regext
Internet-Draft
Intended status: Standards Track
Expires: 3 September 2026

J. Galvin
Identity Digital
M. Bauland
Knipp Medien und Kommunikation GmbH

2 March 2026

Same Entity Set Support for EPP
draft-galvin-regext-epp-variants-05

Abstract

This document defines an EPP extension allowing clients to learn about and manipulate a set of objects in a shared central repository that are necessarily tied to the same entity (typically domain objects whose names are equivalent in a registry-defined way and are tied to a single registrant). The extension supports multiple registries with a shared definition of equivalence using a shared central repository.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Registration Protocols Extensions Working Group mailing list (regext@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/regext/.

Source for this draft and an issue tracker can be found at https://github.com/arnt/regext-epp-variants.

Status of This Memo

Table of Contents

1. Introduction

EPP is defined in [RFC5730]. EPP commands were developed to operate on a single object at a time. This document defines an EPP extension allowing clients to learn about and manipulate a set of objects that have a registry-defined characteristic that makes them equivalent members of that set. As equivalent objects they MUST be tied to the same entity, which this document defines as the "Same Entity Principle". When the Same Entity Principle (SEP) is active, the <create>, <delete>, <transfer>, and <update> transforms operate on all elements of a "Same Entity Set".

Similar to EPP, the original motivation for this protocol was to provide a standard Internet domain name registration protocol for use between domain name registrars and domain name registries. This protocol provides a means of interaction between a registrar’s applications and registry applications. It is expected that this protocol will have additional uses beyond domain name registration.

With this extension, registering (creating) a domain creates a Same Entity Set and the first domain registered in the set becomes the set’s Primary Domain for that registry. (Domains are expected to be the most common kind of objects involved, and thus the word "domain" is generally used throughout this document.) A registry specifies a policy that is shared with registrars that defines the characteristic(s) that make the members of the set equivalent and the options that are relevant to the members of the set. This policy and the method by which it is shared is outside the scope of this specification.

This document exemplifies the use of three types of options that may be specified in the registry policy. These types are neither required nor limiting. They are described to exemplify how a Same Entity Set may be used. What is REQUIRED is that the Primary Domain determines how each option is applied to the remaining members of the set and this(ese) action(s) are agreed between the registry and the registrar.

An option is Prescribed if its value is determined immediately upon creation of the Primary Domain. For example, whether or not another domain name is a member of the set, i.e., is considered equivalent to the Primary Domain, is ordinarily expected to be known when the Primary Domain is created. As another example, when IDNs and their variants are eligible to be registered, the Label Generation Rules for determining the variants may prescribe if a label may be registered at all, i.e., the disposition value of the label, which can be either allocatable or blocked.

An option is Settable if its value may be managed by the registrar. For example, single domains ordinarily may have various attributes set or unset according to the use of the <update> transform, such as the nameserver for a domain. The same is true for members of the Same Entity Set.

An option is Linked if the setting or unsetting of the option for any member of the Same Entity Set applies to all members of the set. For example, Registry Policy can prescribe if the setting or unsetting of an attribute such as the nameserver applies to all members of the Same Entity Set or just the member being acted upon.

After the creation of the Primary Domain, subsequent domains in the same set can only be registered in the central repository by the same registrar. The registrar is responsible for ensuring that each domain in the set is assigned to the same registrant. Each domain in a same entity set may be the target of any EPP command, with the following restrictions.

* A <transfer> of any domain in any Same Entity Set always acts on the entire set. This is required to ensure that the set is always managed via the same entity, i.e., the same registrant at the same registrar. If multiple registries share a common policy using a shared central repository, then the transfer always acts on all sets in all registries.

* The <delete> of a Primary Domain in any same entity set always acts on the entire set in the relevant registry. This is required to support the option where the Primary Domain governs options as defined by the registry.

Note that a domain (or other object) is a member of a set by virtue of a rule, not by virtue of having been created. Creating a Primary Domain creates a set. The other members of the set thereby begin to exist in a conceptual sense, as the set may be extremely large (a set containing 10ツケ竅オ domains has been described in a realistic use case). Therefore, this extension offers no way to enumerate the set's members. The EPP client may list the allocated members of the set, but the EPP server never lists the complete set. This leads to the third restriction:

* A <create> of a later member of a Same Entity Set is not possible; <create> creates a same entity set with at least one allocated member and all other members MUST exist at least conceptually.

* Since all members of a Same Entity Set exist, at least conceptually, upon creation of a Primary Domain, the <update> transform is used to allocate or de-allocate an object in a set, thus making it exist in the central repository.

This extension is backwards compatible with registrars that do not support same entity sets. Registrars that attempt to act on a member of a set inappropriately will receive a compatible error response with which they can continue to function. The compatible error response may not provide sufficient detail to fully understand the rejection but will be sufficient to ensure continuation of normal operations.

The remainder of this document describes the specific details.

TODO: discussion of reference to EPP Extensibility and Extension Analysis https://docs.google.com/document/d/1WR00oB43XZCDqD0zvRvRajuWAq_9wQ3c0RrFKlGC3So/edit?tab=t.0

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terms

Allocated Member: A domain that has been created in the registry, and which is related to an existing Primary Domain according to a registry defined policy.

Allocatable Member: A domain that has not been allocated and exists at least conceptually in a Same Entity Set because it is related to the aforementioned set’s Primary Domain according to a registry defined policy.

Activated Member: An Allocated Member domain that is in the DNS. For objects other than domains Activated MAY NOT be relevant.

Blocked domain: A domain that cannot be allocated due to its disposition value option in relation to the Primary Domain name. This is common when a domain is an IDN and the members of a set are created according to Label Generation Rules.

Exempted domain: A preexisting domain that exists as a stand-alone domain prior to the introduction of support for this extension and would be part of a set if it were created or allocated now. Exempted domains may exist with any registrant at any registrar. The exemption ends in one of two ways.

* When there is at most 1 allocated domain remaining in a given Same Entity Set, at which time the registry MUST make the single allocated domain into a Primary Domain of that set.

* When the registrar asserts knowledge of the Same Entity Set and brings all domains in the Same Entity Set together. See the <update> command for details.

Primary Domain: The chronologically first domain allocated in a Same Entity Set. While the member relationship in a Same Entity Set is symmetric, the option values of its members are not. For example, when an IDN and its equivalent variants are members of a Same Entity Set, the members other than the Primary Domain can have a disposition value of either blocked or allocatable. The Primary Domain name therefore partitions the members of the Same Entity Set into allocatable members and blocked members. In the case of a Same Entity Set of registries, there can be a Same Entity Set with a distinct Primary Domain per Registry, and thus members may have different disposition values in different sets.

Same Entity Principle: A requirement that all domains within a set either belong to the same registrar or are withheld for that registrar. No other registrar is allowed to allocate any domain within the same set.

Same Entity Set: An implicit set of domains defined by a registry policy. The relationship between the members of the set is symmetric. Hence, an arbitrary member of a same entity set defines the whole set. The set is not expressed explicitly in EPP, because it can be impractically large.

Status Value: While a same entity set relationship is symmetric, a set member has its own option values that are not necessarily symmetric. The Status Value is one option that a member may have that can be "allocatable" (i.e., available for the same entity) or "blocked" (i.e., not available for anybody).

4. Architectural Principles

There are three architectural principles REQUIRED to be true at all times when this extension is in use. There MUST NOT be any exceptions at any time.

4.1. Backwards Compatibility

Support for Same Entity Sets is optional and therefore it is REQUIRED that a registry supporting Same Entity Sets MUST be backwards compatible with a registrar that does not support Same Entity Sets. Backwards compatibility is REQUIRED to mean that a registrar will receive a response that is fail-safe including when the registrar may not be able to fully understand the reason for the rejection.

A registry that does not support same entity sets will behave according to the sstandard when interacting with a registrar that supports same entity sets.

4.2. Same Entity Management

When a registry supports this extensions, all domains created are eligible to be in a same entity set MUST be managed by the same entity, including if the set has only one member, i.e., the domain being created. This has three requirements.

1. Registrars MUST ensure that domains in a same entity set are managed by the same registrant.

2. Registries MUST ensure that domains in a same entity set are managed by the same registrar.

3. Registries that are a member of a same entity set MUST be managed such that they share a central repository, which ordinarily is expected to mean they share a single service provider.

4.3. Same Entity Set Management

Most EPP commands may be executed independently on any member of the same entity set. However, commands that change the membership or an option value of one or more members in a same entity set, or change the Same Entity Management requirements, MUST operate on the members of the set as a set.

As explained in detail in later sections, there are currently two commands with explicit requirements: <transfer> and <delete>.

5. Technical Principles

The following technical principles have guided the development of this extension and established operational requirements.

* The members of a same entity set are defined by registry policy and that policy must be agreed by both the registry and the registrar. The establishment of this policy and the method by which the parties agree is outside the scope of this specification. Multiple registries may share a policy and a shared central repository, and thus may themselves be members of a Same Entity Set.

  - The first iteration of this work focused on IDN variants, which have the advantage that there is a relatively formal process for defining the eligible members of a set. However, some Latin characters with diacritic marks are not considered variants of Latin characters without diacritic marks and yet there are circumstances when it is desirable for them to be considerated equivalent. As a result this extension presumes the existence of a set and sets outside its scope the actual definition of the equivalence of the members of the set.

* The registry policy MUST define the options of a Same Entity Set, which MUST include at least the following properties.

  - If the same entity set exists in a registry that itself is a member of a same entity set, then all the same entity sets in any registry in the registry’s same entity set MUST be consistent to each other, i.e., membership in a same entity set is consistent.

    This principle derives directly from the Same Entity Principle.

    In the case of IDNs, the LGR tables may be different in each registry but the tables MUST be harmonized to ensure symmetry.

  - The first domain created in a same entity set is designated the Primary Domain.

    If the registry of the same entity set is itself a member of a same entity set, the Primary Domain in a same entity set MAY be different in each registry.

  - The Primary Domain has at least two REQUIRED functions. First, it defines the members of the Same Entity Set. Second, it defines the options of the members of the set.

    In the case of IDN variants, one of those options is the disposition value of the label. If a registry is itself a member of a same entity set, the Primary Domain MAY indicate different values for an option of a domain in the same entity set of different registries.

* EPP today implicitly defines two status values for any domain: registered and available. This same entity set extension adds the following status values.

  The Allocated status means that the member of the set is active in the registry. It may or may not be delegated in the DNS. It may or may not be registered in the registry.

  The Allocatable status means that the member of the set is available to be allocated by the same entity.

  The Blocked status means that the member of the set is not available to be allocated by anyone.

* The creation of a Primary Domain establishes the implicit existence of all members of the Same Entity Set.

  If the registry of the Same Entity Set is itself a member of a Same Entity Set, the Same Entity Set is implicitly created in all registries in the Same Entity Set of the registry. This ensures that only the same registrar is permitted to register or allocate any member of the set in any registry.

6. EPP Commands

In this section, the behavior of each EPP command when Same Entity Sets are supported is specified.

6.1. EPP <check> command

The <check> command always acts on the target domain in the command. There is no change on the client side when using the <check> command.

When the server receives a <check> command from a same entity agnostic client and the target domain is or could be a member of a same entity set, if that same entity set has at least one Allocated or Exempted member, the server’s response:

* MUST NOT include the <extension> element.

* MUST indicate ’available = "false"’.

* MAY indicate a reason of "Unavailable (except as member of a same entity set)".

When the server receive a <check> command from a same entity aware client and the target domain is or could be a member of a same entity set, if that same entity set has at least one Allocated or Exempted member, the server’s response:

* MUST contain an <extension&gt, which MUST contain a child <var:chkData> element.

* The <fee:chkData> element MUST contain a <var:cd> element for each object referenced in the client <check> command.

* Each <var:cd> (check data) element MUST contain the following child elements:

  - A <var:objID> element, which MUST match an element referenced in the client <check> command.

  - A <var:primary> element matching the Primary Domain for the same entity set if it is known. This is not possible for sets with exempted domains as no (unique) Primary Domain exists, in which case the <var:primary> element MUST NOT be included.

* A <var:status> element, which explains in more detail the availability status of the queried domain.

Example <check> response:

```
S: <?xml version="1.0" encoding="utf-8" standalone="no"?&gt;
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0"&gt;
S:   <response&gt;
S:     <result code="1000"&gt;
S:       <msg&gt;Command completed successfully</msg&gt;
S:     </result&gt;
S:     <resData&gt;
S:       <domain:chkData
S:         xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"&gt;
S:         <domain:cd&gt;
S:           <domain:name avail="1"&gt;examplev1.com</domain:name&gt;
S:         </domain:cd&gt;
S:         <domain:cd&gt;
S:           <domain:name avail="0"&gt;examplev1.net</domain:name&gt;
S:         </domain:cd&gt;
S:         <domain:cd&gt;
S:           <domain:name avail="0"&gt;examplev1.tel</domain:name&gt;
S:         </domain:cd&gt;
S:         <domain:cd&gt;
S:           <domain:name avail="0"&gt;examplev1.swiss</domain:name&gt;
S:         </domain:cd&gt;
```

```
S:       </domain:chkData&gt;
S:     </resData&gt;
S:     <extension&gt;
S:       <var:chkData
S:           xmlns:var="urn:ietf:params:xml:ns:epp:variants-1.0"&gt;
S:         <var:cd avail="1"&gt;
S:           <var:objID&gt;examplev1.com</var:objID&gt;
S:           <var:primary&gt;example.com</var:primary&gt;
S:           <var:status&gt;AllocatableMember</var:status&gt;
S:         </var:cd&gt;
S:         <var:cd avail="0"&gt;
S:           <var:objID&gt;examplev1.net</var:objID&gt;
S:           <var:primary&gt;example.net</var:primary&gt;
S:           <var:status&gt;NotSameEntity</var:status&gt;
S:         </var:cd&gt;
S:         <var:cd avail="0"&gt;
S:           <var:objID&gt;examplev1.tel</var:objID&gt;
S:           <var:status&gt;Exempted</var:status&gt;
S:         </var:cd&gt;
S:         <var:cd avail="0"&gt;
S:           <var:objID&gt;examplev1.swiss</var:objID&gt;
S:           <var:primary&gt;example.swiss</var:primary&gt;
S:           <var:status&gt;PendingTransfer</var:status&gt;
S:         </var:cd&gt;
S:       </var:chkData&gt;
S:     </extension&gt;
S:     <trID&gt;
S:       <clTRID&gt;ABC-12345</clTRID&gt;
S:       <svTRID&gt;54322-XYZ</svTRID&gt;
S:     </trID&gt;
S:   </response&gt;
S: </epp&gt;
```

The EPP <check> command may return six new results:

* AllocatableMember: A member of the same entity set is already active. Provisioning of this domain must be to the same registrant via the same registrar.

* NotSameEntity: The domain cannot be provisioned because it is a member of a Same Entity Set, and the set belongs to a different client

* Blocked: The domain cannot be provisioned because its disposition value is blocked.

* Exempted: The domain cannot be provisioned because it should be a member of a same entity set but the the set contains Exempted members.

* PendingTransfer: The domain cannot be provisioned because it is a member of a same entity set that is currently being transferred to a different registrar.

* Custom: Additional custom value that may be used for server peculiarities.

6.2. EPP <info> command

The <info> command always acts on the target domain in the command. There is no change on the client side when using the <info> command.

The main part of the response MUST contain the actual data of the target domain name (contacts, hosts, status values, etc.).

When the server receives an <info> command from a same entity agnostic client the response MUST contain the actual data of the target domain, independent of whether it is a member of a same entity set. In addition, if the same entity agnostic registrar is inquiring about a domain with a status of Allocatable, the response SHOULD be the same as if the client were inquiring about a reserved name.

When the server receives an <info> command from a same entity aware client and the target domain is or could be a member of a same entity set, if that same entity set has at least one Allocated or Exempted member, the server's response MUST contain an <extension> element with the following child elements:

* A <var:primary> element matching the Primary Domain for the same entity set of the target domain, which MAY match the target domain.

* A list of all the Allocated and Exempted members of the same entity set.

If the registry of the target domain is itself a member of a same entity set and the target domain is or could be a member of a same entity set in any registry in that registry's same entity set, if any one of those target domain same entity sets has at least one Allocated or Exempted member, the server's response MUST contain an <extension> element with the following child elements:

* A <var:primary> element matching the Primary Domain for the same entity set of the target domain, which MAY match the target domain.

* A list of all the same entity sets of the target domain with Allocated or Exempted members such that each same entity set list has its Primary Domain listed first.

Example <info> response when querying a primary domain name:
~~~~~~~~~~~ S: <?xml version="1.0" encoding="UTF-8"?> S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0"> S: <response> S: <result code="1000"> S: <msg lang="en-US">Command completed successfully</msg> S: </result> S: <resData> S: <infData xmlns="urn:ietf:params:xml:ns:domain-1.0"> S: <name>example.com</name> S: <roid>D123456789</roid> S: <status s="active"/> S: <registrant>abc123</registrant> S: <contact type="tech">ghi789</contact> S: <ns> S: <hostObj>ns1.example.net</hostObj> S: <hostObj>ns2.example.net</hostObj> S: </ns> S: <clID>registrar</clID> S: <crID>registrar</crID> S: <crDate>2010-09-08T07:06:05.0Z</crDate> S: <exDate>2012-09-08T23:59:59.0Z</exDate> S: <authInfo> S: <pw>secret</pw> S: </authInfo> S: </infData> S: </resData> S: <extension> S: <var:infData xmlns:var="urn:ietf:params:xml:ns:epp:variants-1.0"> S: <var:primary> S: <var:name>example.com</var:name> S: <var:name>example.comv1</var:name> S: <var:name>example.comv2</var:name> S: </var:primary> S: <var:related> S: <var:name>examplev1.com</var:name> S: <var:name>examplev2.com</var:name> S: </var:related> S: </var:infData> S: </extension> S: <trID> S: <svTRID>ZYX-99958</svTRID> S: </trID> S: </response> S: </epp> ~~~~~~~~~~~

6.3. EPP <transfer> command

The <transfer> command always acts on the target domain in the command. The use of the <transfer> command is extended if both the server and the client support same entity sets.

When the server receives a <transfer> command from a same entity agnostic client and the target domain is or could be a member of a same entity set, if that same entity set has exactly one Allocated or Exempted member the transfer request is acted upon according to the standard. If the target domain has a status of Exempted, it retains that status.
~~~~~~~~~~~

When the server receives a <transfer> command from a same entity agnostic client and the target domain is or could be a member of a same entity set, if that same entity set has more than one Allocated or Exempted member the transfer request MUST be denied using 2305 "Object status prohibits operation".

When the server receives a <transfer> command from a same entity aware client and the target domain is or could be a member of a same entity set, the request must include an <extension> element with a <var:primary> element matching the Primary Domain, including if the Primary Domain is the target domain. If the extension is not present the transfer request MUST be denied using ’2003 "Required parameter missing"’. Note that the <check> or <info> command MAY be used to identify the Primary Domain.

A valid transfer request MUST apply to all members of a same entity set. If the registry of the target domain is itself a member of a same entity set, then the transfer request MUST apply to all same entity sets in all registries of the registry’s same entity set.

The server’s response to the transfer request MUST contain an <extension> element with the following child elements;

* A <var:primary> element matching the Primary Domain for the same entity set of the target domain, which MAY match the target domain.

* A list of all the Allocated and Exempted members of the same entity set.

If the registry of the target domain is itself a member of a same entity set and the target domain is or could be a member of a same entity set in any registry in that registry’s same entity set, if any one of those target domain same entity sets has at least one Allocated or Exempted member, the server’s response MUST contain an <extension> element with the following child elements:

* A <var:primary> element matching the Primary Domain for the same entity set of the target domain, which MAY match the target domain.

* A list of all the same entity sets of the target domain with Allocated or Exempted members such that each same entity set list has its Primary Domain listed first.

TODO: It must be ensured that the poll message to the losing registrar also contains the full list of domains that will be transferred together with the primary domain.

6.4. EPP <create> command

The <create> command always acts on the target domain in the command. There is no change on the client side when using the <create> command.

The EPP <create> command's standard task is to provision a new domain. When same entity sets are supported, the <create> command MUST be used to create the Primary Domain and MUST NOT be used to provision any other member of the Primary Domain's same entity set. The task of converting an allocatable domain into an allocated domain MUST be performed using the <update> command.

When the server receives a <create> command from a same entity agnostic client and the target domain is or could be a member of a same entity set, one of the following actions MUST be completed as appropriate.

* If any member of the same entity set is currently Allocated or Exempted, the command MUST be rejected and the response MUST be the same as if the domain to be created is reserved.

* If there are no members of the same entity set either Allocated or Exempted, the <create> MUST proceed according to the standard with the server implicitly reserving all other members of the same entity set such that they MUST NOT be allocated until such time as the client is same entity aware and the client MUST indicate that the target domain is to be extended to be a Primary Domain as described in the <update> command.

When the server receives a <create> command from a same entity aware client and the target domain is or could be a member of a same entity set, one of the following actions MUST be completed as appropriate.

* If the target domain does not exist and any other member of the same entity set is Allocated or Exempted, the command MUST be rejected and indicate that it is an inappropriate use of the command.

* If the target domain does not exist and no other member of the same entity set is Allocated or Exempted, the <create> command MUST proceed according to the standard with the server implicitly noting to itself the existence of all other members of the same entity set and setting their status value as prescribed by registry policy.

The EPP <create> command may have five new errors, as described in the <check> section above.

TODO: check alignment of the new error codes

6.5. EPP <update> command

The EPP <update> command provides a transform operation that allows a client to change the options of a member of a same entity set. It is extended to cover three new tasks:

* Activating an allocatable domain in an existing same entity set.

* Deactivating an activated domain in an existing same entity set.

* Converting an Allocated or Exempted Domain into a Primary Domain and optionally converting other Exempted Domains that are eligible to be in the same entity set of the stated Primary Domain to be activated domains of the same entity set.

This extended <update> command is not valid for use by a same entity agnostic client. Any such use by a same entity agnostic client MUST be rejected and indicate it is an inappropriate use of the command.

A same entity agnostic client MUST only use the standard defined <update> command and the server MUST only respond as defined by the standard.

The rest of this section specifies behavior when same entity aware servers and same entity aware clients are interacting and describes the three new tasks.

When the target domain of the update command is any member of a same entity set, including the Primary Domain of the same entity set, the client MUST include an <extension> element that MUST include at least the <var:primary> child element indicating the Primary Domain of the corresponding same entity set. The extension MAY include additional elements as indicated below to provision a new task. If the extension is not present the command MUST be rejected and indicate that a required parameter is missing.

If the Primary Domain and the target domain match, all other elements in the extension MUST be ignored and the update command MUST be processed as a standard defined update command acting on the Primary Domain.

The rest of this section specifies behavior when the target domain and the Primary Domain indicated in the extension do not match.

If the <var:status> child element is present in the extension, one of the following actions MUST be completed as appropriate.

* In order to Activate an Allocatable domain, the target domain MUST have a status of Allocatable and the extension MUST include the <var:status> child element with a value of "allocated". The server MUST update the status of the target domain and the response MUST include the extension element with both the Primary Domain indicated and the revised status indicated.

* In order to deactivate an Allocated domain, the target domain MUST have a status of Allocated and the extension MUST include the <var:status> child element with a value of "allocatable". The server MUST update the status of the target domain and the response MUST include the extension element with both the Primary Domain indicated and the revised status indicated.

* In all other cases, if the status element is present the command MUST be rejected and indicate an invalid parameter is present.

If the <var:status> child element is not present in the extension, then all elements other than the Primary Domain indication MUST be ignored and the update command MUST be processed according to the standard acting on the target domain.

Note that depending on registry policy, the target domain may share attributes with the Primary Domain, e.g., nameservers. A registry policy MAY specify rules or guidelines for the set of elements required or permitted for a domain according to the Primary Domain.

The EPP domain mapping from [RFC3915] describes the elements that have to be specified within an <update> command. The requirement to provide at least one <domain:add>, <domain:rem>, or <domain:chg> element is updated by this extension such that at least one empty <domain:add>, <domain:rem>, or <domain:chg> element MUST be present if this extension is specified within an <update> command. This requirement is updated to disallow the possibility of modifying a domain object as part of the deactivation.

If a client wishes to convert an Exempted domain into the Primary Domain of a same entity set, the update command from the client MUST be provided as follows.

* The target domain of the update command and the Primary Domain in the extension MUST match.

* The target domain MUST have the status of Exempted.

* If there exists multiple Exempted domains that would ordinarily be members of the same entity set, they MUST all have the same Registrar of Record and it MUST match the update requesting

registrar, and the extension MUST include a list of all Exempted domains, including the Primary Domain, that MUST match the list maintained by the registry.

If the update command is valid as indicated above, the server MUST change the status of the indicated domains from Exempted to Allocated, and MUST indicate the Primary Domain. The response MUST include an extension indicating the Primary Domain and the list of domains whose status changed from Exempted to Allocated.

6.6. EPP <delete> command

When the server receives a <delete> command from a same entity agnostic client the server MUST respond as defined by the standard.

The rest of this section specifies behavior when same entity aware servers and same entity aware clients are interacting.

The <delete> command is extended to REQUIRE the deletion of all members of a same entity set if the Primary Domain is deleted.

A same entity aware client MUST NOT use the <delete> command to delete any member of a same entity set except the Primary Domain. Any other use MUST be rejected and indicate it is an inappropriate use of the command. Note that the <update> command is used for the purpose of deactivating other members of a same entity set.

When the server receives a <delete> command from a same entity aware client, the command MUST include the <extension> element which MUST include the <var:primary> child element which MUST match the target domain name.

The delete command is extended such that all Allocated members of the same entity set defined by the Primary Domain MUST all be deleted at once. If it is not possible for any member of the same entity set to be deleted for any reason, the delete command MUST fail leaving all members of the same entity set intact.

If the delete command is successful, the response MUST include the extension element which MUST include both an indication of the Primary Domain and the list of all members of the same entity set that were deleted, including the Primary Domain.

6.7. EPP renew command

The renew command is not extended.

The server MAY reject renewals while a same entity set is being transferred.

6.8. EPP <transfer> query command

Same entity sets are transferred as a set and thus the result of a <transfer> query command is necessarily the same for all domains in a set. Therefore, the result of a <transfer> query command for any domain in a same entity set applies to all domains in the set.

The transfer query command is not extended.

7. Result codes

The following additional result codes are defined:

23x1: Change impossible because of a transfer in progress.

23x2: Change impossible because something is not a variant.

This error code is used when a change presupposes that two domains belong to the same variant group, but the EPP server’s implementation disagrees.

23x3: Change impossible due to invalid primary domain

This error code is used when the primary domain specified in the command is not registered, or is not registered via this registrar.

23x4: Change impossible due to unspecified primary domain

This error code is used when a command needs to specify a primary domain, and does not.

23x5: Specified domain is exempted

This error code is used when a domain specifies a primary domain, and the change is impossible because the specified domain is exempted instead.

23x6: Specified domain is allocatable, but not by you.

This result code is used when a domain is a member of a variant set, and the command did not refer to the primary domain.

8. Acknowledgements

The design of this extension is almost completely based on work done by and decisions made by the [EPDP] committee, which was reviewed by a small technical design team chaired by James Galvin. Members of this team included Dennis Tan, Rick Wilhelm, Edmon Chung, and Jennifer Chung. This text (in RFC format) was initially written by Arnt Gulbrandsen based on a conference presentation by James Galvin.

YOU YES YOU (<- insert name) have reviewed it and provided helpful comments or contributed in other ways.

9. Security Considerations

If two domains are different according to the DNS rules and identical in the eyes of the intended audience, then the audience may be confused. Confusion can always have security-related effects.

This extension expresses the relationships between members of a same entity set clearly, making it a little more difficult for a would-be impersonator to register a member of another registrant’s existing domain.

10. IANA Considerations

11. References

11.1. Normative References

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <https://www.rfc-editor.org/rfc/rfc5730>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

11.2. Informative References

[RFC3915] Hollenbeck, S., "Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP)", RFC 3915, DOI 10.17487/RFC3915, September 2004, <https://www.rfc-editor.org/rfc/rfc3915>.

[RFC7940] Davies, K. and A. Freytag, "Representing Label Generation Rulesets Using XML", RFC 7940, DOI 10.17487/RFC7940, August 2016, <https://www.rfc-editor.org/rfc/rfc7940>.

[EPDP] ICANN, "Phase 2 Initial Report of the EPDP on Internationalized Domain Names", 2024, <https://www.icann.org/en/public-comment/proceeding/phase-2-initial-report-of-the-epdp-on-internationalized-domain-names-11-04-2024>.

Appendix A. Open issues

Open issue: Assign numbers to the error codes, properly.

Open issue: Not clear that there are any security considerations here 寰the relationships between the domains may have some, but those exist outside EPP, EPP merely describes them. In Italian, caffe and caff竪 are variants of the same thing, it’s not clear that linking them in a protocol affects security in any way.

Open issue: Check how to insert a DS record in a variant domain.

Open issue: Can a unicode upgrade cause domains to become exempted? Yes, I think, and the terminology covers it, but as of now, it’s difficult for the EPP client to understand the situation. Extending the <info> command would help here, perhaps.

Open issue: <Delete> now cascades and deletes many domains. Should it instead turn any variant domains into exempted domains?

Authors’ Addresses

James Galvin
Identity Digital
Bellevue, Washington,
United States of America
Email: jgalvin@identity.digital

Michael Bauland
Knipp Medien und Kommunikation GmbH
Dortmund
Germany
Email: michael.bauland@knipp.de