

Registration Protocols Extensions
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2017

R. Carney
GoDaddy Inc.
G. Brown
CentralNic Group plc
J. Frakes
March 8, 2017

**Registry Fee Extension for the Extensible Provisioning Protocol (EPP)
draft-ietf-regext-epp-fees-02**

Abstract

This document describes an Extensible Provisioning Protocol (EPP) extension mapping for registry fees.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 9, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Conventions Used in This Document	3
2.	Migrating to Newer Versions of This Extension	4
3.	Extension Elements	4
3.1.	Client Commands	4
3.2.	Currency Codes	5
3.3.	Validity Periods	5
3.4.	Fees and Credits	5
3.4.1.	Refunds	6
3.4.2.	Grace Periods	6
3.4.3.	Correlation between Refundability and Grace Periods	7
3.4.4.	Applicability	7
3.5.	Account Balance	7
3.6.	Credit Limit	7
3.7.	Classification of Objects	8
3.8.	Phase and Subphase Attributes	8
4.	Server Handling of Fee Information	9
5.	EPP Command Mapping	9
5.1.	EPP Query Commands	9
5.1.1.	EPP <check> Command	9
5.1.1.1.	Server Handling of Elements	14
5.1.2.	EPP Transfer Query Command	15
5.2.	EPP Transform Commands	16
5.2.1.	EPP <create> Command	16
5.2.2.	EPP <delete> Command	19
5.2.3.	EPP <renew> Command	20
5.2.4.	EPP <transfer> Command	22
5.2.5.	EPP <update> Command	24
6.	Formal Syntax	26
6.1.	Fee Extension Schema	26
7.	Security Considerations	30
8.	IANA Considerations	30
8.1.	XML Namespace	30
8.2.	EPP Extension Registry	31
9.	Implementation Status	31
9.1.	RegistryEngine EPP Service	32
10.	Acknowledgements	32
11.	Change History	32
11.1.	Change from 01 to 02	32
11.2.	Change from 00 to 01	33
11.3.	Change from draft-brown-00 to draft-ietf-regext-fees-00	33
12.	Normative References	33
	Authors' Addresses	34

1. Introduction

Historically, domain name registries have applied a simple fee structure for billable transactions, namely a basic unit price applied to domain <create>, <renew>, <transfer> and RGP [\[RFC3915\]](#) restore commands. Given the relatively small number of EPP servers to which EPP clients have been required to connect, it has generally been the case that client operators have been able to obtain details of these fees out-of-band by contacting the server operators.

Given the recent expansion of the DNS namespace, and the proliferation of novel business models, it is now desirable to provide a method for EPP clients to query EPP servers for the fees and credits associated with certain commands and specific objects.

This document describes an extension mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) [\[RFC5730\]](#). This EPP mapping provides a mechanism by which EPP clients may query the fees and credits associated with various billable transactions, and also obtain their current account balance.

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [\[RFC2119\]](#).

XML is case sensitive. Unless stated otherwise, XML specifications and examples provided in this document MUST be interpreted in the character case presented in order to develop a conforming implementation.

"fee" is used as an abbreviation for "urn:ietf:params:xml:ns:fee-0.15". The XML namespace prefix "fee" is used, but implementations MUST NOT depend on it and instead employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

In examples, "C:" represents lines sent by a protocol client and "S:" represents lines returned by a protocol server. Indentation and white space in examples are provided only to illustrate element relationships and are not a REQUIRED feature of this protocol.

(Note to RFC Editor: remove the following paragraph before publication as an RFC.)

The XML namespace prefix above contains a version number, specifically "0.15". This version number will increment with successive versions of this document, and will reach 1.0 if and when

this document is published as an RFC. This permits clients to distinguish which version of the extension a server has implemented.

2. Migrating to Newer Versions of This Extension

(Note to RFC Editor: remove this section before publication as an RFC.)

Servers which implement this extension SHOULD provide a way for clients to progressively update their implementations when a new version of the extension is deployed.

Servers SHOULD (for a temporary migration period) provide support for older versions of the extension in parallel to the newest version, and allow clients to select their preferred version via the `<svcExtension>` element of the `<login>` command.

If a client requests multiple versions of the extension at login, then, when preparing responses to commands which do not include extension elements, the server SHOULD only include extension elements in the namespace of the newest version of the extension requested by the client.

When preparing responses to commands which do include extension elements, the server SHOULD only include extension elements for the extension versions present in the command.

3. Extension Elements

3.1. Client Commands

The `<fee:command>` element is used in the EPP `<check>` command to determine the fee which is applicable to the given command.

The list of values include:

- o "create" indicating a `<create>` command as defined in [\[RFC5730\]](#);
- o "delete" indicating a `<delete>` command as defined in [\[RFC5730\]](#);
- o "renew" indicating a `<renew>` command as defined in [\[RFC5730\]](#);
- o "update" indicating a `<update>` command as defined in [\[RFC5730\]](#);
- o "transfer" indicating a `<transfer>` command as defined in [\[RFC5730\]](#);
- o If the server supports the Registry Grace Period Mapping [\[RFC3915\]](#), then the server MUST also support the "restore" value as defined in [\[RFC3915\]](#).

The `<fee:command>` element MAY have an OPTIONAL "phase" attribute specifying a launch phase as described in [\[draft-ietf-eppext-](#)

launchphase]. It may also contain an OPTIONAL "subphase" attribute identifying the custom or sub-phase as described in [[draft-ietf-eppext-launchphase](#)].

3.2. Currency Codes

The <fee:currency> element is used to indicate which currency fees are charged in. This value of this element MUST be a three-character currency code from [ISO4217].

Note that ISO 4217 provides the special "XXX" code, which MAY be used if the server uses a non-currency based system for assessing fees, such as a system of credits.

The use of <fee:currency> elements in commands is OPTIONAL: if a <fee:currency> element is not present in a command, the server MUST determine the currency based on the client's account settings which MUST be agreed by the client and server via an out-of-band channel. However, the <fee:currency> element MUST be present in responses.

Servers SHOULD NOT perform a currency conversion if a client uses an incorrect currency code. Servers SHOULD return a 2004 "Parameter value range" error instead.

3.3. Validity Periods

When querying for fee information using the <check> command, the <fee:period> element is used to indicate the units to be added to the registration period of objects by the <create>, <renew> and <transfer> commands. This element is derived from the <domain:period> element described in [[RFC5731](#)].

The <fee:period> element is OPTIONAL in <check> commands: if omitted, the server MUST determine the fee(s) using a validity period of 1 year. The <fee:period> element MUST be present in <check> responses.

3.4. Fees and Credits

Servers which implement this extension will include elements in responses which provide information about the fees and/or credits associated with a given billable transaction.

The <fee:fee> and <fee:credit> elements are used to provide this information. The presence of a <fee:fee> element in a response indicates a debit against the client's account balance; a <fee:credit> element indicates a credit. A <fee:fee> element MUST have a non-negative value. A <fee:credit> element MUST have a negative value.

A server MAY respond with multiple `<fee:fee>` and `<fee:credit>` elements in the same response. In such cases, the net fee or credit applicable to the transaction is the arithmetic sum of the values of each of the `<fee:fee>` and/or `<fee:credit>` elements. This amount applies to the total additional validity period applied to the object (where applicable) rather than to any incremental unit.

The following attributes are defined for the `<fee:fee>` element. These are described in detail below:

`description`: an OPTIONAL attribute which provides a human-readable description of the fee. Servers should provide documentation on the possible values of this attribute, and their meanings.

`refundable`: an OPTIONAL boolean attribute indicating whether the fee is refundable if the object is deleted.

`grace-period`: an OPTIONAL attribute which provides the time period during which the fee is refundable.

`applied`: an OPTIONAL attribute indicating when the fee will be deducted from the client's account.

The `<fee:credit>` element can take a `"description"` attribute as described above. No other attributes are defined for this element.

3.4.1. Refunds

`<fee:fee>` elements MAY have an OPTIONAL `"refundable"` attribute which takes a boolean value. Fees may be refunded under certain circumstances, such as when a domain application is rejected (as described in [[draft-ietf-epext-launchphase](#)]) or when an object is deleted during the relevant Grace Period (see below).

If the `"refundable"` attribute is omitted, then clients SHOULD NOT make any assumption about the refundability of the fee.

3.4.2. Grace Periods

[RFC3915] describes a system of "grace periods", which are time periods following a billable transaction during which, if an object is deleted, the client receives a refund.

The `"grace-period"` attribute MAY be used to indicate the relevant grace period for a fee. If a server implements the Registry Grace Period extension, it MUST specify the grace period for all relevant transactions.

If the "grace-period" attribute is omitted, then clients SHOULD NOT make any assumption about the grace period of the fee.

3.4.3. Correlation between Refundability and Grace Periods

If a <fee:fee> element has a "grace-period" attribute then it MUST also be refundable. If the "refundable" attribute of a <fee:fee> element is false then it MUST NOT have a "grace-period" attribute.

3.4.4. Applicability

Fees may be applied immediately upon receipt of a command from a client, or may only be applied once an out-of-band process (such as the processing of applications at the end of a launch phase) has taken place.

The "applied" attribute of the <fee:fee> element allows servers to indicate whether a fee will be applied immediately, or whether it will be applied at some point in the future. This attribute takes two possible values: "immediate" or "delayed".

3.5. Account Balance

The <fee:balance> element is an OPTIONAL element which MAY be included in server responses to transform commands. If present, it can be used by the client to determine the remaining credit at the server.

Whether or not the <fee:balance> is included in responses is a matter of server policy. However, if a server chooses to offer support for this element, it MUST be included in responses to all "transform" or billable commands (ie <create>, <renew>, <update>, <delete>, <transfer op="request">).

The value of the <fee:balance> MAY be negative. A negative balance indicates that the server has extended a line of credit to the client (see below).

If a server includes a <fee:balance> element in response to transform commands, the value of the element MUST reflect the client's account balance after any fees or credits associated with that command have been applied.

3.6. Credit Limit

As described above, if a server returns a response containing a <fee:balance> with a negative value, then the server has extended a line of credit to the client. A server MAY also include a

<fee:creditLimit> element in responses which indicates the maximum credit available to a client. A server MAY reject certain transactions if the absolute value of the <fee:balance> is equal to or exceeds the value of the <fee:creditLimit> element.

Whether or not the <fee:creditLimit> is included in responses is a matter of server policy. However, if a server chooses to offer support for this element, it MUST be included in responses to all "transform" commands (ie <create>, <renew>, <update>, <delete>, <transfer op="request">).

3.7. Classification of Objects

Objects may be assigned to a particular class, category, or tier, each of which has a particular fee or set of fees associated with it. The <fee:class> element which appears in <check> responses is used to indicate the classification of an object.

If a server makes use of this element, it should provide clients with a list of all the values that the element may take via an out-of-band channel. Servers MUST NOT use values which do not appear on this list.

Servers which make use of this element MUST use a <fee:class> element with the value "standard" for all objects that are subject to the standard or default fee.

3.8. Phase and Subphase Attributes

The <fee:command> element has two attributes, phase and subphase, that provide additional information related to a specific launch phase as described in [[draft-ietf-eppext-launchphase](#)].

If the client <fee:command> contains no phase attribute the server SHOULD return data (including the phase/subphase attribute(s)) for all valid server combinations of currently active phase(s) and active subphase(s).

If the client <fee:command> contains a server supported combination of phase/subphase the server MUST return fee data (including the phase/subphase attribute(s)) for the specific combination(s).

If the client <fee:command> contains a phase attribute with no subphase, the server SHOULD return data (including the phase/subphase attribute(s)) for all valid server combinations of provided phase and available subphase(s).

If the client `<fee:command>` contains a subphase with no phase attribute the server MUST respond with a 2003 "Required parameter missing" error.

If the client `<fee:command>` contains a phase attribute not defined in [[draft-ietf-eppext-launchphase](#)] or not supported by server the server MUST respond with a 2004 "Parameter value range" error.

If the client `<fee:command>` contains a subphase attribute (or phase/subphase combination) not supported by server the server MUST respond with a 2004 "Parameter value range" error.

4. Server Handling of Fee Information

Depending on server policy, a client MAY be required to include the extension elements described in this document for certain transform commands. Servers must provide clear documentation to clients about the circumstances in which this extension must be used.

If a server receives a command from a client which does not include the extension elements required by the server for that command, then it MUST respond with a 2003 "Required parameter missing" error.

If the currency or total fee provided by the client do not agree with the server's own calculation of the fee for that command, then the server MUST reject the command with a 2004 "Parameter value range" error.

5. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in [[RFC5730](#)].

5.1. EPP Query Commands

This extension does not add any elements to the EPP `<poll>` or `<info>` commands or responses.

5.1.1. EPP `<check>` Command

This extension defines additional elements for the EPP `<check>` command.

The command MAY contain an `<extension>` element which MAY contain a `<fee:check>` element. The `<fee:check>` element MUST contain one or more `<fee:object>` elements containing the following child elements:

- o A <fee:objID> element that MUST match one and only one of the domain names in the <domain:check>;
- o An OPTIONAL <fee:currency> element. If the server does not support <fee:currency> value, it MUST return a 2306 error response;
- o One or more <fee:command> element(s) with a "name" attribute, an OPTIONAL "phase" attribute, and an OPTIONAL "subphase" attribute.

The <fee:command> element(s) MAY have the following child elements:

- o An OPTIONAL <fee:period> element.
- o An OPTIONAL <fee:class> element.

Example <check> command:

```
C: <?xml version="1.0" encoding="utf-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:   <command>
C:     <check>
C:       <domain:check
C:         xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:           <domain:name>example.com</domain:name>
C:           <domain:name>example.net</domain:name>
C:           <domain:name>example.xyz</domain:name>
C:         </domain:check>
C:       </check>
C:     <extension>
C:       <fee:check xmlns:fee="urn:ietf:params:xml:ns:fee-0.15">
C:         <fee:object>
C:           <fee:objID>example.com</fee:objID>
C:           <fee:currency>USD</fee:currency>
C:           <fee:command name="create">
C:             <fee:period unit="y">2</fee:period>
C:           </fee:command>
C:           <fee:command name="renew"/>
C:           <fee:command name="transfer"/>
C:           <fee:command name="restore"/>
C:         </fee:object>
C:         <fee:object>
C:           <fee:objID>example.net</fee:objID>
C:           <fee:currency>USD</fee:currency>
C:           <fee:command name="create">
C:             <fee:period unit="y">11</fee:period>
C:           </fee:command>
C:           <fee:command name="renew"/>
C:         </fee:object>
C:       </fee:check>
C:     </extension>
C:   <clTRID>ABC-12345</clTRID>
C: </command>
C: </epp>
```

When the server receives a <check> command that includes the extension elements described above, its response MUST contain an <extension> element, which MUST contain a child <fee:chkData> element. The <fee:chkData> element MUST contain a <fee:cd> element for each object <fee:object> element referenced in the <fee:check> element of the client command.

Each <fee:cd> element MUST contain the following child elements:

- o A <fee:objID> element, which MUST match a <fee:objID> from the client command.
- o A <fee:currency> element, which contains the same currency code that appeared in the <fee:currency> element of the command. If no <fee:currency> element appeared in the command, then the client's default billing currency should be used.
- o A <fee:command> element matching each <fee:command> that appeared in the corresponding <fee:object> of the command. If a <fee:object> element in the client command contains no <fee:command> element the server SHOULD return a <fee:command> element for each server billable transaction combination of the <fee:object>. This element MAY have the OPTIONAL "phase" and "subphase" attributes, which MUST match the same attributes in the corresponding <fee:object> element of the client command.

The <fee:command> element also has an OPTIONAL "avail" attribute which is a boolean. If the value of this attribute evaluates to false, this indicates that the server cannot calculate the relevant fees, because the object, command, currency, period or class is invalid according to server policy.

The <fee:command> element(s) MAY have the following child elements:

- o An OPTIONAL <fee:period> element, which contains the same unit that appeared in the <fee:period> element of the command. If the value of the preceding <fee:command> element is "restore", this element MUST NOT be included. Otherwise it MUST be included. If no <fee:period> appeared in command (and the command is not "restore") then the server MUST return its default period value.
- o Zero or more <fee:fee> elements.
- o Zero or more <fee:credit> elements.
- o An OPTIONAL <fee:class> element.
- o An OPTIONAL <fee:reason> element.

If no <fee:fee> elements are present in a <fee:cd> element, this indicates that no fee will be assessed by the server for this command.

If the "avail" attribute of the <fee:command> element is false, then the <fee:command> element MUST NOT contain any <fee:fee> or <fee:credit> child elements. If the "avail" attribute is true, then the <fee:command> element MUST NOT contain a <fee:reason> element.

Example <check> response:

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



```
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:chkData
S:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:          <domain:cd>
S:            <domain:name avail="1">example.com</domain:name>
S:          </domain:cd>
S:          <domain:cd>
S:            <domain:name avail="1">example.net</domain:name>
S:          </domain:cd>
S:          <domain:cd>
S:            <domain:name avail="1">example.xyz</domain:name>
S:          </domain:cd>
S:        </domain:chkData>
S:      </resData>
S:    <extension>
S:      <fee:chkData
S:        xmlns:fee="urn:ietf:params:xml:ns:fee-0.15"
S:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:      <fee:cd>
S:        <fee:objID>example.com</fee:objID>
S:        <fee:currency>USD</fee:currency>
S:        <fee:command name="create">
S:          <fee:period unit="y">2</fee:period>
S:          <fee:fee
S:            description="Registration Fee"
S:            refundable="1"
S:            grace-period="P5D">10.00</fee:fee>
S:        </fee:command>
S:        <fee:command name="renew">
S:          <fee:period unit="y">1</fee:period>
S:          <fee:fee
S:            description="Renewal Fee"
S:            refundable="1"
S:            grace-period="P5D">5.00</fee:fee>
S:        </fee:command>
S:        <fee:command name="transfer">
S:          <fee:period unit="y">1</fee:period>
S:          <fee:fee
S:            description="Transfer Fee"
S:            refundable="1"
S:            grace-period="P5D">5.00</fee:fee>
S:        </fee:command>
S:        <fee:command name="restore">
S:          <fee:fee
S:            description="Redemption Fee">5.00</fee:fee>
```



```
S:      </fee:command>
S:      </fee:cd>
S:      <fee:cd>
S:      <fee:objID>example.net</fee:objID>
S:      <fee:currency>USD</fee:currency>
S:      <fee:command avail="0" name="create">
S:      <fee:reason>Maximum period is 10 years</fee:reason>
S:      </fee:command>
S:      <fee:command name="renew">
S:      <fee:period unit="y">1</fee:period>
S:      <fee:fee
S:      description="Renewal Fee"
S:      refundable="1"
S:      grace-period="P5D">5.00</fee:fee>
S:      </fee:command>
S:      </fee:cd>
S:      </fee:chkData>
S:      </extension>
S:      <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:      </trID>
S:      </response>
S: </epp>
```

5.1.1.1. Server Handling of Elements

Clients MAY include a `<fee:class>` in the `<fee:command>` element.

There are three ways in which servers may handle this element:

1. If the server supports the concept of tiers or classes of objects, then the value of this element MUST be validated. If incorrect for the specified object, the "avail" attribute of the corresponding `<fee:command>` element MUST be false.
2. If the server supports different "types" of object registrations (such as a "blocking" registration which does not resolve, or where a registry provides a value-added service that requires an opt-out to disable), then, as with the first model, the server MUST validate the value of the element. If the value is incorrect, the "avail" attribute of the corresponding `<fee:command>` element MUST be false.
3. If the server supports neither of the above models, the element MUST be ignored.

Server operators must provide clear documentation to client operators which of the above models it supports.

5.1.2. EPP Transfer Query Command

This extension does not add any elements to the EPP <transfer> query command, but does include elements in the response, when the extension has been selected during a <login> command.

When the <transfer> query command has been processed successfully, the client selected the extension when it logged in, and the client is authorised by the server to view information about the transfer, the server MAY include in the <extension> section of the EPP response a <fee:trnData> element, which contains the following child elements:

- o A <fee:currency> element.
- o A <fee:period> element.
- o Zero or more <fee:fee> elements containing the fees that will be charged to the gaining client.
- o Zero or more <fee:credit> elements containing the credits that will be refunded to the losing client.

Servers SHOULD omit <fee:credit> when returning a response to the gaining client, and omit <fee:fee> elements when returning a response to the losing client.

If no <fee:trnData> element is included in the response, then no fee will be assessed by the server for the transfer.

Example <transfer> query response:

```
S: <?xml version="1.0" encoding="utf-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:   <response>
S:     <result code="1001">
S:       <msg>Command completed successfully; action pending</msg>
S:     </result>
S:     <resData>
S:       <domain:trnData
S:         xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:         <domain:name>example.com</domain:name>
S:         <domain:trStatus>pending</domain:trStatus>
S:         <domain:reID>ClientX</domain:reID>
S:         <domain:reDate>2000-06-08T22:00:00.0Z</domain:reDate>
S:         <domain:acID>ClientY</domain:acID>
S:         <domain:acDate>2000-06-13T22:00:00.0Z</domain:acDate>
S:         <domain:exDate>2002-09-08T22:00:00.0Z</domain:exDate>
S:       </domain:trnData>
S:     </resData>
S:     <extension>
S:       <fee:trnData xmlns:fee="urn:ietf:params:xml:ns:fee-0.15">
S:         <fee:currency>USD</fee:currency>
S:         <fee:period unit="y">1</fee:period>
S:         <fee:fee>5.00</fee:fee>
S:       </fee:trnData>
S:     </extension>
S:     <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>54322-XYZ</svTRID>
S:     </trID>
S:   </response>
S: </epp>
```

5.2. EPP Transform Commands

5.2.1. EPP <create> Command

This extension adds elements to both the EPP <create> command and response, when the extension has been selected during a <login> command.

If a <create> command is received with no <fee:create> extension and the server requires a <fee:create> extension for the <domain:name> the server MUST respond with a 2003 "Required parameter missing" error.

When submitting a <create> command to the server, the client MAY include in the <extension> element a <fee:create> element which includes the following child elements:

- o An OPTIONAL <fee:currency> element;
- o One or more <fee:fee> elements.

When the <create> command has been processed successfully, and the client selected the extension when it logged in, and a fee was assessed by the server for the transaction, the server MUST include in the <extension> section of the EPP response a <fee:creData> element, which contains the following child elements:

- o A <fee:currency> element;
- o Zero or more <fee:fee> elements;
- o Zero or more <fee:credit> elements;
- o An OPTIONAL <fee:balance> element;
- o An OPTIONAL <fee:creditLimit> element.

Example <create> command:

```
C: <?xml version="1.0" encoding="utf-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:   <command>
C:     <create>
C:       <domain:create
C:         xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:           <domain:name>example.com</domain:name>
C:           <domain:period unit="y">2</domain:period>
C:           <domain:ns>
C:             <domain:hostObj>ns1.example.net</domain:hostObj>
C:             <domain:hostObj>ns2.example.net</domain:hostObj>
C:           </domain:ns>
C:           <domain:registrant>jd1234</domain:registrant>
C:           <domain:contact type="admin">sh8013</domain:contact>
C:           <domain:contact type="tech">sh8013</domain:contact>
C:           <domain:authInfo>
C:             <domain:pw>2fooBAR</domain:pw>
C:           </domain:authInfo>
C:         </domain:create>
C:       </create>
C:     <extension>
C:       <fee:create xmlns:fee="urn:ietf:params:xml:ns:fee-0.15">
C:         <fee:currency>USD</fee:currency>
C:         <fee:fee>5.00</fee:fee>
C:       </fee:create>
C:     </extension>
C:     <clTRID>ABC-12345</clTRID>
C:   </command>
C: </epp>
```


Example <create> response:

```
S: <?xml version="1.0" encoding="utf-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:   <response>
S:     <result code="1000">
S:       <msg>Command completed successfully</msg>
S:     </result>
S:     <resData>
S:       <domain:creData
S:         xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:         <domain:name>example.com</domain:name>
S:         <domain:crDate>1999-04-03T22:00:00.0Z</domain:crDate>
S:         <domain:exDate>2001-04-03T22:00:00.0Z</domain:exDate>
S:       </domain:creData>
S:     </resData>
S:     <extension>
S:       <fee:creData xmlns:fee="urn:ietf:params:xml:ns:fee-0.15">
S:         <fee:currency>USD</fee:currency>
S:         <fee:fee
S:           description="Registration Fee"
S:           refundable="1"
S:           grace-period="P5D">5.00</fee:fee>
S:         <fee:balance>-5.00</fee:balance>
S:         <fee:creditLimit>1000.00</fee:creditLimit>
S:       </fee:creData>
S:     </extension>
S:     <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>54321-XYZ</svTRID>
S:     </trID>
S:   </response>
S: </epp>
```

5.2.2. EPP <delete> Command

This extension does not add any elements to the EPP <delete> command, but does include elements in the response, when the extension has been selected during the <login> command.

When the <delete> command has been processed successfully, and the client selected the extension when it logged in, the server MAY include in the <extension> section of the EPP response a <fee:delData> element, which contains the following child elements:

- o A <fee:currency> element;
- o Zero or more <fee:fee> elements;
- o Zero or more <fee:credit> elements;

- o An OPTIONAL <fee:balance> element;
- o An OPTIONAL <fee:creditLimit> element.

Example <delete> response:

```
S: <?xml version="1.0" encoding="utf-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:   <response>
S:     <result code="1000">
S:       <msg>Command completed successfully</msg>
S:     </result>
S:     <extension>
S:       <fee:delData
S:         xmlns:fee="urn:ietf:params:xml:ns:fee-0.15">
S:         <fee:currency>USD</fee:currency>
S:         <fee:credit description="AGP Credit">-5.00</fee:credit>
S:         <fee:balance>1005.00</fee:balance>
S:       </fee:delData>
S:     </extension>
S:     <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>54321-XYZ</svTRID>
S:     </trID>
S:   </response>
S: </epp>
```

5.2.3. EPP <renew> Command

This extension adds elements to both the EPP <renew> command and response, when the extension has been selected during a <login> command.

When submitting a <renew> command to the server, the client MAY include in the <extension> element a <fee:renew> element which includes the following child elements:

- o An OPTIONAL <fee:currency> element;
- o One or more <fee:fee> elements.

When the <renew> command has been processed successfully, and the client selected the extension when it logged in, the server MAY include in the <extension> section of the EPP response a <fee:renData> element, which contains the following child elements:

- o A <fee:currency> element;
- o Zero or more <fee:fee> elements;
- o Zero or more <fee:credit> elements;
- o An OPTIONAL <fee:balance> element;

- o An OPTIONAL <fee:creditLimit> element.

Example <renew> command:

```
C: <?xml version="1.0" encoding="utf-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:   <command>
C:     <renew>
C:       <domain:renew
C:         xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:           <domain:name>example.com</domain:name>
C:           <domain:curExpDate>2000-04-03</domain:curExpDate>
C:           <domain:period unit="y">5</domain:period>
C:         </domain:renew>
C:       </renew>
C:     <extension>
C:       <fee:renew xmlns:fee="urn:ietf:params:xml:ns:fee-0.15">
C:         <fee:currency>USD</fee:currency>
C:         <fee:fee>5.00</fee:fee>
C:       </fee:renew>
C:     </extension>
C:     <clTRID>ABC-12345</clTRID>
C:   </command>
C: </epp>
```


Example <renew> response:

```
S: <?xml version="1.0" encoding="utf-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:   <response>
S:     <result code="1000">
S:       <msg>Command completed successfully</msg>
S:     </result>
S:     <resData>
S:       <domain:renData
S:         xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:         <domain:name>example.com</domain:name>
S:         <domain:exDate>2005-04-03T22:00:00.0Z</domain:exDate>
S:       </domain:renData>
S:     </resData>
S:     <extension>
S:       <fee:renData xmlns:fee="urn:ietf:params:xml:ns:fee-0.15">
S:         <fee:currency>USD</fee:currency>
S:         <fee:fee
S:           refundable="1"
S:           grace-period="P5D">5.00</fee:fee>
S:         <fee:balance>1000.00</fee:balance>
S:       </fee:renData>
S:     </extension>
S:     <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>54322-XYZ</svTRID>
S:     </trID>
S:   </response>
S: </epp>
```

5.2.4. EPP <transfer> Command

This extension adds elements to both the EPP <transfer> command and response, when the value of the "op" attribute of the <transfer> command element is "request", and the extension has been selected during the <login> command.

When submitting a <transfer> command to the server, the client MAY include in the <extension> element a <fee:transfer> element which includes the following child elements:

- o An OPTIONAL <fee:currency> element;
- o One or more <fee:fee> elements.

When the <transfer> command has been processed successfully, and the client selected the extension when it logged in, the server MAY

include in the <extension> section of the EPP response a <fee:trnData> element, which contains the following child elements:

- o A <fee:currency> element;
- o Zero or more <fee:fee> elements;
- o Zero or more <fee:credit> elements;
- o An OPTIONAL <fee:balance> element;
- o An OPTIONAL <fee:creditLimit> element.

Example <transfer> command:

```
C: <?xml version="1.0" encoding="utf-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:   <command>
C:     <transfer op="request">
C:       <domain:transfer
C:         xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:           <domain:name>example.com</domain:name>
C:           <domain:period unit="y">1</domain:period>
C:           <domain:authInfo>
C:             <domain:pw roid="JD1234-REP">2fooBAR</domain:pw>
C:           </domain:authInfo>
C:         </domain:transfer>
C:       </transfer>
C:     <extension>
C:       <fee:transfer xmlns:fee="urn:ietf:params:xml:ns:fee-0.15">
C:         <fee:currency>USD</fee:currency>
C:         <fee:fee>5.00</fee:fee>
C:       </fee:transfer>
C:     </extension>
C:     <clTRID>ABC-12345</clTRID>
C:   </command>
C: </epp>
```


Example <transfer> response:

```
S: <?xml version="1.0" encoding="utf-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:   <response>
S:     <result code="1001">
S:       <msg>Command completed successfully; action pending</msg>
S:     </result>
S:     <resData>
S:       <domain:trnData
S:         xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:           <domain:name>example.com</domain:name>
S:           <domain:trStatus>pending</domain:trStatus>
S:           <domain:reID>ClientX</domain:reID>
S:           <domain:reDate>2000-06-08T22:00:00.0Z</domain:reDate>
S:           <domain:acID>ClientY</domain:acID>
S:           <domain:acDate>2000-06-13T22:00:00.0Z</domain:acDate>
S:           <domain:exDate>2002-09-08T22:00:00.0Z</domain:exDate>
S:         </domain:trnData>
S:       </resData>
S:       <extension>
S:         <fee:trnData xmlns:fee="urn:ietf:params:xml:ns:fee-0.15">
S:           <fee:currency>USD</fee:currency>
S:           <fee:fee
S:             refundable="1"
S:             grace-period="P5D">5.00</fee:fee>
S:         </fee:trnData>
S:       </extension>
S:     <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>54322-XYZ</svTRID>
S:     </trID>
S:   </response>
S: </epp>
```

5.2.5. EPP <update> Command

This extension adds elements to both the EPP <update> command and response, when the extension has been selected during the <login> command.

When submitting a <update> command to the server, the client MAY include in the <extension> element a <fee:update> element which includes the following child elements:

- o An OPTIONAL <fee:currency> element;
- o One or more <fee:fee> elements.

When the <update> command has been processed successfully, and the client selected the extension when it logged in, the server MAY include in the <extension> section of the EPP response a <fee:updData> element, which contains the following child elements:

- o A <fee:currency> element;
- o Zero or more <fee:fee> elements;
- o Zero or more <fee:credit> elements;
- o An OPTIONAL <fee:balance> element;
- o An OPTIONAL <fee:creditLimit> element.

Example <update> command:

```
C: <?xml version="1.0" encoding="utf-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:   <command>
C:     <update>
C:       <domain:update
C:         xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:           <domain:name>example.com</domain:name>
C:           <domain:chg>
C:             <domain:registrant>sh8013</domain:registrant>
C:           </domain:chg>
C:         </domain:update>
C:       </update>
C:     <extension>
C:       <fee:update xmlns:fee="urn:ietf:params:xml:ns:fee-0.15">
C:         <fee:currency>USD</fee:currency>
C:         <fee:fee>5.00</fee:fee>
C:       </fee:update>
C:     </extension>
C:     <clTRID>ABC-12345</clTRID>
C:   </command>
C: </epp>
```


Example <update> response:

```
S: <?xml version="1.0" encoding="utf-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:   <response>
S:     <result code="1000">
S:       <msg>Command completed successfully</msg>
S:     </result>
S:     <extension>
S:       <fee:updData xmlns:fee="urn:ietf:params:xml:ns:fee-0.15">
S:         <fee:currency>USD</fee:currency>
S:         <fee:fee>5.00</fee:fee>
S:       </fee:updData>
S:     </extension>
S:     <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>54321-XYZ</svTRID>
S:     </trID>
S:   </response>
S: </epp>
```

6. Formal Syntax

One schema is presented here that is the EPP Fee Extension schema.

The formal syntax presented here is a complete schema representation of the object mapping suitable for automated validation of EPP XML instances. The BEGIN and END tags are not part of the schema; they are used to note the beginning and ending of the schema for URI registration purposes.

6.1. Fee Extension Schema

BEGIN

```
<?xml version="1.0" encoding="utf-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:fee="urn:ietf:params:xml:ns:fee-0.15"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
  targetNamespace="urn:ietf:params:xml:ns:fee-0.15"
  elementFormDefault="qualified">

  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0" />
  <import namespace="urn:ietf:params:xml:ns:domain-1.0" />

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0 Fee Extension
```



```
</documentation>
</annotation>

<!-- Child elements found in EPP commands and responses -->
<element name="check" type="fee:checkType" />
<element name="chkData" type="fee:chkDataType" />
<element name="create" type="fee:transformCommandType" />
<element name="creData" type="fee:transformResultType" />
<element name="renew" type="fee:transformCommandType" />
<element name="renData" type="fee:transformResultType" />
<element name="transfer" type="fee:transformCommandType" />
<element name="trnData" type="fee:transformResultType" />
<element name="update" type="fee:transformCommandType" />
<element name="updData" type="fee:transformResultType" />
<element name="delData" type="fee:transformResultType" />

<!-- client <check> command -->
<complexType name="checkType">
  <sequence>
    <element name="object" type="fee:objectCheckType"
      maxOccurs="unbounded" />
  </sequence>
</complexType>

<complexType name="objectCheckType">
  <sequence>
    <element name="objID" type="fee:objectIdentifierType" />
    <element name="currency" type="fee:currencyType"
      minOccurs="0" />
    <element name="command" type="fee:commandType"
      maxOccurs="unbounded" />
  </sequence>
</complexType>

<complexType name="objectIdentifierType">
  <simpleContent>
    <extension base="eppcom:labelType">
      <attribute name="element"
        type="NMTOKEN" default="name" />
    </extension>
  </simpleContent>
</complexType>

<!-- server <check> result -->
<complexType name="chkDataType">
  <sequence>
    <element name="cd" type="fee:objectCDType"
      maxOccurs="unbounded" />
```



```
</sequence>
</complexType>

<complexType name="objectCDType">
  <sequence>
    <element name="objID" type="fee:objectIdentifierType" />
    <element name="currency" type="fee:currencyType" />
    <element name="command" type="fee:commandType"
      maxOccurs="unbounded" />
  </sequence>
</complexType>

<!-- general transform (create, renew, update, transfer) command -->
<complexType name="transformCommandType">
  <sequence>
    <element name="currency" type="fee:currencyType"
      minOccurs="0" />
    <element name="fee" type="fee:feeType"
      maxOccurs="unbounded" />
    <element name="credit" type="fee:creditType"
      minOccurs="0" maxOccurs="unbounded" />
  </sequence>
</complexType>

<!-- general transform (create, renew, update) result -->
<complexType name="transformResultType">
  <sequence>
    <element name="currency" type="fee:currencyType"
      minOccurs="0" />
    <element name="period" type="domain:periodType"
      minOccurs="0" />
    <element name="fee" type="fee:feeType"
      minOccurs="0" maxOccurs="unbounded" />
    <element name="credit" type="fee:creditType"
      minOccurs="0" maxOccurs="unbounded" />
    <element name="balance" type="fee:balanceType"
      minOccurs="0" />
    <element name="creditLimit" type="fee:creditLimitType"
      minOccurs="0" />
  </sequence>
</complexType>

<!-- common types -->
<simpleType name="currencyType">
  <restriction base="string">
    <pattern value="[A-Z]{3}" />
  </restriction>
</simpleType>
```



```
<complexType name="commandType">
  <sequence>
    <element name="period" type="domain:periodType"
      minOccurs="0" maxOccurs="1" />
    <element name="fee" type="fee:feeType"
      minOccurs="0" maxOccurs="unbounded" />
    <element name="credit" type="fee:creditType"
      minOccurs="0" maxOccurs="unbounded" />
    <element name="class" type="token" minOccurs="0" />
    <element name="reason" type="token" minOccurs="0" />
  </sequence>
  <attribute name="name" type="fee:commandTypeValue"/>
  <attribute name="phase" type="token" />
  <attribute name="subphase" type="token" />
  <attribute name="avail" type="boolean" default="1" />
</complexType>

<simpleType name="commandTypeValue">
  <restriction base="token">
    <minLength value="3"/>
    <maxLength value="16"/>
  </restriction>
</simpleType>

<simpleType name="nonNegativeDecimal">
  <restriction base="decimal">
    <minInclusive value="0" />
  </restriction>
</simpleType>

<simpleType name="negativeDecimal">
  <restriction base="decimal">
    <maxInclusive value="0" />
  </restriction>
</simpleType>

<complexType name="feeType">
  <simpleContent>
    <extension base="fee:nonNegativeDecimal">
      <attribute name="description"/>
      <attribute name="refundable" type="boolean" />
      <attribute name="grace-period" type="duration" />
      <attribute name="applied">
        <simpleType>
          <restriction base="token">
            <enumeration value="immediate" />
            <enumeration value="delayed" />
          </restriction>
        </simpleType>
      </attribute>
    </extension>
  </simpleContent>
</complexType>
```



```
        </simpleType>
      </attribute>
    </extension>
  </simpleContent>
</complexType>

<complexType name="creditType">
  <simpleContent>
    <extension base="fee:negativeDecimal">
      <attribute name="description"/>
    </extension>
  </simpleContent>
</complexType>

<simpleType name="balanceType">
  <restriction base="decimal" />
</simpleType>

<simpleType name="creditLimitType">
  <restriction base="decimal" />
</simpleType>

</schema>
END
```

7. Security Considerations

The mapping extensions described in this document do not provide any security services beyond those described by EPP [[RFC5730](#)], the EPP domain name mapping [[RFC5731](#)], and protocol layers used by EPP. The security considerations described in these other specifications apply to this specification as well.

8. IANA Considerations

8.1. XML Namespace

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [[RFC3688](#)]. The following URI assignment is requested of IANA:

URI: ietf:params:xml:ns:fee-0.15

Registrant Contact: See the "Author's Address" section of this document.

XML: See the "Formal Syntax" section of this document.

8.2. EPP Extension Registry

The EPP extension described in this document should be registered by the IANA in the EPP Extension Registry described in [[RFC7451](#)]. The details of the registration are as follows:

Name of Extension: EPP Fee Extension

Document status: Standards Track

Reference: (insert reference to RFC version of this document)

Registrant Name and Email Address: See the "Author's Address" section of this document.

TLDs: any

IPR Disclosure: none

Status: active

Notes: none

9. Implemntation Status

Note to RFC Editor: Please remove this section and the reference to [[RFC6982](#)] before publication.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [[RFC6982](#)]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [[RFC6982](#)], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

9.1. RegistryEngine EPP Service

Organization: CentralNic

Name: RegistryEngine EPP Service

Description: Generic high-volume EPP service for gTLDs, ccTLDs and SLDs

Level of maturity: Deployed in CentralNic's production environment as well as two other gTLD registry systems, and two ccTLD registry systems.

Coverage: All aspects of the protocol are implemented.

Licensing: Proprietary In-House software

Contact: epp@centralnic.com

URL: <https://www.centralnic.com>

10. Acknowledgements

The authors wish to thank the following persons for their feedback and suggestions:

- o James Gould of Verisign Inc
- o Luis Munoz of ISC
- o Michael Young of Architelos
- o Ben Levac and Jeff Eckhaus of Demand Media
- o Seth Goldman of Google
- o Klaus Malorny and Michael Bauland of Knipp
- o Jody Kolker, Joe Snitker and Kevin Allendorf of Go Daddy
- o Michael Holloway of Com Laude
- o Santosh Kalsangrah of Impetus Infotech
- o Alex Mayrhofer of Nic.at

11. Change History

11.1. Change from 01 to 02

Updated scheme to version 0.15 to fix errors in CommandType, objectCDType, transformCommandType and transformResultType definitions.

11.2. Change from 00 to 01

Added Roger Carney as author to finish draft. Moved Formal Syntax section to main level numbering. Various grammar, typos, and administrative edits for clarity. Removed default value for the "applied" attribute of <fee:fee> so that it can truly be optional. Added support for the <delete> command to return a <fee:fee> element as well. Modified default response on the <check> command for the optional <fee:period> when it was not provided in the command, leaving it to the server to provide the default period value. Extensive edits were done to the <check> command, the <check> response and to the fee extension schema (checkType, objectCheckType, objectIdentifierType, objectCDType, commandType) to support requesting and returning multiple transformation fees in a single call. Added section on Phase/Subphase to provide more context on the uses.

11.3. Change from draft-brown-00 to draft-ietf-regext-fees-00

Updated to be REGEXT WG document.

12. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), DOI 10.17487/RFC3688, January 2004, <<http://www.rfc-editor.org/info/rfc3688>>.
- [RFC3915] Hollenbeck, S., "Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP)", [RFC 3915](#), DOI 10.17487/RFC3915, September 2004, <<http://www.rfc-editor.org/info/rfc3915>>.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, [RFC 5730](#), DOI 10.17487/RFC5730, August 2009, <<http://www.rfc-editor.org/info/rfc5730>>.
- [RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, [RFC 5731](#), DOI 10.17487/RFC5731, August 2009, <<http://www.rfc-editor.org/info/rfc5731>>.

[RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", [RFC 6982](#), DOI 10.17487/RFC6982, July 2013, <<http://www.rfc-editor.org/info/rfc6982>>.

[RFC7451] Hollenbeck, S., "Extension Registry for the Extensible Provisioning Protocol", [RFC 7451](#), DOI 10.17487/RFC7451, February 2015, <<http://www.rfc-editor.org/info/rfc7451>>.

Authors' Addresses

Roger Carney
GoDaddy Inc.
14455 N. Hayden Rd. #219
Scottsdale, AZ 85260
US

Email: rcarney@godaddy.com
URI: <http://www.godaddy.com>

Gavin Brown
CentralNic Group plc
35-39 Moorgate
London, England EC2R 6AR
GB

Phone: +44 20 33 88 0600
Email: gavin.brown@centralnic.com
URI: <http://www.centralnic.com>

Jothan Frakes

Email: jothan@jothan.com
URI: <http://jothan.com>

