:voice>
        <contact:fax x="">+39.058375124</contact:fax>
```

```
            <contact:email>andrea.bianchi@demoreg.it</contact:email>
            <contact:clID>DEMO-REG</contact:clID>
            <contact:crID>DEMO-REG</contact:crID>
            <contact:crDate>2013-01-24T16:41:53.000+01:00</contact:crDate>
          </extdom:infContact>
          <extdom:extInfo>
            <extcon:consentForPublishing>true</extcon:consentForPublishing>
          </extdom:extInfo>
        </extdom:contact>
      </extdom:infContactsData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>c33f38aa-48ce-43ad-a86d-775b3d239b6c</svTRID>
    </trID>
  </response>
</epp>
```

Example 3

Response to a Domain Info request, relating to the domain name esempio.it, submitted by the Registrar with the extension infContacts op="all". In this case the server, in addition to the information relating to the domain name, also returns the data associated with all the contacts referenced in it (registrant, admin e tech):

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0" xmlns:extcon="http://www.nic.it/ITNIC-
 EPP/extcon-1.0" xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData>
        <domain:name>esempio.it</domain:name>
        <domain:roid>ITNIC-162761</domain:roid>
        <domain:status lang="en" s="ok"/>
        <domain:registrant>MR0001</domain:registrant>
        <domain:contact type="admin">MR0001</domain:contact>
        <domain:contact type="tech">TECH001</domain:contact>
        <domain:contact type="tech">TECH002</domain:contact>
        <domain:ns>
          <domain:hostAttr>
            <domain:hostName>ns1.esempio.it</domain:hostName>
            <domain:hostAddr ip="v4">193.205.245.6</domain:hostAddr>
          </domain:hostAttr>
          <domain:hostAttr>
            <domain:hostName>ns2.esempio.it</domain:hostName>
            <domain:hostAddr ip="v4">193.205.245.7</domain:hostAddr>
          </domain:hostAttr>
        </domain:ns>
        <domain:host>ns1.esempio.it</domain:host>
        <domain:host>ns2.esempio.it</domain:host>
        <domain:clID>DEMO-REG</domain:clID>
        <domain:crID>DEMO-REG</domain:crID>
        <domain:crDate>2013-01-24T16:41:53.000+01:00</domain:crDate>
        <domain:exDate>2014-01-24T16:41:53.000+01:00</domain:exDate>
        <domain:authInfo>
```

```
      <domain:pw>22fooBAR</domain:pw>
    </domain:authInfo>
  </domain:infData>
</resData>
<extension>
  <extdom:infContactsData>
    <extdom:registrant>
      <extdom:infContact>
        <contact:id>MR0001</contact:id>
        <contact:roid>ITNIC-326982</contact:roid>
        <contact:status lang="en" s="ok"/>
        <contact:status lang="en" s="linked"/>
        <contact:postalInfo type="loc">
          <contact:name>Mario Rossi</contact:name>
          <contact:org>Mario Rossi</contact:org>
          <contact:addr>
            <contact:street>Via Moruzzi, 1</contact:street>
            <contact:city>Pisa</contact:city>
            <contact:sp>PI</contact:sp>
            <contact:pc>56100</contact:pc>
            <contact:cc>IT</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice x="">+39.050315111</contact:voice>
        <contact:fax x="">+39.050315111</contact:fax>
        <contact:email>mario.rossi@esempio.it</contact:email>
        <contact:clID>DEMO-REG</contact:clID>
        <contact:crID>DEMO-REG</contact:crID>
        <contact:crDate>2013-01-24T16:41:53.000+01:00</contact:crDate>
      </extdom:infContact>
      <extdom:extInfo>
        <extcon:consentForPublishing>true</extcon:consentForPublishing>
        <extcon:registrant>
          <extcon:nationalityCode>IT</extcon:nationalityCode>
          <extcon:entityType>1</extcon:entityType>
          <extcon:regCode>RSSMRA64C14G702Q</extcon:regCode>
        </extcon:registrant>
      </extdom:extInfo>
    </extdom:registrant>
    <extdom:contact type="admin">
      <extdom:infContact>
        <contact:id>MR0001</contact:id>
        <contact:roid>ITNIC-326982</contact:roid>
        <contact:status lang="en" s="ok"/>
        <contact:status lang="en" s="linked"/>
        <contact:postalInfo type="loc">
          <contact:name>Mario Rossi</contact:name>
          <contact:org>Mario Rossi</contact:org>
          <contact:addr>
            <contact:street>Via Moruzzi, 1</contact:street>
            <contact:city>Pisa</contact:city>
            <contact:sp>PI</contact:sp>
            <contact:pc>56100</contact:pc>
            <contact:cc>IT</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice x="">+39.050315111</contact:voice>
        <contact:fax x="">+39.050315111</contact:fax>
        <contact:email>mario.rossi@esempio.it</contact:email>
        <contact:clID>DEMO-REG</contact:clID>
        <contact:crID>DEMO-REG</contact:crID>
```

```
    <contact:crDate>2013-01-24T16:41:53.000+01:00</contact:crDate>
  </extdom:infContact>
  <extdom:extInfo>
    <extcon:consentForPublishing>true</extcon:consentForPublishing>
    <extcon:registrant>
      <extcon:nationalityCode>IT</extcon:nationalityCode>
      <extcon:entityType>1</extcon:entityType>
      <extcon:regCode>RSSMRA64C14G702Q</extcon:regCode>
    </extcon:registrant>
  </extdom:extInfo>
</extdom:contact>
<extdom:contact type="tech">
  <extdom:infContact>
    <contact:id>TECH001</contact:id>
    <contact:roid>ITNIC-326980</contact:roid>
    <contact:status lang="en" s="ok"/>
    <contact:status lang="en" s="linked"/>
    <contact:postalInfo type="loc">
      <contact:name>Mirco Bartolini</contact:name>
      <contact:org>Demo Registrar Srl</contact:org>
      <contact:addr>
        <contact:street>via 4 Novembre, 12</contact:street>
        <contact:city>Barga</contact:city>
        <contact:sp>LU</contact:sp>
        <contact:pc>55052</contact:pc>
        <contact:cc>IT</contact:cc>
      </contact:addr>
    </contact:postalInfo>
    <contact:voice x="">+39.0583123456</contact:voice>
    <contact:fax x="">+39.058375124</contact:fax>
    <contact:email>mirco.bartolini@demoreg.it</contact:email>
    <contact:clID>DEMO-REG</contact:clID>
    <contact:crID>DEMO-REG</contact:crID>
    <contact:crDate>2013-01-24T16:41:53.000+01:00</contact:crDate>
  </extdom:infContact>
  <extdom:extInfo>
    <extcon:consentForPublishing>true</extcon:consentForPublishing>
  </extdom:extInfo>
</extdom:contact>
<extdom:contact type="tech">
  <extdom:infContact>
    <contact:id>TECH002</contact:id>
    <contact:roid>ITNIC-326982</contact:roid>
    <contact:status lang="en" s="ok"/>
    <contact:status lang="en" s="linked"/>
    <contact:postalInfo type="loc">
      <contact:name>Andrea Bianchi</contact:name>
      <contact:org>Demo Registrar Srl</contact:org>
      <contact:addr>
        <contact:street>via 4 Novembre, 12</contact:street>
        <contact:city>Barga</contact:city>
        <contact:sp>LU</contact:sp>
        <contact:pc>55052</contact:pc>
        <contact:cc>IT</contact:cc>
      </contact:addr>
    </contact:postalInfo>
    <contact:voice x="">+39.0583123458</contact:voice>
    <contact:fax x="">+39.058375124</contact:fax>
    <contact:email>andrea.bianchi@demoreg.it</contact:email>
    <contact:clID>DEMO-REG</contact:clID>
    <contact:crID>DEMO-REG</contact:crID>
```

```
            <contact:crDate>2013-01-24T16:41:53.000+01:00</contact:crDate>
          </extdom:infContact>
          <extdom:extInfo>
            <extcon:consentForPublishing>true</extcon:consentForPublishing>
          </extdom:extInfo>
        </extdom:contact>
      </extdom:infContactsData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>02cd9dbe-82d1-475f-ab66-4b49c5223a1c</svTRID>
    </trID>
  </response>
</epp>
```

10.2.3 Domain Transfer Query

The Domain Transfer Query command enables the client to request information on a single domain name for which a request to change the Registrar is under way or has been completed. The domain object associated with the domain name specified in the request must be present in the Registry Database and the client that sends the request must be the one that currently manages the domain name or the new Registrar who must take over and that is in possession of the AuthInfo. Otherwise the server sends a response error. In its response, the server reports the information of the object relative to the transfer, including any extensions that have been automatically assigned by the system.
For as long as the domain name is in pendingTransfer, the command can be submitted without AuthInfo both by the Registrar requesting the transfer (i.e. the new Registrar) and by the one currently associated with the domain name (i.e. the previous Registrar). Once the transfer has been completed, successfully or unsuccessfully, only the Registrar who holds the domain name can submit the Transfer Query command without the AuthInfo.
For a change of the Registrar with the simultaneous change in the Registrant which is still in progress, the server's response will include the identity of the new Registrant and the new AuthInfo. For the sake of privacy, this information is provided only if the request for Transfer Query came from the same Registrar as sent the request for transfer.
The information that the server will include in the response is shown in the following table:

Field	Description	XML Tag	XML Tag Attribute	Cardinality	Notes
Domain name	Unique ID of the domain name in the Registry's Database	domain:name			
Transfer status	Transfer substatus to describe the transfer situation	domain:trStatus			Assigned automatically by the system. Possible values: pending clientApproved clientRejected clientCancelled serverApproved
Client ID of the request	ID of Registrar that submitted the transfer request	domain:reID			Assigned automatically by the system
Request date	Date when the transfer request was submitted	domain:reDate			Assigned automatically by the system
Request acceptation Client ID	ID of Registrar to whom the possible transfer validation is requested	domain:acID			Assigned automatically by the system
Request acceptation date	Transfer must be accepted before this date	domain:acDate			Assigned automatically by the system. Calculated starting from the request date, adding the maximum period in which the domain can stay in the pendingTransfer
New Registrant data		extdom:transferTrade			Extension containing information to change the Registrant with the simultaneous change in Registrar
New Registrant	New Registrant identifier	extdom:newRegistrant			
New domain name AuthInfo	Identifies the new authorization password for the domain name specific operation request	extdom:newAuthInfo			

10.2.3.1 Example of Domain Transfer Query request

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <transfer op="query">
      <domain:transfer
       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>example.it</domain:name>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:transfer>
    </transfer>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

10.2.3.2 Examples of responses to a Domain Transfer Query request

Example 1
Response to a Domain Transfer Query request on a domain name that is not in pendingTransfer:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2301">
      <msg lang="en">Object not pending transfer</msg>
      <value>
        <extepp:wrongValue>
          <extepp:element>name</extepp:element>
          <extepp:namespace>urn:ietf:params:xml:ns:domain-1.0</extepp:namespace>
          <extepp:value>example.it</extepp:value>
        </extepp:wrongValue>
      </value>
      <extValue>
        <value>
          <extepp:reasonCode>9054</extepp:reasonCode>
        </value>
        <reason lang="en">Domain transfer not pending</reason>
      </extValue>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>10e0d6f8-5987-4784-8f76-baf8fb479d0f</svTRID>
    </trID>
  </response>
</epp>
```

Example 2

Response to a Domain Transfer Query request submitted by an unauthorized Registrar: the Registrar that is making the request is not the one that currently operates the domain name nor the one that made the request to change the Registrar:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="2201">
      <msg lang="en">Authorization error</msg>
      <value>
        <extepp:wrongValue>
          <extepp:element>name</extepp:element>
          <extepp:namespace>urn:ietf:params:xml:ns:domain-1.0</extepp:namespace>
          <extepp:value>example.it</extepp:value>
        </extepp:wrongValue>
      </value>
      <extValue>
        <value>
          <extepp:reasonCode>9051</extepp:reasonCode>
        </value>
        <reason lang="en">Lack of permissions to view status of domain transfer
request</reason>
      </extValue>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>0c98654e-b138-4fc7-ac03-fa61f40edf57</svTRID>
    </trID>
  </response>
</epp>
```

Example 3
Response to a Domain Transfer Query request on a domain name that is in pendingTransfer:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:trnData>
        <domain:name>example.it</domain:name>
        <domain:trStatus>pending</domain:trStatus>
        <domain:reID>NEW-REGISTRAR</domain:reID>
        <domain:reDate>2013-02-25T07:40:00+01:00</domain:reDate>
        <domain:acID>DEMO-REGISTRAR</domain:acID>
        <domain:acDate>2013-02-25T23:59:59+01:00</domain:acDate>
```

```
      </domain:trnData>
    </resData>
    <trID>
       <clTRID>ABC-12345</clTRID>
       <svTRID>6017a420-0c4c-4c5d-afd8-a9b5b1c73400</svTRID>
    </trID>
  </response>
</epp>
```

Example 4

Response to a Domain Transfer Query request on a domain name for which the last request for change of Registrar has been approved:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:trnData>
        <domain:name>example.it</domain:name>
        <domain:trStatus>clientApproved</domain:trStatus>
        <domain:reID>NEW-REGISTRAR</domain:reID>
        <domain:reDate>2013-02-25T07:54:21+01:00</domain:reDate>
        <domain:acID>DEMO-REGISTRAR</domain:acID>
        <domain:acDate>2013-02-25T08:54:49+01:00</domain:acDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>fbffe593-fe79-4794-80c8-63724f211f8b</svTRID>
    </trID>
  </response>
</epp>
```

Example 5

Response to a Domain Transfer Query requested by the new Registrar that has submitted the Domain Transfer-Trade command for the change of the Registrar with the simultaneous change of the registrant of a domain name (example.it).

In this case the domain name is in pendingTransfer.

The response to the Domain Transfer Query command contains "extdom:trade" because the Registrar that made a Domain Transfer Query request is the same as the one submitted the Domain Transfer Request with "extdom:trade":

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
```

```
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:trnData>
        <domain:name>example.it</domain:name>
        <domain:trStatus>pending</domain:trStatus>
        <domain:reID>NEW-REGISTRAR</domain:reID>
        <domain:reDate>2013-07-29T15:02:34+02:00</domain:reDate>
        <domain:acID>DEMO-REGISTRAR</domain:acID>
        <domain:acDate>2013-07-29T23:59:59+02:00</domain:acDate>
      </domain:trnData>
      </resData>
      <extension>
        <extdom:trade>
          <extdom:transferTrade>
            <extdom:newRegistrant>MM2-001</extdom:newRegistrant>
            <extdom:newAuthInfo>
               <extdom:pw>NEW2fooBAR</extdom:pw>
            </extdom:newAuthInfo>
          </extdom:transferTrade>
        </extdom:trade>
     </extension>
     <trID>
       <clTRID>ABC-12345</clTRID>
       <svTRID>c6959039-2eee-4aba-9162-a51153af2ecc</svTRID>
     </trID>
  </response>
</epp>
```

Example 6

Response to a Domain Transfer Query submitted by the old Registrar on a domain name maintained by this old Registrar (example.it) that is subject to a Domain Transfer-Trade request for the change of the Registrar with the simultaneous change of the Registrant.
In this case the domain name is in pendingTransfer.
The response to the command Domain Transfer Query does not contain "extdom:trade" because the Registrar that made a Domain Transfer Query request is the same one that manages the domain name:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:trnData>
        <domain:name>example.it</domain:name>
        <domain:trStatus>pending</domain:trStatus>
        <domain:reID>NEW-REGISTRAR</domain:reID>
        <domain:reDate>2013-07-29T15:02:34+02:00</domain:reDate>
        <domain:acID>DEMO-REGISTRAR</domain:acID>
```

```
        <domain:acDate>2013-07-29T23:59:59+02:00</domain:acDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>3a82464f-9d0a-4078-9c38-38db12501f0e</svTRID>
    </trID>
  </response>
</epp>
```

10.3 Polling

10.3.1 Example of a Poll Req request

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <poll op="req"/>
    <clTRID>ABC-12345</clTRID>
   </command>
 </epp>
```

10.3.2 Examples of responses to a Poll Req request

Example 1
Response to a Req Poll completed successfully. The message informs the client in relation to the imminent expiry of the password.
The response contains extepp:passwordReminder which shows the expiry date of the password.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ id="26" count="1">
      <qDate>2013-07-22T09:07:43+02:00</qDate>
      <msg lang="en">Password will expire soon</msg>
    </msgQ>
    <extension>
      <extepp:passwdReminder>
        <extepp:exDate>2013-07-30T12:28:42+02:00</extepp:exDate>
      </extepp:passwdReminder>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>8bc32a50-89cb-4bb8-a216-a1dc75f18204</svTRID>
    </trID>
  </response>
</epp>
```

Example 2
Response to a Poll Req completed successfully. The message informs the client that in the Login request some obsolete namespaces were mentioned.
The answer contains the extepp:wrongNamespaceReminder extension which shows the wrong namespaces and the correct ones.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ id="26" count="1">
      <qDate>2012-04-12T10:00:42.000+02:00</qDate>
      <msg lang="en">Wrong namespace in Login Request</msg>
    </msgQ>
    <extension>
      <extepp:wrongNamespaceReminder>
        <extepp:wrongNamespaceInfo>
          <extepp:wrongNamespace>http://www.nic.it/ITNIC-EPP/extepp-
1.0</extepp:wrongNamespace>
          <extepp:rightNamespace>http://www.nic.it/ITNIC-EPP/extepp-
2.0</extepp:rightNamespace>
        </extepp:wrongNamespaceInfo>
        <extepp:wrongNamespaceInfo>
          <extepp:wrongNamespace>http://www.nic.it/ITNIC-EPP/extdom-
1.0</extepp:wrongNamespace>
          <extepp:rightNamespace>http://www.nic.it/ITNIC-EPP/extdom-
2.0</extepp:rightNamespace>
        </extepp:wrongNamespaceInfo>
      </extepp:wrongNamespaceReminder></extension>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>358813aa-c12d-4871-8a8b-0410f7dba4b0</svTRID>
    </trID>
  </response>
</epp>
```

Example 3
Response to a Poll Req completed successfully. The message informs the client that the nameservers associated with the domain name are not configured correctly.
The response contains extdom:dnsErrorMsgData which contains information on the verification of the correctness of the DNS configuration of the domain name operated by the synchronous server of the Registry:

- **extdom:domain:** name of the domain undergoing check
- **extdom:status:** final check outcome
- **extdom:validationId**: univocal identification code of the check request internally carried out by the synchronous server of the Registry

- **extdom:validationDate**: date of the last check carried out
- extdom:nameservers:
  - *extdom:nameserver*: (list of items)
    - **name:** name of the nameserver (attribute)
    - **extdom:address:** possible IP address (list of items)
      - *type***:** one of the following values (attribute):
        - *IPV4*
        - *IPv6*
- extdom:tests*:*
  - *extdom:test:* (list of items)
    - **name:** name of the test (attribute)
    - **status:** final outcome of the specific test (attribute)
    - **skipped:** test non executed (attribute)
    - extdom:nameserver: (list of items)
      - *name***:** name of the nameserver undergoing a test (attribute)
      - *status***:** outcome of the specific test on the nameserver (attribute)
      - *extdom:detail:* (list of items)
        - **name***: host name in the query for the CNAMEHostTest (attribute)*
        - **status***: outcome of the query (attribute)*
        - **queryId***: query identifier (attribute)*
- extdom:queries*:*
  - *extdom:query*: (list of items)
    - **id:** query identifier (attribute)
    - **extdom:queryFor:** object of the query
    - **extdom:type:** one of the following values:
      - ANY
      - SOA
      - NS
      - A
      - MX
      - AAAA
      - CNAME
      - PTR
      - TXT
    - extdom:destination: query receiver
    - **extdom:result:** outcome of the query

In each element the *status* attribute can take the following values: FAILED, SUCCEEDED, WARNING

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:host="urn:ietf:params:xml:ns:host-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
```

```
  </result>
  <msgQ id="4" count="36">
    <qDate>2012-04-04T18:25:14.000+02:00</qDate>
    <msg lang="en">DNS check ended unsuccessfully</msg>
  </msgQ>
  <extension>
    <extdom:dnsErrorMsgData version="2.0">
      <extdom:domain>esempio.it</extdom:domain>
      <extdom:status>FAILED</extdom:status>
      <extdom:validationId>e7edc45c-7e38-4d98-bf40-
96c9f604dec8</extdom:validationId>
      <extdom:validationDate>2012-04-
04T18:20:13.993+02:00</extdom:validationDate>
      <extdom:nameservers>
        <extdom:nameserver name="ns1.esempio.it.">
          <extdom:address type="IPv4">192.12.192.23</extdom:address>
        </extdom:nameserver>
        <extdom:nameserver name="ns2.esempio.it.">
          <extdom:address type="IPv4">192.12.192.24</extdom:address>
        </extdom:nameserver>
      </extdom:nameservers>
      <extdom:tests>
        <extdom:test status="SUCCEEDED" name="NameserversResolvableTest">
          <extdom:nameserver status="SUCCEEDED" name="ns1.esempio.it."/>
          <extdom:nameserver status="SUCCEEDED" name="ns2.esempio.it."/>
        </extdom:test>
        <extdom:test status="FAILED" name="NameserversAnswerTest">
          <extdom:nameserver status="FAILED" name="ns2.esempio.it.">
            <extdom:detail queryId="2">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
          <extdom:nameserver status="FAILED" name="ns1.esempio.it.">
            <extdom:detail queryId="1">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
        </extdom:test>
        <extdom:test status="FAILED" name="NameserverReturnCodeTest">
          <extdom:nameserver status="FAILED" name="ns2.esempio.it.">
            <extdom:detail queryId="2">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
          <extdom:nameserver status="FAILED" name="ns1.esempio.it.">
            <extdom:detail queryId="1">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
        </extdom:test>
        <extdom:test status="FAILED" name="AATest">
          <extdom:nameserver status="FAILED" name="ns2.esempio.it.">
            <extdom:detail queryId="2">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
          <extdom:nameserver status="FAILED" name="ns1.esempio.it.">
            <extdom:detail queryId="1">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
        </extdom:test>
        <extdom:test status="FAILED" name="NSCompareTest">
          <extdom:nameserver status="FAILED" name="ns2.esempio.it.">
            <extdom:detail queryId="2">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
```

```
          <extdom:nameserver status="FAILED" name="ns1.esempio.it.">
            <extdom:detail queryId="1">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
        </extdom:test>
        <extdom:test status="FAILED" name="NSCountTest">
          <extdom:nameserver status="FAILED" name="ns2.esempio.it.">
            <extdom:detail queryId="2">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
          <extdom:nameserver status="FAILED" name="ns1.esempio.it.">
            <extdom:detail queryId="1">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
        </extdom:test>
        <extdom:test status="WARNING" name="CNAMEHostTest"/>
        <extdom:test status="FAILED" name="IPCompareTest">
          <extdom:nameserver status="FAILED" name="ns1.esempio.it.">
            <extdom:detail>Unresolveable ns1.esempio.it.</extdom:detail>
          </extdom:nameserver>
          <extdom:nameserver status="FAILED" name="ns2.esempio.it.">
            <extdom:detail>Unresolveable ns2.esempio.it.</extdom:detail>
          </extdom:nameserver>
        </extdom:test>
        <extdom:test status="FAILED" name="MXCompareTest">
          <extdom:nameserver status="FAILED" name="ns2.esempio.it.">
            <extdom:detail queryId="2">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
          <extdom:nameserver status="FAILED" name="ns1.esempio.it.">
            <extdom:detail queryId="1">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
        </extdom:test>
        <extdom:test status="WARNING" name="MXRecordIsPresentTest">
          <extdom:nameserver status="WARNING" name="ns2.esempio.it.">
            <extdom:detail queryId="2">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
          <extdom:nameserver status="WARNING" name="ns1.esempio.it.">
            <extdom:detail queryId="1">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
        </extdom:test>
        <extdom:test status="WARNING" name="SOAMasterCompareTest">
          <extdom:nameserver status="WARNING" name="ns2.esempio.it.">
            <extdom:detail queryId="2">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
          <extdom:nameserver status="WARNING" name="ns1.esempio.it.">
            <extdom:detail queryId="1">Nameserver test skipped for error in
query: java.net.SocketTimeoutException</extdom:detail>
          </extdom:nameserver>
        </extdom:test>
        <extdom:test skipped="true" name="IPSoaTest"/>
      </extdom:tests>
      <extdom:queries>
        <extdom:query id="1">
          <extdom:queryFor>esempio.it.</extdom:queryFor>
          <extdom:type>ANY</extdom:type>
          <extdom:destination>
```

```
            ns1.esempio.it./[IPAddress(address=/192.12.192.23, family=1)]
            </extdom:destination>
            <extdom:result>java.net.SocketTimeoutException</extdom:result>
          </extdom:query>
          <extdom:query id="2">
            <extdom:queryFor>esempio.it.</extdom:queryFor>
            <extdom:type>ANY</extdom:type>
            <extdom:destination>
            ns2.esempio.it./[IPAddress(address=/192.12.192.24, family=1)]
            </extdom:destination>
            <extdom:result>java.net.SocketTimeoutException</extdom:result>
          </extdom:query>
          <extdom:query id="3">
            <extdom:queryFor>ns1.esempio.it.</extdom:queryFor>
            <extdom:type>ANY</extdom:type>
            <extdom:destination>
            ns1.esempio.it./[IPAddress(address=/192.12.192.23, family=1)],
            ns2.esempio.it./[IPAddress(address=/192.12.192.24, family=1)]
            </extdom:destination>
            <extdom:result>java.net.SocketTimeoutException</extdom:result>
          </extdom:query>
          <extdom:query id="4">
            <extdom:queryFor>ns2.esempio.it.</extdom:queryFor>
            <extdom:type>ANY</extdom:type>
            <extdom:destination>
            ns1.esempio.it./[IPAddress(address=/192.12.192.23, family=1)],
            ns2.esempio.it./[IPAddress(address=/192.12.192.24, family=1)]
            </extdom:destination>
            <extdom:result>java.net.SocketTimeoutException</extdom:result>
          </extdom:query>
        </extdom:queries>
      </extdom:dnsErrorMsgData>
    </extension>
    <trID>
      <clTRID>RTRT-00037</clTRID>
      <svTRID>4f8a0382-95e3-402e-a120-83aee6a41038</svTRID>
    </trID>
  </response>
</epp>
```

Example 4

Response to a Poll Req completed successfully. The server notifies the client of an event that took place on a domain name. In this case, the message informs the Registrar that the domain name, which he managed, has been deleted. The response contains extdom:simpleMsgData, which only shows the domain name in question.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ id="24" count="1">
      <qDate>2013-07-21T12:44:37+02:00</qDate>
```

```
      <msg lang="en">Domain has been deleted</msg>
    </msgQ>
    <extension>
      <extdom:simpleMsgData>
            <extdom:name>esempio.it</extdom:name>
      </extdom: simpleMsgData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>2b421937-c235-4a92-9209-613deb593ac2</svTRID>
    </trID>
  </response>
</epp>
```

Example 5

Response to a Poll Req completed successfully. The server sends the switch status of a domain name. The message is sent following a report from a client following an event. This example considers a request for deletion and subsequent placing of the domain name in pendingDelete/redemptionPeriod.
The response contains extdom:chgStatusMsgData, which contains the following information:

- *extdom:name*: The name of the domain to which the message refers and which underwent a switch status;
- *extdom:targetStatus*: the status of arrival of the domain name contains the statuses that may belong to the standard ones described in XML Schema domain-1.0.xsd (tag domain:status) to the extension of the protocol for the grace period and are described in rgp-1.0.xsd (tag rgp:rgpStatus) or the extension extdom-2.0.xsd (tag extdom:ownStatus).

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ id="84" count="1">
      <qDate>2013-08-04T18:57:45+02:00</qDate>
      <msg lang="en">redemptionPeriod is started</msg>
    </msgQ>
    <extension>
      <extdom:chgStatusMsgData>
        <extdom:name>esempio.it</extdom:name>
        <extdom:targetStatus>
          <domain:status lang="en" s="pendingDelete"/>
          <rgp:rgpStatus lang="en" s="redemptionPeriod"/>
        </extdom:targetStatus>
      </extdom:chgStatusMsgData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>fbf2aa9a-f195-4fee-a23b-8e4af6a1d7e1</svTRID>
    </trID>
  </response>
</epp>
```

Example 6
Response to a Poll Req completed successfully. The server notifies the interruption of the proxy.
The message is sent following the deletion of a domain name registered with nameservers subordinate to it and referenced in the registration of other domain names.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ id="24" count="1">
      <qDate>2013-07-21T12:50:57+02:00</qDate>
      <msg lang="en">Lost delegation</msg>
    </msgQ>
    <extension>
      <extdom:dlgMsgData>
        <extdom:name>dominio.it</extdom:name>
       <extdom:ns>ns1.esempio.it</extdom:ns>
      </extdom:dlgMsgData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>aa5b4315-6d24-4c10-8db9-c38732ec8140</svTRID>
    </trID>
  </response>
</epp>
```

Example 7
Response to a Poll Req completed successfully. The server notifies the current Registrar of the domain name of the request for change of Registrar received from the new Registrar.
In this case the change of Registrar was requested by the “NEW-REGISTRAR” using Domain Transfer “op=request”. The message is inserted in the polling queue of the Registrar “DEMO-REGISTRAR” that manages the domain name.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ id="33" count="1">
      <qDate>2013-07-29T10:19:16+02:00</qDate>
      <msg lang="en">Domain transfer has been requested: pendingTransfer is
started</msg>
```

```
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name>esempio.it</domain:name>
        <domain:trStatus>pending</domain:trStatus>
        <domain:reID>NEW-REGISTRAR</domain:reID>
        <domain:reDate>2013-07-29T10:19:16+02:00</domain:reDate>
        <domain:acID>DEMO-REGISTRAR</domain:acID>
        <domain:acDate>2013-07-29T23:59:59+02:00</domain:acDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>f8c470d4-e23e-412d-9073-5f1ec43bf088</svTRID>
    </trID>
  </response>
</epp>
```

Example 8

Response to a Poll Req completed successfully. The server notifies the current Registrar of the domain name that the request for change of Registrar has been annulled by the new Registrar.
The request is submitted by the Registrar “NEW-REGISTRAR” using Domain Transfer “op=cancel”. The message is inserted in the polling queue of the Registrar “DEMO-REGISTRAR” that manages the domain name.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ id="34" count="1">
      <qDate>2013-07-29T10:23:34+02:00</qDate>
      <msg lang="en">Domain transfer has been cancelled</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name>esempio.it</domain:name>
        <domain:trStatus>clientCancelled</domain:trStatus>
        <domain:reID>NEW-REGISTRAR</domain:reID>
        <domain:reDate>2013-07-29T10:19:16+02:00</domain:reDate>
        <domain:acID>DEMO-REGISTRAR</domain:acID>
        <domain:acDate>2013-07-29T10:23:34+02:00</domain:acDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>f4ac9065-1514-461f-b47e-ce49d76f0a1c</svTRID>
    </trID>
  </response>
</epp>
```

Example 9

Response to a Poll Req completed successfully. The server notifies the new Registrar that the request for change of Registrar has been rejected by the current Registrar of the domain name.
The request is submitted by the Registrar “DEMO-REGISTRAR” using Domain Transfer “op=reject”. The message is inserted in the polling queue of the Registrar “NEW-REGISTRAR” that previously made the request for change of Registrar.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ id="36" count="1">
      <qDate>2013-07-29T10:31:23+02:00</qDate>
      <msg lang="en">Domain transfer has been rejected</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name>esempio.it</domain:name>
        <domain:trStatus>clientRejected</domain:trStatus>
        <domain:reID>NEW-REGISTRAR</domain:reID>
        <domain:reDate>2013-07-29T10:30:39+02:00</domain:reDate>
        <domain:acID>DEMO-REGISTRAR</domain:acID>
        <domain:acDate>2013-07-29T10:50:22+02:00</domain:acDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>22a4ebad-cbdc-484e-b2af-daabbef0e04e</svTRID>
    </trID>
  </response>
</epp>
```

Example 10
Response to a Poll Req completed successfully. The server notifies the new Registrar that the request for change of Registrar has been approved by the current Registrar of the domain name.
The request is submitted by the Registrar “DEMO-REGISTRAR” using Domain Transfer “op=approve”. The message is inserted in the polling queue of the Registrar “NEW-REGISTRAR” and, with a different svTRID, in that of the old Registrar “DEMO-REGISTRAR”.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
    </result>
```

```
    <msgQ id="36" count="1">
      <qDate>2013-07-29T10:31:23+02:00</qDate>
      <msg lang="en">Domain transfer has been executed</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name>esempio.it</domain:name>
        <domain:trStatus>clientApproved</domain:trStatus>
        <domain:reID>NEW-REGISTRAR</domain:reID>
        <domain:reDate>2013-07-29T10:30:39+02:00</domain:reDate>
        <domain:acID>DEMO-REGISTRAR</domain:acID>
        <domain:acDate>2013-07-29T10:31:22+02:00</domain:acDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>0f5c62c1-244c-4fb7-9c46-ff96bd2dcd08</svTRID>
    </trID>
  </response>
</epp>
```

Example 11

Response to a Poll Req completed successfully.
In this case the change of Registrar is made automatically by the server at the expiry of pendingTransfer. The message is inserted in the polling queue of the Registrar “NEW-REGISTRAR” and, with a different svTRID, in that of the old Registrar “DEMO-REGISTRAR”.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ id="36" count="1">
      <qDate>2013-07-29T10:31:23+02:00</qDate>
      <msg lang="en">Domain transfer is expired: transfer has been executed</msg>
    </msgQ>
    <resData>
      <domain:trnData>
        <domain:name>esempio.it</domain:name>
        <domain:trStatus>serverApproved</domain:trStatus>
        <domain:reID>NEW-REGISTRAR</domain:reID>
        <domain:reDate>2013-07-29T10:30:39+02:00</domain:reDate>
        <domain:acID>DEMO-REGISTRAR</domain:acID>
        <domain:acDate>2013-07-30T00:45:00+02:00</domain:acDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>ad4ccc2c-33e2-40b8-93c9-1d4ff959a3e5</svTRID>
    </trID>
  </response>
</epp>
```

Example 12
Response to a Poll Req completed successfully. The server notifies the Registrar about a domain name registration which contains "*remapped*" characters (Section 5.1.2):

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0" xmlns:extcon="http://www.nic.it/ITNIC-
 EPP/extcon-1.0" xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0">
  <response>
    <result code="1301">
      <msg lang="en">Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ id="32969079" count="54">
       <qDate>2012-03-27T14:07:04.000+01:00</qDate>
       <msg lang="en">Requested IDN domain contains remapped chars</msg>
    </msgQ>
    <extension>
      <extdom:remappedIdnData>
        <extdom:idnRequested>άιάιάιάι.it</extdom:idnRequested>
        <extdom:idnCreated>ΰΰΰΰ.it</extdom:idnCreated>
      </extdom:remappedIdnData>
    </extension>
    <trID>
      <svTRID>25a61bec-0c95-4c8c-b389-6c10fz00ub74</svTRID>
    </trID>
  </response>
</epp>
```

The response includes the extension "extdom:remappedIdnData" that contains the IDN domain name requested in the Domain Create command ("extdom:idnRequested") and the domain name actually recorded as a result of transformation ("extdom:idnCreated").

10.3.3 Example of a Poll Ack request

Poll Ack for deleting the message with msgID="227" from the client's polling queue:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
  <command>
    <poll op="ack" msgID="227"/>
    <clTRID>ABC-12346</clTRID>
  </command>
</epp>
```

10.3.4 Examples of responses to a Poll Ack request

Example 1
Response to successful Poll Ack:

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
```

```
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>ABC-12346</clTRID>
      <svTRID>a5c3bc6c-0b8e-469c-a208-ec0a37cc1b91</svTRID>
   </trID>
  </response>
</epp>
```

Example 2
Response to successful Poll Ack sent by client whose polling queue is empty.

```
<?xml version="1.0" encoding="UTF-8" ?>
<epp xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1300">
      <msg lang="en">Command completed successfully; no messages</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>3239ef1b-d3da-486c-a160-301c3efe256c</svTRID>
    </trID>
  </response>
</epp>
```

11 DNSSEC in the ccTLD .it

The DNS (Domain Name System) protocol (RFC1035) defines the specifications to provide a domain name resolution service that has no form of authentication nor implements mechanisms to ensure data integrity. In order to overcome this limitation, the IETF has defined (RFC4033 e RFC 4641) a protocol known as Domain Name System Security Extensions (DNSSEC). DNSSEC uses public/private keys for cryptography to ensure that the information is coming from an authoritative source, and has not been altered during its transport through the network.
DNSSEC enables:

- DNS servers to sign their own resource records (RRs) with a private key;
- DNS resolvers to verify the information through its public key.

The public keys are stored in the “parent” zone of the digitally signed zone.

To facilitate the verification of signatures, DNSSEC has established some new RRs:

- *RRSIG*: containing a cryptographic signature for a set of RRs of the same type (RRset);

- *DNSKEY*: containing a public key.

DNSSEC also introduces the Delegation Signer (DS) record to implement the "chain-of-trust" between a parent zone and a child zone. A zone manager generates a "digest" of the public key (DNSKEY record) associated with the digitally signed domain name and transmits it to the parent zone manager who associates the delegation of the domain name through a DS record.

11.1 Delegation Signer (DS) record

In general, the most commonly methods used to obtain the DS record to be associated with a domain name and create the "chain-of-trust" in a TLD zone are:

- the Registrar (or the person who manages the authoritative name server of the zone you wish to sign) generates and transmits it (via EPP, via Web portal, etc.) to the Registry that manages the TLD;
- the Registrar (or the person who manages the authoritative name server of the zone you wish to sign) generates and transmits (e.g. via EPP, via Web portal) to the Registry that manages the TLD the public key associated with the zone and consequently, the DS record generation is ascribed to the TLD Registry.

The ccTLD .it has implemented the first solution, that is: the Registrar shall communicate to the .it Registry the DS records associated with a domain name. Even if a domain name has the DNS service managed by someone other than the Registrar, the DS record transmission is still under the responsibility of the Registrar of the domain name, and must be made exclusively through EPP.
In practice, the Registrar must communicate to the .it Registry, via EPP, the following four fields, which compose the DS record associated with the domain name that has been signed:

- **keytag**: this value is automatically calculated when the DS record is generated and depends strictly on the information related to the public key;
- **algorithm**: the values supported by the .it Registry are the following:
  - 3 (DSA/SHA-1)
  - 5 (RSA/SHA-1)
  - 6 (DSA-NSEC3-SHA1)
  - 7 (RSASHA1-NSEC3-SHA1)
  - 8 (RSA/SHA-256)
  - 10 (RSA/SHA-512)
  - 12 (ECC-GOST)
  - 13 (ECDSAP256SHA256)
  - 14 (ECDSAP384SHA384)
- **digest type**: the values supported by the .it Registry are the following:
  - 1 (SHA-1)
  - 2 (SHA-256)
  - 3 (GOST R 34.11-94)
  - 4 (SHA-384)
- **digest**: is the hash generated from the public key according to the algorithm and digest type.

11.2 Registrars and DNSSEC

In the .it ccTLD, the adoption of DNSSEC protocol by Registrars is neither mandatory nor binding.
The Registrars interested in providing this service to their customers will have to carry out a “DNSSEC technical accreditation test” whose specifications are outlined in a dedicated document.
The Registrars which, on the contrary, do not wish to make use of this service, can continue to operate as at present.
A Registrar has the possibility to carry out the DNSSEC technical accreditation test, if and only if, it is an accredited Registrar in the .it ccTLD and in “active” status.
Registrars are therefore divided into two categories:

- “DNSSEC accredited”;
- “non DNSSEC accredited”.

The “DNSSEC accredited” Registrars are identified on the Registry website (www.nic.it) by a special logo.
The Whois service contains, in the “Registrar” section, the “DNSSEC:” field with “yes” or “no” values depending on whether the Registrar is “DNSSEC accredited” or not.
Similarly, the Whois service includes, for domain names, the “Signed:” field with “yes” or “no” values depending on whether the domain name has been signed or not.
In the .it ccTLD the transmission of DS records associated with digitally signed zones, for their publication in the .it zone file, is carried out by the “DNSSEC accredited” Registrars and this must take place exclusively with EPP.

11.3 EPP and DNSSEC

Let us now see the implications of the introduction of DNSSEC on the requests that a “DNSSEC accredited” Registrar must send to the EPP server of .it Registry through its client.
The **secDNS-1.1** EPP extension (RFC5910) describes two different ways to enable a Registrar to transmit to the Registry the information regarding the DS records:

1. the first, referred to as “DS Data Interface” foresees the transmission via EPP of DS records information to the Registry. This transmission takes place concurrently with the registration of a “signed” domain name (through the Domain Create operation) or a subsequent modification of the DS records associated with it (addition, removal or replacement by an Domain Update operation). The EPP server of the Registry will report such information in the response to the Domain Info request;

2. the second, referred to as “Key Data Interface”, is quite similar to the first with the difference that the Registrar, instead of providing information about the DS records, must provide data on the public key associated with the “signed” domain name (flags, protocol, alg, pubkey).

Optionally, the protocol provides that the “DS Data Interface” can be provided, along with the DS record information, including those relating to the “Key Data Interface”. This is to facilitate an eventual consistency check by the Registry, including the public key and the DS records associated with the domain name.
It is mandatory that the EPP server supports a single mode of transmission of information within a single request or response.
The .it ccTLD has chosen the “DS Data Interface” way of transmission, by which the Registrar sends to the Registry information concerning only DS records.

The next sections describe the main EPP operations that are involved in the introduction of DNSSEC in .it, where a Registrar has passed the DNSSEC technical accreditation test and consequently is a "DNSSEC accredited" Registrar.

11.3.1 Login

Only a "DNSSEC accredited" Registrar can indicate in the EPP Login request the two following namespaces:

- **urn:ietf:params:xml:ns:secDNS-1.1**, concerning the standard extensions introduced by EPP;
- **http://www.nic.it/ITNIC-EPP/extsecDNS-1.0**, concerning extensions introduced by the .it Registry.

If a "DNSSEC accredited" Registrar does not indicate in the EPP Login request (Section 7.2.2) the above mentioned namespaces, they cannot send, in the EPP session, further requests containing extensions related to them.

The responses obtained by the EPP server have a different header depending on whether the Registrar has indicated (Section 0) or has not indicated (Section 0) in the EPP Login request the above mentioned two namespaces.

11.3.2 Domain Create

The Domain Create command has been extended with the addition, in the **<extension>** section, of the **<secDNS:create>** element (where secDNS is the prefix that identifies the reference to the secDNS-1.1 namespace), which can contain up to a maximum of 2 <**secDNS:Dsdata**> elements corresponding to DS records.
Therefore, a Domain Create request for a .it domain name with the DNSSEC extension that makes use of the DS Data Interface, has the following XML format:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 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>esempio.it</domain:name>
        <domain:period unit="y">1</domain:period>
        <domain:ns>
          <domain:hostAttr>
            <domain:hostName>x.dns.it</domain:hostName>
          </domain:hostAttr>
          <domain:hostAttr>
            <domain:hostName>y.dns.it</domain:hostName>
          </domain:hostAttr>
        </domain:ns>
        <domain:registrant>mm001</domain:registrant>
        <domain:contact type="admin">mm001</domain:contact>
        <domain:contact type="tech">mb001</domain:contact>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
```

```
        </domain:authInfo>
      </domain:create>
    </create>
    <extension>
      <secDNS:create
       xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
        <secDNS:dsData>
          <secDNS:keyTag>12345</secDNS:keyTag>
          <secDNS:alg>3</secDNS:alg>
          <secDNS:digestType>1</secDNS:digestType>
          <secDNS:digest>4347d0f8ba661234a8eadc005e2e1d1b646c9682</secDNS:digest>
        </secDNS:dsData>
      </secDNS:create>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

11.3.3 Domain Update

The Domain Update command has been extended, indicating, in the **<extension>** section, the **<secDNS:update>** element.
Therefore, a Domain Update request for a .it domain name, with the DNSSEC extension that makes use of the DS Data Interface, for which the replacement of the DS record currently associated with another DS record was requested, has the following XML format (in the example below it is requested to replace the current DS record with a new one):

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <command>
    <update>
      <domain:update
       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
       xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
        <domain:name>esempio.it</domain:name>
      </domain:update>
    </update>
    <extension>
      <secDNS:update xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
        <secDNS:rem>
          <secDNS:dsData>
            <secDNS:keyTag>12345</secDNS:keyTag>
            <secDNS:alg>3</secDNS:alg>
            <secDNS:digestType>1</secDNS:digestType>
            <secDNS:digest>
            4347d0f8ba661234a8eadc005e2e1d1b646c9682
            </secDNS:digest>
          </secDNS:dsData>
        </secDNS:rem>
        <secDNS:add>
          <secDNS:dsData>
            <secDNS:keyTag>45063</secDNS:keyTag>
            <secDNS:alg>3</secDNS:alg>
            <secDNS:digestType>2</secDNS:digestType>
            <secDNS:digest>
             E9B696C3AC8644735BF0A6409BE6D77BBFB4142D667E0EB0D41AD75BCC9D0D43
            </secDNS:digest>
          </secDNS:dsData>
```

```
        </secDNS:add>
      </secDNS:update>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

If the Registrar intends to request the removal of all DS records associated with the domain name, this is allowed through a Domain Update operation in which, in **<secDNS:rem>** section, the **<secDNS:all>** element is used.

Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <command>
    <update>
      <domain:update
       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
       xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
        <domain:name>esempio.it</domain:name>
      </domain:update>
    </update>
    <extension>
      <secDNS:update xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
        <secDNS:rem>
          <secDNS:all>true</secDNS:all>
        </secDNS:rem>
      </secDNS:update>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

11.3.4 Domain Transfer

The introduction of DNSSEC does not imply any change to the formats of the Domain Transfer request and response.
No restrictions are applied to transfers between "DNSSEC accredited" Registrars and "non DNSSEC accredited" Registrars.
The Domain Transfer operation does not alter the DNS configuration: if the new Registrar wishes to change it, a new DNS configuration (with or without DNSSEC extension) can be submitted through a Domain Update operation.

11.3.5 Domain Delete

The introduction of DNSSEC does not imply any change to the formats of Domain Delete response and request.

11.3.6 Domain Info

In case of a Domain Info request for a "signed" domain name, the request involves two distinct XML formats depending on whether the DNS configuration of the domain name is in the process of being validated or has already been validated.

The introduction of DNSSEC has involved the introduction of the element <**extsecDNS:infDsOrKeyToValidateData**> (described in the extsecDNS-1.0 namespace), which shows, for a given domain name, the configuration of the DS records that are in the process of validation by the DNS check service of the .it Registry.
A Domain Info response of a "signed" domain name that has been registered but not validated by the DNS check service, and which therefore is in inactive/*dnsHold* status, has the following XML format:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extsecDNS="http://www.nic.it/ITNIC-EPP/extsecDNS-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData>
        <domain:name>esempio.it</domain:name>
        <domain:roid>ITNIC-306194</domain:roid>
        <domain:status s="inactive" lang="en"/>
        <domain:registrant>MM001</domain:registrant>
        <domain:contact type="admin">MM001</domain:contact>
        <domain:contact type="tech">MB001</domain:contact>
        <domain:clID>DEMO-REG</domain:clID>
        <domain:crID>DEMO-REG</domain:crID>
        <domain:crDate>2016-06-29T08:26:44.000+02:00</domain:crDate>
        <domain:exDate>2017-06-29T23:59:59.000+02:00</domain:exDate>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:infData>
    </resData>
    <extension>
      <extdom:infData>
        <extdom:ownStatus s="dnsHold" lang="en"/>
      </extdom:infData>
      <extdom:infNsToValidateData>
        <extdom:nsToValidate>
          <domain:hostAttr>
            <domain:hostName>m.dns.it</domain:149 hostName >
          </domain:hostAttr>
          <domain:hostAttr>
            <domain:hostName>j.dns.it</domain:hostName>
          </domain:hostAttr>
        </extdom:nsToValidate>
      </extdom:infNsToValidateData>
      <extsecDNS:infDsOrKeyToValidateData>
        <extsecDNS:dsOrKeysToValidate>
          <secDNS:dsData>
            <secDNS:keyTag>12345</secDNS:keyTag>
            <secDNS:alg>3</secDNS:alg>
            <secDNS:digestType>1</secDNS:digestType>
            <secDNS:digest>
            4347d0f8ba661234a8eadc005e2e1d1b646c9682
```

```
            </secDNS:digest>
          </secDNS:dsData>
        </extsecDNS:dsOrKeysToValidate>
      </extsecDNS:infDsOrKeyToValidateData>
    </extension>
    <trID>
      <svTRID>9141b61b-5272-4d63-90b1-7cb2348f5b40</svTRID>
    </trID>
  </response>
</epp>
```

If the DNS validation (both authoritative name servers and DS records) is successful, the domain name is put into ok status and the Domain Info response, in this case, takes the following XML format:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extsecDNS="http://www.nic.it/ITNIC-EPP/extsecDNS-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData>
        <domain:name>esenpio.it</domain:name>
        <domain:roid>ITNIC-306194</domain:roid>
        <domain:status s="ok" lang="en"/>
        <domain:registrant>MM001</domain:registrant>
        <domain:contact type="admin">MM001</domain:contact>
        <domain:contact type="tech">MB001</domain:contact>
        <domain:ns>
          <domain:hostAttr>
            <domain:hostName>m.dns.it</domain:hostName>
          </domain:hostAttr>
          <domain:hostAttr>
            <domain:hostName>j.dns.it</domain:hostName>
          </domain:hostAttr>
        </domain:ns>
        <domain:clID>DEMO-REG</domain:clID>
        <domain:crID>DEMO-REG</domain:crID>
        <domain:crDate>2016-06-29T08:26:44.000+02:00</domain:crDate>
        <domain:upID>DEMO-REG</domain:upID>
        <domain:upDate>2016-06-29T08:26:45.000+02:00</domain:upDate>
        <domain:exDate>2017-06-29T23:59:59.000+02:00</domain:exDate>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:infData>
    </resData>
    <extension>
      <secDNS:infData>
        <secDNS:dsData>
          <secDNS:keyTag>12345</secDNS:keyTag>
          <secDNS:alg>3</secDNS:alg>
          <secDNS:digestType>1</secDNS:digestType>
```

```
          <secDNS:digest>4347d0f8ba661234a8eadc005e2e1d1b646c9682</secDNS:digest>
        </secDNS:dsData>
      </secDNS:infData>
    </extension>
    <trID>
      <svTRID>615ec859-f80d-41f2-b55f-0d7108b91cb6</svTRID>
    </trID>
  </response>
</epp>
```

If a "signed" domain name is subject to a Domain Update operation in order to change the authoritative name servers and/or DS records, since in the pendingUpdate status a successfully validated DNS configuration already exists, the Domain Info response can contain the <**domain:ns**> and <**extdom:infNsToValidateData**> elements (if name server change is requested) and <**secDNS:infData**> and <**extsecDNS:infDsOrKeyToValidateData**> elements (if DS records change is requested), at the same time.
The following example shows the result of a Domain Info operation on a domain name for which authoritative name servers and DS records update were requested:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
 xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
 xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
 xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
 xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1"
 xmlns:extcon="http://www.nic.it/ITNIC-EPP/extcon-1.0"
 xmlns:extdom="http://www.nic.it/ITNIC-EPP/extdom-2.0"
 xmlns:extsecDNS="http://www.nic.it/ITNIC-EPP/extsecDNS-1.0"
 xmlns:extepp="http://www.nic.it/ITNIC-EPP/extepp-2.0">
  <response>
    <result code="1000">
      <msg lang="en">Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData>
        <domain:name>esenpio.it</domain:name>
        <domain:roid>ITNIC-306194</domain:roid>
        <domain:status s="pendingUpdate" lang="en"/>
        <domain:registrant>MM001</domain:registrant>
        <domain:contact type="admin">MM001</domain:contact>
        <domain:contact type="tech">MB001</domain:contact>
        <domain:ns>
          <domain:hostAttr>
            <domain:hostName>m.dns.it</domain:hostName>
          </domain:hostAttr>
          <domain:hostAttr>
            <domain:hostName>j.dns.it</domain:hostName>
          </domain:hostAttr>
        </domain:ns>
        <domain:clID>DEMO-REG</domain:clID>
        <domain:crID>DEMO-REG</domain:crID>
        <domain:crDate>2016-06-29T08:26:44.000+02:00</domain:crDate>
        <domain:upID>DEMO-REG</domain:upID>
        <domain:upDate>2016-06-29T08:26:45.000+02:00</domain:upDate>
        <domain:exDate>2017-06-29T23:59:59.000+02:00</domain:exDate>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:infData>
```

```
    </resData>
    <extension>
      <extdom:infNsToValidateData>
        <extdom:nsToValidate>
          <domain:hostAttr>
            <domain:hostName>n.dns.it</domain:hostName>
          </domain:hostAttr>
          <domain:hostAttr>
            <domain:hostName>k.dns.it</domain:hostName >
          </domain:hostAttr>
        </extdom:nsToValidate>
      </extdom:infNsToValidateData>
      <secDNS:infData>
        <secDNS:dsData>
          <secDNS:keyTag>12345</secDNS:keyTag>
          <secDNS:alg>3</secDNS:alg>
          <secDNS:digestType>1</secDNS:digestType>
          <secDNS:digest>4347d0f8ba661234a8eadc005e2e1d1b646c9682</secDNS:digest>
        </secDNS:dsData>
      </secDNS:infData>
      <extsecDNS:infDsOrKeyToValidateData>
        <extsecDNS:dsOrKeysToValidate>
          <secDNS:dsData>
            <secDNS:keyTag>45063</secDNS:keyTag>
            <secDNS:alg>3</secDNS:alg>
            <secDNS:digestType>2</secDNS:digestType>
            <secDNS:digest>
            E9B696C3AC8644735BF0A6409BE6D77BBFB4142D667E0EB0D41AD75BCC9D0D43
            </secDNS:digest>
          </secDNS:dsData>
        </extsecDNS:dsOrKeysToValidate>
      </extsecDNS:infDsOrKeyToValidateData>
    </extension>
    <trID>
      <svTRID>1e53552c-585a-4a48-8c45-4b2068ea057d</svTRID>
    </trID>
  </response>
</epp>
```

The following example shows, on the contrary, the result of a Domain Info operation on a domain name for which only the removal of all DS records was requested:

```
<?xml version=”1.0” encoding=”UTF-8” standalone=”no”?>
<epp xmlns=”urn:ietf:params:xml:ns:epp-1.0”
 xmlns:domain=”urn:ietf:params:xml:ns:domain-1.0”
 xmlns:contact=”urn:ietf:params:xml:ns:contact-1.0”
 xmlns:rgp=”urn:ietf:params:xml:ns:rgp-1.0”
 xmlns:secDNS=”urn:ietf:params:xml:ns:secDNS-1.1”
 xmlns:extcon=”http://www.nic.it/ITNIC-EPP/extcon-1.0”
 xmlns:extdom=”http://www.nic.it/ITNIC-EPP/extdom-2.0”
 xmlns:extsecDNS=”http://www.nic.it/ITNIC-EPP/extsecDNS-1.0”
 xmlns:extepp=”http://www.nic.it/ITNIC-EPP/extepp-2.0”>
  <response>
    <result code=”1000”>
      <msg lang=”en”>Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData>
        <domain:name>esenpio.it</domain:name>
        <domain:roid>ITNIC-306194</domain:roid>
        <domain:status s=”pendingUpdate” lang=”en”/>
```

```
        <domain:registrant>MM001</domain:registrant>
        <domain:contact type="admin">MM001</domain:contact>
        <domain:contact type="tech">MB001</domain:contact>
        <domain:ns>
          <domain:hostAttr>
            <domain:hostName>m.dns.it</domain:hostName>
          </domain:hostAttr>
          <domain:hostAttr>
            <domain:hostName>j.dns.it</domain:hostName>
          </domain:hostAttr>
        </domain:ns>
        <domain:clID>DEMO-REG</domain:clID>
        <domain:crID>DEMO-REG</domain:crID>
        <domain:crDate>2016-06-29T08:26:44.000+02:00</domain:crDate>
        <domain:upID>DEMO-REG</domain:upID>
        <domain:upDate>2016-06-29T08:26:45.000+02:00</domain:upDate>
        <domain:exDate>2017-06-29T23:59:59.000+02:00</domain:exDate>
        <domain:authInfo>
          <domain:pw>22fooBAR</domain:pw>
        </domain:authInfo>
      </domain:infData>
    </resData>
    <extension>
      <secDNS:infData>
        <secDNS:dsData>
          <secDNS:keyTag>12345</secDNS:keyTag>
          <secDNS:alg>3</secDNS:alg>
          <secDNS:digestType>1</secDNS:digestType>
          <secDNS:digest>4347d0f8ba661234a8eadc005e2e1d1b646c9682</secDNS:digest>
        </secDNS:dsData>
      </secDNS:infData>
      <extsecDNS:infDsOrKeyToValidateData>
        <extsecDNS:remAll/>
      </extsecDNS:infDsOrKeyToValidateData>
    </extension>
    <trID>
      <svTRID>3774a765-5418-4f43-a999-5d2f337560c0</svTRID>
    </trID>
  </response>
</epp>
```

11.3.7 Polling

In case of an EPP Poll (op="req") request, the XML format of the messages concerning the result of the DNS validation which foresees the validation of the DS records other than the one of the nameservers, has been modified.
In particular, the formats of the two following messages have been modified:

- **DNS check ended unsuccessfully**, message of DNS check ended with failure;
- **DNS check ended successfully with warning**, message of DNS check ended successfully but with warning.

The XML format of the above mentioned messages has been modified by adding, in the <extension> section, the **<extsecDNS:secDnsErrorMsgData>** element. This element appears after the **<extdom:dnsErrorMsgData>** element in the case that the message is "DNS check ended unsuccessfully" and after the <extdom:dnsWarningMsgData> element in the case that the message is "DNS check ended successfully with warning". In the following two examples, the

<extsecDNS:secDnsErrorMsgData> element is reported.

The following example shows the case where the "alg" field value of the DS record submitted via a Domain Create/Update command is different from the corresponding value in the KSK ("Key Signing Key"):

```
…
<extsecDNS:secDnsErrorMsgData>
  <extsecDNS:dsOrKeys>
    <secDNS:dsData>
      <secDNS:keyTag>33138</secDNS:keyTag>
      <secDNS:alg>8</secDNS:alg>
      <secDNS:digestType>2</secDNS:digestType>
      <secDNS:digest>
      516DE15A6897FE168F34FBE7E084B852D78321A89542A28C4DE19A6C696C0C97
      </secDNS:digest>
    </secDNS:dsData>
  </extsecDNS:dsOrKeys>
  <extsecDNS:tests>
    <extsecDNS:test name="DNSKEYQueryAnswerTest" status="SUCCEEDED">
    …
    </extsecDNS:test>
    <extsecDNS:test name="DSRecordValidationTest" status="FAILED">
      <extsecDNS:nameserver name="ns1.pubtest.nic.it." status="FAILED">
        <extsecDNS:detail status="FAILED" queryId="11">Algorithm mismatch for Key-
id '33138': KSK algorithm '10' doesn't match with DS algorithm '8'</extsecDNS:detail>
      </extsecDNS:nameserver>
      <extsecDNS:nameserver name="ns2.pubtest.nic.it." status="FAILED">
        <extsecDNS:detail status="FAILED" queryId="12">Algorithm mismatch for Key-
id '33138': KSK algorithm '10' doesn't match with DS algorithm '8'</extsecDNS:detail>
      </extsecDNS:nameserver>
    </extsecDNS:test>
    <extsecDNS:test name="DNSKEYSignatureValidationTest" status="SUCCEEDED">
    …
    </extsecDNS:test>
    <extsecDNS:test name="SOASignatureValidationTest" status="SUCCEEDED">
    …
    </extsecDNS:test>
    <extsecDNS:test name="NSSignatureValidationTest" status="SUCCEEDED">
    …
    </extsecDNS:test>
  </extsecDNS:tests>
  <extsecDNS:queries>
  …
  </extsecDNS:queries>
</extsecDNS:secDnsErrorMsgData>
…
```

The following example shows the case where the digest value in the DS record submitted via Domain Create/Update is different from the corresponding value generated by the KSK ("Key Signing Key"):

```
…
<extsecDNS:secDnsErrorMsgData>
  <extsecDNS:dsOrKeys>
    <secDNS:dsData>
      <secDNS:keyTag>55416</secDNS:keyTag>
      <secDNS:alg>10</secDNS:alg>
      <secDNS:digestType>2</secDNS:digestType>
      <secDNS:digest>
```

```
      516DE15A6897FE168F34FBE7E084B852D78321A89542A28C4DE19A6C696C0C97
      </secDNS:digest>
    </secDNS:dsData>
  </extsecDNS:dsOrKeys>
  <extsecDNS:tests>
    <extsecDNS:test name="DNSKEYQueryAnswerTest" status="SUCCEEDED">
    …
    </extsecDNS:test>
    <extsecDNS:test name="DSRecordValidationTest" status="FAILED">
      <extsecDNS:nameserver name="ns1.pubtest.nic.it." status="FAILED">
        <extsecDNS:detail status="FAILED" queryId="11">Digest mismatch for Key-id
'55416': digest generated from KSK
'9D2E9A5DA30CBE4606DDC7B450F70243BFD18A68ECC53D112A6F9915617B2556' is different
from DS record digest
'516DE15A6897FE168F34FBE7E084B852D78321A89542A28C4DE19A6C696C0C97'</extsecDNS:deta
il>
      </extsecDNS:nameserver>
      <extsecDNS:nameserver name="ns2.pubtest.nic.it." status="FAILED">
        <extsecDNS:detail status="FAILED" queryId="12">Digest mismatch for Key-id
'55416': digest generated from KSK
'9D2E9A5DA30CBE4606DDC7B450F70243BFD18A68ECC53D112A6F9915617B2556' is different
from DS record digest
'516DE15A6897FE168F34FBE7E084B852D78321A89542A28C4DE19A6C696C0C97'</extsecDNS:deta
il>
      </extsecDNS:nameserver>
    </extsecDNS:test>
    <extsecDNS:test name="DNSKEYSignatureValidationTest" status="SUCCEEDED">
    …
    </extsecDNS:test>
    <extsecDNS:test name="SOASignatureValidationTest" status="SUCCEEDED">
    …
    </extsecDNS:test>
    <extsecDNS:test name="NSSignatureValidationTest" status="SUCCEEDED">
    …
    </extsecDNS:test>
  </extsecDNS:tests>
  <extsecDNS:queries>
  …
  </extsecDNS:queries>
</extsecDNS:secDnsErrorMsgData>
…
```

If, in the EPP Login request, the "DNSSEC accredited" Registrar has not indicated the two namespaces related to the DNSSEC extensions and they wish to examine, through an EPP Poll (op="req") request, a message containing the extension concerning the extsecDNS-1.0 namespace, they will receive a message of error.

11.4 Checks performed by the EPP server on DS records

The system verifies that the requests are compatible with:

- the constraints present in the XML Schema secDNS-1.1;
- the system verifies the following additional restrictions:
  - the Login request must contain also the following two namespaces:
    - urn:ietf:params:xml:ns:secDNS-1.1;
    - http://www.nic.it/ITNIC-EPP/extsecDNS-1.0;

- the value of the keyTag field of a <secDNS:dsData> element must be in the interval from 0 to 65535;
- the value of the digestType field of a <secDNS:dsData> element must fulfil what is indicated in Section 11.1;
- the value of the alg field of a <secDNS:dsData> element must fulfil what is indicated in Section 11.1;
- the length of the value of the digest field of a <secDNS:dsData> element must be compatible with the digest type;
- there must not be 2 <secDNS:dsData> elements containing the same values for the foreseen 4 values;
- in the Domain Create request:
  - the <secDNS:create> element must not contain the <secDNS:maxSigLife> element;
  - the <secDNS:create> element must not contain a <secDNS:keyData> element in place of or within a <secDNS:dsData> element;
  - the number of DS records per domain name after a Domain Create must not exceed the maximum limit specified in the MAX_DS_IN_CREATE parameter described in the table included in Section 12.8;
- in the Domain Update request:
  - the <secDNS:update> element must not contain the <secDNS:maxSigLife> element;
  - the <secDNS:update> element must not contain a <secDNS:keyData> element in place of or within a <secDNS:dsData> element;
  - the <secDNS:update> element must not contain the urgent attribute;
  - the <secDNS:update> element must contain the <secDNS:add> element and/or the <secDNS:rem> element;
  - the <secDNS:rem> element must contain single <secDNS:dsData> elements or the <secDNS:all> element;
  - the <secDNS:add> element must not contain a DS record already associated with the domain name;
  - the <secDNS:rem> element must not contain a DS record not yet associated with the domain name;
  - the number of DS records per domain name after a Domain Update must not exceed the maximum limit specified in the MAX_DS_IN_UPDATE parameter described in the table included in Section 12.8.

11.5 Validation of DNS configuration

The introduction of DNSSEC has also implications on the procedure of the DNS configuration validation. In fact, in case of “signed” domain names, it foresees further checks in addition to those already existing and described in Section 8.1.2.7.
In particular, it verifies that:

- the algorithm that appears in the DS record must be the same as the one which appears in the DNSKEY 257 record;
- the digest of the DS records indicated in the registration/modification of a domain name are congruent with the content of DNSKEY 257 record:
  - the above control is carried out for all declared authoritative name servers for the zone concerned;
- the digest of SOA record corresponds to the one indicated in the RRSIG SOA record:
  - the above control is carried out for all declared authoritative name servers for the zone concerned;
- the digest of the NS records corresponds to the one indicated in the RRSIG NS records:
  - the above control is carried out for all declared authoritative name servers for the zone concerned;
- the digest of the DNSKEY records corresponds to the one indicated in the RRSIG DNSKEY records:

- the above control is carried out for all declared authoritative name servers for the zone concerned;
- the signatures of the RRSIG records are not expired or in the future.

The list of the DNSSEC checks carried out for each host subjected to the DNS validation by the DNS validator of the Registry is the following:

- *DNSKEYQueryAnswerTest*: it verifies which nameservers have replied to the DNSKEY queries. The answer must be authoritative and containing a NOERROR return code.
- *DSRecordValidationTest*: it verifies the correctness of the DS records compared to the keys contained in the zone
- *DNSKEYSignatureValidationTest*: it verifies the correctness of the signature of the DNSKEY records
- *SOASignatureValidationTest*: it verifies the correctness of the signature of the SOA record
- *NSSignatureValidationTest*: it verifies the correctness of the signature of the NS records.

12 Further details about operations

12.1 Pending Actions

The EPP server of the Registry provides for three pending actions subsequent to a request being sent:

- Registering a domain name with DNS configuration validated
- Changing the DNS configuration of an existing domain name
- Changing the Registrar (with or without simultaneous change in the Registrant) of an existing domain name

There are no pending actions affecting transactions on the contacts.
The notification of completion of the pending (with success or failure) takes place via a message that the server includes in the "polling queue" (Section 12.4).
Other notifications sent by the server to the client regard the status switches of domain objects after delays.

12.2 Charging and Billing

The implementation of the registration system of the Registry stipulates that the following transactions on domain names shall be billed:

- Registration of a new domain name
- Change of Registrar (with or without the simultaneous change of the Registrant)
- Recovery from redemption period (following a request for deletion)
- Maintenance (automatic renewal of a domain name registered in the Registry Database)

The first three transactions are requests made by the Registrar through EPP, while the fourth is performed

by the Registry on the expiry of the period of validity of the domain name. In the event of a change of Registrar with a trade extension (change of the Registrar with the simultaneous change of the Registrant), the cost charged is only related to the change of the Registrar.
It is necessary to distinguish a charge for a transaction from its actual billing:

- **debit**, means the action of taking from the Registrar's credit, the cost of a transaction including any VAT. In order to keep the amount of the credit of the Registrar constantly updated, the transactions listed above are immediately debited.
- **invoicing/billing**, this means reporting such costs in an invoice sent to the Registrar. The invoice contains all the transactions carried out by the Registrar in respect of a particular payment.

This separation is necessary since billing may not take place at the same time as charging. For example, for changes of the Registrar (case 2) and automatic renewal (case 4), charging and billing are not simultaneous:

- **automatic renewal**: if the domain name is in a status that permits it, the debit takes place at the expiry of the domain name. The billing, however, takes place on the expiry of the grace period - autoRenewPeriod. If during this period, the domain name is transferred to another Registrar or deleted, the cost of renewal is re-accredited to the Registrar and the renewal is not invoiced.
- **change of Registrar**: the debit of the transaction takes place at the time of the request to change the Registrar. Invoicing only occurs when the transaction is successful. If the transaction fails, the cost of the change of Registrar will be re-credited to the Registrar that made the request, and the transaction is not invoiced.

In cases 1 and 3 charging and billing take place at the same time.

12.3 Time Periods

For some status switches and for the management of certain information regarding the objects registered in the Registry Database (registrars, contacts, domains), the Registry's synchronous server refers to specific time periods which are listed below:

Name	Meaning	Unit	Value
PENDING_UPDATE_PERIOD	Maximum period in pendingUpdate	dd	5
REDEMPTION_PERIOD	Maximum period in pendingDelete/redemption period	dd	30
PENDING_TRANSFER_PERIOD	Maximum period in pendingTransfer	dd	1
AUTO_RENEW_PERIOD	Grace period after domain name automatical renewal	dd	15
PENDING_DELETE_PERIOD	Maximum period in pendingDelete/pendingDelete established by the Drop Time process that provides the cancellation of the domain names at scheduled times	-	-
CHALLENGED_PERIOD	Maximum period in *challenged*	dd	180
REVOKED_PERIOD	Maximum period in inactive/*revoked*	dd	30
TO_BE_REASSIGNED_PERIOD	Maximum period in inactive/*toBeReassigned*	dd	30

NO_REGISTRAR_PERIOD	Maximum period in inactive/*noRegistrar*	dd	60
NOT_RENEWED_PERIOD	Maximum period in inactive/*notRenewed*	dd	30
UNLINKED_CONTACT_PERIOD	Maximum user disconnection time length	dd	60
BILLING_LOW_CREDIT_WARNING_PERIOD	Number of days taken into account by the system to calculate the credit threshold of warning required to maintain their own domain names.	dd	4
PASSWD_VALIDITY_PERIOD	Password validity period	dd	180
PASSWD_REMINDER_PERIOD	Time period to calculate the date when the server will notify the client of imminent expiry of password	dd	15
OLD_MESSAGE_PERIOD	Maximum message stay period in the polling queue	dd	60
RECENTLY_DELETED_DOMAINS_PERIOD	Minimum period during which it is not allowed to send to the production server a Domain Create request for a domain that can be registered once more after its deletion. The Domain Create request must be sent to a special server.	dd	7
ACCREDITATION_TEST_PERIOD	Maximum period for executing the accreditation test	min	60

12.4 Polling queue

EPP requires that the server informs the client of all the events that occur offline, with respect to the normal "request-response".
The client receives a notification through the insertion of messages in the Registrar's polling queue, for which the protocol makes available two commands:

- **Poll Req**: for consulting the first message in the queue (the oldest);
- **Poll Ack**: for removing a message identified by a specific ID from the queue.

The synchronous system server implements two classes of messages:

- the first concerns the Registrar i.e. those messages that relate to the authentication or the Registrar's credit;
- the second concerns domain names of a Registrar i.e. those messages that relate to actions started, currently under way, or completed on a particular domain name.

12.4.1 Messages concerning the Registrar

The table summarizes the messages that relate to the Registrar.

Event	**Message**
Registrar password is about to expire	Password will expire soon
The Registrar sent a Login Request that contains obsolete namespaces	Wrong namespace in Login Request

Current credit is under the personal threshold set by the Registrar	Credit is under the threshold set by the Registrar
Low credit: credit threshold of warning reached	The Registrar is in low credit
Credit ~ 0	Out of funds: only not invoiced operations are allowed

In case of events related to the Registrar credit, as shown in the previous table, the system not only inserts a message in the polling queue but also sends an e-mail to the billing contacts that the Registrar has created in the RAIN-NG portal.

12.4.2 Messages concerning the domain names of a Registrar

The table summarizes the correspondences between the events of the EPP server and messages posted in the polling queue.

Event	**Start status**	**Destination status**	**Message**
Successful registration of a IDN domain name containing *remapped* characters	**--**	**inactive/*dnsHold***	Requested IDN domain contains remapped chars
DNS check OK	**inactive/*dnsHold***	**ok**	DNS check ended successfully
			DNS check ended successfully with warning
DNS check KO	**inactive/*dnsHold***	**inactive/*dnsHold***	DNS check ended unsuccessfully
Reception of a Domain Update for change host and/or DS record	**ok**	**pendingUpdate**	pendingUpdate is started
DNS check OK	**pendingUpdate**	**ok** (new DNS configuration)	DNS check ended successfully
			DNS check ended successfully with warning
DNS check KO	**pendingUpdate**	**pendingUpdate**	DNS check ended unsuccessfully
Expiry pendingUpdate	**pendingUpdate**	**ok** (old DNS configuration)	pendingUpdate is expired
Reception of a Domain Delete	**ok**	**pendingDelete/redemptionPeriod**	redemptionPeriod is started
	ok/autoRenewPeriod		

	inactive/*dnsHold*		
	inactive/*dnsHold*/autoRenewPeriod		
Expiry redemptionPeriod	**pendingDelete/redemptionPeriod**	**pendingDelete/pendingDelete**	redemption Period is expired
	pendingDelete/ redemptionPeriod/ *challenged*	**inactive/*toBeReassigned***	
Expiry pendingDelete	**pendingDelete/pendingDelete**	**--**	Domain has been deleted (for the Registrar that manages the domain name)
			Lost delegation (for the Registrars that manage those domain names whose nameservers are subordinate to the cancelled domain name)
Reception of a Domain Transfer (op=request)	**ok**	**pendingTransfer**	Domain transfer has been requested: pendingTransfer is started (for the Registrar that manages the domain name)
	ok/*noRegistrar*		
	inactive/*noRegistrar*		
	inactive/*notRenewed*		
	inactive/*dnsHold*		
	inactive/*dnsHold*/*noRegistrar*		
Reception of a Domain Transfer (op=reject)	**pendingTransfer**	**inactive/*noRegistrar*** (if the reject operation was performed after the expiry of the auto renew period)	Domain transfer has been rejected (for the Registrar that requested the transfer)
		ok (if the transfer did not begin in auto renew period)	
		ok/*noRegistrar* (if the transfer did not begin in ok/*noRegistrar*)	
		inactive/*notRenewed* (if credit = 0)	

		inactive/*dnsHold* (if the transfer began in inactive/*dnsHold*)	
		inactive/*dnsHold*/*noRegistrar* (if the transfer began in inactive/*dnsHold/noRegistrar*)	
	pendingTransfer/autoRenewPeriod	**ok/autoRenewPeriod** (if the domain name has a valid DNS configuration)	
		inactive/*dnsHold*/autoRenewPeriod (if the domain name does not have a valid DNS configuration)	
Reception of a Domain Transfer (op=cancel)	**pendingTransfer**	**inactive/*noRegistrar*** (if the cancel operation was performed after the expiry of the auto renew period)	Domain transfer has been cancelled (for the Registrar that manages the domain name)
		ok (if the transfer did not begin in auto renew period)	
		ok/*noRegistrar* (if the transfer began in ok/*noRegistrar*)	
		inactive/*notRenewed* (if credit = 0)	
		inactive/*dnsHold* (if the transfer began in inactive/*dnsHold*)	
		inactive/*dnsHold*/*noRegistrar* (if the transfer began in inactive/*dnsHold/noRegistrar*)	
	pendingTransfer/autoRenewPeriod	**ok/autoRenewPeriod** (if the domain name has a valid DNS configuration)	
		inactive/*dnsHold*/autoRenewPeriod (if the domain name does not have a valid DNS configuration)	
Reception of a Domain Transfer (op=approve)	**pendingTransfer**	**ok** (if the domain name has a valid DNS configuration)	Domain transfer has been executed (for the Registrar that requested the transfer)

		inactive/*dnsHold* (if the domain name does not have a valid DNS configuration)	Domain transfer has been executed. You should therefore remove the records contained on your nameservers for such domain name (for the Registrar that manages the domain name)
Expiry pendingTransfer	**pendingTransfer**	**ok** (if the domain name has a valid DNS configuration)	Domain transfer is expired: transfer has been executed (for the Registrar that requested the transfer)
		inactive/*dnsHold* (if the domain name does not have a valid DNS configuration)	Domain transfer is expired: transfer has been executed. You should therefore remove the records contained on your nameservers for such domain name (for the Registrar that manages the domain name)
Reception of a Domain Transfer (op=request with ext. Trade)	**ok**	**pendingTransfer**	Domain and trade transfer has been requested: pendingTransfer is started (for the Registrar that manages the domain name)
	ok/*noRegistrar*		
	inactive/*noRegistrar*		
	inactive/*notRenewed*		
	inactive/*dnsHold*		
	inactive/*dnsHold*/*noRegistrar*		
Reception of a Domain Transfer (op=reject) for Domain and Trade Transfer	**pendingTransfer**	**inactive/*noRegistrar*** (if the reject operation was performed after the expiry of the auto renew period)	Domain and trade transfer has been rejected (for the Registrar that requested the transfer)
		ok (if the transfer did not begin in auto renew period)	

		ok/*noRegistrar* (if the transfer began in ok/*noRegistrar*)	
		inactive/*notRenewed* (if credit = 0)	
		inactive/*dnsHold* (if the transfer began in inactive/*dnsHold*)	
		inactive/*dnsHold*/*noRegistrar* (if the transfer began in inactive/*dnsHold/noRegistrar*)	
	pendingTransfer/autoRenewPeriod	**ok/autoRenewPeriod** (if the domain name has a valid DNS configuration)	
		inactive/*dnsHold*/autoRenewPeriod (if the domain name does not have a valid DNS configuration)	
Reception of a Domain Transfer (op=cancel) for Domain and Trade Transfer	**pendingTransfer**	**inactive/*noRegistrar*** (if the cancel operation was performed after the expiry of the auto renew period)	Domain and trade transfer has been cancelled (for the Registrar that manages the domain name)
		ok (if the transfer did not begin in auto renew period)	
		ok/*noRegistrar* (if the transfer began in ok/*noRegistrar*)	
		inactive/*notRenewed* (if credit = 0)	
		inactive/*dnsHold* (if the transfer began in inactive/*dnsHold*)	
		inactive/*dnsHold*/*noRegistrar* (if the transfer began in inactive/*dnsHold/noRegistrar*)	
	pendingTransfer/autoRenewPeriod	**ok/autoRenewPeriod** (if the domain name has a valid DNS configuration)	

<table>
<tr><td></td><td></td><td colspan="2">inactive/dnsHold/autoRenewPeriod
(if the domain name does not have a valid DNS configuration)</td><td></td></tr>
<tr><td rowspan="2">Reception of a Domain Transfer (op=approve) for Domain and Trade Transfer</td><td rowspan="2">pendingTransfer</td><td colspan="2">ok
(if the domain name has a valid DNS configuration)</td><td>Domain and trade transfer has been executed
(for the Registrar that requested the transfer)</td></tr>
<tr><td colspan="2">inactive/dnsHold
(if the domain name does not have a valid DNS configuration)</td><td>Domain and trade transfer has been executed. You should therefore remove the records contained on your nameservers for such domain name
(for the Registrar that manages the domain name)</td></tr>
<tr><td rowspan="2">Expiry pendingTransfer for Domain and Trade Transfer</td><td rowspan="2">pendingTransfer</td><td colspan="2" rowspan="2">ok
(if the domain name has a valid DNS configuration)</td><td>Domain and trade transfer is expired: transfer has been executed
(for the Registrar that requested the transfer)</td></tr>
<tr><td rowspan="2">Domain and trade transfer is expired: transfer has been executed. You should therefore remove the records contained on your nameservers for such domain name
(for the Registrar that manages the domain name)</td></tr>
<tr><td colspan="2">inactive/dnsHold
(if the domain name does not have a valid DNS configuration)</td></tr>
<tr><td rowspan="5">Expiry domain name and credit > 0</td><td>ok</td><td>ok</td><td rowspan="4">auto Renew Period</td><td rowspan="5">autoRenewPeriod is started</td></tr>
<tr><td>pendingUpdate</td><td>pendingUpdate</td></tr>
<tr><td>pendingTransfer</td><td>pendingTransfer</td></tr>
<tr><td>inactive/dnsHold</td><td>inactive/dnsHold</td></tr>
<tr><td>pendingDelete/redemptionPeriod</td><td colspan="2">pendingDelete/redemptionPeriod</td></tr>
</table>

Expiry domain name and credit = 0	**ok**		**inactive/*notRenewed***	Not Renewed is started
Expiry autoRenewPeriod	**ok**	***auto Renew Period***	**ok**	autoRenewPeriod is expired
	pendingUpdate		**pendingUpdate**	
	pendingTransfer		**pendingTransfer**	
	inactive/*dnsHold*		**inactive/*dnsHold***	
	pendingDelete/redemptionPeriod		**pendingDelete/redemptionPeriod**	
Expiry toBeReassigned	**inactive/*toBeReassigned***		**pendingDelete/pendingDelete**	Reassignation is expired
Expiry revoked	**inactive/*revoked***		**pendingDelete/pendingDelete**	Revoke is expired
	inactive/*revoked/challenged*		**inactive/*toBeReassigned***	
Expiry notRenewed	**inactive/*notRenewed***		**pendingDelete/pendingDelete**	Not Renewed is expired
	inactive/*notRenewed/challenged*		**inactive/*toBeReassigned***	
Registrar is not active	**--**		**ok/*noRegistrar***	No Registrar is started
			inactive/dnsHold/noRegistrar	
Reception of a Domain Transfer (op=reject) for Domain and Trade Transfer *(if the transfer began in autoRenewPeriod and the domain name is not in autoRenewPeriod)*	**pendingTransfer**		**inactive/*noRegistrar***	No Registrar is started
Expiry noRegistrar	**inactive/*noRegistrar***		**pendingDelete/pendingDelete**	No Registrar is expired
	inactive/*noRegistrar/challenged*		**inactive/*toBeReassigned***	
Credit refunded	**inactive/*notRenewed***		**ok** *(if the domain name is not in auto renew period* and has a valid DNS configuration*)*	Not Renewed is ended

<table>
<tr><td></td><td></td><td colspan="2">ok/autoRenewPeriod
(if the domain name is in auto renew period and has a valid DNS configuration)</td><td></td></tr>
<tr><td></td><td></td><td colspan="2">inactive/dnsHold
(if the domain name is not in auto renew period and does not have a valid DNS configuration)</td><td></td></tr>
<tr><td></td><td></td><td colspan="2">inactive/dnsHold/autoRenewPeriod
(if the domain name is in auto renew period and does not have a valid DNS configuration)</td><td></td></tr>
<tr><td rowspan="8">Revocation by the Registry</td><td>ok</td><td colspan="2" rowspan="8">inactive/revoked</td><td rowspan="8">Revoke is started</td></tr>
<tr><td>inactive/dnsHold</td></tr>
<tr><td>pendingTransfer</td></tr>
<tr><td>pendingUpdate</td></tr>
<tr><td>pendingDelete/redemptionPeriod</td></tr>
<tr><td>ok/noRegistrar</td></tr>
<tr><td>inactive/noRegistrar</td></tr>
<tr><td>inactive/dnsHold/noRegistrar</td></tr>
<tr><td>inactive/notRenewed</td></tr>
<tr><td rowspan="8">Reception of a challenge request</td><td>ok</td><td>ok</td><td rowspan="8">challenged</td><td rowspan="8">Challenge procedure is started</td></tr>
<tr><td>inactive/dnsHold</td><td>inactive/dnsHold</td></tr>
<tr><td>pendingTransfer</td><td>pendingTransfer</td></tr>
<tr><td>pendingUpdate</td><td>pendingUpdate</td></tr>
<tr><td>pendingDelete/ redemptionPeriod</td><td>pendingDelete/ redemptionPeriod</td></tr>
<tr><td>ok/noRegistrar</td><td>ok/noRegistrar</td></tr>
<tr><td>inactive/dnsHold/noRegistrar</td><td>inactive/dnsHold/noRegistrar</td></tr>
<tr><td>inactive/noRegistrar</td><td>inactive/noRegistrar</td></tr>
<tr><td rowspan="5">Challenge procedure terminated</td><td>ok</td><td rowspan="5">challenged</td><td colspan="2">ok</td><td rowspan="5">Challenge procedure is ended</td></tr>
<tr><td>inactive/dnsHold</td><td colspan="2">inactive/dnsHold</td></tr>
<tr><td>pendingTransfer</td><td colspan="2">pendingTransfer</td></tr>
<tr><td>pendingUpdate</td><td colspan="2">pendingUpdate</td></tr>
<tr><td>pendingDelete/ redemptionPeriod</td><td colspan="2">pendingDelete/ redemptionPeriod</td></tr>
</table>

	ok/*noRegistrar*		**ok/*noRegistrar***	
	inactive/*dnsHold/noRegistrar*		**inactive/*dnsHold/noRegistrar***	
	inactive/*noRegistrar*		**inactive/*noRegistrar***	
	inactive/ *notRenewed*		**inactive/ *notRenewed***	
Reception of a hold request from a Registrant	**ok**		**inactive/serverHold**	Hold by Registrant is started
Removal of hold by a Registrant	**inactive/serverHold**		**ok**	Hold by Registrant is ended
Reception of a hold request from a third party	**ok**		**inactive/serverHold**	Hold by third party is started
Removal of hold by a third party	**inactive/serverHold**		**ok**	Hold by third party is ended
Domain name is put in hold by the Registry	**ok**		**inactive/serverHold**	Hold by server is started
Removal of hold by the Registry	**inactive/serverHold**		**ok**	Hold by server is ended
Reception of a lock request from a Registrant	**ok**		**serverUpdateProhibited/serverDeleteProhibited/serverTransferProhibited**	Lock by Registrant is started
Removal of lock by a Registrant	**serverUpdateProhibited/serverDeleteProhibited/serverTransferProhibited**		**ok**	Lock by Registrant is ended
Reception of a lock request from a third party	**ok**		**serverUpdateProhibited/serverDeleteProhibited/serverTransferProhibited**	Lock by third party is started
Removal of lock by a third party	**serverUpdateProhibited/serverDeleteProhibited/serverTransferProhibited**		**ok**	Lock by third party is ended
Domain is put in lock by the Registry	**ok**		**serverUpdateProhibited/serverDeleteProhibited/serverTransferProhibited**	Lock by server is started

Removal of lock by the Registry	**serverUpdateProhibited/serverDeleteProhibited/serverTransferProhibited**	**ok**	Lock by server is ended
Reception of a Domain Delete	**ok/autoRenewPeriod**	**pendingDelete/redemptionPeriod**	Refund renew for deleting domain in autoRenewPeriod
	inactive/*dnsHold*/autoRenewPeriod		
Cancellation by the Registry	**-/autoRenewPeriod**	**-**	Refund renew for deleting domain by Registry in autoRenewPeriod
Revocation by the Registry	**-/autoRenewPeriod**	**-**	Refund renew for revoking domain in autoRenewPeriod
Reception of a Domain Transfer (op=approve) for Domain Transfer	**pendingTransfer/autoRenewPeriod**	**ok**	Refund renew for transferring domain to a registrar in autoRenewPeriod
Expiry pendingTransfer for Domain Transfer		**inactive/*dnsHold***	
Reception of a Domain Transfer (op=approve) for Domain and Trade Transfer	**pendingTransfer/autoRenewPeriod**	**ok**	Refund renew for transferring and trading domain to a registrar in autoRenewPeriod
Expiry pendingTransfer for Domain and Trade Transfer		**inactive/*dnsHold***	
Reception of a Domain Transfer (op=approve) for Domain and Trade Transfer	**ok/autoRenewPeriod**	**ok**	Refund renew for transferring and trading domain to a registrar in autoRenewPeriod
Expiry pendingTransfer for Domain and Trade Transfer			
Expiry of autoRenewPeriod during pendingTransfer	**pendingTransfer/autoRenewPeriod**	**pendingTransfer**	Refund renew for autoRenewPeriod expired during pendingTransfer
	pendingTransfer	**ok**	

Reception of a Domain Transfer (op=reject) for Domain Transfer		**inactive/*dnsHold***	Refund domain transfer to a registrar for rejecting transfer
Reception of a Domain Transfer (op=reject) for Domain and Trade Transfer	**pendingTransfer**	**ok**	Refund domain transfer and trade to a registrar for rejecting transfer
		inactive/*dnsHold*	
Reception of a Domain Transfer (op=cancel) for Domain Transfer	**pendingTransfer**	**ok**	Refund domain transfer to a registrar for cancelling transfer
		inactive/*dnsHold*	
Reception of a Domain Transfer (op=cancel) for Domain and Trade Transfer	**pendingTransfer**	**ok**	Refund domain transfer and trade to a registrar for cancelling transfer
		inactive/*dnsHold*	
Successful end of a bulk transfer (one only message for all the domains transferred in auto renew period)		**ok**	Refund renews for bulk transferring domains in autoRenewPeriod
		inactive/dnsHold	
Reception of a Restore Domain for expired domain name in noRegistrar	**inactive/*noRegistrar***	**ok** (if the domain name is not in auto renew period and has a valid DNS configuration)	Debit renew for restoring expired domain in inactive/*noRegistrar*
		ok/autoRenewPeriod (if the domain name is in auto renew period and has a valid DNS configuration)	
		inactive/*dnsHold* (if the domain name is not in auto renew period and does not have a valid DNS configuration)	
		inactive/*dnsHold*/autoRenewPeriod (if the domain name is in auto renew period and does not have a valid DNS configuration)	
Reception of a Restore Domain for expired and	**pendingDelete/redemptionPeriod**	**ok** (if the domain name is not in auto renew period and has a valid DNS configuration)	Debit renew for restoring expired domain in

cancelled domain name		**ok/autoRenewPeriod** (if the domain name is in auto renew period and has a valid DNS configuration)	pendingDelete/redemptionPeriod
		inactive/*dnsHold* (if the domain name is not in auto renew period and does not have a valid DNS configuration)	
		inactive/*dnsHold*/autoRenewPeriod (if the domain name is in auto renew period and does not have a valid DNS configuration)	

12.4.3 Correspondence between messages and XML Schema

Below is the correspondence between a message, XML Schema and type within the schema.

Message	**XML Schema**	**Type**
Password will expire soon	extepp-2.0.xsd	extepp:passwdReminder
Wrong namespace in Login Request	extepp-2.0.xsd	extepp:wrongNamespaceReminder
Credit is under the threshold set by the Registrar	extepp-2.0.xsd	extepp:creditMsgData
Normal balance: all operations are allowed	extepp-2.0.xsd	extepp:creditMsgData
The Registrar is in low credit	extepp-2.0.xsd	extepp:creditMsgData
Out of funds: only not invoiced operations are allowed	extepp-2.0.xsd	extepp:creditMsgData
Requested IDN domain contains remapped chars	extdom-2.0.xsd	extdom: remappedIdnData
DNS check ended unsuccessfully	extdom-2.0.xsd	extdom:dnsErrorMsgData
DNS check ended successfully	extdom-2.0.xsd	extdom:chgStatusMsgData
DNS check ended successfully with warning	extdom-2.0.xsd	extdom:dnsWarningMsgData
pendingUpdate is started	extdom-2.0.xsd	extdom:chgStatusMsgData
pendingUpdate is expired	extdom-2.0.xsd	extdom:chgStatusMsgData
redemptionPeriod is started	extdom-2.0.xsd	extdom:chgStatusMsgData
redemptionPeriod is expired	extdom-2.0.xsd	extdom:chgStatusMsgData
pendingDelete is started	extdom-2.0.xsd	extdom:chgStatusMsgData
Domain has been deleted	extdom-2.0.xsd	extdom:simpleMsgData
Lost delegation	extdom-2.0.xsd	extdom: dlgMsgData
autoRenewPeriod is started	extdom-2.0.xsd	extdom:chgStatusMsgData
autoRenewPeriod is expired	extdom-2.0.xsd	extdom:chgStatusMsgData
Revoke is started	extdom-2.0.xsd	extdom:chgStatusMsgData

Revoke is expired	extdom-2.0.xsd	extdom:chgStatusMsgData
No Registrar is started	extdom-2.0.xsd	extdom:chgStatusMsgData
No Registrar is expired	extdom-2.0.xsd	extdom:chgStatusMsgData
Reassignation is expired	extdom-2.0.xsd	extdom:chgStatusMsgData
Not Renewed is started	extdom-2.0.xsd	extdom:chgStatusMsgData
Not Renewed is ended	extdom-2.0.xsd	extdom:chgStatusMsgData
Not Renewed is expired	extdom-2.0.xsd	extdom:chgStatusMsgData
Challenge procedure is started	extdom-2.0.xsd	extdom:chgStatusMsgData
Challenge procedure is ended	extdom-2.0.xsd	extdom:chgStatusMsgData
Hold by registrant is started	extdom-2.0.xsd	extdom:chgStatusMsgData
Hold by third party is started	extdom-2.0.xsd	extdom:chgStatusMsgData
Hold by server is started	extdom-2.0.xsd	extdom:chgStatusMsgData
Hold by registrant is ended	extdom-2.0.xsd	extdom:chgStatusMsgData
Hold by third party is ended	extdom-2.0.xsd	extdom:chgStatusMsgData
Hold by server is ended	extdom-2.0.xsd	extdom:chgStatusMsgData
Lock by registrant is started	extdom-2.0.xsd	extdom:chgStatusMsgData
Lock by third party is started	extdom-2.0.xsd	extdom:chgStatusMsgData
Lock by server is started	extdom-2.0.xsd	extdom:chgStatusMsgData
Lock by registrant is ended	extdom-2.0.xsd	extdom:chgStatusMsgData
Lock by third party is ended	extdom-2.0.xsd	extdom:chgStatusMsgData
Lock by server is ended	extdom-2.0.xsd	extdom:chgStatusMsgData
Domain transfer has been requested: pendingTransfer is started	domain-1.0.xsd	domain:trnData
Domain transfer has been rejected	domain-1.0.xsd	domain:trnData
Domain transfer has been cancelled	domain-1.0.xsd	domain:trnData
Domain transfer has been executed	domain-1.0.xsd	domain:trnData
Domain transfer has been executed. You should therefore remove the records contained on your nameservers for such domain name.	domain-1.0.xsd	domain:trnData
Domain transfer is expired: transfer has been executed	domain-1.0.xsd	domain:trnData
Domain transfer is expired: transfer has been executed. You should therefore remove the records contained on your nameservers for such domain name.	domain-1.0.xsd	domain:trnData
Domain and trade transfer has been requested: pendingTransfer is started	domain-1.0.xsd extdom-2.0.xsd	domain:trnData extdom:trade
Domain and trade transfer has been rejected	domain-1.0.xsd extdom-2.0.xsd	domain:trnData extdom:trade
Domain and trade transfer has been cancelled	domain-1.0.xsd extdom-2.0.xsd	domain:trnData extdom:trade

Domain and trade transfer has been executed	domain-1.0.xsd extdom-2.0.xsd	domain:trnData extdom:trade
Domain and trade transfer has been executed. You should therefore remove the records contained on your nameservers for such domain name.	domain-1.0.xsd extdom-2.0.xsd	domain:trnData extdom:trade
Domain and trade transfer is expired: transfer has been executed	domain-1.0.xsd extdom-2.0.xsd	domain:trnData extdom:trade
Domain and trade transfer is expired: transfer has been executed. You should therefore remove the records contained on your nameservers for such domain name.	domain-1.0.xsd extdom-2.0.xsd	domain:trnData extdom:trade
Refund renew for deleting domain in autoRenewPeriod	extdom-2.0.xsd	extdom:delayedDebitAndRefundMsgData
Refund renew for transferring domain to a registrar in autoRenewPeriod	extdom-2.0.xsd	extdom:delayedDebitAndRefundMsgData
Refund renew for transferring and trading domain to a registrar in autoRenewPeriod	extdom-2.0.xsd	extdom:delayedDebitAndRefundMsgData
Refund renew for autoRenewPeriod expired during pendingTransfer	extdom-2.0.xsd	extdom:delayedDebitAndRefundMsgData
Refund domain transfer to a registrar for rejecting transfer	extdom-2.0.xsd	extdom:delayedDebitAndRefundMsgData
Refund domain transfer and trade to a registrar for rejecting transfer	extdom-2.0.xsd	extdom:delayedDebitAndRefundMsgData
Refund domain transfer to a registrar for cancelling transfer	extdom-2.0.xsd	extdom:delayedDebitAndRefundMsgData
Refund domain transfer and trade to a registrar for cancelling transfer	extdom-2.0.xsd	extdom:delayedDebitAndRefundMsgData
Refund renew for revoking domain in autoRenewPeriod	extdom-2.0.xsd	extdom:delayedDebitAndRefundMsgData
Refund renew for deleting domain by Registry in autoRenewPeriod	extdom-2.0.xsd	extdom:delayedDebitAndRefundMsgData
Refund renews for bulk transferring domains in autoRenewPeriod	extdom-2.0.xsd	extdom: refundRenewsForBulkTransferMsgData
Debit renew for restoring expired domain in inactive/*noRegistrar*	extdom-2.0.xsd	extdom:delayedDebitAndRefundMsgData
Debit renew for restoring expired domain in pendingDelete/redemptionPeriod	extdom-2.0.xsd	extdom:delayedDebitAndRefundMsgData

12.5 Emails to the Registrant

The EPP server sends an email directly to the Registrant in the following cases:

- Registration of a domain name completed successfully
- Change of Registrant completed successfully (the mail is sent to both the old and the new Registrant)
- Change of the Registrar with the simultaneous change of the Registrant completed successfully (the mail is sent to both the old and the new Registrant)
- Change of the domain name to “noRegistrar” status

12.6 Return codes and reasons for errors

To be able to interact efficiently with the EPP server, it is necessary that the Registrar is aware of the return codes of the standard EPP and the reasons for rejection, which further specify return codes and relate to the EPP server.
The return codes are set out in Appendix C and the reasons in Appendix D.

12.7 Format of dates

For the "Date" fields of the contact and domain objects, EPP provides two formats:

- one that expresses the date in CUT (Coordinated Universal Time);
- one that expresses the local date by adding the difference in hours (offset), positive or negative, compared to the CUT.

The implementation of the Registry's synchronous system uses the second format:

yyyy-mm-dd 'T' hh:mm:ss+<offset>

where <offset> may take the following values:

- +01:00 - when DST is not applied
- +02:00 - when DST is applied

Example: 2008-07-07T15:13:18+02:00

12.8 Other useful parameters

The following table shows other parameters that may be useful for clients when interacting with the Registry's synchronous server.

Name	Meaning	Value
MIN_IP	Minimum number of IP addresses for each name server subordinate to the domain name (one Ipv4 address)	1
MAX_IP	Maximum number of IP addresses for each name server subordinate to the domain name (one Ipv4 address and one Ipv6 address)	2
MIN_NS	Minimum number of name servers for each domain name	2
MAX_NS	Maximum number of name servers for each domain name	6
MAX_DS_IN_CREATE	Maximum number of DS records for each domain name after a Domain Create	1
MAX_DS_IN_UPDATE	Maximum number of DS records for each domain name after a Domain Update	2
MIN_CONTACT_TECH	Minimum number of tech contacts for each domain name	1
MAX_CONTACT_TECH	Maximum number of tech contacts for each domain name	6

MIN_CONTACT_ADMIN	Minimum number of admin contacts for each domain name	1
MAX_CONTACT_ADMIN	Maximum number of admin contacts for each domain name	1
MAX_CMD	Maximum number of Domain Check commands for Registrar that can be sent daily to the epp.nic.it server	20000
	Maximum number of Domain Check commands for Registrar that can be sent daily to the epp-deleted.nic.it server Maximum number of Domain Create commands for Registrar that can be sent daily to the epp-deleted.nic.it server	3000
MAX_CHECK	Maximum number of referred domain names/contacts for each Domain Check/Contact Check command	5
MIN_PWAUTHINFO_LENGTH	Minimum length of the AuthInfo associated with the domain name	8
MAX_PWAUTHINFO_LENGTH	Maximum length of the AuthInfo associated with the domain name	32
MIN_PWUSER_LENGTH	Minimum length of the user password	6
MIN_NEWPWUSER_LENGTH	Minimum length of new user password	8
MAX_PWUSER_LENGTH	Maximum length of the user password	16
USER_SESSIONS_LIMIT	Maximum number of contemporary active sessions for each Registrar	5
USER_CONNECTIONS_LIMIT	Maximum number of TCP connections simultaneously opened per IP address	3
USER_SESSION_TIMEOUT	Session timeout in minutes	5
USER_IP_ADDRESSES	Maximum number of static IP addresses of clients per Registrar	5

13 Securing the EPP servers and the information

Being the ccTLD .it an organization providing a strategic service for the Italian Internet community, the Italian Government has declared the ccTLD .it to be Operator of Essential Service (OES) as defined in the EU Network and Information Security directive [EU-NIS]. Therefore, the ccTLD .it has been required to implement all the measures needed to boost the overall level of cybersecurity.
To secure the EPP servers and the information, the following measures have been implemented:

- the EPP servers cannot be accessed by everyone. Before submitting the first EPP request, the Registrars are required to declare on the RAIN-NG portal the IP addresses (max 5) of the machines hosting their EPP clients. The complete list of incoming IP addresses is used to define an IPTABLES rule on the NGINX proxies to reject connection requests coming from unknown origins;
- the EPP commands are issued on a secure channel, i.e. HTTPS, therefore, they cannot be sniffed;
- the Registrars can start an EPP session after being authenticated through a Login request;
- before processing every EPP request other than Login, the server checks that the IP address of the client is the same as the client that has previously started the session;
- each pool of real EPP servers can be accessed only from the corresponding NGINX proxies;
- the EPP servers use 2 different connections to the MySQL cluster: one towards the master node allowing both read and write SQL commands and the other towards the read-only nodes allowing only queries. The former is used by the EPP transform commands, the latter is used by the EPP query commands;
- the MySQL cluster can be accessed only from the internal network of ccTLD .it and, in particular, only from those servers hosting applications that need to interact with the DBAN;
- the Registrar' password as well as the domain AuthInfo are stored after being encrypted and their values are redacted in the logs. Therefore, they cannot be uncovered by the MySQL administrator or any application accessing the DBAN;
- some limits have been set to mitigate the risk of DoS attacks and resource exhaustion:
  - maximum number of simultaneous TCP connections;
  - maximum number of simultaneous EPP sessions;
  - maximum time an unused session can live;
- the EPP servers are constantly monitored through Nagios and every suspected activity is signaled to the system administrators;
- the EPP servers are periodically subjected to both vulnerability assessment and penetration testing.

14 References

[EU-NIS] DIRECTIVE (EU) 2016/1148 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL OF 6 JULY 2016 CONCERNING MEASURES FOR A HIGH COMMON LEVEL OF SECURITY OF NETWORK AND INFORMATION SYSTEMS ACROSS THE UNION, <HTTPS://EUR-LEX.EUROPA.EU/LEGAL-CONTENT/EN/TXT/PDF/?URI=CELEX:32016L1148&FROM=EN>

[RFC1035] Mockapetris P.: Domain Names – Implementation and Specification, RFC1035, <https://www.rfc-editor.org/info/rfc1035>

[RFC3375] Hollenbeck S.: Generic Registry-Registrar Protocol Requirements, RFC3375, <https://www.rfc-editor.org/info/rfc3375>

[RFC3735] Hollenbeck S.: Guidelines for Extending the Extensible Provisioning Protocol (EPP), RFC3735, <https://www.rfc-editor.org/info/rfc3735>

[RFC3915] Hollenbeck S.: Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP), RFC3915, <https://www.rfc-editor.org/info/rfc3915>

[RFC4033] Arends R., Austein R., Larson M., Massey D., Rose S.: DNS Security Introduction and Requirements, RFC4033, <https://www.rfc-editor.org/info/rfc4033>

[RFC4641] Kolkman O., Gieben R.: DNSSEC Operational Practices, RFC4641, <https://www.rfc-editor.org/info/rfc4641>

[RFC5322] Resnick P.: Internet Message Format, RFC5322, <https://www.rfc-editor.org/info/rfc5322>

[RFC5730] Hollenbeck S.: Extensible Provisioning Protocol (EPP), RFC5730, <https://www.rfc-editor.org/info/rfc5730>

[RFC5731] Hollenbeck S.: Extensible Provisioning Protocol (EPP) Domain Name Mapping, RFC5731, <https://www.rfc-editor.org/info/rfc5731>

[RFC5732] Hollenbeck S.: Extensible Provisioning Protocol (EPP) Host Mapping, RFC5732, <https://www.rfc-editor.org/info/rfc5732>

[RFC5733] Hollenbeck S.: Extensible Provisioning Protocol (EPP) Contact Mapping, RFC5733, <https://www.rfc-editor.org/info/rfc5733>

[RFC5910] Gould J., Hollenbeck S.: Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP), RFC5910, <https://www.rfc-editor.org/info/rfc5910>

15 Appendix A - Return codes

Below are the return codes submitted by the client to the EPP server that are adopted by the Registry's synchronous server. The codes and meanings are established by the EPP standard.

1000=Command completed successfully
Response to a command completed successfully.

1001=Command completed successfully; action pending
Response to a command completed successfully and notification that a consequent action must be carried out asynchronously by the server.
Example: response to the command Domain Create, which creates the domain and puts it in dnsHold awaiting the successful DNS configuration.
The same response occurs when the configuration of the name servers associated with a domain is changed by the command Domain Update (wait for the outcome of the control of the new DNS configuration).

1300=Command completed successfully; no messages
Response to a command Poll(op="req") completed successfully and notification that the polling queue contains NO messages.

1301=Command completed successfully; ack to dequeue
Response to a command Poll(op="req") completed successfully and notification that the polling queue contains at least one message.

1500=Command completed successfully; ending session
Response to a command Logout completed successfully.

2001=Command syntax error
Response to a command whose execution fails because a parameter (or a value of a parameter) inserted it the command is incorrect.
The same response occurs when the command is unknown to the server (i.e. it does NOT belong to the version of EPP implemented by the server).

2002=Command use error
Response to a command whose execution fails due to errors regarding the context and/or sequence of the commands.
Example: issuing any command when the session is not active or when the session is terminated, performing Login or Logout twice in succession or logging out without first logging in.

2003=Required parameter missing
Response to a command whose execution fails because the required parameter is missing

2004=Parameter value range error
Response to a command whose execution fails because a "parameter" (a value of an xml) element in the command is not in the value range allowed.

2005=Parameter value syntax error
Response to a command whose execution fails because a "parameter" (a value of an xml) element in the command contains a syntax error.
Example: response to a Contact Create with an ID contact containing characters not permitted.

2100=Unimplemented protocol version
Response to a Login command whose execution fails because a protocol version has been declared which is different from the one shown by the server.

2101=Unimplemented command
Response to a command whose execution fails because NOT implemented by the server.
Example: response to a Domain Renew.

2102=Unimplemented option
Response to a command whose execution fails because it uses an option (op) NOT implemented by the server.

2103=Unimplemented extension
Response to a command whose execution fails because it uses an extension NOT implemented by the server.

2104=Billing failure
Response to a command whose execution fails because it is NOT allowed by the Registrar's credit situation.

2106=Object is not eligible for transfer
Response to a Domain Transfer command (op="request") whose execution fails because transfer for that domain is NOT allowed.

2200=Authentication error
Response to a command (typically Login) whose execution fails because the credentials supplied are NOT valid

2201=Authorization error
Response to a command whose execution fails because AuthInfo has not been supplied.

2202=Invalid authorization information
Response to a command whose execution fails because the AuthInfo supplied does NOT coincide with that associated with the domain.

2300=Object pending transfer
Response to a command whose execution fails because the domain is pending transfer.

2301=Object not pending transfer
Response to a command whose execution fails because the domain is NOT pending transfer.

2302=Object exists
Response to a creation command whose execution fails because the object already exists.

2303=Object does not exist
Response to a command whose execution fails because the object does NOT exist.

2304=Object status prohibits operation
Response to a command whose execution fails because of the current status of the object.

2305=Object association prohibits operation
Response to a command whose execution fails due to the associations that the object has with the other objects.
Example: attempt to delete a contact that is associated with one or more domains.

2306=Parameter value policy error
Response to a command whose execution fails due to the value of a parameter specified in the request that does NOT conform to system policy.

2308=Data management policy violation
Response to a command whose execution fails due to one or more parameters in the request which would violate the system's policies of data management.

Example: attempt to create a domain with a number of name servers that is lower than the minimum defined by the system policy.

2400=Command failed
Response to a command whose execution fails without termination of current session.

2500=Command failed; server ending session
Response to a command whose execution fails with termination of current session.

2502=Session limit exceeded; server closing connection
Response to a command whose execution fails because the maximum limit of simultaneous sessions per Registrar has been reached.

16 Appendix B - Reasons for errors

Below are the reasons used by the synchronous server of the Italian Registry detailing further the return codes reported in the previous section, numbered from 1000 to 2502.
The reasons are divided up by category:

- Reasons >= 4000: generic errors
- Reasons >= 5000: session errors
- Reasons >= 6000: accounting errors
- Reasons >= 7000: errors regarding the DNS configuration proposed in the Domain Create and Domain Update commands
- Reasons >= 8000: Contact object errors
- Reasons >= 9000: Domain object errors
- Reasons >= 10000: errors regarding the DNSSEC

The meaning of each reason is established by the Italian Registry and can be subject to change.

16.1 Reasons for errors about .it contacts and domains

```
(2001=Command syntax error 4003=<The syntax error message coming from the XML parser>):
Response to any command whose execution fails because it is NOT syntactically correct.

(2002=Command use error 4004=Command has been already executed successfully in the accreditation test. This error does not affect the test result):
Response to any command submitted to the accreditation server whose execution fails because command has been already executed successfully in the accreditation test.

(2002=Command use error 4005=Unexpected command in accreditation test):
Response to any command submitted to the accreditation server whose execution fails because it is unexpected in the accreditation test.

(2002=Command use error 4006=Unexpected command in accreditation test - Test completed):
Response to any command submitted to the accreditation server whose execution fails because the test has been completed.

(2002=Command use error 4007=Command is prohibited on this server):
Response to a command Update/Delete/Domain Transfer whose execution fails because it has been submitted to the server dedicated to the registration of domain names deleted less than 7 days ago.

(2002=Command use error 4014=Login request was sent on a session already opened):
Response to a Login command whose execution fails because the command report a session ID associated with an opened session.

(2002=Command use error 4015=First request on a new session was not Login):
Response to any command, except Login and Hello, whose execution fails because the command has not been submitted on an opened session.

(2002=Command use error 5058=The Registrar is suspended):
```

Response to a Domain Create, Domain Transfer, Domain Transfer-Trade or Domain Update for the change of Registrant command whose execution fails because the command has been sent by a suspended Registrar.

(2003=Required parameter missing 4011=Object URI missing):
Response to a Login command whose execution fails because an URI of a required EPP object is missing.

(2003=Required parameter missing 4012=Extension URI missing):
Response to a Login command whose execution fails because an URI of a required extension is missing or, in the case of a "DNSSEC accredited" Registrar who has indicated only one between the following namespaces:

- urn:ietf:params:xml:ns:secDNS-1.1;
- http://www.nic.it/ITNIC-EPP/extsecDNS-1.0.

(2003=Required parameter missing 5001=Message ID missing):
Response to a command Poll (op="ack") whose execution fails because the command does NOT contain the ID message of the message to confirm.

(2003=Required parameter missing 5005=Message refers to a namespace URI missing in Login request):
Response to a Poll (op="req") command whose execution fails because the Registrar has not indicated, in the Login request, a namespace related to an extension which is in the message.

(2003=Required parameter missing 8004=There is nothing to update):
Response to a command Contact Update whose execution fails because the <chg> parameter has NOT been specified or filled.

(2003=Required parameter missing 8019=Email address missing):
Response to a command Contact Create whose execution fails because the contact's email is missing.

(2003=Required parameter missing 8020=Consent for publishing missing):
Response to a command Contact Create whose execution fails because ConsentForPublishing has not been specified.

(2003=Required parameter missing 8022=Voice number missing):
Response to a command Create/Contact Update whose execution fails because <contact:voice> is missing or empty.

(2003=Required parameter missing 8023=Registrant: entity type missing):
Response to a command Create/Contact Update whose execution fails because <extcon:entityType> is missing or empty.

(2003=Required parameter missing 8025=Registrant: nationality code missing):
Response to a command Create/Contact Update whose execution fails because <extcon:nationalityCode> is missing.

(2003=Required parameter missing 8026=Registrant: reg code missing):
Response to a command Create/Contact Update whose execution fails because <extcon:regCode> is missing or empty.

(2003=Required parameter missing 8032=Postal information missing):
Response to a command Contact Create whose execution fails because <contact:PostalInfo> is missing.

(2003=Required parameter missing 8034=Postal information: name missing):
Response to a command Create/Contact Update whose execution fails because NO name has been specified in <contact:name>.

(2003=Required parameter missing 8035=Postal information: org missing):
Response to a command Create/Contact Update whose execution fails because <contact:org> is missing or empty.
NB: The org field is only mandatory if the contact is a Registrant with EntityType<>1.

(2003=Required parameter missing 8036=Postal information: addr missing):
Response to a command Create/Contact Update whose execution fails because <contact:addr> is missing or empty.

(2003=Required parameter missing 8037=Postal information: street missing):
Response to a command Create/Contact Update whose execution fails because <contact:street> is missing or empty.

(2003=Required parameter missing 8039=Postal information: city missing):
Response to a command Create/Contact Update whose execution fails because <contact:city> is empty.

(2003=Required parameter missing 8040=Postal information: sp missing):
Response to a command Create/Contact Update whose execution fails because <contact:sp> is missing or empty.

(2003=Required parameter missing 8041=Postal information: pc missing):
Response to a command Create/Contact Update whose execution fails because <contact:pc> is missing or empty.

(2003=Required parameter missing 8042=Postal information: cc missing):
Response to a command Create/Contact Update whose execution fails because <contact:cc> is missing or empty.

(2003=Required parameter missing 8061=Contact: add element is empty):
Response to a command Contact Update whose execution fails because <add> does NOT contain anything to add (is empty).

(2003=Required parameter missing 8062=Contact: rem element is empty):
Response to a command Contact Update whose execution fails because <rem> does NOT contain anything to remove (is empty).

(2003=Required parameter missing 8064=Contact: chg element is empty):
Response to a command Contact Update whose execution fails because <chg> does NOT contain anything to change (is empty).

(2003=Required parameter missing 9016=Registrant missing):
Response to a command Create/Domain Update whose execution fails because <domain:registrant> has not been specified or is empty.

(2003=Required parameter missing 9019=There is nothing to update):
Response to a command Domain Update whose execution fails because <add>, <rem> or <chg> are missing and there is thus nothing to update.

(2003=Required parameter missing 9038=Domain: add element is empty):
Response to a command Domain Update whose execution fails because <add> does NOT contain anything to add (is empty).

(2003=Required parameter missing 9039=Domain: rem element is empty):
Response to a command Domain Update whose execution fails because <rem> does NOT contain anything to remove (is empty).

(2003=Required parameter missing 9040=Domain: chg element is empty):
Response to a command Domain Update whose execution fails because <chg> does NOT contain anything to change (is empty).

(2003=Required parameter missing 9068=Authorization information missing in Domain Update):
Response to a Domain Update command of a domain name that requires a modification of the Registrant whose execution fails because the new authInfo is missing, or a simple modification that only involves a change of authInfo.

(2004=Parameter value range error 4002=Invalid values):
An element in a XML request has an invalid value.

(2004=Parameter value range error 5053=Property is mandatory):
An element in a XML request is a mandatory property.

(2004=Parameter value range error 7004=Host does not exist):
A name server reported in a XML request does not exist.

(2004=Parameter value range error 8012=Status to add has not "client" prefix):
Response to a command Contact Update for the addition of a status whose execution fails because the status to add does NOT have the prefix “client”.
NB: only if the status to add is one of the valid ones (serverDeleteProhibited, ok, linked, etc.).

(2004=Parameter value range error 8013=Status to remove has not "client" prefix):
Response to a command Contact Update to remove a status whose execution fails because the status to remove does NOT have the “client”.
NB: only if the status to remove is one of the valid ones (serverDeleteProhibited, ok, linked, etc.).

(2004=Parameter value range error 8021=Too many contact identifiers):
Response to a command Contact Check whose execution fails because the number of contacts specified in the command is higher than the maximum specified by system policy.

(2004=Parameter value range error 8024=Registrant: invalid entity type):
Response to a command Create/Contact Update whose execution fails because <extcon:entityType> contains an invalid value (out of range [1,..,7]).

(2004=Parameter value range error 8027=Registrant: invalid reg code):
Response to a command Create/Contact Update whose execution fails because <extcon: regCode> contains an invalid value.

(2004=Parameter value range error 8046=Email cannot be changed with an empty value):
Response to a command Contact Update whose execution fails because <contact:email> is empty.

(2004=Parameter value range error 8047=Voice cannot be changed with an empty value):
Response to a command Contact Update whose execution fails because <contact:voice> is empty.

(2004=Parameter value range error 8048=Postal information: invalid cc value):
Response to a command Create/Contact Update whose execution fails because <contact:cc> contains a country code (of 2 characters) that does NOT exist.

(2004=Parameter value range error 8049=Postal information: invalid sp value):
Response to a command Create/Contact Update whose execution fails because <contact:sp> contains an invalid value.

(2004=Parameter value range error 8050=Registrant: invalid nationality code):
Response to a command Create/Contact Update whose execution fails because <registrant:nationalityCode> a country code (of 2 characters) that does NOT exist.

(2004=Parameter value range error 8051=Registrant: nationality code is not allowed):
Response to a command Create/Contact Update whose execution fails because <registrant:nationalityCode> a country code (of 2 characters) that is NOT valid for the Registrant.

(2004=Parameter value range error 8059=Contact status is not implemented by the server):
Response to a command Contact Update whose execution fails because <contact:status> in the add section contains a status that is not implemented by the server.

(2004=Parameter value range error 8064=Registrant: entity type is not compatible with nationality code):
Response to a command Create/Contact Update whose execution fails because the value of <registrant:entityType> is not compatible with the value of <registrant:nationalityCode>.

(2004=Parameter value range error 8065=Postal information: invalid pc value):
Response to a command Create/Contact Update whose execution fails because the value of element <contact:pc> is not valid.

(2004=Parameter value range error 9003=Contact does not exist):
Response to a command Domain Create whose execution fails because one or more contacts in the command do NOT exist.

(2004=Parameter value range error 9030=Status to add has not "client" prefix):
Response to a command Domain Update to add a status whose execution fails because the status to add does NOT have the “client” prefix.
NB: to have such a response, the status must in any case be one of those existing in the system: serverDeleteProhibited, inactive, etc.

(2004=Parameter value range error 9031=Status to remove has not "client" prefix):
Response to a command Domain Update to remove a status whose execution fails because the status to remove does NOT have the “client” prefix.
NB: to have such a response, the status must in any case be one of those existing in the system: serverDeleteProhibited, inactive, etc.

(2004=Parameter value range error 9049=Invalid length of authInfo element):
Response to a command whose execution fails because the length of the password of authInfo specified is greater than the maximum length specified by the system policy.

(2004=Parameter value range error 9050=Too many domain names):
Response to a command Domain Check whose execution fails because the number of domain names specified in the command is greater than the maximum specified by the system policy.

(2004=Parameter value range error 9067=New authorization information is current authorization information):
Response to a command Domain Update of a domain name that requests the modification of the Registrant whose execution fails because the new authInfo is the same as the current authorization.

(2004=Parameter value range error 9073=Domain status is not implemented by the server):
Response to a command Domain Update whose execution fails because <domain:status> in the add section contains a status not implemented by the server.

(2005=Parameter value syntax error 7001=Host name syntax error):
Response to a command Create/Domain Update whose execution fails because one or more hostnames in the command are NOT syntactically correct.

(2005=Parameter value syntax error 7003=IP address syntax error):

Response to a command Create/Domain Update whose execution fails because one or more IP addresses in the command are NOT syntactically correct.

(2005=Parameter value syntax error 8001=Contact ID syntax error):
Response to a command Contact Create whose execution fails because the contact ID specified in the command is NOT syntactically correct.

(2005=Parameter value syntax error 8018=Email address syntax error):
Response to a command Create/Contact Update whose execution fails because the Email specified is NOT syntactically correct.

(2005=Parameter value syntax error 8053=Voice number syntax error):
Response to a command Create/Contact Update whose execution fails because <contact:voice> specified is NOT syntactically correct.

(2005=Parameter value syntax error 8054=Fax number syntax error):
Response to a command Create/Contact Update whose execution fails because <contact:fax> specified is NOT syntactically correct.

(2005=Parameter value syntax error 8066=Voice extension syntax error):
Response to a command Create/Contact Update whose execution fails because the attribute x of <contact:voice> specified is NOT syntactically correct.

(2005=Parameter value syntax error 8067=Fax extension syntax error):
Response to a command Create/Contact Update whose execution fails because the attribute x of <contact:fax> specified is NOT syntactically correct.

(2005=Parameter value syntax error 8070=Postal information: invalid org value):
Response to a command Create/Contact Update whose execution fails because <contact:org> contains an invalid value.

(2005=Parameter value syntax error 9007=Domain name syntax error):
Response to a command to create a domain name whose execution fails because the domain name is NOT syntactically correct.

(2102=Unimplemented option 4008=Unsupported language):
Response to a Login command whose execution fails because the value of <lang> element is unsupported.

(2102=Unimplemented option 4009=Unsupported object URI):
Response to a Login command whose execution fails because the value of <objURI> element is unsupported.

(2102=Unimplemented option 4010=Unsupported extension URI):
Response to a Login command whose execution fails because the value of <extURI> element is unsupported.

(2102=Unimplemented option 9020=Unsupported transfer option):
Response to a command Domain Transfer whose execution fails because the option requested does NOT exist.

(2102=Unimplemented option 9086=Unsupported hostObj option):
Response to a command Domain Create whose execution fails because the hostObj option is NOT supported.

(2102=Unimplemented option 9087=Unsupported report option):
Response to a command Domain Update whose execution fails because the report option of rgp:update extension is NOT supported.

(2102=Unimplemented option 10002=DNSSEC: unsupported maxSigLife element):
Response to a Create/Domain Update command, from a “DNSSEC accredited” Registrar,

whose execution fails because within the <secDNS:create> or <secDNS:update> element there is <secDNS:maxSigLife>.

(2102=Unimplemented option 10003=DNSSEC: unsupported keyData element):
Response to a Create/Domain Update command, from a "DNSSEC accredited" Registrar, whose execution fails because within the <secDNS:create> or <secDNS:update> element there is a <secDNS:keyData> element in place of or within <secDNS:dsData>.

(2102=Unimplemented option 10004=DNSSEC: unsupported urgent attribute):
Response to a Domain Update command, from a "DNSSEC accredited" Registrar, whose execution fails because within the <secDNS:update> element there is the ***urgent*** attribute.

Deprecated (2104=Billing failure 5054=Low credit: only auto renew and unbillable commands will be processed):
Response to a command (for a payment) whose execution fails because the residual credit of the Registrar is too low; it is only enough to renew the domain names maintained.

(2104=Billing failure 5055=Out of funds):
Response to a command (for a payment) whose execution fails because the Registrar is out of funds.

Deprecated (2104=Billing failure 5056=Credit is going below threshold limit due to the operation cost):
Response to a command (for a payment) whose execution fails because the residual credit, due to the operation cost, will go under the low credit threshold.

(2106=Object is not eligible for transfer 9018=Destination client of the transfer operation is the domain name sponsoring client):
Response to a command Domain Transfer (op="request") whose execution fails because it has been submitted by the same Registrar who owns the domain name.

(2200=Authentication error 6002=Object does not exist):
Response to a Login command whose execution fails because the Registrar does not exist.

(2200=Authentication error 6003=Account expired):
Response to a Login command whose execution fails because the account is expired.

(2200=Authentication error 6004=Password expired):
Response to a Login command whose execution fails because the password is expired.

(2200=Authentication error 6005=Invalid username or password):
Response to a command Login whose execution fails because the username and/or password are incorrect.

(2200=Authentication error 6007=Account disabled):
Response to a command Login whose execution fails because the account has been disabled.

(2200=Authentication error 6008=Invalid new password):
Response to a Login command whose execution fails because the new password is the same as the old one or its length is below the minimum length specified in parameter MIN_NEWPWUSER_LENGTH described in the table included in Section 12.8.

(2201=Authorization error 6001=Lack of permissions to process command):
Response to a Contact Info, Contact Delete or Contact Update command whose execution fails because the contact requested in the command does NOT belong to the current Registrar.

(2201=Authorization error 6009=Lack of permissions to process command or object does not exist):
Response to a Domain Delete or Domain Update command whose execution fails because the domain name specified in the command does not belong to the current Registrar.

(2201=Authorization error 9051=Lack of permissions to view status of domain transfer request):
Response to a command Domain Transfer (op="query") whose execution fails because the Registrar is NOT permitted to see the progress status of the transfer.

(2201=Authorization error 9053=Lack of permissions to cancel domain transfer request):
Response to a command Domain Transfer (op="cancel") whose execution fails because the Registrar is NOT permitted to cancel the transfer.

(2201=Authorization error 9071=Lack of permissions to approve domain transfer request):
Response to a command Domain Transfer (op="approve") whose execution fails because the Registrar is NOT permitted to approve the transfer.

(2201=Authorization error 9072=Lack of permissions to reject domain transfer request):
Response to a command Domain Transfer (op="reject") whose execution fails because the Registrar is NOT permitted to reject the transfer.

(2202=Invalid authorization information 9001=Authorization information missing):
Response to a command whose execution fails due to missing AuthInfo associated with the object referred to in the command.
Example: execution of the command Domain Info on a domain name that belongs to another Registrar without specifying the AuthInfo.

(2202=Invalid authorization information 9085=Invalid domain authorization information or domain does not exist):
Response to a Domain Info or Transfer Query Domain command whose execution fails due to the lack of correspondence between the AuthInfo associated with the domain name and the one inserted in the command.
Es: execution of the Domain Info command on a domain name that belongs to another Registrar specifying an incorrect AuthInfo or without specifying it.

(2301=Object not pending transfer 9054=Domain transfer not pending):
Response to a command Domain Transfer (op="query") whose execution fails because the domain name specified in the query has NEVER been involved in a transfer.

(2302=Object exists 8058=Contact already exists):
Response to a command Contact Create whose execution fails because the contact specified already exists.

Deprecated (2302=Object exists 8068=Contact is registered in the asynchronous system):
Response to a command Contact Info whose execution fails because the contact is registered in the asynchronous system.

(2302=Object exists 9042=Domain is registered):
Response to a command Domain Create whose execution fails because the domain name is registered. This reason is used also in the Domain Check response when the domain name is not available.

Deprecated (2302=Object exists 9082=Domain is in pending create status in the asynchronous system):
Response to a command Domain Create whose execution fails because the domain name is in pending create in the asynchronous system. This reason is used also in the

Domain Check response when the domain name is not available.

Deprecated (2302=Object exists 9084=Domain is registered in the asynchronous system):
Response to a command Transfer or Domain Info whose execution fails because the domain name is in pending create in the asynchronous system.

(2303=Object does not exist 5004=There are no messages in the queue):
Response to a command Poll (op="ack") whose execution fails because the queue does not contain any messages to confirm.

(2303=Object does not exist 9003=Contact does not exist):
Response to a command Contact Info whose execution fails because the contact in the command does NOT exist.

(2302=Object does not exist 9021=Domain is reserved):
Response to a command Domain Create whose execution fails because the domain name is reserved. This reason is used also in the Domain Check response when domain name is not available.

(2303=Object does not exist 9043=Domain is unassignable):
Response to a command Domain Create whose execution fails because the domain name CANNOT be assigned. This reason is used also in the Domain Check response when the domain name is not available.

(2303=Object does not exist 9044=Domain is geographic):
Response to a command Domain Create whose execution fails because the domain name is geographical. This reason is used also in the Domain Check response when the domain name is not available.

(2304=Object status prohibits operation 8006=Contact has status clientDeleteProhibited):
Response to a command Contact Delete whose execution fails because the contact object has the status clientDeleteProhibited.

(2304=Object status prohibits operation 8007=Contact has status serverDeleteProhibited):
Response to a command Contact Delete whose execution fails because the contact object has the status serverDeleteProhibited.

(2304=Object status prohibits operation 8008=Contact has status clientUpdateProhibited):
Response to a command Contact Update whose execution fails because the contact object has the status clientUpdateProhibited.

(2304=Object status prohibits operation 8009=Contact has status serverUpdateProhibited):
Response to a command Contact Update whose execution fails because the contact object has the status serverUpdateProhibited.

(2304=Object status prohibits operation 9022=Domain has status clientTransferProhibited):
Response to a command Domain Transfer (op="request") whose execution fails because the domain object has the status clientTransferProhibited.

(2304=Object status prohibits operation 9023=Domain has status serverTransferProhibited):
Response to a command Domain Transfer (op="request") whose execution fails because the domain object has the status serverTransferProhibited.

(2304=Object status prohibits operation 9024=Domain has status

clientDeleteProhibited):
Response to a command Domain Delete whose execution fails because the domain object has the status clientDeleteProhibited.

(2304=Object status prohibits operation 9025=Domain has status serverDeleteProhibited):
Response to a command Domain Delete whose execution fails because the domain object has the status serverDeleteProhibited.

(2304=Object status prohibits operation 9026=Domain has status clientUpdateProhibited):
Response to a command whose execution fails because the domain object has the status clientUpdateProhibited.
Example: attempt to update the nameservers of a domain name with the Domain Update when in clientUpdateProhibited.

(2304=Object status prohibits operation 9027=Domain has status serverUpdateProhibited):
Response to a command whose execution fails because the domain object has the status serverUpdateProhibited.
Example: attempt to update the nameservers of a domain name with the Domain Update when in serverUpdateProhibited.

(2304=Object status prohibits operation 9045=Domain has status clientHold):
Response to a command whose execution fails because the domain object has the status clientHold.
Example: attempt to change the Regisrant of a domain name (with the Domain Update) when in clientHold.

(2304=Object status prohibits operation 9047=Domain has status serverHold):
Response to a command whose execution fails because the domain object has the status serverHold.

(2304=Object status prohibits operation 9055=Domain has status ok):
Response to a command whose execution fails because the domain object has the status ok.

(2304=Object status prohibits operation 9056=Domain has status inactive):
Response to a command whose execution fails because the domain object has the status inactive.

(2304=Object status prohibits operation 9057=Domain has status dnsHold):
Response to a command whose execution fails because the domain object has the status dnsHold.

(2304=Object status prohibits operation 9058=Domain has status autoRenewPeriod):
Response to a command whose execution fails because the domain object has the status autoRenewPeriod.

(2304=Object status prohibits operation 9059=Domain has status pendingUpdate):
Response to a command whose execution fails because the domain object has the status pendingUpdate.

(2304=Object status prohibits operation 9060=Domain has status pendingTransfer):
Response to a command whose execution fails because the domain object has the status pendingTransfer.

(2304=Object status prohibits operation 9061=Domain has status noRegistrar):
Response to a command whose execution fails because the domain object has the status noRegistrar.

(2304=Object status prohibits operation 9062=Domain has status toBeReassigned):
Response to a command whose execution fails because the domain object has the status toBeReassigned.

(2304=Object status prohibits operation 9063=Domain has status challenged):
Response to a command whose execution fails because the domain object has the status challenged.

(2304=Object status prohibits operation 9064=Domain has status redemptionPeriod):
Response to a command whose execution fails because the domain object has the status redemptionPeriod.

(2304=Object status prohibits operation 9064=Domain has status serverHold):
Response to a command whose execution fails because the domain object has the status serverHold.

(2304=Object status prohibits operation 9065=Domain has status revoked):
Response to a command whose execution fails because the domain object has the status revoked.

(2304=Object status prohibits operation 9066=Domain has status pendingDelete):
Response to a command whose execution fails because the domain object has the status pendingDelete.

(2304=Object status prohibits operation 9077=Domain has status notRenewed):
Response to a command whose execution fails because the domain object has the status notRenewed.

(2304=Object status prohibits operation 9081=Domain has status notRenewed):
Response to a command whose execution fails because the domain object is subjected to a bulk operation.

(2305=Object association prohibits operation 8005=Contact is associated with domains):
Response to a command Contact Delete whose execution fails because the contact object is still associated with one or more domain names.

(2306=Parameter value policy error 5002=Message ID is not allowed):
Response to a command Poll(op=”req”) whose execution fails because the <msgID> must NOT be specified in the command.

(2306=Parameter value policy error 5003=Message ID is not the ID of the first message in the queue):
Response to a command Poll(op=”ack”) whose execution fails because the ID of the message to remove from the queue (confirming reading) does NOT coincide with the one actually at the head of the queue.

(2306=Parameter value policy error 7002=Duplicate IP addresses):
Response to a command Create/Domain Update whose execution fails because the same IP address has been specified more than once for different hosts.

(2306=Parameter value policy error 7008=IP address to add already exists):
Response to a command Domain Update whose execution fails because an IP address has been specified that is already present in an existing host and which will not be removed with the same command.

(2306=Parameter value policy error 8002=Contact ID prefix not allowed):
Response to a command Contact Create whose execution fails because the contact ID specified contains an invalid prefix.

(2306=Parameter value policy error 8010=Duplicate statuses to add):

Response to a command Contact Update whose execution fails because the same status to add is inserted more than once in the command.

(2306=Parameter value policy error 8011=Duplicate statuses to remove):
Response to a command Contact Update whose execution fails because the same status to remove is inserted more than once in the command.

(2306=Parameter value policy error 8031=Postal information in international form is not allowed):
Response to a command Create/Contact Update whose execution fails because "int" PostalInfo has been specified.
NB: we only accept "loc" (local) addresses.

(2306=Parameter value policy error 8043=Postal information: name cannot be changed for a Registrant with the entity type = 1):
Response to a command Contact Update for a Registrant (with EntityType=1) whose execution fails because in this case the <contact:name> field cannot be changed.

(2306=Parameter value policy error 8044=Postal information: org cannot be changed for a registrant):
Response to a command Contact Update for a Registrant whose execution fails because the <contact:org> field cannot be changed.

(2306=Parameter value policy error 8045=Postal information: cc cannot be changed for a registrant with the entity type != 1):
Response to a command Contact Update for a Registrant (with EntityType<>1) whose execution fails because the <contact:cc> field cannot be changed.

(2306=Parameter value policy error 8055=Registrant generated from admin/tech contact must have entity type = 1):
Response to a command Contact Update sent in order to transform an admin/tech contact into a registrant contact whose execution fails because the <extcon:entityType> element contains a value different from 1.

(2306=Parameter value policy error 8056=Registrant: contact already present as registrant - update is prohibited):
Response to a command Contact Update for a Registrant whose execution fails because Registrant data cannot be changed once they have been set either via a Contact Create or Contact Update after a Contact Create.

(2306=Parameter value policy error 8057=Registrant: registrant with the entity type = 1 org and name are different):
Response to a command Create or Contact Update for a type 1 Registrant whose execution fails because the org and name fields are different.

(2306=Parameter value policy error 9004=Duplicate names of name server):
Response to a command Create/Domain Update whose execution fails because the same name server has been inserted several times.

Deprecated (2306=Parameter value policy error 9008=Zone is not managed):
Response to a command Domain Create whose execution fails because the domain name belongs to a zone that is NOT managed by the Registry.
Example: paperino.net

(2306=Parameter value policy error 9009=New registrant ID is current registrant ID):
Response to a command Domain Update (to modify Registrant) whose execution fails because the new Registrant submitted coincides with the current one.

(2306=Parameter value policy error 9037=Duplicate contacts):
Response to a command Domain Create whose execution fails because the same contact has been referred several times with the same role.

Example: creation of a domain name with the same two technical contacts.

(2306=Parameter value policy error 9075=Duplicate statuses to add):
Response to a Domain/Contact Update command whose execution fails because in the command the status to add has been inserted several times.

(2306=Parameter value policy error 9076=Duplicate statuses to remove):
Response to a Domain/Contact Update command whose execution fails because in the command the status to remove has been inserted several times.

(2306=Parameter value policy error 10001=DNSSEC: Registrar is not DNSSEC accredited):
Response to any EPP request, from a "non DNSSEC accredited" Registrar, whose execution fails because one or both the following namespaces have been inserted:

- urn:ietf:params:xml:ns:secDNS-1.1;
- http://www.nic.it/ITNIC-EPP/extsecDNS-1.0.

(2306=Parameter value policy error 10005=DNSSEC: no dsData to remove or add):
Response to a Domain Update command, from a "DNSSEC accredited" Registrar, whose execution fails because within the <secDNS:update> element there is neither the <secDNS:add> element nor the <secDNS:rem> element, or there is the <secDNS:rem> element which contains neither single <secDNS:dsData> elements nor the <secDNS:all> element.

(2306=Parameter value policy error 10007=DNSSEC: invalid digestType value):
Response to a Create/Domain Update command, from a "DNSSEC accredited" Registrar, whose execution fails because a <secDNS:dsData> element contains an unsupported or invalid ***digestType*** value (Section 11.1).

(2306=Parameter value policy error 10008=DNSSEC: invalid alg value):
Response to a Create/Domain Update command, from a "DNSSEC accredited" Registrar, whose execution fails because a <secDNS:dsData> element contains an unsupported or invalid ***alg*** value (Section 11.1).

(2306=Parameter value policy error 10009=DNSSEC: invalid digest value):
Response to a Create/Domain Update command, from a "DNSSEC accredited" Registrar, whose execution fails because a <secDNS:dsData> element contains a value of the ***digest*** field whose length is not compatible with the chosen ***digest type***.

(2306=Parameter value policy error 10010=DNSSEC: duplicate dsData):
Response to a Create/Domain Update command, from a "DNSSEC accredited" Registrar, whose execution fails because there are 2 <secDNS:dsData> elements containing the same values for the foreseen 4 fields.

(2306=Parameter value policy error 10011=DNSSEC: dsData to add is already associated with the domain):
Response to a Domain Update command, from a "DNSSEC accredited" Registrar, whose execution fails because in the <secDNS:add> element there is a DS record already associated with the domain name.

(2306=Parameter value policy error 10012=DNSSEC: dsData to remove is not associated with the domain):
Response to a Domain Update command, from a "DNSSEC accredited" Registrar, whose execution fails because in the <secDNS:rem> element there is a DS record not associated with the domain name.

(2308=Data management policy violation 5050=Command limit exceeded):
Response to a command whose execution fails because the limit of MAX_CMD parameter defined by the system policy, specified in the table at paragraph 12.8, has been exceeded.

(2308=Data management policy violation 7005=Too few IP addresses):
Response to a command Create/Domain Update whose execution fails because (one or more of the name servers) have been specified with a number of IP addresses lower than the number defined by the system policy.

(2308=Data management policy violation 7006=Too many IP addresses):
Response to a command Create/Domain Update whose execution fails because one or more of the name servers) have been specified with a number of IP addresses higher than the number defined by the system policy.

(2308=Data management policy violation 7007=One v4 IP address for this host is required):
Response to a command Create/Domain Update whose execution fails because at least one IPv4 address has NOT been specified for a name server.

Deprecated (2308=Data management policy violation 7009=IP V6 address currently unsupported):
Response to a command Create/Domain Update whose execution fails because one IPv6 address has been specified for a name server.

(2308=Data management policy violation 8009=Contact has status serverUpdateProhibited):
Response to a Create/Update/Transfer-Trade Domain command whose execution fails because the contact specified in the request is in serverUpdateProhibited status.

(2308=Data management policy violation 8014=Status to add is already associated with the contact):
Response to a command Contact Update for the addition of a status whose execution fails because the status to add is already associated with the contact.

(2308=Data management policy violation 8015=Status to remove is not associated with the contact):
Response to a command Contact Update for the removal status whose execution fails because the status to remove is NOT associated with the contact.

(2308=Data management policy violation 8017=Too many postal information elements in localized form):
Response to a command Create/Contact Update whose execution fails because a number of PostalInfo addresses have been inserted that is greater than the maximum defined by the system policy.

(2308=Data management policy violation 8028=Registrant: ConsentForPublishing cannot be set to false if entity type != 1 and entity type != 3):
Response to a command Create/Domain Update whose execution fails because for Registrants other than natural persons, individual firm and professional person (EntityType<>1 and EntityType<>3) the ConsentForPublishing element cannot be set to false/0.

(2308=Data management policy violation 8029=Registrant: registrant with the entity type = 1 and admin are different):
Response to a command Create/Domain Update whose execution fails because the constraint of the Registrant (with entityType=1) coinciding with the admin contact (admin) of the domain name has not been respected.

(2308=Data management policy violation 8030=Contact is not a registrant):
Response to a command Create/Domain Update whose execution fails because the ID contact specified in <domain:registrant> is NOT in reality a Registrant.

(2308=Data management policy violation 8038=Postal information: too many streets):
Response to a command Create/Contact Update whose execution fails because a number

of <contact:street> have been specified that is greater than the maximum number defined by the system policy.

(2308=Data management policy violation 8050=Contact is not sponsored by the registrar):
Response to a command whose execution fails because one or more contacts in the command belong to another Registrar.
Example: in the command Domain Create, Domain Update and also in the Contact Update.

(2308=Data management policy violation 8060=Registrant: registrant cannot be a minor):
Response to a command Create or Contact Update for a type 1 Registrant whose execution fails because the Registrant is a minor.

(2308=Data management policy violation 8069=Registrant: country code is not allowed):
Response to a command Create or Contact Update for a non type 1 Registrant (the country code always overrides the nationality code) or type 1 Registrant (if nationality code is not enabled then country code is considered) whose execution fails because country code is not enabled.

(2308=Data management policy violation 9005=Too few name servers):
Response to a command Create/Domain Update whose execution fails because the number of nameservers is lower than the minimum allowed by the system policy.

(2308=Data management policy violation 9006=Too many name servers):
Response to a command Create/Domain Update whose execution fails because the number of name server is greater than the maximum allowed by the system policy.

(2308=Data management policy violation 9010=At least one administrative contact is required):
Response to a command Create/Domain Update whose execution fails because the administrative contact has NOT been specified.
(2308=Data management policy violation 9011=Too few administrative contacts):
Response to a command Create/Domain Update whose execution fails because the number of administrative contacts is lower than the minimum allowed by the system policy.

(2308=Data management policy violation 9012=Too many administrative contacts):
Response to a command Create/Domain Update whose execution fails because the number of administrative contacts is greater than the maximum allowed by the system policy.

(2308=Data management policy violation 9013=At least one tech contact is required):
Response to a command Create/Domain Update whose execution fails because the technical contact has NOT been specified.

(2308=Data management policy violation 9014=Too few technical contacts):
Response to a command Create/Domain Update whose execution fails because the number of technical contacts is lower than the minimum allowed by the system policy.

(2308=Data management policy violation 9015=Too many technical contacts):
Response to a command Create/Domain Update whose execution fails because the number of technical contacts is greater than the maximum allowed by the system policy.

(2308=Data management policy violation 9017=Domain cannot be associated with a Registrant with entity type != 1 and entity type != 3, and consentForPublishing set to false):
Response to a command Domain Create/Update/Transfer-Trade whose execution fails because a domain name cannot be associated with a Registrant other than a natural person, individual firm and professional person (EntityType<>1 and EntityType<>3) whose ConsentForPublishing element is set to false / 0.

(2308=Data management policy violation 9028=Contact to add is already associated with the domain):
Response to a command Domain Update whose execution fails because the contact to add is already associated with the domain name.

(2308=Data management policy violation 9029=Contact to remove is not associated with the domain):
Response to a command Domain Update whose execution fails because the contact to remove is NOT currently associated with the domain name.

(2308=Data management policy violation 9032=Status to add is already associated with the domain):
Response to a command Domain Update whose execution fails because the status to add is already associated with the domain name.

(2308=Data management policy violation 9033=Status to remove is not associated with the domain):
Response to a command Domain Update whose execution fails because the status to remove is NOT associated with the domain name.

(2308=Data management policy violation 9034=Name server to add is already associated with the domain):
Response to a command Domain Update whose execution fails because the name server to add is already associated with the domain name.

(2308=Data management policy violation 9035=Name server to remove is not associated with the domain):
Response to a command Domain Update whose execution fails because the name server to remove is NOT associated with the domain name.

(2308=Data management policy violation 9041=Domain Update combination of status, name server and registrant is not allowed):
Response to a command Domain Update whose execution fails because simultaneous operations between any two of the following: change of the status, change of the name servers/DS records, restoration of a cancelled domain name and change of the Registrant are not allowed.

(2308=Data management policy violation 9048=Name server to add is subordinate for the domain but has no IP addresses):
Response to a Create/Domain Update command whose execution fails because the nameserver to associate is subordinate to the domain name and no IP addresses were specified.

(2308=Data management policy violation 9070=Billing contacts prohibited):
Response to a command Create/Domain Update whose execution fails because the billing contacts have been specified.

(2308=Data management policy violation 9074=At least two name servers are required):
Response to a command Domain Create whose execution fails because no name server has been specified.

(2308=Data management policy violation 9078=Domain names deleted by less than 7 days must be registered on epp-deleted.nic.it (epp-deleted.pubtest.nic.it for test environment)):
Response to a command Domain Create whose execution fails because the domain name to create is deleted less than 7 days ago and the request has been sent to epp.nic.it (or epp.pubtest.nic.it for test enviroment).

(2308=Data management policy violation 9079=Request for domain references an uncompleted contact. A mandatory field is empty or has a wrong value):
Response to any command requested on a domain name whose execution fails because the

requests references a contact with uncompleted data. The contact has been migrated from the old “asynchronous” registration system and it should be normalized before being referred in a new acquisition done by the synchronous system.

(2308=Data management policy violation 9080=Request for domain references an uncompleted registrant. A mandatory field is empty or has a wrong value):
Response to any command requested on a domain name whose execution fails because the requests references a Registrant with uncompleted or wrong data. The Registrant has been migrated from the old “asynchronous” registration system and it should be normalized before being referred in a new acquisition done by the synchronous system.

(2308=Data management policy violation 9083=Only domain names deleted by less than 7 days can be registered on this server):
Response to a command Domain Create whose execution fails because the doman name to create is NOT deleted less than 7 days ago and the request has been sent to epp-deleted.nic.it (epp-deleted.pubtest.nic.it for test environment).

(2308=Data management policy violation 10006=DNSSEC: too many dsData items):
Response to a Create/Domain Update command, from a “DNSSEC accredited” Registrar, whose execution fails because the number of <secDNS:dsData> elements are not compatible with MAX_DS_IN_CREATE/MAX_DS_IN_UPDATE parameters described in the table included in Section 12.8.

(2400=Command failed 4000=Database error):
Response to a command whose execution fails due to an access to the database error. It is an error in the system and does NOT depend on the command sent by the client.

(2400=Command failed 4001=Concurrency error):
Response to a command whose execution fails due to concurrency problems on the EPP transaction.

(2400=Command failed 4013=Unexpected session ID inserted in Hello or Login request):
Response to a command Hello/Login whose execution fails (causing the ending of session) because the client has used in the request a session identifier before receiving it in the Greeting/Login response.

(2400=Command failed 4017=Internal error):
Response to a command whose execution fails due to an internal system error, which does NOT depend on the command sent by the client.

(2400=Command failed 5052=User IP address is not allowed):
Response to a command Login whose execution fails (causing termination of current session) because the client IP address is NOT allowed to integrate with the system.

(2400=Command failed 5057=Command IP address is not equal to Login IP address):
Response to a command whose execution fails because sent from an IP address different from the one used to send the Login command but with the same session ID.

(2400=Command failed 6006=Login command failed):
Response to a command Login whose execution fails because the username is incorrect.

(2502=Session limit exceeded; server closing connection 5051=Session opened limit exceeded):
Response to a command whose execution fails (causing termination of current session) because the maximum number of simultaneous sessions permitted by the system has been exceeded.

16.2 Specific reasons for errors about edu.it contacts and domains

(2004=Parameter value range error 8100=Registrant: entity type is not consistent with school code):
Response to an EPP Create Contact command whose execution is supposed to fail because the entityType is not compatible with the presence of the schoolCode field.

(2005=Parameter value syntax error 8101=Registrant: school code is not valid):
Response to an EPP Create Contact command whose execution is supposed to fail because the schoolCode field is not valid.

(2004=Parameter value range error 8102=Registrant: school code does not exist):
Response to an EPP Create Contact command whose execution is supposed to fail because the schoolCode field does not exist.

(2004=Parameter value range error 8103=Registrant: school code is inactive):
Response to an EPP Create Contact command whose execution is supposed to fail because the schoolCode field is not active.

(2004=Parameter value range error 8104=Registrant: reg code is not consistent with school code):
Response to an EPP Create Contact command whose execution is supposed to fail because the regCode is not consistent with the schoolCode field.

(2400=Command failed 8105=Registrant: unable to validate school code):
Response to an EPP Create Contact command whose execution is supposed to fail because the EPP server could not validate the schoolCode.

(2308=Data management policy violation 9100=Registrar is not enabled to manage 'edu.it' domains):
Response to an EPP Create Contact command whose execution is supposed to fail because a Registrar that is not enabled to manage edu.it domains has reported the schoolCode in the registrant extension. The same error reason is obtained in response to an EPP Create Domain command for an edu.it domain name when the Registrar is not enabled to manage such domains.

(2400=Command failed 9101=Unable to check if domain 'edu.it' is reserved):
Response to an EPP Create Contact command whose execution is supposed to fail because the EPP server could not check whether the edu.it domain name has previously been registered by the registration system that is made available by MIUR.

(2308=Data management policy violation 9102=Domain 'edu.it' cannot be associated with a registrant without school code):
Response to an EPP Create Contact command whose execution is supposed to fail because the Registrant contact doesn’t include the schoolCode.

(2308=Data management policy violation 9103=Domain 'edu.it' is reserved to a registrant with a different school code):
Response to an EPP Create Contact command whose execution is supposed to fail because the domain is reserved to a registrant with a different schoolCode.

16.3 Specific reasons for errors about gov.it contacts and domains

(2004=Parameter value range error 8200=Registrant 'gov.it': entity type is not consistent with IPA code):
Response to an EPP Create Contact command whose execution fails because the entityType is not compatible with the presence of the ipaCode field.

(2004=Parameter value range error 8201=Registrant 'gov.it': invalid IPA code or UO code):

Response to an EPP Create Contact command whose execution fails because the ipaCode and/or uoCode do not match the values obtained from the Validation Procedure or validation has expired.

(2004=Parameter value range error 8202=Registrant 'gov.it': reg code is not consistent with IPA code and/or UO code):
Response to an EPP Create Contact command whose execution fails because the regCode does not match the ipaCode and/or uoCode.

(2004=Parameter value range error 8203=Registrant 'gov.it': email is not consistent with IPA code and/or UO code):
Response to an EPP Create Contact command whose execution fails because the email does not match the value obtained from the Validation Procedure.

(2308= Data management policy violation 8204=Registrant 'gov.it': email update is prohibited):
Response to an EPP Update Contact command whose execution fails because an email change was requested.

(2003=Required parameter missing 9200=Request on 'gov.it' domain: missing validation code):
Response to an EPP Create Domain/Update Domain (trade)/Transfer Domain (transfertrade) command whose execution fails because the validationCode has not been sent.

(2004=Parameter value range error 9201=Request on 'gov.it' domain: invalid validation code):
Response to an EPP Create Domain/Update Domain (trade)/Transfer Domain (transfertrade) command whose execution fails because the validationCode does not match the value obtained from the Validation Procedure, does not match the ipaCode and/or uoCode, or validation has expired.

(2308=Data management policy violation 9202=Domain 'gov.it' cannot be associated to a registrant without IPA code):
Response to an EPP Create Domain/Update Domain (trade)/Transfer Domain (transfertrade) command whose execution fails because the contact registrant (or new registrant in the case of trade) does not include the ipaCode.

(2308=Data management policy violation 9203=Domain 'gov.it': the 'gov.it' registrant specified as tech/admin is different from the domain registrant):
Response to an EPP Create Domain/Update Domain (trade)/Transfer Domain (transfertrade) command whose execution fails because the tech/admin contact refers to a registrant other than the one specified in the registrant field of the domain name.

(2003=Required parameter missing 4016=Request refers to a namespace URI missing in Login request):
Response to an EPP command whose execution fails because the Registrar has not reported, in the EPP Login request, a namespace related to an extension present in the command.

17 Appendix C - Latin-1 Supplement Charset

Unicode Encoding	Character	Description
U+00E0	à	LATIN SMALL LETTER A WITH GRAVE
U+00E1	á	LATIN SMALL LETTER A WITH ACUTE
U+00E2	â	LATIN SMALL LETTER A WITH CIRCUMFLEX
U+00E3	ã	LATIN SMALL LETTER A WITH TILDE
U+00E4	ä	LATIN SMALL LETTER A WITH DIAERESIS
U+00E5	å	LATIN SMALL LETTER A WITH RING ABOVE
U+00E6	æ	LATIN SMALL LETTER AE
U+00E7	ç	LATIN SMALL LETTER C WITH CEDILLA
U+00E8	è	LATIN SMALL LETTER E WITH GRAVE
U+00E9	é	LATIN SMALL LETTER E WITH ACUTE
U+00EA	ê	LATIN SMALL LETTER E WITH CIRCUMFLEX
U+00EB	ë	LATIN SMALL LETTER E WITH DIAERESIS
U+00EC	ì	LATIN SMALL LETTER I WITH GRAVE
U+00ED	í	LATIN SMALL LETTER I WITH ACUTE
U+00EE	î	LATIN SMALL LETTER I WITH CIRCUMFLEX
U+00EF	ï	LATIN SMALL LETTER I WITH DIAERESIS
U+00F0	ð	LATIN SMALL LETTER ETH
U+00F1	ñ	LATIN SMALL LETTER N WITH TILDE
U+00F2	ò	LATIN SMALL LETTER O WITH GRAVE
U+00F3	ó	LATIN SMALL LETTER O WITH ACUTE
U+00F4	ô	LATIN SMALL LETTER O WITH CIRCUMFLEX
U+00F5	õ	LATIN SMALL LETTER O WITH TILDE
U+00F6	ö	LATIN SMALL LETTER O WITH DIAERESIS
U+00F8	ø	LATIN SMALL LETTER O WITH STROKE
U+00F9	ù	LATIN SMALL LETTER U WITH GRAVE
U+00FA	ú	LATIN SMALL LETTER U WITH ACUTE
U+00FB	û	LATIN SMALL LETTER U WITH CIRCUMFLEX
U+00FC	ü	LATIN SMALL LETTER U WITH DIAERESIS
U+00FD	ý	LATIN SMALL LETTER Y WITH ACUTE
U+00FE	þ	LATIN SMALL LETTER THORN
U+00FF	ÿ	LATIN SMALL LETTER Y WITH DIAERESIS
U+00DF	ß	LATIN SMALL LETTER SHARP S

18 Appendix D - Latin Extended-A Charset

Unicode Encoding	Character	Description
U+0101	ā	LATIN SMALL LETTER A WITH MACRON
U+0103	ă	LATIN SMALL LETTER A WITH BREVE
U+0105	ą	LATIN SMALL LETTER A WITH OGONEK
U+0107	ć	LATIN SMALL LETTER C WITH ACUTE
U+0109	ĉ	LATIN SMALL LETTER C WITH CIRCUMFLEX
U+010B	ċ	LATIN SMALL LETTER C WITH DOT ABOVE
U+010D	č	LATIN SMALL LETTER C WITH CARON
U+010F	ď	LATIN SMALL LETTER D WITH CARON
U+0111	đ	LATIN SMALL LETTER D WITH STROKE
U+0113	ē	LATIN SMALL LETTER E WITH MACRON
U+0115	ĕ	LATIN SMALL LETTER E WITH BREVE
U+0117	ė	LATIN SMALL LETTER E WITH DOT ABOVE
U+0119	ę	LATIN SMALL LETTER E WITH OGONEK
U+011B	ě	LATIN SMALL LETTER E WITH CARON
U+011D	ĝ	LATIN SMALL LETTER G WITH CIRCUMFLEX
U+011F	ğ	LATIN SMALL LETTER G WITH BREVE
U+0121	ġ	LATIN SMALL LETTER G WITH DOT ABOVE
U+0123	ģ	LATIN SMALL LETTER G WITH CEDILLA
U+0125	ĥ	LATIN SMALL LETTER H WITH CIRCUMFLEX
U+0127	ħ	LATIN SMALL LETTER H WITH STROKE
U+0129	ĩ	LATIN SMALL LETTER I WITH TILDE
U+012B	ī	LATIN SMALL LETTER I WITH MACRON
U+012D	ĭ	LATIN SMALL LETTER I WITH BREVE
U+012F	į	LATIN SMALL LETTER I WITH OGONEK
U+0131	ı	LATIN SMALL LETTER DOTLESS I
U+0135	ĵ	LATIN SMALL LETTER J WITH CIRCUMFLEX
U+0137	ķ	LATIN SMALL LETTER K WITH CEDILLA
U+013°	ĺ	LATIN SMALL LETTER L WITH ACUTE
U+013C	ļ	LATIN SMALL LETTER L WITH CEDILLA
U+013E	ľ	LATIN SMALL LETTER L WITH CARON
U+0140	l·	LATIN SMALL LETTER L WITH MIDDLE DOT
U+0142	ł	LATIN SMALL LETTER L WITH STROKE
U+0144	ń	LATIN SMALL LETTER N WITH ACUTE
U+0146	ņ	LATIN SMALL LETTER N WITH CEDILLA
U+0148	ň	LATIN SMALL LETTER N WITH CARON
U+0149	ʼn	LATIN SMALL LETTER N PRECEDED BY APOSTROPHE
U+014B	ŋ	LATIN SMALL LETTER ENG
U+014D	ō	LATIN SMALL LETTER O WITH MACRON
U+014F	ŏ	LATIN SMALL LETTER O WITH BREVE
U+0151	ő	LATIN SMALL LETTER O WITH DOUBLE ACUTE
U+0153	œ	LATIN SMALL LIGATURE OE
U+0155	ŕ	LATIN SMALL LETTER R WITH ACUTE
U+0157	ŗ	LATIN SMALL LETTER R WITH CEDILLA
U+0159	ř	LATIN SMALL LETTER R WITH CARON
U+015B	ś	LATIN SMALL LETTER S WITH ACUTE
U+015D	ŝ	LATIN SMALL LETTER S WITH CIRCUMFLEX
U+0161	š	LATIN SMALL LETTER S WITH CARON

U+0165	ť	LATIN SMALL LETTER T WITH CARON
U+0167	ŧ	LATIN SMALL LETTER T WITH STROKE
U+0169	ũ	LATIN SMALL LETTER U WITH TILDE
U+016B	ū	LATIN SMALL LETTER U WITH MACRON
U+016D	ŭ	LATIN SMALL LETTER U WITH BREVE
U+016F	ů	LATIN SMALL LETTER U WITH RING ABOVE
U+0171	ű	LATIN SMALL LETTER U WITH DOUBLE ACUTE
U+0173	ų	LATIN SMALL LETTER U WITH OGONEK
U+0175	ŵ	LATIN SMALL LETTER W WITH CIRCUMFLEX
U+0177	ŷ	LATIN SMALL LETTER Y WITH CIRCUMFLEX
U+017°	ź	LATIN SMALL LETTER Z WITH ACUTE
U+017C	ż	LATIN SMALL LETTER Z WITH DOT ABOVE
U+017E	ž	LATIN SMALL LETTER Z WITH CARON

19 Appendix E - Latin Extended-B Charset

Unicode Encoding	Character	Description
U+0219	ș	LATIN SMALL LETTER S WITH COMMA BELOW
U+021B	ț	LATIN SMALL LETTER T WITH COMMA BELOW

20 Appendix F - Greek Charset

Unicode Encoding	Character	Description
U+0390	ΐ	GREEK SMALL LETTER IOTA WITH DIALYTIKA AND TONOS
U+03AC	ά	GREEK SMALL LETTER ALPHA WITH TONOS
U+03AD	έ	GREEK SMALL LETTER EPSILON WITH TONOS
U+03AE	ή	GREEK SMALL LETTER ETA WITH TONOS
U+03AF	ί	GREEK SMALL LETTER IOTA WITH TONOS
U+03B0	ΰ	GREEK SMALL LETTER UPSILON WITH DIALYTIKA AND TONOS
U+03B1	α	GREEK SMALL LETTER ALPHA
U+03B2	β	GREEK SMALL LETTER BETA
U+03B3	γ	GREEK SMALL LETTER GAMMA
U+03B4	δ	GREEK SMALL LETTER DELTA
U+03B5	ε	GREEK SMALL LETTER EPSILON
U+03B6	ζ	GREEK SMALL LETTER ZETA
U+03B7	η	GREEK SMALL LETTER ETA
U+03B8	θ	GREEK SMALL LETTER THETA
U+03B9	ι	GREEK SMALL LETTER IOTA
U+03BA	κ	GREEK SMALL LETTER KAPPA
U+03BB	λ	GREEK SMALL LETTER LAMBDA
U+03BC	μ	GREEK SMALL LETTER MU
U+03BD	ν	GREEK SMALL LETTER NU
U+03BE	ξ	GREEK SMALL LETTER XI
U+03BF	ο	GREEK SMALL LETTER OMICRON
U+03C0	π	GREEK SMALL LETTER PI
U+03C1	ρ	GREEK SMALL LETTER RHO
U+03C2	ς	GREEK SMALL LETTER FINAL SIGMA
U+03C3	σ	GREEK SMALL LETTER SIGMA
U+03C4	τ	GREEK SMALL LETTER TAU
U+03C5	υ	GREEK SMALL LETTER UPSILON
U+03C6	φ	GREEK SMALL LETTER PHI
U+03C7	χ	GREEK SMALL LETTER CHI
U+03C8	ψ	GREEK SMALL LETTER PSI
U+03C9	ω	GREEK SMALL LETTER OMEGA
U+03CA	ϊ	GREEK SMALL LETTER IOTA WITH DIALYTIKA
U+03CB	ϋ	GREEK SMALL LETTER UPSILON WITH DIALYTIKA
U+03CC	ό	GREEK SMALL LETTER OMICRON WITH TONOS
U+03CD	ύ	GREEK SMALL LETTER UPSILON WITH TONOS
U+03CE	ώ	GREEK SMALL LETTER OMEGA WITH TONOS

21 Appendix G - Greek Extended Charset

Unicode Encoding	Character	Description
U+1F00	ἀ	GREEK SMALL LETTER ALPHA WITH PSILI
U+1F01	ἁ	GREEK SMALL LETTER ALPHA WITH DASIA
U+1F02	ἂ	GREEK SMALL LETTER ALPHA WITH PSILI AND VARIA
U+1F03	ἃ	GREEK SMALL LETTER ALPHA WITH DASIA AND VARIA
U+1F04	ἄ	GREEK SMALL LETTER ALPHA WITH PSILI AND OXIA
U+1F05	ἅ	GREEK SMALL LETTER ALPHA WITH DASIA AND OXIA
U+1F06	ἆ	GREEK SMALL LETTER ALPHA WITH PSILI AND PERISPOMENI
U+1F07	ἇ	GREEK SMALL LETTER ALPHA WITH DASIA AND PERISPOMENI
U+1F10	ἐ	GREEK SMALL LETTER EPSILON WITH PSILI
U+1F11	ἑ	GREEK SMALL LETTER EPSILON WITH DASIA
U+1F12	ἒ	GREEK SMALL LETTER EPSILON WITH PSILI AND VARIA
U+1F13	ἓ	GREEK SMALL LETTER EPSILON WITH DASIA AND VARIA
U+1F14	ἔ	GREEK SMALL LETTER EPSILON WITH PSILI AND OXIA
U+1F15	ἕ	GREEK SMALL LETTER EPSILON WITH DASIA AND OXIA
U+1F20	ἠ	GREEK SMALL LETTER ETA WITH PSILI
U+1F21	ἡ	GREEK SMALL LETTER ETA WITH DASIA
U+1F22	ἢ	GREEK SMALL LETTER ETA WITH PSILI AND VARIA
U+1F23	ἣ	GREEK SMALL LETTER ETA WITH DASIA AND VARIA
U+1F24	ἤ	GREEK SMALL LETTER ETA WITH PSILI AND OXIA
U+1F25	ἥ	GREEK SMALL LETTER ETA WITH DASIA AND OXIA
U+1F26	ἦ	GREEK SMALL LETTER ETA WITH PSILI AND PERISPOMENI
U+1F27	ἧ	GREEK SMALL LETTER ETA WITH DASIA AND PERISPOMENI
U+1F30	ἰ	GREEK SMALL LETTER IOTA WITH PSILI
U+1F31	ἱ	GREEK SMALL LETTER IOTA WITH DASIA
U+1F32	ἲ	GREEK SMALL LETTER IOTA WITH PSILI AND VARIA
U+1F33	ἳ	GREEK SMALL LETTER IOTA WITH DASIA AND VARIA
U+1F34	ἴ	GREEK SMALL LETTER IOTA WITH PSILI AND OXIA
U+1F35	ἵ	GREEK SMALL LETTER IOTA WITH DASIA AND OXIA
U+1F36	ἶ	GREEK SMALL LETTER IOTA WITH PSILI AND PERISPOMENI
U+1F37	ἷ	GREEK SMALL LETTER IOTA WITH DASIA AND PERISPOMENI
U+1F40	ὀ	GREEK SMALL LETTER OMICRON WITH PSILI
U+1F41	ὁ	GREEK SMALL LETTER OMICRON WITH DASIA
U+1F42	ὂ	GREEK SMALL LETTER OMICRON WITH PSILI AND VARIA
U+1F43	ὃ	GREEK SMALL LETTER OMICRON WITH DASIA AND VARIA
U+1F44	ὄ	GREEK SMALL LETTER OMICRON WITH PSILI AND OXIA
U+1F45	ὅ	GREEK SMALL LETTER OMICRON WITH DASIA AND OXIA
U+1F50	ὐ	GREEK SMALL LETTER UPSILON WITH PSILI
U+1F51	ὑ	GREEK SMALL LETTER UPSILON WITH DASIA
U+1F52	ὒ	GREEK SMALL LETTER UPSILON WITH PSILI AND VARIA
U+1F53	ὓ	GREEK SMALL LETTER UPSILON WITH DASIA AND VARIA
U+1F54	ὔ	GREEK SMALL LETTER UPSILON WITH PSILI AND OXIA
U+1F55	ὕ	GREEK SMALL LETTER UPSILON WITH DASIA AND OXIA
U+1F56	ὖ	GREEK SMALL LETTER UPSILON WITH PSILI AND PERISPOMENI
U+1F57	ὗ	GREEK SMALL LETTER UPSILON WITH DASIA AND PERISPOMENI

U+1F60	ὠ	GREEK SMALL LETTER OMEGA WITH PSILI
U+1F61	ὡ	GREEK SMALL LETTER OMEGA WITH DASIA
U+1F62	ὢ	GREEK SMALL LETTER OMEGA WITH PSILI AND VARIA
U+1F63	ὣ	GREEK SMALL LETTER OMEGA WITH DASIA AND VARIA
U+1F64	ὤ	GREEK SMALL LETTER OMEGA WITH PSILI AND OXIA
U+1F65	ὥ	GREEK SMALL LETTER OMEGA WITH DASIA AND OXIA
U+1F66	ὦ	GREEK SMALL LETTER OMEGA WITH PSILI AND PERISPOMENI
U+1F67	ὧ	GREEK SMALL LETTER OMEGA WITH DASIA AND PERISPOMENI
U+1F70	ὰ	GREEK SMALL LETTER ALPHA WITH VARIA
U+1F71	ά	GREEK SMALL LETTER ALPHA WITH OXIA
U+1F72	ὲ	GREEK SMALL LETTER EPSILON WITH VARIA
U+1F73	έ	GREEK SMALL LETTER EPSILON WITH OXIA
U+1F74	ὴ	GREEK SMALL LETTER ETA WITH VARIA
U+1F75	ή	GREEK SMALL LETTER ETA WITH OXIA
U+1F76	ὶ	GREEK SMALL LETTER IOTA WITH VARIA
U+1F77	ί	GREEK SMALL LETTER IOTA WITH OXIA
U+1F78	ὸ	GREEK SMALL LETTER OMICRON WITH VARIA
U+1F79	ό	GREEK SMALL LETTER OMICRON WITH OXIA
U+1F7A	ὺ	GREEK SMALL LETTER UPSILON WITH VARIA
U+1F7B	ύ	GREEK SMALL LETTER UPSILON WITH OXIA
U+1F7C	ὼ	GREEK SMALL LETTER OMEGA WITH VARIA
U+1F7D	ώ	GREEK SMALL LETTER OMEGA WITH OXIA
U+1F80	ᾀ	GREEK SMALL LETTER ALPHA WITH PSILI AND YPOGEGRAMMENI
U+1F81	ᾁ	GREEK SMALL LETTER ALPHA WITH DASIA AND YPOGEGRAMMENI
U+1F82	ᾂ	GREEK SMALL LETTER ALPHA WITH PSILI AND VARIA AND YPOGEGRAMMENI
U+1F83	ᾃ	GREEK SMALL LETTER ALPHA WITH DASIA AND VARIA AND YPOGEGRAMMENI
U+1F84	ᾄ	GREEK SMALL LETTER ALPHA WITH PSILI AND OXIA AND YPOGEGRAMMENI
U+1F85	ᾅ	GREEK SMALL LETTER ALPHA WITH DASIA AND OXIA AND YPOGEGRAMMENI
U+1F86	ᾆ	GREEK SMALL LETTER ALPHA WITH PSILI AND PERISPOMENI AND YPOGEGRAMMENI
U+1F87	ᾇ	GREEK SMALL LETTER ALPHA WITH DASIA AND PERISPOMENI AND YPOGEGRAMMENI
U+1F90	ᾐ	GREEK SMALL LETTER ETA WITH PSILI AND YPOGEGRAMMENI
U+1F91	ᾑ	GREEK SMALL LETTER ETA WITH DASIA AND YPOGEGRAMMENI
U+1F92	ᾒ	GREEK SMALL LETTER ETA WITH PSILI AND VARIA AND YPOGEGRAMMENI
U+1F93	ᾓ	GREEK SMALL LETTER ETA WITH DASIA AND VARIA AND YPOGEGRAMMENI
U+1F94	ᾔ	GREEK SMALL LETTER ETA WITH PSILI AND OXIA AND YPOGEGRAMMENI
U+1F95	ᾕ	GREEK SMALL LETTER ETA WITH DASIA AND OXIA AND YPOGEGRAMMENI
U+1F96	ᾖ	GREEK SMALL LETTER ETA WITH PSILI AND PERISPOMENI

		AND YPOGEGRAMMENI
U+1F97	ῇ	GREEK SMALL LETTER ETA WITH DASIA AND PERISPOMENI AND YPOGEGRAMMENI
U+1FA0	ᾠ	GREEK SMALL LETTER OMEGA WITH PSILI AND YPOGEGRAMMENI
U+1FA1	ᾡ	GREEK SMALL LETTER OMEGA WITH DASIA AND YPOGEGRAMMENI
U+1FA2	ᾢ	GREEK SMALL LETTER OMEGA WITH PSILI AND VARIA AND YPOGEGRAMMENI
U+1FA3	ᾣ	GREEK SMALL LETTER OMEGA WITH DASIA AND VARIA AND YPOGEGRAMMENI
U+1FA4	ᾤ	GREEK SMALL LETTER OMEGA WITH PSILI AND OXIA AND YPOGEGRAMMENI
U+1FA5	ᾥ	GREEK SMALL LETTER OMEGA WITH DASIA AND OXIA AND YPOGEGRAMMENI
U+1FA6	ᾦ	GREEK SMALL LETTER OMEGA WITH PSILI AND PERISPOMENI AND YPOGEGRAMMENI
U+1FA7	ᾧ	GREEK SMALL LETTER OMEGA WITH DASIA AND PERISPOMENI AND YPOGEGRAMMENI
U+1FB0	ᾰ	GREEK SMALL LETTER ALPHA WITH VRACHY
U+1FB1	ᾱ	GREEK SMALL LETTER ALPHA WITH MACRON
U+1FB2	ᾲ	GREEK SMALL LETTER ALPHA WITH VARIA AND YPOGEGRAMMENI
U+1FB3	ᾳ	GREEK SMALL LETTER ALPHA WITH YPOGEGRAMMENI
U+1FB4	ᾴ	GREEK SMALL LETTER ALPHA WITH OXIA AND YPOGEGRAMMENI
U+1FB6	ᾶ	GREEK SMALL LETTER ALPHA WITH PERISPOMENI
U+1FB7	ᾷ	GREEK SMALL LETTER ALPHA WITH PERISPOMENI AND YPOGEGRAMMENI
U+1FC2	ῂ	GREEK SMALL LETTER ETA WITH VARIA AND YPOGEGRAMMENI
U+1FC3	ῃ	GREEK SMALL LETTER ETA WITH YPOGEGRAMMENI
U+1FC4	ῄ	GREEK SMALL LETTER ETA WITH OXIA AND YPOGEGRAMMENI
U+1FC6	ῆ	GREEK SMALL LETTER ETA WITH PERISPOMENI
U+1FC7	ῇ	GREEK SMALL LETTER ETA WITH PERISPOMENI AND YPOGEGRAMMENI
U+1FD0	ῐ	GREEK SMALL LETTER IOTA WITH VRACHY
U+1FD1	ῑ	GREEK SMALL LETTER IOTA WITH MACRON
U+1FD2	ῒ	GREEK SMALL LETTER IOTA WITH DIALYTIKA AND VARIA
U+1FD3	ΐ	GREEK SMALL LETTER IOTA WITH DIALYTIKA AND OXIA
U+1FD6	ῖ	GREEK SMALL LETTER IOTA WITH PERISPOMENI
U+1FD7	ῗ	GREEK SMALL LETTER IOTA WITH DIALYTIKA AND PERISPOMENI
U+1FE0	ῠ	GREEK SMALL LETTER UPSILON WITH VRACHY
U+1FE1	ῡ	GREEK SMALL LETTER UPSILON WITH MACRON
U+1FE2	ῢ	GREEK SMALL LETTER UPSILON WITH DIALYTIKA AND VARIA
U+1FE3	ΰ	GREEK SMALL LETTER UPSILON WITH DIALYTIKA AND OXIA
U+1FE4	ῤ	GREEK SMALL LETTER RHO WITH PSILI
U+1FE5	ῥ	GREEK SMALL LETTER RHO WITH DASIA
U+1FE6	ῦ	GREEK SMALL LETTER UPSILON WITH PERISPOMENI
U+1FE7	ῧ	GREEK SMALL LETTER UPSILON WITH DIALYTIKA AND

		PERISPOMENI
U+1FF2	ὼ	GREEK SMALL LETTER OMEGA WITH VARIA AND YPOGEGRAMMENI
U+1FF3	ῳ	GREEK SMALL LETTER OMEGA WITH YPOGEGRAMMENI
U+1FF4	ώ	GREEK SMALL LETTER OMEGA WITH OXIA AND YPOGEGRAMMENI
U+1FF6	ῶ	GREEK SMALL LETTER OMEGA WITH PERISPOMENI
U+1FF7	ῷ	GREEK SMALL LETTER OMEGA WITH PERISPOMENI AND YPOGEGRAMMENI

22 Appendix H - Cyrillic Charset

Unicode Encoding	Character	Description
U+0430	а	CYRILLIC SMALL LETTER A
U+0431	б	CYRILLIC SMALL LETTER BE
U+0432	в	CYRILLIC SMALL LETTER VE
U+0433	г	CYRILLIC SMALL LETTER GHE
U+0434	д	CYRILLIC SMALL LETTER DE
U+0435	е	CYRILLIC SMALL LETTER IE
U+0436	ж	CYRILLIC SMALL LETTER ZHE
U+0437	з	CYRILLIC SMALL LETTER ZE
U+0438	и	CYRILLIC SMALL LETTER I
U+0439	й	CYRILLIC SMALL LETTER SHORT I
U+043A	к	CYRILLIC SMALL LETTER KA
U+043B	л	CYRILLIC SMALL LETTER EL
U+043C	м	CYRILLIC SMALL LETTER EM
U+043D	н	CYRILLIC SMALL LETTER EN
U+043E	о	CYRILLIC SMALL LETTER O
U+043F	п	CYRILLIC SMALL LETTER PE
U+0440	р	CYRILLIC SMALL LETTER ER
U+0441	с	CYRILLIC SMALL LETTER ES
U+0442	т	CYRILLIC SMALL LETTER TE
U+0443	у	CYRILLIC SMALL LETTER U
U+0444	ф	CYRILLIC SMALL LETTER EF
U+0445	х	CYRILLIC SMALL LETTER HA
U+0446	ц	CYRILLIC SMALL LETTER TSE
U+0447	ч	CYRILLIC SMALL LETTER CHE
U+0448	ш	CYRILLIC SMALL LETTER SHA
U+0449	щ	CYRILLIC SMALL LETTER SHCHA
U+044A	ъ	CYRILLIC SMALL LETTER HARD SIGN
U+044B	ы	CYRILLIC SMALL LETTER YERU
U+044C	ь	CYRILLIC SMALL LETTER SOFT SIGN
U+044D	э	CYRILLIC SMALL LETTER E
U+044E	ю	CYRILLIC SMALL LETTER YU
U+044F	я	CYRILLIC SMALL LETTER YA