

Description of EPP Extensions

Version 1.1

Addendum 1 to the Specifications for Registrars' Interaction with the Domain Registration System During Sunrise and Other Limited Registration Periods

Table of Contents

1. Introduction	1
1.1. Purpose	1
1.2. Sources	1
2. Permanent Extensions	2
2.1. Contact	2
2.1.1. Attributes for a Legal Entity	3
2.1.2. Attributes for an Individual	4
2.2. Schema for Extension of Create and Update Commands for Contact Object	5
2.3. Use of Attributes Examples	8
2.3.1. Creating a Contact for a Legal Entity	8
2.3.2. Creating a Contact for an Individual	9

1. Introduction

1.1. Purpose

This document is an addendum to the Specifications for Registrars' Interaction with the Domain Registration System During Sunrise and Other Limited Registration Periods.

The document features all extensions of the standard protocol, which are implemented for Contact objects.

This document does not feature the EPP protocol in full, since the protocol itself is the industry standard and is implemented in accordance with the RFC.

1.2. Sources

EPP protocol, which is the main channel of interaction between registrar and registry, is implemented in accordance with the following standards:

- RFC 3375 – Generic Registry-Registrar Protocol Requirements.
- RFC 5730 – Extensible Provisioning Protocol.

- RFC 5731 – EPP Domain Name Mapping.
- RFC 5732 – EPP Host Mapping.
- RFC 5733 – EPP Contact Mapping.
- RFC 5734 – EPP Transport over TCP.
- RFC 3915 – Domain Registry Grace Period Mapping for EPP.
- RFC 4114 – E.164 Number Mapping for EPP.
- RFC 4310 – DNS Security Extensions Mapping for EPP.

This document does not contain any descriptions from the standard protocol. This documents contains only detailed descriptions of extensions implemented for Contact objects in the Domain Registration System.

These extensions are implemented in accordance with the industry standards:

- RFC 3735 – Guidelines for extending EPP.
- RFC 5730 - Extensible Provisioning Protocol (EPP). Appendix A, Object Mapping Template. Composition of extension, composition of attributes, features of attributes are implemented in accordance with the following drafts:
 - draft-lozano-tmch-smd-03 – Mark and Signed Mark Objects Mapping;
 - draft-tan-epp-launchphase-12 – LaunchPhase Mapping for the Extensible Provisioning Protocol (EPP).

Country codes are implemented in accordance with the following standard:

ISO3166-2 – ISO, "International Standard for country codes and codes for their subdivisions", 2006.

Codes for the state legal authorities, businessmen, and nongovernmental organizations are implemented in accordance with the following standard:

WIPO, "Recommended standard on two-letter codes for the representation of states, other entities and intergovernmental organizations", March 2000
(<http://www.wipo.int/export/sites/www/standards/en/pdf/03-03-01.pdf>).

Codes for goods and services of trademarks are implemented in accordance with the following standard:

International classification of goods and services under the Nice Agreement, tenth edition –
<http://www.wipo.int/classifications/nivilo/nice/index.htm>.

2. Permanent Extensions

2.1. Contact

Commands for creating and editing a Contact object are extended with a new mandatory element – type of contact. Depending on a type of contact, objects will have different sets of attributes. It is necessary to specify one of the two types of contact:

- "person". This type of contact means that the contact is for an individual.
- "organization". This type of contact means that the contact is for an organization (legal entity). A set of attributes of a Contact object is subject to the specified type.

2.1.1. Attributes for a Legal Entity

Contact object with Organization type is intended for description of a legal entity's contact. The object contains the following attributes:

- id – contact's identifier.
- name – contact's name. The attribute is a text field. It may contain Cyrillic and Latin letters, spaces, punctuation marks.
- org – organization's name. The attribute is a text field. It may contain Cyrillic and Latin letters, spaces, punctuation marks.
- addr – complex attribute for postal address. The contact may contain several addresses, including addresses in different languages (with different values of Type attribute). It includes the following elements and attributes:
 - type – contact's language. There are two ways for using this attribute:
 - local – the address will be specified in Russian, using the Cyrillic characters and digits.
 - int – the address will be specified using the Latin characters and digits.
 - street – name of the street. The attribute is a text string. It may be filled with Latin or Cyrillic letters, depending on the value of Type attribute. Spaces and punctuation marks are also allowed. This is a mandatory attribute.
 - city – name of the city. The attribute is a text string. It may be filled with Latin or Cyrillic letters, depending on the value of Type attribute. Spaces and punctuation marks are also allowed. This is a mandatory attribute.
 - sp – region's code. The attribute shall be filled with a two-letter code.
 - pc – postal code.
 - cc – two-letter country code. This is a mandatory attribute.
- voice – contact's phone number for voice communication. This is not a mandatory attribute. It may be left blank.
- email – contact's email address. This is a mandatory attribute.
- authInfo – complex attribute for authentication details. For a Contact object in the registry, this attribute contains one nested element:
 - pw – contact holder's password.
- extension – section containing additional attributes of the extension:
 - contact type – for legal entities it must be <contact:organization>;
 - legalAddr – complex attribute for organization's legal address. The contact may contain several addresses, including addresses in different languages (with different values of Type attribute). It includes the following elements and attributes:
 - type – contact's language. There are two ways for using this attribute:

- local – the address will be specified in Russian, using the Cyrillic characters and digits.
- int – the address will be specified using the Latin characters and digits.
- street – name of the street. The attribute is a text string. It may be filled with Latin or Cyrillic letters, depending on the value of Type attribute. Spaces and punctuation marks are also allowed. This is a mandatory attribute.
- city – name of the city. The attribute is a text string. It may be filled with Latin or Cyrillic letters, depending on the value of Type attribute. Spaces and punctuation marks are also allowed. This is a mandatory attribute.
- sp – region's code. The attribute shall be filled with a two-letter code.
- pc – postal code.
- cc – two-letter country code. This is a mandatory attribute.
- TIN – organization's INN (Russian taxpayer identification number). This is a mandatory attribute. The length of this attribute is 22 characters.

2.1.2. Attributes for an Individual

For an individual's contact there are the following attributes:

- id – contact's identifier.
- name – contact's name. The attribute is a text field. It may contain Cyrillic and Latin letters, spaces, punctuation marks.
- postalInfo, nested attribute of addr – complex attribute for postal address in English with Latin characters. The contact may contain several addresses, including addresses in different languages (with different values of Type attribute). It includes the following elements:
 - type – contact's language. There are two ways for using this attribute:
 - local – the address will be specified in Russian, using the Cyrillic characters and digits.
 - int – the address will be specified using the Latin characters and digits.
 - street – name of the street. The attribute is a text string. It may be filled with Latin or Cyrillic letters, depending on the value of Type attribute. Spaces and punctuation marks are also allowed. This is a mandatory attribute.
 - city – name of the city. The attribute is a text string. It may be filled with Latin or Cyrillic letters, depending on the value of Type attribute. Spaces and punctuation marks are also allowed. This is a mandatory attribute.
 - sp – region's code. The attribute shall be filled with a two-letter code.
 - pc – postal code.
 - cc – two-letter country code. This is a mandatory attribute.

- voice – contact's phone number for voice communication. This is not a mandatory attribute. It may be left blank.
- email – contact holder's email address. This is a mandatory attribute.
- authInfo – complex attribute for authentication details. For a Contact object in the registry, this attribute contains one nested element:
 - pw – contact holder's password.
- extension – section containing additional attributes of the extension:
 - For an individual's contact the type must be <contact:person>.
 - birthday – date of birth. The attribute must be filled with digits using this format: YYYY-MM-DD.
 - passport – string with details from the ID. The attribute must contain the name of document in English and the number of document.
 - TIN – individual's INN (Russian taxpayer identification number). This is not a mandatory attribute. It may be left blank.

2.2. Schema for Extension of Create and Update Commands for Contact Object

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
  targetNamespace="http://www.tcinet.ru/epp/tci-contact-ext-1.0"
  xmlns:contExt="http://www.tcinet.ru/epp/tci-contact-ext-1.0"
  xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <import namespace="urn:ietf:params:xml:ns:contact-1.0"
    schemaLocation="contact-1.0.xsd"/>
  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      contact extensions provisioning schema.
    </documentation>
  </annotation>
  <!--
  Child elements found in EPP commands.
  -->
  <element name="create" type="contExt:createType"/>
  <element name="update" type="contExt:updateType"/>
  <!--
  Child elements of the <create> command.
  -->
  <complexType name="createType">
    <sequence>
      <choice>
        <element name="person"
          type="contExt:personType"/>
        <element name="organization"
          type="contExt:orgType"/>
      </choice>
    </sequence>
  </complexType>
  <!--
```

```

Child elements of organization interface
-->
<complexType name="legalAddrType">
  <sequence>
    <element name="street"
type="contact:optPostalLineType"
      minOccurs="1" maxOccurs="3"/>
    <element name="city"
type="contact:postalLineType"/>
    <element name="sp"
type="contact:optPostalLineType"
      minOccurs="0"/>
    <element name="pc" type="contact:pcType"
      minOccurs="0"/>
    <element name="cc" type="contact:ccType"/>
  </sequence>
  <attribute name="type"
type="contact:postalInfoEnumType"
      use="required"/>
</complexType>
<simpleType name="TINType">
  <restriction base="token">
    <maxLength value="22"/>
  </restriction>
</simpleType>
<complexType name="orgType">
  <sequence>
    <element name="legalAddr"
type="contExt:legalAddrType" minOccurs="1"
maxOccurs="2"/>
    <element name="TIN" type="contExt:TINType"/>
    <element name="disclose"
type="contExt:orgDiscloseType" minOccurs="0"/>
  </sequence>
</complexType>
<complexType name="orgDiscloseType">
  <sequence>
    <element name="legalAddr"
type="contact:intLocType"
      minOccurs="0" maxOccurs="2"/>
    <element name="TIN" minOccurs="0"/>
  </sequence>

  <attribute name="flag" type="boolean"
use="required"/>
</complexType>
<!--
Child elements of person interface
-->
<complexType name="personType">
  <sequence>
    <element name="birthday" type="date"/>
    <element name="passport"
type="contExt:passportType"/>
    <element name="TIN" type="contExt:TINType"
minOccurs="0"/>
    <element name="disclose"
type="contExt:personDiscloseType"

```

```

        minOccurs="0"/>
    </sequence>
</complexType>
<simpleType name="passportType">
    <restriction base="token">
        <minLength value="1"/>
        <maxLength value="512"/>
    </restriction>
</simpleType>
<complexType name="personDiscloseType">
    <sequence>
        <element name="birthday" minOccurs="0"/>
        <element name="passport" minOccurs="0"/>
        <element name="TIN" minOccurs="0"/>
    </sequence>

    <attribute name="flag" type="boolean"
use="required"/>
</complexType>
<!--
Child elements of the <update> command.
-->
<complexType name="updateType">
    <sequence>
        <element name="chg" type="contExt:chgType"
minOccurs="0"/>
    </sequence>
</complexType>
<!--
Data elements that can be changed.
-->
<complexType name="chgOrgType">
<sequence>
    <element name="legalAddr"
type="contExt:legalAddrType"
minOccurs="0" maxOccurs="2"/>
    <element name="TIN" type="contExt:TINType"
minOccurs="0"/>
    <element name="disclose"
type="contExt:orgDiscloseType"
minOccurs="0"/>
</sequence>
</complexType>
<complexType name="chgPersonType">
<sequence>
    <element name="birthday" type="date"
minOccurs="0"/>
    <element name="passport"
type="contExt:passportType"
minOccurs="0"/>
    <element name="TIN" type="contExt:TINType"
minOccurs="0"/>
    <element name="disclose"
type="contExt:personDiscloseType"
minOccurs="0"/>
</sequence>
</complexType>
<!--

```

Data elements that can be changed.

```
-->
<complexType name="chgType">
  <sequence>
    <choice>
      <element name="organization"
type="contExt:chgOrgType"/>
      <element name="person"
type="contExt:chgPersonType"/>
    </choice>
  </sequence>
</complexType>
<!--
Child response elements.
-->
<element name="infData" type="contExt:createType"/>
</schema>
```

2.3. Use of Attributes Examples

2.3.1. Creating a Contact for a Legal Entity

```
<?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>h3PA2YBI-vrdev</contact:id>
        <contact:postallInfo type="int">
          <contact:name>AAA LTD</contact:name>
          <contact:org>AAA LTD</contact:org>
          <contact:addr>
            <contact:street>Tverskaya 101 </contact:street>
            <contact:city>Moscow</contact:city>
            <contact:sp>Moscow</contact:sp>
            <contact:pc>107140</contact:pc>
            <contact:cc>RU</contact:cc>
          </contact:addr>
        </contact:postallInfo>
        <contact:postallInfo type="loc">
          <contact:name>OOO AAA</contact:name>
          <contact:org>OOO AAA</contact:org>
          <contact:addr>
            <contact:street>Тверская 101</contact:street>
            <contact:city>Москва</contact:city>
            <contact:sp>Москва</contact:sp>
            <contact:pc>107140</contact:pc>
            <contact:cc>RU</contact:cc>
          </contact:addr>
        </contact:postallInfo>
        <contact:voice>+7.4951241438</contact:voice>
        <contact:email>someone@tcinet.ru</contact:email>
        <contact:authInfo>
          <contact:pw>EujGiCwW5UwzikUw</contact:pw>
        </contact:authInfo>
      </contact:create>
    </create>
```



```

<extension>
<contact:create xmlns:contact="http://www.tcinet.ru/epp/tci-contact-ext-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.tcinet.ru/epp/tci-
contact-ext-1.0 tci-contact-ext-1.0.xsd">
<contact:organization>
  <contact:legalAddr type="loc">
    <contact:street>Новая 101</contact:street>
    <contact:city>Москва</contact:city>
    <contact:sp>Москва</contact:sp>
    <contact:pc>107140</contact:pc>
    <contact:cc>RU</contact:cc>
  </contact:legalAddr>
  <contact:TIN/>
</contact:organization>
</contact:create>
</extension>
<cITRID>TCI_GTLID-1584-1382620336153907</cITRID>
</command>
</epp>

```

2.3.2. Creating a Contact for an Individual

```

<?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>
<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>con-1-1384434788</contact:id>
  <contact:postallInfo type="int">
    <contact:name>Testov T Test</contact:name>
    <contact:addr>
      <contact:street>Prospect of Peace
</contact:street>
      <contact:street>32</contact:street>
      <contact:street>building
6</contact:street>
      <contact:city>Moscow</contact:city>
      <contact:sp>Russian
Federation</contact:sp>
      <contact:pc>122345</contact:pc>
      <contact:cc>ru</contact:cc>
    </contact:addr>
  </contact:postallInfo>
  <contact:postallInfo type="loc">
    <contact:name>Тестов Тест Тестович</contact:name>
    <contact:addr>
      <contact:street>Проспект Мира</contact:street>
      <contact:street>дом 32</contact:street>
      <contact:street>строение 6</contact:street>
      <contact:city>Москва</contact:city>

```

```
<contact:sp>Российская Федерация</contact:sp>
<contact:pc>122345</contact:pc>
<contact:cc>ru</contact:cc>
</contact:addr>
</contact:postallInfo>
<contact:voice x="ext123">+7.4951234567</contact:voice>
<contact:fax x="факс
эктеншен">+7.4950004567</contact:fax>
<contact:email>test@test.ru</contact:email>
<contact:authInfo>
<contact:pw>password</contact:pw>
</contact:authInfo>
</contact:create>
</create>
<extension>
  <contact:create
xmlns:contact="http://www.tcinet.ru/epp/tci-contact-
ext-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:schemaLocation="http://www.tcinet.ru/epp/tci-
contact-ext-1.0 tci-contact-ext-1.0.xsd">
  <contact:person>
  <contact:birthday>1970-11-11</contact:birthday>
  <contact:passport>строка паспорта</contact:passport>
  <contact:TIN>4444444444444444</contact:TIN>
  </contact:person>
</contact:create>
</extension>
<cITRID>RtkToolkitTest-1384434788-4</cITRID>
</command>
</epp>
```