Фонд содействия развитию технологий и инфраструктуры Интернета
Введено в действие: 27 ноября 2014 г.

Foundation for Assistance for Internet Technologies and Infrastructure Development

Specifications for Registrars' Interaction with the Domain Registration System During the General Registration Period With Claims Services

Version 1.3

2014

Table of Contents

1. Terms and definitions

Attribute is a string (field) of information that has a predefined identifier.

Object is a given set of attributes that has a unique identifier in the Registry.

Registry is the collection of records structured as objects and stored in the database.

Subordinate level means a level that is lower than the previous level. For example, a second-level domain is subordinate to a top-level domain, while a third-level domain is subordinate to a second-level domain, and so on.

Domain Registration System is a set of hardware and software intended for registration of domains in the Registry, storage of information about domain registrants, and delegation of domains.

ICANN is an international non-profit organization responsible for discussing and developing policies regulating technical matters of the coordination of the Internet domain name system.

Trademark / service mark is a registered object of intellectual property. Trademark must be registered by the Federal Service for Intellectual Property
(Rospatent) or by the World Intellectual Property Organization (WIPO).

Trademark Clearinghouse (TMCH) is the centralized depository of trademarks registered in ICANN.

Registry Operator is the administrator of the top-level domain authorized to develop rules and policies for domain name registration.

Registrar is a registrar accredited by ICANN for providing domain name registration services in the top-level domains.

Registrant is the administrator of a domain name and owner of rights for domain name management.

2. Introduction

This document provides information required for interaction of registrars with the Domain Registration System during the General Registration Period with Claims Services.

If you have any questions on this document, please contact us at adm.contact@faitid.org.

Registrars get access to two Registration Systems: Test Registration System and Operational Registration System.

Operational Registration System stores all data on registered domains and generates a zone file for delegation of domains on DNS servers.

3. Getting Access to Domain Registration System

To get access to the Domain Registration System, a registrar shall:

- Provide required information about its entity to Registry Operator and execute an agreement for getting access to the domain name registry with Registry Operator of the top-level domain. After executing the agreement, Registry Operator will provide authentication details and other information for getting access to the Test Registration System.
- Undergo the process of operational test and evaluation, which is necessary to test the interaction of registrar's software with the Registration System. The testing will be done using

the Test Registration System in accordance with the Operational Test and Evaluation Certification Policy published on the website of Registry Operator at the address specified in the Technical Policy of the Registry of .REGISTRY TLD, where .REGISTRY is the name of the top-level domain.

- If the testing is successful, Registry Operator will activate the registrar's access to the Operational Registration System and give to the registrar the access details.

Access to the Test Registration System remains for the effective term of the agreement between the registrar and Registry Operator.

4. Features of Domain Registration System

4.1. Requirements for implementation

Registration System is implemented in accordance with ICANN requirements and the following standards:

- RFC 5730 – Extensible Provisioning Protocol (EPP).
- RFC 5731 – EPP-Domain Name Mapping.
- RFC 5732 – Extensible Provisioning Protocol (EPP) Host Mapping.
- RFC 5733 – Extensible Provisioning Protocol (EPP) Contact Mapping.
- RFC 3915 – Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP).
- RFC 3735 – Guidelines for Extending the Extensible Provisioning Protocol (EPP).
- RFC 4033 – DNS Security Introduction and Requirements.
- RFC 6480 – Clarifications and Implementation Notes for DNS Security (DNSSEC).
- IRTP (Inter-Registrar Transfer Policy). The requirements from the following documents are also met:
- draft-lozano-tmch-smd-03 – Mark and Signed Mark Objects Mapping;
- draft-tan-epp-launchphase-11 – Launch Phase Mapping for the Extensible Provisioning Protocol (EPP).
- draft-ar-tmch-epp-mapping-02 – Trademark Clearinghouse Extension Mapping for the Extensible Provisioning Protocol (EPP).

4.2. Composition of Registration System

The Operational Registration System is composed of two identical nodes (for the purpose of redundancy).

Registrar's queries to the Registration System go the Application Servers. EPP interface is used for interaction with the domain name registry. Web interface is used for access to the Registrar's account manager.

Application Servers exchange information with the database of the domain name registry.

Data stored in the registry is used to generate a zone file that is transferred to the DNS servers in order to delegate the registered domains.

To give information about registered domains to the Internet users, some of the information from the registry is copied to the Whois server.

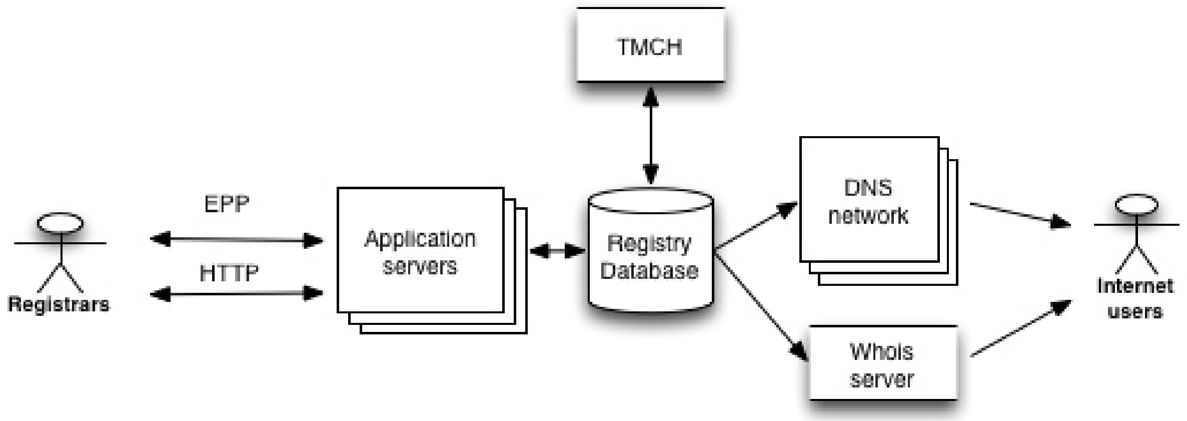

Figure 1

Test Registration System is similar to Operational Registration System. Test Registration System is intended for registrar's operational test and evaluation. Domains registered in this system will not be actually delegated.

4.3. Application Servers

Application Servers are used to organize interaction of registrars with the database containing information of the registries of top-level domains. Application Servers use Web and EPP interfaces of the Registration System.

4.3.1. Database

This database contains information of the registries of top-level domains. This database is intended for:

- storing information about registrars;
- checking if domain names that are to be registered match trademarks included in the TMCH and exchanging that information with the TMCH;
- registration of second-level domains;
- storing information about registrants, so that they could be identified;
- monitoring the registration terms;
- storing information about DNS servers to which registered domains are delegated;
- monitoring the time frames for storing information and performing procedures on information deletion;
- managing delegation of domains;
- generating a signed root zone file and public keys; support of DNSSEC;
- informing the Internet users about registered domains and their registrants;
- updating information about domain names, registrants, DNS servers, registrars.

This database is intended for performance of the following operations pursuant to requests from registrars:

- o checking domain name registrations;
- o registering Contact objects;
- o getting information from Contact objects;

- changing information in Contact objects;
- deleting Contact objects;
- registering Domain objects;
- getting information from Domain objects;
- changing information in Domain objects;
- transferring Domain objects to other registrars;
- deleting Domain objects;
- redeeming Domain objects during Redemption Grace period;
- registering Host objects;
- getting information from Host objects;
- changing information in Host objects;
- deleting Host objects;
- managing delegation of domains.

4.4. DNS Network

Domains are delegated to the networks of DNS nodes located in different regions and countries, which ensures 100-percent availability of the service.

Information in zone files is updated at least once per hour.

4.5. DNSSEC Support

All zone files of top-level domains supported by the Registration System are signed using DNSSEC. Policy of using the DNSSEC for the TLD is published in the Technical Policy of the Registry of .REGISTRY TLD.

4.6. Whois Service

The Registration System includes Whois servers that may be used by any Internet user for getting information about registered domains.

5. Interfaces of Registration System

Registrars interact with the Registration System through the following interfaces:

- Web interface for access to the registrar's account manager
- EPP interface for access to the Registration System
- Whois interface for access to the reference system with information about registered domains.

5.1. Limitations of Access Through Interfaces

Access to the Registration System through EPP and Web interfaces is limited by IP addresses and SSL certificates. Initially, the registrar gets access through IP addresses it lists when executing the Registry-Registrar Agreement. The registrar may later change this information by submitting a request to the technical support service.

Specific parameters of access limitation are shown in the Technical Policy of the Registry of .REGISTRY TLD.

Users' access to the Whois interface may be restricted when the frequency of requests is above the value that may overload the server (in case of a network attack). Restrictions are implemented pursuant to the publicly available Whois Service Terms of Use. Registrars with fixed IP addresses get unlimited access to a separate Whois server.

5.2. Web Interface

Web interface is implemented on the Application Server and intended for registrar's access to its account manager with information about:

- Operations performed by the registrar.
- Money transferred by the registrar to its personal account in the Registration System, use of money for getting the services, and available balance.
- Events related to the operation of the domain registration system.

5.3. EPP Interface

EPP interface of the Registration System is implemented on the Application Server and intended for registrar's interaction with the Registration System through the Extensible Provisioning Protocol (EPP).

Access through this interface will be provided only after registrar's authentication in the Registration System.

Interaction through EPP is done with information blocks structured in accordance with the XML specification. Three types of blocks are used:

- Request. To perform a command, the registrar sends to the Registration System an information block containing an object's identifier, a command to be performed with this object, and, if necessary, parameters.
- Response to request. In response to a request, the registry returns to the registrar an information block with the result of command performance, containing either confirmation of command performance, or an error message with the code of error.
- Notification. An information block generated by the Registration System for the registrar if any event in the registry requires a notification for the registrar.

Each subsequent request to the Registration System may be submitted only after a response to the previous request.

EPP extensions used for each top-level domain and maximum number of requests to the Registration System per unit of time are specified in the Technical Policy of the Registry of .REGISTRY TLD.

EPP extensions are described in this document and in Addendum 1 (Description of EPP Extensions).

5.4. Whois Interface

Whois interface is implemented on the Whois server in accordance with RFC 3912. Its purpose is to inform the Internet users about second-level domain names registered in the TLD, as well as their registrants and statuses.
For each of the TLDs supported by Registry Operator, Whois server is available at WHOIS.NIC.TLD, where TLD is the name of a top-level domain (including IDN). Interfaces are available through port 43 and port 80 (Web-based Whois).

6. Registry Objects of a Top-Level Domain

The registry of a top-level domain contains four types of objects:

- Registrar. Contains information about registrar.
- Domain. Contains information about domain name, status of delegation, and connections with Registrar, Contact, and Host objects.

- Contact. Three types of contacts are supported:
    - Registrant. Contains information about domain registrant; mandatory for registration.
    - Admin. Contains information about administrative contact.
    - Tech. Contains information about technical contact.
- Host. Contains information about the DNS server that may be used for domain delegation and connections with Registrar and Domain objects.

All objects are implemented pursuant to the documents listed in clause 4.1.

Model of object connections:

- Registrar may manage the set of unique Domain, Contact, Host objects.
- Domain object managed by the registrar must be connected only with one Registrant-type Contact object;
- Domain object managed by the registrar may be connected with one or many Admin-type and Tech-type Contact objects;
- Domain object managed by the registrar may be connected with unlimited number of Host objects.
- Contact object managed by the registrar may be connected with many Domain objects of this registrar.
- Host object managed by the registrar may be connected with many Domain objects of this registrar.

Contact and Host objects not connected with any Domain object will be deleted from the registry.

Every object in the registry has a mandatory attribute – a set of statuses.

Statuses starting with "server" are set and removed by the server procedures of the Registration System (server statuses).

Statuses starting with "client" may be set and removed by the registrar (client statuses) using the requests to the registry.

Object's statuses define its state, and whether or not certain operations with this objects may be performed.

7. Registrar Object

Registrar object contains information about domain name registrar that has access to the Registration System. This object is created by Registry Operator (administrator) of a top-level domain when giving access to such registrar.

Registrar's identifier in the registry assigned by Registry Operator is used as this object's identifier. This identifier is used by the database to identify the registrar that manages other registry objects.

8. Domain Object

Identifier of Domain object is the name of domain chosen by the registrar when submitting a registration request. Requirements for the name of domain are subject to the Terms and Conditions of Registration in a particular TLD, which are specified in the Technical Policy of the Registry of .REGISTRY TLD.

Registrar's request for registration of a Domain object will not be satisfied and will return an error if the registry already has a Domain object with such identifier. This applies to any limited registration period during which a domain is registered.

8.1. Domain Object Life Cycle

Domain object life cycle (hereinafter referred to as the "domain life cycle") includes the following main periods:

- Registration period;
- Redemption grace period;
- Pending delete period;

Main periods of the domain life cycle are provided in Figure 2:

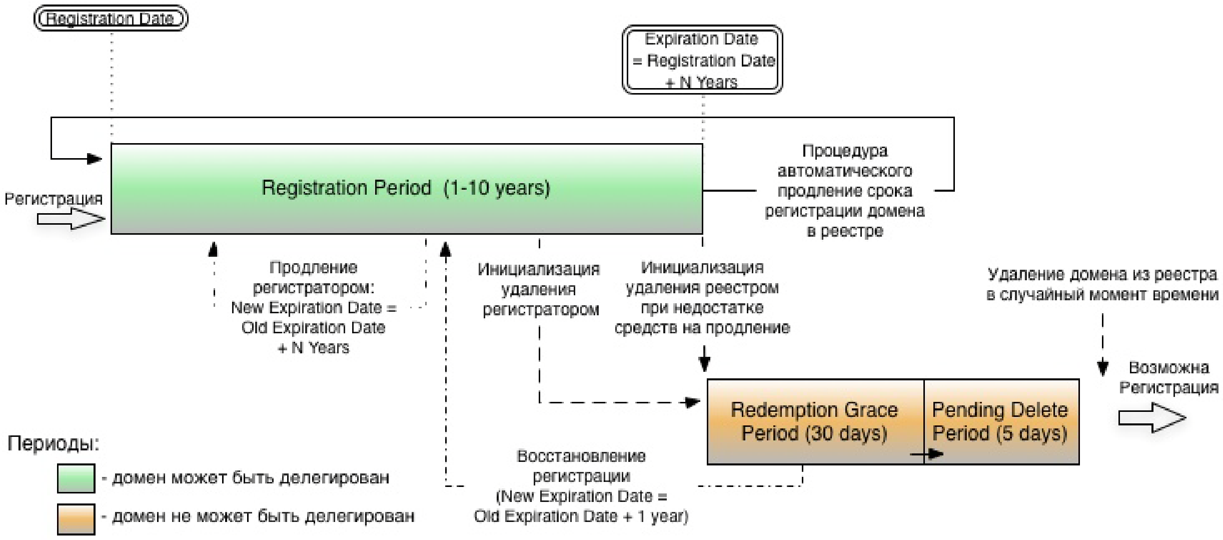
Registration Date
Expiration Date = Registration Date + N Years
Регистрация
Registration Period (1-10 years)
Процедура автоматического продление срока регистрации домена в реестре
Продление регистратором: New Expiration Date = Old Expiration Date + N Years
Инициализация удаления регистратором
Инициализация удаления реестром при недостатке средств на продление
Удаление домена из реестра в случайный момент времени
Возможна Регистрация
Redemption Grace Period (30 days)
Pending Delete Period (5 days)
Восстановление регистрации (New Expiration Date = Old Expiration Date + 1 year)
Периоды:
🟩 - домен может быть делегирован
🟧 - домен не может быть делегирован

Figure 2

Each period of the domain life cycle is characterized by its set of domain statuses and operations that may be executed with the domain.

During the Registration Period domain objects **may** have the following EPP statuses: ok, or inactive, or clientHold, clientRenewProhibited, clientTransferProhibited, clientUpdateProhibited.

During the Redemption Grace and Pending Delete periods domain objects **will** have the following EPP statuses: serverHold, pendingDelete, serverRenewProhibited, serverTransferProhibited, serverUpdateProhibited.

Main periods of the domain object life cycle may be combined with a set of additional periods:

- Auto Renew Grace Period;
- Renew Grace Period;
- Transfer Grace Period;
- Pending Restore Period;
- Pending Transfer Period.

8.1.1. Main Registration Period

Upon domain registration (from the moment of domain record entry) the Registration Period starts. The duration of the registration period must be a multiple of one (1) year and may last from one (1) year to ten (10) years.

Domain name expiration date is stored in the Expiration date attribute. The date format must be as follows: DD.MM.YYYY hh:mm:ss, Moscow time. All dates in responses to requests will be given in UTC. Domain name expiration date will be changed after domain renewal, automatic domain renewal, domain transfer with renewal and domain recovery with automatic renewal, as well as after renewal cancellation accompanying domain name deletion within Grace Period. Domain name expiration date is changed by modifying YYYY parameter of Expiration date.

Upon registration, all operations on a registered domain, except for domain recovery and expressly prohibited by the specified statuses, are available for the Registrar.

Setting serverDeleteProhibited status for the domain may increase the duration of the registration period - in this case, registration period will expire on the next day, after removing serverDeleteProhibited status (provided that registration period has already expired at that moment).

8.1.2. Main Redemption Grace Period

When domain deletion from the Registry is initialized by the Registrar's EPP command <delete> or by server procedure, domain delegation shall be terminated and Redemption Grace Period (RGP) will start for the domain name.

RGP lasts thirty (30) days. RGP will be terminated ahead of time upon execution of domain redemption operation.

During RGP the Registrar may use domain redemption operation with automatic 1 year renewal.

8.1.3. Pending Delete Period

Upon completion of RGP, Pending Delete Period starts. Pending Delete Period lasts five (5) days and is completed by domain name deletion from the Registry.

During Pending Delete Period any operations on a domain name are unavailable for the Registrar.

8.1.4. Auto Renew Grace Period

Upon completion of the domain registration period and if there is enough money available on the Registrar's personal account, server procedure for automatic domain renewal for 1 year will be completed and a new registration period will start.

Auto Renew Grace Period (ARGP) will start simultaneously.

ARGP lasts forty-five (45) days. ARGP will be terminated ahead of time upon execution of domain renewal and domain deletion operations by the Registrar. When domain is deleted by the Registrar within ARGP, funds allocated by the automatic renewal procedure for the domain renewal will be remitted to the Registrar's Agreement Account (Registrar's personal account).

During operation of domain transfer to another registrar ARGP will be completed at the time of transfer operation completion.

8.1.5. Renew Grace Period

Renew Grace Period (RenewGP) starts after the domain is renewed by the Registrar with the use of EPP command <renew>.

RenewGP lasts five (5) days. RenewGP will be terminated ahead of time upon domain deletion or successful completion of its transfer.
Domain renewal is permitted within RenewGP. In this case the previous RenewGP is interrupted and a new RenewGP starts. Domain deletion within RenewGP will result in a refund of all renewals, if neither RenewGP is completed by the expiry of a 5-day period.

When domain deletion is initialized upon the Registrar's request in that period, domain renewal will be canceled and renewal fee will be returned to the Registrar's personal account. In this case, domain registration term will decrease by the renewal term.

When domain is transferred to another registrar within RenewGP, Transfer Grace Period will start and RenewGP will be interrupted. In this case, renewal fee will not be refunded and it will not effect the domain registration term.

8.1.6. Transfer Grace Period

Transfer Grace Period (TGP) starts upon completion of the domain transfer to another registrar.

TGP lasts five (5) days. TGP will be terminated ahead of time upon domain deletion.

Domain renewal upon the Registrar's request within TGP will launch RenewGP in parallel.

When domain deletion is initialized upon the Registrar's request within TGP, transfer fee will be returned to the gaining Registrar's personal account, and domain registration term will decrease by one (1) year.

If domain is in TGP and RenewGP simultaneously, and domain deletion is initialized upon the Registrar's request, domain renewal fee and fee for domain transfer to another registrar will be returned to the Registrar's personal account.

During TGP all operations on a domain, except for domain redemption from RGP, domain transfer to another registrar and expressly prohibited by the Registry by specified server statuses, are available for the Registrar.

8.1.7. Pending Transfer Period

Pending Transfer Period starts upon receipt of the request to transfer a domain to another registrar by the Registry.

Pending Transfer Period lasts five (5) days. Pending Transfer Period will be terminated ahead of time upon transfer or failure to transfer a domain to another registrar.

During Pending Transfer Period the following operations: domain renewal, domain deletion, change of registrar and registrant are prohibited to the transferring registrar.

8.1.8. Pending Restore

Pending Restore Period starts upon the receipt of the EPP command <update restore_request> from the Registrar.

Pending Restore Period lasts five (5) days. Pending Restore Period will be terminated ahead of time upon the receipt of the EPP command <update restore_report> from the Registrar to complete domain restore operation.

During Pending Restore Period any modifying operations on a domain name are unavailable for the Registrar, except for EPP command <update restore_report> containing description of the reason for domain restore.

8.2. Operations With Domain Objects

8.2.1. Domain registration

Domain registration is performed in the registry stored in the database.

Domain registration term starts from the moment of domain registration.

To create a Domain object (to register a domain) in the top-level domain Registry the Policy Guide for a corresponding TLD Registry, which impose requirements on the composition of characters allowed in a domain name, the minimum amount of characters in a domain name, etc., should be complied.

Before registering a domain name the Registrar shall generate a unique access code (Authinfo code) and provide it to the Registrant. Registrant may later use this access code for transferring domain to another registrar. Registrar shall specify this unique code in the request for registration of a domain name.

To register a domain Registrant Contact object with Registrant's information should be previously created in the Registry.

Domain may have one mandatory reference to the Registrant Contact object and up to three references to the different types of Contact objects.

Domain may be registered for a period of one (1) year to ten (10) years depending on the registration term specified in the request.

Domains are registered using the <domain:create> request. Registration is performed in Claims operating mode of the registration system as follows:

- Registrar shall check if the domain is registered in the Registry using the <domain:check> request;
- if the domain is already registered, the Registrant may not register the domain.
- if the domain is not registered, the Registrar shall analyze launch:claimKey attribute received in response to <domain:check> request;
- if launch:claimKey is empty, the Registrar may register the domain using <domain:create> command without specifying <extension>;
- if launch:claimKey is not empty, the Registrar shall contact a third party - TMCH service and according to a launch:claimKey identifier get a warning Claim Notice message and the unique identifier of the message - launch:noticeID. Then the Registrar is obliged to make this message available for the Registrant, and upon the Registrant's consent continue registering the application by <domain:create> command by adding launch:noticeID attribute to the request. If there is no consent from the Registrant, the domain shall not be registered.

Examples of requests and responses are given in clause 12.4.

8.2.2. Getting domain information

Registrar may get information about attribute values of the registered domain using <domain:info> command.

8.2.3. Updating domain attributes

During the registration period the registrar can change the Domain object attributes if such operation is not prohibited by serverUpdateProhibited or clientUpdateProhibited statuses.

The following attributes may be updated: client statuses, links to Contact objects, links to Host objects, and DNSSec parameters. Update is performed via <domain:update> request.

8.2.4. Domain delegation

Domain name delegation is exercised by the registry automatically by creating a top-level domain zone file and its placement on DNS servers. Delegation may be performed during the registration period and, at the registrar's discretion, during ARGP.

The domain will be delegated if the following conditions are met:

- the domain name has been registered in the registry;
- the domain name is linked to two or more DNS servers (the Domain object is related to two or more Host objects);
- if the indicated DNS server is subordinate to the registered domain name, IP addresses should be specified for such DNS server;
- statuses prohibiting the domain name delegation (serverHold and clientHold) should not be enabled for the domain name.

8.2.5. Domain deletion

Domain name deletion is the process of the domain name registration cancellation in the registry.

The domain name deletion process is initialized after:

- EPP registrar's request <domain:delete> has been received;
- the server procedure has automatically renewed the domain name, but the funds available on the registrar's personal account are insufficient to render the service on domain name renewal for a period of 1 year.

The registrar can initialize the domain name deletion during the registration period provided, however, that the domain name is not in the process of outbound transfer to another registrar (Pending Transfer period).

Initialization of the domain name deletion can be canceled by the registrar during RGP period by a sequence of EPP requests <update restore_request> and <update restore_report> (domain name registration redemption).

After the initialization is complete the domain name acquires "pendingDelete" status and Redemption Grace Period (RGP) starts.

The enabled serverDeleteProhibited or clientDeleteProhibited statuses prohibit to run a deletion initialization request.

The domain name is deleted from the registry by the server procedure upon expiration of the Pending Delete period. When the domain name is deleted, the money is not refunded to the registrar.

8.2.6. Domain renewal

Domain name can be renewed:

- by the registrar during the registration period (at rendering service on domain name renewal) by EPP request <domain:renew> for the period of 1 to 10 years inclusive (depending on the renewal term indicated in the request) provided that the domain name registration term does not exceed 10 years;
- for the period of 1 year (upon expiry of the domain name registration period) in the domain name automatic renewal procedure provided that the funds available on the registrar's personal account are sufficient to pay for the service;
- for the period of 1 year at rendering services on domain name registration redemption from RGP (upon its successful completion) provided that the domain name registration term does not exceed 10 years.

serverRenewProhibited or clientRenewProhibited statuses prohibit execution of domain name renewal requests by the registrar and domain name registration term automatic renewal procedure.

8.2.7. Domain outbound transfer to another registrar

The domain outbound transfer is initiated by EPP gaining registrar's request <transfer request> with AuthInfo code.

In this case the following occurs in the registry:

- "pendingTransfer" status is enabled for the domain name and Pending Transfer period starts;
- by enabling the corresponding server statuses, operations on initializing domain name deletion, renewal, outbound transfer are prohibited;
- the registrant change is prohibited by updating the domain name Contact-Id attribute;
- the losing registrar is informed on the receipt of the outbound transfer request.

During Pending Transfer period:

- the losing registrar may submit a request confirming acceptance of the domain name under the gaining registrar's control, in this case the Pending Transfer period is discontinued immediately and the domain name is transferred to the gaining registrar;
- the losing registrar may submit a request with refusal to transfer the domain name, in this case the Pending Transfer period is discontinued immediately and the domain name remains under control of the losing registrar;
- the gaining registrar may submit a request which stops execution of the domain name outbound transfer to the gaining registrar, in this case the Pending Transfer period is discontinued immediately and the domain name remains under control of the losing registrar.

If during Pending Transfer period neither losing, nor gaining registrar submit any request, then when the period expires the domain name is transferred to the gaining registrar.

At rendering the outbound transfer service, the registration period may be extended for 1 year provided that the domain name registration term does not exceed 10 years. In this case the funds for rendering the service will be deducted from the registrar's personal account regardless of whether the domain registration period has been extended or not.

serverTransferProhibited or clientTransferProhibited statuses prohibit execution of the outbound transfer request.

serverTransferProhibited status is enabled automatically by the server procedure at the moment of the domain name registration and at the moment of the outbound transfer. serverTransferProhibited status is disabled automatically by the server procedure in 60 days after the events.

8.2.8. Domain name registration redemption from Redemption Grace Period

The registrar can cancel initialization of the domain name deletion process and redeem the domain name registration during Redemption Grace Period (RGP).

Domain name redemption is initialized by <update restore_request> request. In this case the domain name acquires "pendingRestore" status and Pending Restore period starts.

During Pending Restore period the registrar may send <update restore_report> request containing information on the reasons for domain name redemption. In this case:

- Pending Restore period discontinues and pendingRestore status is disabled;
- the domain name registration is redeemed while preserving the domain name registration date, Expiration date, links to Contact and Host objects as of the moment preceding execution of the domain name deletion initialization operation;
- the domain name is renewed for 1 year provided that the domain name registration period does not exceed 10 years. When the registration renewal procedure is performed serverRenewProhibited and clientRenewProhibited statuses are ignored and RenewGP is not counted;
- the domain name renewal service shall be deemed provided regardless of whether the domain name registration period has been extended or not.

The domain name registration redemption service may be provided only if the funds available on the registrar's personal account are sufficient to pay for the service.

If no <update restore_report> request has been submitted by the registrar during Pending Restore period, then upon its expiry the domain name registration redemption operation is no longer available. If the time period between termination date and RGP expiry date is less than Pending Restore period duration, RGP is discontinued immediately and Pending Delete period starts counting.

9. Contact object

The object identifier is assigned by the registrar at creating a request for object registration in the registry. The identifier shall consist of a sequence of ASCII case-insensitive characters.

The registrar's request to the registry to register Contact object will not be performed or finished with errors, if an object with such ID is already available in the registry.

9.1. Contact object life cycle

Only one period is available for Contact object – Registration Period of indeterminate duration. The registration period starts after the Contact object has been created in the registry at the registrar's request <create> to the registry.

The Contact object registration period finishes under the following circumstances:

- the object has been deleted by the registrar using <delete> request;
- the object has been deleted by server procedures in 20 days after registration in the registry if during the indicated time period neither of the registered domain names had links to the object, or at outbound transfer - if neither of domain names under the losing registrar's control had links to the Contact object.

9.2. Creating, updating, outbound transfer and deleting the Contact object

The object is created by <contact:create> request and updated by <contact:update> request. The request shall include information on the sub-type which the given object is related to, <person> or <organization>, see Appendix No. 1 hereto for Description of EPP Extensions .

As the result of outbound transfer of a domain name which has links to the Contact object, the object becomes managed by the gaining registrar. In this case, if:

- neither of the losing registrar' domain names has links to the Contact object being transferred, the latter retains its Contact-Id during the outbound transfer. The losing registrar is not permitted to manage and obtain information on the transferred object;
- other domain names of the losing registrar has links to the Contact object being transferred, then a copy of Contact object with a new ID assigned by the registry (and not the object itself) is transferred to the gaining registrar.

The Contact object which has no links to it from any Domain object can be transferred to another registrar at the initiative of the gaining registrar using Authinfo code.

The Contact object is deleted from the registry upon expiry of its registration period.

10. Host object

Each Host object in the registry has a composite internal identifier consisting of two parts:

- DNS server name which is assigned by the registrar at creating the request for Host object registration;
- sequence of characters assigned by the database which supports the registry.

This solution enables the registrar to control Host objects being created by him independently of other registrars and all Host objects are unique within the registry framework. In this case the registrar can modify Host object attributes by indicating only DNS server name (assigned by registrar) as the object identifier.

10.1. Host object life cycle

Only one period is available for Host object – Registration Period of indeterminate duration.

The Host object registration period finishes under the following circumstances:

- the object has been deleted by the registrar using <delete> request;

- the object has been deleted by server procedures in 20 days after registration in the registry if during the indicated time period neither of the registered domain names had links to the object, or at outbound transfer - if neither of domain names under the losing registrar's control had links to the Host object.

10.2. Creating, updating, outbound transfer and deleting the Host object

The object is registered by <host:create> request and updated by <host:update> request. The Host object is transferred to another registrar automatically when the Domain object which has links to it is transferred to another registrar. During transfer the Host object internal identifier in the registration system changes, and if:

- neither of the losing registrar's domain names has links to NS server indicated in the object being transferred, the object is deleted from the list of objects managed by the losing registrar;
- other domain names of the losing registrar have links to NS server indicated in the object being transferred, a copy of the object is transferred to the gaining registrar.

According to ICANN recommendations on prevention of Orphan Glue Records, if the registrar has indicated IP address of non-subordinate NS server with respect to the domain name containing a link on Host object, such IP addresses are retained at creation and modification of the Host object but ignored during the domain name delegation and are not included into the zone file.

The Host object is deleted from the registry upon expiry of its registration period.

11. Messages and notifications

According to EPP protocol, the following types of messages are automatically sent to the registrar:

- At domain name outbound transfer to another registrar:
  - initiation (to the losing registrar);
  - rejection (to the gaining registrar);
  - cancelation (to the losing registrar);
  - successful completion of transfer (to the gaining registrar);
  - rejection to perform the transfer due to insufficiency of funds on the gaining registrar's personal account (to the losing and gaining registrars).
- At automatic renewal:
  - successful automatic renewal;
  - automatic renewal failed due to insufficiency of funds on the personal account (upon expiry of the registration period);
  - automatic renewal failed due to enabled serverRenewProhibited or clientRenewProhibited statuses (upon expiry of the registration period).
- At automatic transfer of the non-renewed domain name to RGP.
- At failure to restore a domain name upon expiration of Pending Restore period.
- At enabling or disabling any server status.

12. EPP extensions of Domain object

During the General Registration period, there are special EPP extensions for checking the registration of a Trademark and notifying the Trademark holder about the registration of a domain name matching their Trademark.

If checking (using <check> command) shows that the domain name does not match any Trademark, then there is no need to use the EPP extensions.

The full specification is available at https://tools.ietf.org/html/draft-ietf-eppext-launchphase-02

12.1. <launch:phase> element

Launch:phase element is used for indication of the registration system operating mode (phase). <launch:phase> element must have the following value: <launch:phase>claims</launch:phase>

12.2. <launch:claimKey> element

launch:claimKey element contains a pointer to TMCH information on availability of Claim Notices with respect to the domain name.

12.3. <launch:notice> element

<launch:notice> element contains the following child elements:

12.3.1. <launch:noticeID> element

<launch:noticeID> extension contains a unique identifier of the claim with respect to a domain name in TMCH system. It shall be included into <create domain> request for the domain names which TMCH has claims to.

12.3.2. <launch:notAfter> element

Contains date and time of Claim Notice expiry.

12.3.3. <launch:acceptedDate> element

Contains date and time when Claim Notice was received.

12.4. Examples of using extensions

These examples show specific parameters that must be specified in requests for interaction with .MOSCOW and .МОСКВА registries.

Examples of requests are given for domain names matching Trademarks.

12.4.1. <check> request for checking if domain name may be registered in the registry

This request checks if such domain name may be registered in the registry. <check> request without <extension> has type=”avail”. If <check> has <extension> without type specified, by default type=”claims”.

Format of <check> request for checking if domain name may be registered:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>nic.moscow</domain:name>
<domain:name>test123.moscow</domain:name>
<domain:name>testandvalidate.moscow</domain:name>
      </domain:check>
    </check>
    <extension>
      <launch:check xmlns:launch="urn:ietf:params:xml:ns:launch-1.0"
type="avail">
```

```
           <launch:phase>claims</launch:phase>
          </launch:check>
         </extension>
       <clTRID>check-avail-perlcli-avt-20141126-123851</clTRID>
        </command>
       </epp>
```

Response:

```
       <?xml version="1.0" encoding="UTF-8"?>
       <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">
        <response>
         <result code="1000">
          <msg lang="en">Command completed successfully</msg>
         </result>
         <resData>
          <domain:chkData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
           <domain:cd>
            <domain:name avail="1">nic.moscow</domain:name>
           </domain:cd>
           <domain:cd>
            <domain:name avail="1">test123.moscow</domain:name>
           </domain:cd>
           <domain:cd>
            <domain:name avail="1">testandvalidate.moscow</domain:name>
           </domain:cd>
          </domain:chkData>
         </resData>
         <trID>
          <clTRID>check-avail-perlcli-avt-20141126-123851</clTRID>
          <svTRID>136926539</svTRID>
         </trID>
        </response>
       </epp>
```

12.4.2. <check> request for checking if domain name matches a Trademark

```
<?xml version="1.0" encoding="UTF-8"?>
<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>nic.moscow</domain:name>
<domain:name>test123.moscow</domain:name>
<domain:name>testandvalidate.moscow</domain:name>
   </domain:check>
      </check>
    <extension>
      <launch:check xmlns:launch="urn:ietf:params:xml:ns:launch-1.0"
       type="claims">
```

```
        <launch:phase>claims</launch:phase>
      </launch:check>
    </extension>
   <clTRID>check-claims-perlcli-avt-20141126-123851</clTRID>
  </command>
</epp>
```

In the response to the request the registrar will get information about claimKey:

```
<?xml version="1.0" encoding="UTF-8"?>
<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">
 <response>
  <result code="1000">
   <msg lang="en">Command completed successfully</msg>
  </result>
  <extension>
   <launch:chkData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:launch-1.0 launch-1.0.xsd">
    <launch:phase>claims</launch:phase>
    <launch:cd>
     <launch:name exists="0">nic.moscow</launch:name>
    </launch:cd>
    <launch:cd>
     <launch:name exists="0">test123.moscow</launch:name>
    </launch:cd>
    <launch:cd>
     <launch:name exists="1">testandvalidate.moscow</launch:name>
<launch:claimKey>2014072400/D/9/F/2fEoxs9QtB_QcemCwbGn3Sg90000000016</launch:claimKey>
    </launch:cd>
   </launch:chkData>
  </extension>
  <trID>
   <clTRID>check-claims-perlcli-avt-20141126-123851</clTRID>
   <svTRID>136926540</svTRID>
  </trID>
 </response>
</epp>
```

12.4.3. <domain:create> request for domain name registration

```
<?xml version="1.0"?>
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
  xmlns="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <command>
    <create>
      <domain:create xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
domain-1.0.xsd" xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>testandvalidate.moscow</domain:name>
          <domain:period unit="y">1</domain:period>
          <domain:registrant>faitid-0001</domain:registrant>
          <domain:contact type="admin">faitid-0001</domain:contact>
          <domain:contact type="tech">faitid-0001</domain:contact>
          <domain:contact type="billing">faitid-0001</domain:contact>
          <domain:authInfo>
            <domain:pw>domainpasswd</domain:pw>
          </domain:authInfo>
      </domain:create>
    </create>
    <extension>
      <launch:create xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase>claims</launch:phase>
        <launch:notice>
          <launch:noticeID>5db29e0d000000000000058752936</launch:noticeID>
          <launch:notAfter>2014-11-27T00:00:00.0Z</launch:notAfter>
          <launch:acceptedDate>2014-11-25T12:00:00.0Z</launch:acceptedDate>
        </launch:notice>
      </launch:create>
    </extension>
    <clTRID>create-dom-perlcli-avt-20141126-123851</clTRID>
</command>
</epp>
```

Response:

```
       <?xml version="1.0" encoding="UTF-8"?>
       <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">
        <response>
         <result code="1000">
          <msg lang="en">Command completed successfully</msg>
         </result>
         <resData>
          <domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
           <domain:name>testandvalidate.moscow</domain:name>
           <domain:crDate>2014-11-26T09:38:52.129Z</domain:crDate>
          </domain:creData>
         </resData>
         <extension>
          <launch:creData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:launch-1.0 launch-1.0.xsd">
           <launch:phase>claims</launch:phase>
           <launch:applicationID>7649399</launch:applicationID>
          </launch:creData>
         </extension>
         <trID>
          <clTRID>create-dom-perlcli-avt-20141126-123851</clTRID>
          <svTRID>136926541</svTRID>
         </trID>
        </response>
       </epp>
```

12.4.4. <domain:info> request

If you need the registry's response to contain information about a trademark, you need to specify <extension>. Information about a trademark requested with includeMark="true" will only be displayed it there is a trademark. If there is no trademark, the infromation will not be displayed, but there will be no error either.

```
<?xml version="1.0" encoding="UTF-8"?>
<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">testandvalidate.moscow</domain:name>
      </domain:info>
    </info>
    <extension>
      <launch:info includeMark="false"
        xmlns:launch="urn:ietf:params:xml:ns:launch-1.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:ietf:params:xml:ns:launch-1.0
        launch-1.0.xsd">
        <launch:phase>claims</launch:phase>
      </launch:info>
    </extension>
  <clTRID>info-dom-perlcli-avt-20141126-184044</clTRID>
  </command>
</epp>
```

```
        Response:
        <?xml version="1.0" encoding="UTF-8"?>
        <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">
         <response>
          <result code="1000">
           <msg lang="en">Command completed successfully</msg>
          </result>
          <resData>
           <domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
            <domain:name>testandvalidate.moscow</domain:name>
            <domain:roid>3199999992350601-REGDETI</domain:roid>
            <domain:status s="inactive"/>
            <domain:registrant>faitid-0001</domain:registrant>
            <domain:contact type="admin">faitid-0001</domain:contact>
            <domain:contact type="tech">faitid-0001</domain:contact>
            <domain:contact type="billing">faitid-0001</domain:contact>
            <domain:clID>FAITID-TST-MSK</domain:clID>
            <domain:crID>FAITID-TST-MSK</domain:crID>
            <domain:crDate>2014-11-26T09:38:52.0Z</domain:crDate>
            <domain:exDate>2015-11-26T09:38:52.0Z</domain:exDate>
            <domain:authInfo>
             <domain:pw> domainpasswd </domain:pw>
            </domain:authInfo>
           </domain:infData>
          </resData>
          <extension>
           <launch:infData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:launch-1.0 launch-1.0.xsd">
            <launch:phase>claims</launch:phase>
            <launch:applicationID>7649399</launch:applicationID>
            <launch:status s="allocated"/>
           </launch:infData>
          </extension>
          <trID>
           <clTRID>info-dom-perlcli-avt-20141126-184044</clTRID>
           <svTRID>136957202</svTRID>
          </trID>
         </response>
        </epp>
```

13. EPP extensions of Contact object

Additional information about Registrant and other contacts related to the domain is stored using EPP Extensions. Description of the used EPP Extensions is provided in Appendix No. 1 “Description of EPP Extensions”.

Providing Information in russian language in the extension element <contact:postalInfo type="loc"> is obligatory for the residents of the Russian Federation and for the legal bodies established or resided in the Russian Federation.

Providing Information in english language in the extension element <contact:postalInfo type="int"> is obligatory for all Registrations, not limiting it to the residents or established legal bodies of any jurisdiction.

14. Technical Support for Registrars

Technical support for registrars is provided 24/7.

Technical issues may be addressed via e-mail to tech.contact@faitid.org. When composing technical questions through EPP it is recommended and in some case it is a matter of necessity - to attach to the questions fragments of EPP (the queries to registration system and the answers obtained from the registration system). It will at least enable to reduce the time of processing registrar's query.

Registrars may submit administrative and financial questions via e-mail to adm.contact@faitid.org.

Registrars may also ask the required question by phone **+7 (495) 789-82-07**, but in cases requiring technical investigations or modeling registration system behavior, this communication option is not recommended.

15. Maintenance outages

Parts of registration system require maintenance from time to time. This may be related both to maintenance of the available hardware and due to putting the new ones into operation. Similar operations are performed in a manner so that the registration system and other associated services remain functional.

During similar operations appropriate notifications about work start, their duration and completion will be sent in advance to Registrar's contact e-mail specified in the Service Agreement with the Registry Operator. In case of occurrence of unforeseen failures in system operation notifications with the problem, planned outage period will also be sent to the registrars.

16. Registrar's personal account

Registrar shall independently check the availability of funds required for payment of the delivered services on the personal account and recharge the account in advance, if necessary. The information about balance on the personal account is available in the registrar's web interface.