

PDT Input Data Instructions

Instructions for the data to be prepared and submitted to the PDT System, <https://pdt.iis.se/>.

Version: 2.6

2015-09-28

Introduction	3
1.1 Methodology	3
1.2 Data entry into the PDT System	3
1.3 Document changes	3
1.3.1 New fields in pdtwhois.xml.....	3
1.3.2 Corrected Test Node name.....	4
1.3.3 Corrected information on Data Escrow files	4
PDT Test Nodes	5
2.1 IP addresses	5
DNS test	6
3.1 Introduction	6
3.2 Expected input	6
3.2.1 XML schemas	6
3.2.2 Example of input	6
3.2.3 Schema files and sample XML file.....	6
Whois test	7
4.1 Expected input	7
4.1.1 XML schemas	7
4.1.2 EPP Repository Identifier	7
4.1.3 Character encoding	8
4.1.4 Example of input	8
4.1.5 Schema files and sample XML file.....	8
EPP test	9
5.1 Reserved registrars	9
5.2 Expected input	10
5.2.1 XML schemas	10
5.2.2 Example of input	10
5.2.3 Schema files and sample XML file.....	10
5.3 Submitting EPP extensions	10
IDN test	11
6.1 Introduction	11
6.2 Expected input	11
6.3 IDN Self-certification template	11
6.4 Policy statement and EPP information	12
6.5 IDN tables	12
6.6 File format	12



6.7	Table structure and processing	12
6.8	Special consideration on code points	12
6.9	General considerations	13
Data Escrow test		14
7.1	Introduction	14
7.2	Expected input	14
7.3	Sample file names	14
Self-certification documentation test		15
8.1	Introduction	15
8.2	Expected input	15
8.2.1	File names	15
8.2.2	Template	15
Appendix 1: Attachments		17
Appendix 2: pdtescrow.pub		18
Appendix 3: Example of EPP extensions		19
11.1.1	A normal extension with two Name:Value fields	19
11.1.2	An IDN extension with a single text node	19
11.1.3	An IDN extension with a single Name:Value field	19
11.1.4	An extension with fields within field	19

Introduction

This document contains information about what the applicants need to prepare for the pre-delegation tests. All applicants are urged to carefully read this document.

The purpose of the pre-delegation testing is to verify that the applicant has met its commitment to establish registry operations in accordance with the technical and operational criteria described in the gTLD Applicant Guidebook (AGB) and that applicant can operate the gTLD in a stable and secure manner. Each applicant will be required to complete pre-delegation testing as a prerequisite to delegation into the root zone.

The test elements cover both the DNS server operational infrastructure and registry system operations. The tests are based on the AGB, specifically Module 5 (*Transition to Delegation*) and specification 2, 4, 6 and 10, and are described in detail in the document: Pre-delegation testing, Master Test Plan.

1.1 Methodology

The tests are documented according to the standard IEEE 829-2008, as described in the Master Test Plan. The standard allows for different types of tests, e.g. unit, system, and acceptance tests. This test environment will focus on acceptance testing. Multiple areas have been identified within the system requirements:

- DNS
- Whois
- EPP
- IDN
- Data Escrow
- Documentation

Each test area is further described in its own Level Test Plan and one or more Level Test Case documents, which will be published by ICANN.

1.2 Data entry into the PDT System

All input data to the tests shall be provided by the applicant via the PDT System, available as a web application at <https://pdt.iis.se/>. Login credentials will be given to the applicant once the testing has been scheduled by ICANN.

The PDT System is further described in a separate User guide, which is published by ICANN.

1.3 Document changes

There are three updates of this document compared to version 2.5 (2015-07-15):

1.3.1 New fields in pdtwhois.xml

There are two new fields to be added to the pdtwhois.xml file. The change is described below in the Whois section. The change takes affect for PDT appointments after 2015-11-02, but should not be implemented for appointments before that date.

1.3.2 Corrected Test Node name

In the previous version of this document, one of the Test Nodes had an incorrect hostname, but correct IP addresses. The hostname has been corrected in this version of this document.

1.3.3 Corrected information on Data Escrow files

The previous version of this file stated that the Data Escrow files must be 40 days old or newer. The correct age limit is 30 days, which is corrected in this version of this document.

PDT Test Nodes

The PDT Service Provider will verify applicants' infrastructure (Whois, EPP and DNS) from the PDT Test Nodes. The applicant must make sure that firewalls and access lists are open for the following IP addresses (both IPv4 and IPv6). The applicant's infrastructure must be available for testing during the whole test period.

Please make sure that you look at the latest version of this document for the most current set of IP addresses of the PDT Test Nodes.

2.1 IP addresses

PDT Test Node	IPv4	IPv6	Hostname
North America	74.63.16.34	2620:171:f0::34	na-pao.pdt.iis.se
	74.63.17.34	2620:171:f1::34	na-iad.pdt.iis.se
	204.45.70.50	2001:49fo:a000:21::42	na-den.pdt.iis.se
Latin America and Caribbean	74.63.19.34	2620:171:f3::34	lac-gru.pdt.iis.se
	74.63.18.34	2620:171:f2::34	lac-eze1.pdt.iis.se
	200.1.120.31	2001:1398:3:120::31	lac-scl.pdt.iis.se
	131.255.4.95	2803:eb80::12	lac-eze2.pdt.iis.se
Asia/ Australia/ Pacific	203.178.137.9	2001:200:0:180a::502	ap-nrt.pdt.iis.se
	74.63.20.34	2620:171:f4::34	ap-sgw.pdt.iis.se
	103.6.87.155	2403:2500:4000::afd	ap-maa.pdt.iis.se
	103.16.26.69	2001:df0:465::69	ap-hkg.pdt.iis.se
Africa	74.63.22.34	2620:171:f6::34	af-cai.pdt.iis.se
	197.96.55.76	2cof:fc00:a000:1100::4	af-jnb.pdt.iis.se
	74.63.23.34	2620:171:f7::34	af-cpt.pdt.iis.se
Europe	74.63.25.34	2620:171:f9::34	eur-ams.pdt.iis.se
	74.63.24.34	2620:171:f8::34	eur-fra.pdt.iis.se
	77.72.225.37	2a01:3f0:0:4c::37	eur-arn1.pdt.iis.se
	80.252.187.84	2001:16d8:4:7b::84	eur-arn2.pdt.iis.se

DNS test

3.1 Introduction

The PDT Service Provider will verify applicant's DNS infrastructure over both UDP and TCP, and that DNSSEC is supported including life cycle management of Zone and Key signing keys. All tests will be carried out for both IPv4 and IPv6.

- For the DNS tests, applicant must supply the following information to the PDT Service Provider: FQDN of all authoritative name servers
- IPv4 and IPv6 addresses for same
- Delegation Signer (DS) information
- A domain with NS and DS records published in the TLD. This domain must be delegated to a name server different from the TLD name servers, but it does not have to exist on this name server.

3.2 Expected input

The applicant shall provide the input data for the DNS test in a single XML file according to the XML schemas provided by the PDT Service Provider:

- Mandatory file format: XML
- File name on submitted file: pdtdns.xml

Your XML file will be validated against the schema after uploading and you will be notified immediately if there are any errors. However, for better error reporting you are encouraged to validate your XML yourself before uploading.

3.2.1 XML schemas

There are two different XML schemas of which the applicant can choose from. These are the W3C XML Schema (XSD) and the RELAX NG (RNG). The submitted XML file must match the schema.

3.2.2 Example of input

For your convenience a sample XML file containing input data for the DNS test is provided.

3.2.3 Schema files and sample XML file

Schema files and sample XML files are found in the “PDT Input Data Templates” zip file found on <http://newgtlds.icann.org/en/applicants/pdt#resources>. Always use the latest version.

- pdtdns.rng (DNS RELAX NG Schema)
- pdtdns.rnc (DNS RELAX NG Compact Schema)
- pdtdns.xsd (DNS XML Schema Definition)
- pdtdns.xml (DNS Sample XML)

Whois test

Note! Below is an updated format of the pdtwhois.xml file. The updated format is mandatory for all PDT appointments 2015-11-02 or later. The updated format should not be used for any appointment before that date.

The PDT Service Provider will verify that Whois data is accessible over IPv4 and IPv6, both via TCP port 43 and via a web interface. If applicant states that searching in Whois data is supported, this functionality will also be evaluated. For the Whois tests, applicant must supply the following information to the PDT Service Provider:

[REQUIRED]

- An existing domain name for this TLD, which has Whois data
- An existing registrar which has Whois data
- The domain name of an existing name server, which has Whois data
- IPv4 or IPv6 address of an existing name server, which has Whois data
- The *EPP Repository Identifier* used on the ROID on *Domain ID* field in the Whois response
- The character encoding used on Whois CLI (port 43) responses

[OPTIONAL]

- Credentials, e.g. username and password, if required for accessing the Whois search service as a logged in user

4.1 Expected input

The applicant shall provide the input data for the Whois test in a single XML file according to the XML schemas provided by the PDT Service Provider.

Your XML file will be validated against the schema after uploading and you will be notified immediately if there are any errors. However, for better error reporting you are encouraged to validate the XML yourself before uploading.

- Mandatory file format: XML
- File name on submitted file: pdtwhois.xml

4.1.1 XML schemas

There are two different XML schemas of which the applicant can choose from. These are the W3C XML Schema (XSD) and the RELAX NG (RNG).

4.1.2 EPP Repository Identifier

The *EPP Repository Identifier* is used as the "suffix" of the ROIDs (after the mandatory dash "-"). The *Identifier* must be registered at IANA. See <<https://www.iana.org/assignments/epp-repository-ids/epp-repository-ids.xhtml>> for registered *Identifiers* and more information.

For PDT Whois testing, the *Identifier* used in ROID in the *Domain ID* field must be provided in the pdtwhois.xml file.

4.1.3 Character encoding

For PDT Whois testing, the character encoding of the Whois response over TCP port 43 (see the Whois CLI Test Case document) must be provided in the XML file. If the character encoding is UTF-8 or ASCII, then the string "UTF-8" or "UTF8" can be used, which is also used in the example `pdtwhois.xml` file.

If any other character encoding is used (neither ASCII nor UTF-8) then the correct code for that encoding must be used in the `pdtwhois.xml` file. Valid codes for encodings are found in the XML Schema for the XML file.

The encoding must match the code provided as the `iconv` program of *Ubuntu Linux* version 12.04 defines it. The `iconv` program comes with the `glibc` library with that version of *Ubuntu Linux*. To verify that correct brand and version of `iconv` is used, run it on a *Ubuntu Linux* version 12.04 command line and verify that the same or a compatible version output is received:

```
$ iconv -V | head -1
iconv (Ubuntu EGLIBC 2.15-0ubuntu10.12) 2.15
```

To find all supported codes for character encodings, see the `pdtwhois.xsd` file or run the following command on a *Ubuntu Linux* version 12.04 command line:

```
$ iconv --list
```

4.1.4 Example of input

For your convenience, a sample XML file containing input data for the Whois test is provided.

4.1.5 Schema files and sample XML file

Schema files and sample XML files are found in the “PDT Input Data Templates” zip file found on <http://newgtlds.icann.org/en/applicants/pdt#resources>. Always use the latest version.

- `pdtwhois.rng` (Whois RELAX NG Schema)
- `pdtwhois.rnc` (Whois RELAX NG Compact Schema)
- `pdtwhois.xsd` (Whois XML Schema Definition)
- `pdtwhois.xml` (Whois Sample XML)

EPP test

The PDT Service Provider will verify that applicant's EPP Service conforms to appropriate RFCs, including EPP extensions for DNSSEC. For Create, the new object has to be visible in zone file and Whois within 24 hours. For changes, the zone data must be visible in applicant's zone file and Whois service within 60 minutes. If applicant states support for EPP over IPv6, the PDT Service Provider will verify this too.

If your EPP server requires a client certificate for a client to be able to login, make sure to include the pkcs12 file and password in the Client/KeyPair part of the XML.

The address to your EPP server can be given as an IP address or as a hostname, once for IPv4 and once for IPv6, if supported. If a hostname is given, the test script client will use that name as SNI parameter (Server Name Indication, RFC 6066) in the TLS handshake. The hostname should either be resolvable in public DNS or be a name under the TLD in question and resolvable using the DNS delegation information provided in the PDT DNS XML file.

For the EPP tests, applicant must supply the following information to the PDT Service Provider:

[REQUIRED SETUP INFORMATION]

- Information about the EPP server: IPv4 address+port, server certificate and optionally IPv6 address+port. Instead of an IP address a hostname can be provided in one or both of the IP address fields.
- Valid login ID and password required to access the EPP service
- URI and Schema Location for Domain, Contact and Host objects
- URI and Schema Location for the SecDns extension (DNSSEC)
- URI and Schema Location for other extensions, if applicable

[FOR THE TESTS]

- FQDN for at least two name servers
- Three non-registered domain names, ready to be created
- One registered domain name ready to be renewed
- Two registered domain names ready for transfer
- One registered domain name ready to be updated
- One domain name that can be deleted
- A new Contact object, to be created
- Contact and Host objects to be updated
- Contact and Host objects to be deleted

5.1 Reserved registrars

There are two reserved registrar ID's. It is recommended to use those for the PDT. The data for the reserved registrar ID's are below.

- NEW GURID, Account Add #1 (for use by new gTLD registries)

- GURID: 9995
- "Registrar Name": Pre-Delegation Testing Registrar #1
- NEW GURID, Account Add #2 (for use by new gTLD registries)
 - GURID: 9996
 - "Registrar Name": Pre-Delegation Testing Registrar #2

5.2 Expected input

The applicant shall provide the input data for the EPP test in a single XML file according to the XML schemas provided by the PDT Service Provider.

Your XML file will be validated against the schema after uploading and you will be notified immediately if there are any errors. However, for better error reporting you are encouraged to validate your XML yourself before uploading.

While EPP allows for a large number of extensions to each object, the applicant should only submit values for those extensions that are mandatory for their registration system. Please refer to the XML schema below for a precise description of the various objects and their attributes.

- Mandatory file format: XML
- File name on submitted file: pdtepp.xml

5.2.1 XML schemas

There are two different XML schemas of which the applicant can choose from. These are the W3C XML Schema (XSD) and the RELAX NG (RNG).

5.2.2 Example of input

For your convenience, a sample XML file containing input data for the EPP test is provided.

5.2.3 Schema files and sample XML file

Schema files and sample XML files are found in the “PDT Input Data Templates” zip file found on <http://newgtlds.icann.org/en/applicants/pdt#resources>. Always use the latest version.

- pdtepp.rng (EPP RELAX NG Schema)
- pdtepp.rnc (EPP RELAX NG Compact Schema)
- pdtepp.xsd (EPP XML Schema Definition)
- pdtepp.xml (EPP Sample XML)

5.3 Submitting EPP extensions

If applicant’s EPP server needs extra data not found in the standard object definitions, corresponding EPP extensions have to be specified as part of the input data.

In Appendix 3 you will find working XML sample code to input three different EPP extensions and their associated data.

IDN test

6.1 Introduction

The IDN testing has two basic components. The first is a review of the submitted table(s) and associated policy statements, together with the EPP documentation described below. This is both to verify conformance with the reference specifications and to enable a determination of the expected response of the back-end registry to a request for the registration of a given label. The second component of the IDN testing verifies that the registry responds in the anticipated manner, rejecting labels that are not permitted and accepting those that are.

The conformance tests are performed on all IDN tables listed in Exhibit A of the applicant's Registry Agreement. Testing the EPP response of the back-end registry (test cases IDNvalid10, IDNvalid11 and part of IDNvalid07) is restricted to the tables that are also listed in Section 5 of the Self-Certification Document.

There are eleven IDN test cases described in detail in the Test Case document for IDN found on the ICANN Micro-Site for PDT, <http://newgtlds.icann.org/en/applicants/pdt#resources>. Applicants should study that document in detail when preparing for the TLD level of the PDT.

6.2 Expected input

The following material must be submitted in order for the IDN testing to be conducted:

- Every IDN table that is listed in Exhibit A of the Applicant's Registry Agreement, in tables formatted according to RFC 4290 or RFC 3743.
- A complete statement of the policies that apply to IDN registration, i.e. a summary of responses to questions asked in the Applicant Guidebook necessary for the understanding of a submitted IDN table. If variant relationships between codepoints exist, the policy document(s) should describe how these variants are managed. If contextual rules are applicable to any code point in a table, those rules should be explicitly stated.
- A complete statement of which languages and scripts that are to be supported at General Registration.
- A complete example of an EPP Domain Create command for a valid, unregistered, IDN label, applicable to all tables that are listed both in Exhibit A of the Registry Agreement and in the IDN Self-Certification Document.
- A list of all EPP extensions, such as language tags that may be needed for the registration of IDN labels, covering all tables that are listed both in Exhibit A of the Registry Agreement and in the IDN Self-Certification Document.

6.3 IDN Self-certification template

It is recommended that the IDN Self-certification template is used for the information below. It also contains useful information and instructions.

- Policy statement
- List of languages and scripts

- EPP example
- EPP extensions

6.4 Policy statement and EPP information

If no IDN self-certification document is provided, then a separate policy statement must be submitted together with a separate file with the EPP example.

6.5 IDN tables

The IDN tables are expected to be submitted as separate files, one table per file. The submission of material that is not referenced in Exhibit A of the Registry Agreement will delay the testing. Applicants should take care to ensure a one-to-one correspondence between the listed tables and those that are forwarded for testing.

6.6 File format

The tables and associated documents are subject to automated processing and must be submitted as TXT files. Any table or document including non-ASCII characters must be encoded in Unicode UTF-8. The IDN Self-certification document is expected to be submitted as a PDF/A file.

6.7 Table structure and processing

The IDN tables are parsed algorithmically using a script that is publicly available at <https://github.com/dotse/idn-properties>. It is based on the requirements of the reference specifications for IDN table format, RFC 4290 and RFC 3743. The script provides a degree of additional latitude in order to deal with submissions which, for documented local reasons, are not strictly conformant to those RFCs.

The basic requirements are that every row in a table starts with a code point indicated in the form “U+nnnn”, or with a hash mark “#” indicating a remark, or is blank. The notation of a continuous sequence of code points in the form “nnnn..mmmm” is not permitted. If a row starting with a code point contains any additional data, this must either be initiated with a hash mark, or use a notational syntax described in one of the reference RFCs. Every submitted table must be parsable by this script.

More details are found in the IDN Self-certification template.

6.8 Special consideration on code points

Pay special attention on the code points in the IDN table. Investigate the need and requirement of contextual rules for them and the mixing of those with other code points.

- Code points with Unicode script property value COMMON or INHERITED.
- Code points with IDN property CONTEXTJ or CONTEXTO.
- Right-to-left code points.
- Code points with general Unicode category Modifier Letter or Mark.
- Mixing of code points from different explicit Unicode scripts.

- The use of listed code points in the specific language (if language table).

6.9 General considerations

The documentation needed for the IDN testing is not intended to provide background detail about the nature and purpose of IDN nor does it need to restate the terms of reference for the testing. The submitted documentation need provide only the detail necessary to conduct the individual IDN tests. Noting again that the Test Case document cited above should be studied carefully when preparing for the IDN testing.

Data Escrow test

7.1 Introduction

The PDT Service Provider will verify that the Data Escrow deposit conforms to the relevant IETF draft or RFC document. This includes, among other things, file formats, file names and methods for encryption and signing of the escrow file(s).

7.2 Expected input

Both the full deposit and the differential deposit may be split over several files. In this case each data file shall be individually signed and accompanied by a corresponding signature file.

The deposit shall be supplied in files formatted according to the rules in <http://tools.ietf.org/html/draft-arias-noguchi-registry-data-escrow> (refer to latest version) or corresponding RFC as well as RFC 4880.

Encryption shall be done using the PDT Service Provider's public test key, `pdtescrow.pub`, which is found in appendix 2.

For the Data Escrow tests, applicant should supply the following information to the PDT Service Provider. The file name patterns below must be followed.

- Filename: {gTLD}_{YYYY-MM-DD}_full_S{#}_R{rev}.ryde
- Filename: {gTLD}_{YYYY-MM-DD}_full_S{#}_R{rev}.sig
- Filename: {gTLD}.pub
- Filename: {gTLD}_{YYYY-MM-DD}_diff_S{#}_R{rev}.ryde
- Filename: {gTLD}_{YYYY-MM-DD}_diff_S{#}_R{rev}.sig

In the file name patterns above:

1. {gTLD} is equal to the TLD string. If it is an IDN TLD, then this must be the A label.
2. {YYYY-MM-DD} is equal to year, month, and day. The file must be maximum 30 days old.
3. {#} is a number greater than or equal to 1. Leading zeroes are not allowed.
4. {rev} is a number greater than or equal to 0. Leading zeroes are not allowed.

7.3 Sample file names

File names for sample TLD "example":

```
example_2013-11-13_full_S1_R0.ryde
example_2013-11-13_full_S1_R0.ryde
example.pub
example_2013-11-13_diff_S1_R0.ryde
example_2013-11-13_diff_S1_R0.ryde
```

File names for sample TLD "xn--example":

```
xn--example_2013-11-13_full_S1_R0.ryde
xn--example_2013-11-13_full_S1_R0.ryde
xn--example.pub
xn--example_2013-11-13_diff_S1_R0.ryde
xn--example_2013-11-13_diff_S1_R0.ryde
```

Self-certification documentation test

8.1 Introduction

This section contains the specification for the self-certification data that the applicant shall provide as part of the pre-delegation test.

The documentation tests are divided into the following groups:

- Self-certification of DNS, Whois and EPP
- DNSSEC Practice Statement
- Data Escrow

8.2 Expected input

The applicant shall provide the following input data for the different groups of self-certification documentation tests:

Documentation Test	Input data
Self-certification	A single PDF/A file adhering to the chapter structure of the template provided by the PDT Service Provider
DNSSEC Practice Statement	A single PDF/A file containing the DNSSEC Practice Statement
Data Escrow	A single PDF/A file containing the escrow provider agreement with all appendices

8.2.1 File names

Documentation Test	Name of file to be submitted	Sample file names (TLD is “example” and “xn--example”, respectively)
Self-certification	Self-certification_{gTLD}.pdf	Self-certification_example.pdf Self-certification_xn--example.pdf
DNSSEC Practice Statement	DPS_{gTLD}.pdf	DPS_example.pdf DPS_xn--example.pdf
Data Escrow	Data-Escrow_{gTLD}.pdf	Data-Escrow_example.pdf Data-Escrow_xn--example.pdf

{gTLD} is equal to the TLD string. If it is an IDN TLD, then this must be the A label. The string must be in lower case.

8.2.2 Template

Use that for the self-certification document to be submitted. Always preserve chapter structure and headlines.

Documentation Test	Files (same file in different formats)
Self-certification	Self-certification_document.docx Self-certification_document.pdf Self-certification_document.rtf

The template is found in the “PDT Input Data Templates” zip file found on <http://newgtlds.icann.org/en/applicants/pdt#resources>. Always use the latest version.

Appendix 1: Attachments

The following files are attachments to this document and are found in the “PDT Input Data Templates” zip file found on <http://newgtlds.icann.org/en/applicants/pdt#resources>. Always use the latest version.

- Self-certification_document.docx
- Self-certification_document.pdf
- Self-certification_document.rtf
- PDT_IDN_Self-certification_document.docx
- PDT_IDN_Self-certification_document.pdf
- PDT_IDN_Self-certification_document.rtf
- ptdns.rnc
- ptdns.rng
- ptdns.xml
- ptdns.xsd
- pdtepp.rnc
- pdtepp.rng
- pdtepp.xml
- pdtepp.xsd
- pdtwhois.rnc
- pdtwhois.rng
- pdtwhois.xml
- pdtwhois.xsd

Appendix 2: pdtescrow.pub

Public key for encryption of the .ryde files for the Data Escrow tests. Copy everything from and including “-----BEGIN PGP PUBLIC KEY BLOCK-----” up to and including “-----END PGP PUBLIC KEY BLOCK-----” to a file called “pdtescrow.pub”. Preserve all line breaks.

The file can also be found from <https://pdt.iis.se/files/pdtescrow.pub>.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----  
Version: GnuPG v2.0.17 (GNU/Linux)  
  
mQENBFerLLgBCAdm7s6Z4AuZMC3ANIiwZC9ANwObdKM15CjDfZUpsoE9r8uIFNnn  
aeTQENyhpl/WhA1PV/XBTWAmqhxYGPJTUAcoZneJREu+oBcw6IA3mNFvWJqOgm  
T7OWY2SPA+Odjy4/kluAA+WsK3u77curZsTOXDAs0u13FvuKgixMaSFu6dxDh5Qz  
lmkIYfLqrHz8FVd4Ecy+GoLpMf53OpMd45zoRaSEEIL/trc5dvJYDdggDSuW15zW  
E0KCBWvICaCs/lHA/DZi8OeGbRanOF07LdS5M1ANRLnk5NH00gHbm0nrgNwZYctm  
bQMiNpvyngnVanO14tEDL/3mJ5UZ5B7X3HKfABEBAAG0S1BEVCBQcm92aWRlciAo  
S2V5IHVzZWQgZm9yIHRlc3RpbmcgZGF0YSBlc2NyY3cpIDxwZHQtZXNjcm93LXRl  
c3RAaWNhbm4ub3JnPokBOAQTAQIAIgcUCUSssuAIBAwYLCQgHAWIGFQgCCQoLBBYC  
AwECHgECF4AACgkQk0U4NfvtGWLkPgF/d7jAWCOQqng+h0VkoACbCvod5t7edB3O  
B1hfW06VhbtpyCYbZQCrHs/c8FhIOtUA/YDKTuGMJpOis710LKKD4ZBXJqpsLIvm  
qQNwZfudLkQl506fI9lbEjTmIX0fMQekVwMvHwgWKzcpO7HUES19ovvJBAGzJOq  
NiVzwp9i+pKgbp6i3i8FpkAEDTdfjqek+++l6VA6ky/sW67jHUhXRvHW5cLz2rdx  
06wLkdjIrn1h9OTlYivRqQL5+jZxu+ToYpStovLmiVlqE8kMnNrQhizOa9K/TVs0  
KqwmcK1UUh8hE97RLRk2BDEEgDRs7c6ufXRm5pbI0dRvKSmlnleVj0LkBDQRRKy4  
AQgApGTtrw5q+v9uweEmySjGSm3/iySgJ/wvt8QmH5kcz+XbuEn/cV6/Rb9WZLSR  
dFrc77GVVgpybGudevlhaJbXUuH7nQ1QA6TPn/JewQqVz+aQTmvlr4B9tp+ryp3  
YtEoNHjRNNeaTPCO+SQRZD801K7ft0bvoF18c+qsxxIko+/kCXdmCu57YpnhsWY6  
pxqqP3Hy0Ay15NIweJk5kAa69p25mBV/T6r+oxoZ89bwC1bihTBVPM6tTAcRsKwS  
VKqQKI5spq+qNcMx18SNWEYulAREdrd230FlzrCwq2yLGTmZenAtrYmq87HX+Q/V  
60vXmWmFVnYxAlsJtjlmK9em6wARAQABiQEfBBgBAgAJBQJRKYy4AhsMAAoJEJNF  
ODX77RlpJxoIAJra+QnqKZnmGzyu8kJMoEy3uM2gnB88xbpvaZ4JFFhCScJn3F55  
+xppN5Tu2+fUbqm68F9r5tQtXQI1URqvKcg1qZhbWxTqXLLQqNEkbQ/zEbjfXwj  
nz1MBehLfYlpuqACXGJWYj5h+EarR8EY1OdrLZdY7AIkp0I/ot3bxDbao7ujpR/3  
aXiKfhvVvGo8B7I+cYv7k0wiBGDWMNgcwDvQZdeegED/0gYP683TppQlj0AwaZ0I  
Bau7Uox1AjvS8/UTaEYwRyFr8quwZ0QpADk7J+MSLJv3MRD1MtOmuvJrQyYtLFO1  
BxEQw0+2u+GxbCIC4qoQjAVLUUhib1hDh4Q=  
=DnGp  
-----END PGP PUBLIC KEY BLOCK-----
```

Appendix 3: Example of EPP extensions

Below are working XML sample code to input three different EPP extensions and their associated data. Together with each example, the Extensions part of the resulting EPP command that will be sent to applicant's server during testing is shown.

11.1.1 A normal extension with two Name:Value fields

```
<Extension>
  <URI>urn:se:iis:xml:epp:iis-1.2</URI>
  <SL>urn:se:iis:xml:epp:iis-1.2 iis-1.2-xsd</SL>
  <Field>
    <Name>orgno</Name>
    <Value>[SE]551112-3282</Value>
  </Field>
  <Field>
    <Name>vatno</Name>
    <Value>SE551112328201</Value>
  </Field>
</Extension>
```

Resulting EPP fragment

```
<extension>
  <ex01:create xmlns:ex01="urn:se:iis:xml:epp:iis-1.2"
    xsi:schemaLocation="urn:se:iis:xml:epp:iis-1.2 iis-1.2-xsd">
    <ex01:orgno>[SE]551112-3282</ex01:orgno>
    <ex01:vatno>SE551112328201</ex01:vatno>
  </ex01:create>
</extension>
```

11.1.2 An IDN extension with a single text node

```
<Extension>
  <ExtName>language</ExtName>
  <URI>urn:ietf:params:xml:ns:idn-1.0</URI>
  <SL>urn:ietf:params:xml:ns:idn-1.0 idn-1.0.xsd</SL>
  <ExtValue>ger</ExtValue>
</Extension>
```

Resulting EPP fragment

```
<extension>
  <ex01:language xmlns:ex01="urn:ietf:params:xml:ns:idn-1.0"
    xsi:schemaLocation="urn:ietf:params:xml:ns:idn-1.0 idn-1.0.xsd">ger</ex01:language>
</extension>
```

11.1.3 An IDN extension with a single Name:Value field

```
<Extension>
  <URI>urn:ar:params:xml:ns:idn-1.0</URI>
  <SL></SL>
  <Field>
    <Name>languageTag</Name>
    <Value>ar</Value>
  </Field>
</Extension>
```

Resulting EPP fragment

```
<extension>
  <ex01:create xmlns:ex01="urn:ar:params:xml:ns:idn-1.0" xsi:schemaLocation="">
    <ex01:languageTag>ar</ex01:languageTag>
  </ex01:create>
</extension>
```

11.1.4 An extension with fields within field

```
<Extension>
```

```
<URI>http://www.tcinet.ru/epp/tci-contact-ext-1.0</URI>
<SL>http://www.tcinet.ru/epp/tci-contact-ext-1.0 tci-contact-ext-1.0.xsd</SL>
<Field>
  <Name>person</Name>
  <Field>
    <Name>birthday</Name>
    <Value>1970-11-11</Value>
  </Field>
  <Field>
    <Name>passport</Name>
    <Value>passport string</Value>
  </Field>
  <Field>
    <Name>TIN</Name>
    <Value>4444444444444444</Value>
  </Field>
</Field>
</Extension>
```

Resulting EPP fragment

```
<extension>
  <ex01:create xmlns:ex01="http://www.tcinet.ru/epp/tci-contact-ext-1.0"
xsi:schemaLocation="http://www.tcinet.ru/epp/tci-contact-ext-1.0 tci-contact-ext-1.0.xsd">
  <ex01:person>
    <ex01:birthday>1970-11-11</ex01:birthday>
    <ex01:passport>passport string</ex01:passport>
    <ex01:TIN>4444444444444444</ex01:TIN>
  </ex01:person>
</ex01:create>
</extension>
```