Network Working Group
Request for Comments: 2832
Category: Informational

S. Hollenbeck
M. Srivastava
Network Solutions, Inc. Registry
May 2000

NSI Registry Registrar Protocol (RRP) Version 1.1.0

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document describes a protocol for the registration and management of second level domain names and associated name servers in both generic Top Level Domains (gTLDs) and country code Top Level Domains (ccTLDs). This protocol was developed by the Network Solutions Registry for use within the Shared Registration System and is being published "as-is" to document the protocol implementation developed by the Network Solutions, Inc. Registry.

Internet domain name registration typically involves three entities: a registrant who wishes to register a domain name, a registrar who provides services to the registrant, and a registry that provides services to the registrar while serving as the authoritative repository of all functional information required to resolve names registered in the registry's TLDs. This document describes a protocol for registry-registrar communication only. The protocol does not provide any registrant services.

This document is being discussed on the "rrp" mailing list. To join the list, send a message to <majordomo@NSIRegistry.com> with the words "subscribe rrp" in the body of the message. There is also a web site for the mailing list archives at <http://www.NSIRegistry.net/maillist/rrp>.

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 [MUSTSHOULD]. Further,

the term "implicit attribute" refers to an entity attribute whose value is derived either from another attribute or is dependent on an established RRP session.

In examples, "C:" represents lines sent by the registrar client and "S:" represents lines sent by the registry server.

The term "System" is used in this document to collectively refer to this protocol and the software and hardware that implements the protocol.

Table of Contents

1. Introduction

This document describes the specifications for the NSI Registry Registrar Protocol (RRP) version 1.1.0, a TCP-based, 7-bit US-ASCII text protocol that permits multiple registrars to provide second level Internet domain name registration services in the top level domains (TLDs) administered by a TLD registry. RRP is specified using Augmented Backus-Nauer Form (ABNF) as described in [ABNF]. Note that all ABNF string literals are case-insensitive and the examples provided in this document may use mixed case to improve readability.

RRP was developed by the Network Solutions, Inc. Registry under the auspices of the Shared Registration System program. The protocol was initially deployed in April 1999 as part of a test bed implementation of the Shared Registration System with five registrars. Additional registrars began using the protocol in July 1999. The operational experiences of both the registry and the registrars identified several "lessons learned" which have been documented here as "Known Issues".

This document provides both a description of a protocol and notice of learned operational issues that may be useful as first steps in developing a standards track domain registration services protocol. This document and the protocol it describes may be modified in the future based on continued operational experience and community reaction.

The registry stores information about registered domain names and associated name servers. A domain name’s data includes its name, name servers, registrar, registration expiration date, and status. A name server’s data includes its server name, IP addresses, and registrar. A registrar MAY perform the following registration service procedures using RRP:

- Determine if a domain name has been registered.
- Register a domain name.
- Renew the registration of a domain name.
- Cancel the registration of a domain name.
- Update the name servers of a domain name.
- Transfer a domain name from another registrar.
- Examine the status of domain names that the registrar has registered.
- Modify the status of domain names that the registrar has registered.
- Determine if a name server has been registered.
- Register a name server.
- Update the IP addresses of a name server.

- Delete a name server.
- Examine the status of name servers that the registrar has registered.

All RRP commands include features to provide idempotency. That is, the effect of each command is the same if the command is executed once or if the command is executed multiple times. This property is extremely useful in situations when a command is retried due to an error condition that results in a missed command response and a command retry is attempted. Command retries will be caught by the System and rejected with an appropriate error response code. Command parameters that do not provide idempotency will be explained fully as part of the appropriate command description.

2. Security Services

RRP provides only basic password-based registrar authentication services. Additional security services, including privacy and registrar authentication using public key cryptography, are provided through other System features.

2.1 Connection Security

Each RRP session MUST be encrypted using the Secure Socket Layer (SSL) v3.0 protocol as specified in [SSL]. SSL provides privacy services that reduce the risk of inadvertent disclosure of registrar-sensitive information, such as the registrar’s user identifier and password.

SSL supports mutual authentication of both the client and server using signed digital certificates. The Shared Registration System implemented by the NSI Registry requires digital certificates issued by a commercial certification authority for both registrar clients and public registry RRP servers. Both the registrar client and the public registry RRP server are authenticated when establishing an SSL connection. Further, a registrar MUST be authenticated when establishing an RRP connection via the RRP SESSION command by providing a registrar user identifier and password known only to the registrar and the System. Registrars may change their session password at any time using the RRP SESSION command.

The SSL protocol is not an IETF Standards Track protocol. The Transport Layer Security protocol, specified in [TLS], is a Standards Track protocol that provides SSL v3.0 compatibility features.

2.2 System Data Security

The System stores information about the registered domain names and their name servers. Only the current registrar of a registered domain name is authorized to query it, update its name servers, and cancel or renew it. Any registrar can request a transfer of a domain name and its associated name servers from another registrar to the requesting registrar. Only the current sponsoring registrar can receive and explicitly approve or reject domain transfer requests.

Only a name server's registrar can query, update, and delete it. In general, name servers must be registered through the current registrar of the name server's parent domain name, though an implementation MAY allow use of name servers registered in other TLDs without specifying IP addresses or requiring parent domain registration. Use of ccTLD name servers for a gTLD domain name is one such example.

Name servers are implicitly transferred by the System when their parent domain name is transferred. In addition, a name server cannot be deleted if it is hosting domain names.

3. Connection Model

IANA has assigned TCP port 648 for RRP use. All RRP implementations MUST provide RRP services over SSL on TCP port 648. An RRP server MUST return a banner in the following format to confirm that a connection has been established:

<registry name> RRP Server version <version><crlf>
<server build date and time><crlf>

Each line ends with carriage return and line feed characters. The server build date and time string includes the day, month, date, time (specified in hours, minutes, and seconds), the local time zone, and the four-digit year. A dot (".") in column one on a line by itself marks the end of banner text.

Example

A registrar successfully establishes a connection with the NSI Registry on TCP port 648:

S:NSI RRP Server version 1.1.0
S:Mon Oct 25 20:20:34 EDT 1999
S:.

4. Protocol Description

A typical RRP session will go through a number of states during its lifetime. Figure 1 illustrates the possible states of an RRP server.

Initially, the server waits for a client connection and authentication (PRE). All client connections MUST be authenticated.

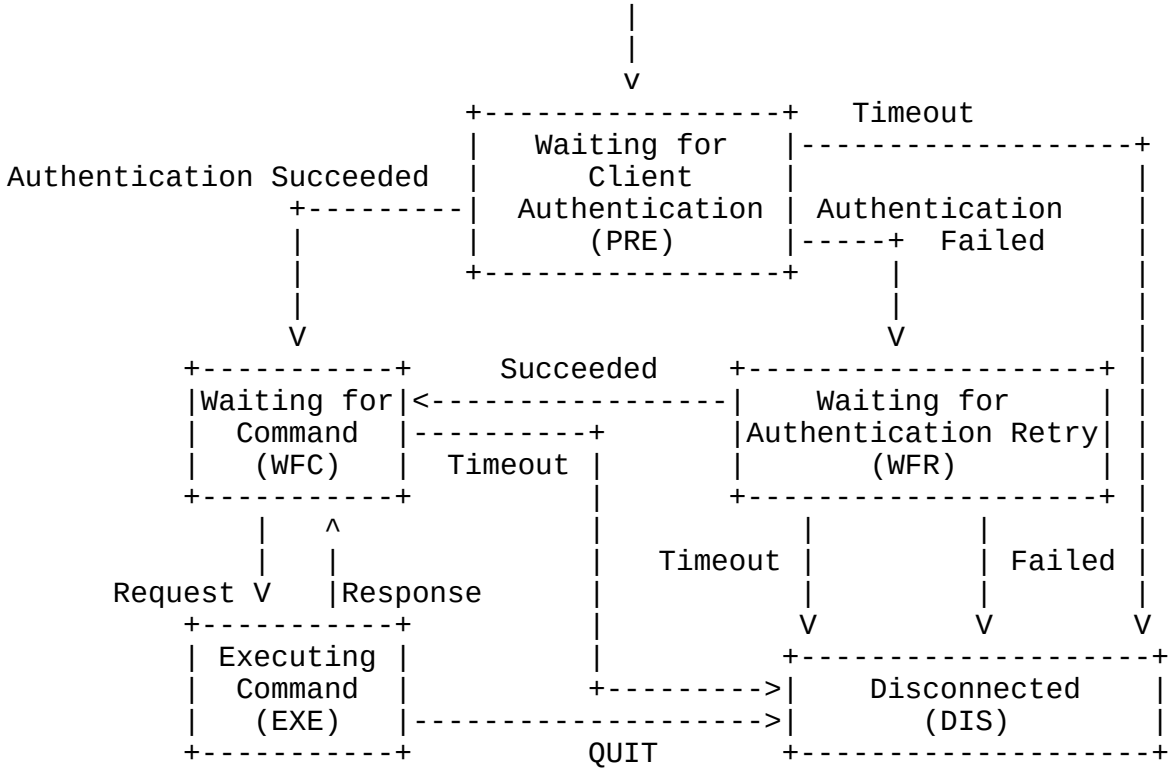

Figure 1: RRP Server Finite State Machine

If the authentication fails, the server gives the client another chance to identify itself (WFR). If the authentication fails again, the server disconnects (DIS). Otherwise, the server waits for a request from the client (WFC). Upon receiving a request, the server executes it and responds to the client with the result (EXE). The server then waits again for another request from the client (WFC). If the client sends a QUIT command, the server ends the session and disconnects (DIS). To keep its state in sync with that of the server, the client SHOULD wait for a response from the server before sending another request on the same connection. The following table summarizes these states:

PRE	Waiting for client connection and authentication
WFR	Waiting for authentication retry
WFC	Waiting for a command from an authenticated client
EXE	Executing a command
DIS	Disconnected

The WFR and WFC states MAY time out. An implementation SHOULD define inactivity timeout periods for these states based on System-specific factors, including (but not limited to) resource availability and security risk. In the absence of other factors, a default timeout period of 10 minutes SHOULD be used. The server MAY disconnect if the server is in one of these states and no message is received from the client during the timeout period.

4.1 Request Format

An RRP request nominally consists of a command name, an entity block, command options, and an end-of-command delimiter. Command options and entity blocks collectively define command parameters and their specification is order independent; examples provided in this document specify entity blocks before command options.

```
CommandName [EntityBlock] [CommandOptions] EndOfCommand
```

A command name specifies the type of an RRP request. A command is a word or abbreviation terminated by a carriage-return linefeed (crlf) sequence.

```
CommandName<crlf>
```

An entity block specifies the data in an RRP request. It consists of attribute name-value pairs specifying the entity and all of the attributes of the entity. Each attribute name-value pair starts with the attribute name, followed by a colon, the attribute value, and is finally terminated by a carriage-return linefeed sequence. Entity blocks are optional for some requests.

```
entityName:entityValue<crlf>
attributeName:attributeValue<crlf>
```

Command options specify control parameters for an RRP request. A command option starts with a dash, followed by the option name, a colon, the option value, and is finally terminated by a carriage-return linefeed sequence.

```
-commandOptionName:commandOptionValue<crlf>
```

An EndOfCommand delimiter specifies the end of an RRP request. It consists of a dot (".") in column one followed by a carriage-return linefeed sequence.

```
.<crlf>
```

4.2 Response Format

An RRP response starts with a three-digit response code, followed by a space, an ASCII text description of the response, a carriage-return linefeed sequence, and zero or more attribute name-value pair lines. An RRP response is terminated by a dot in column one followed by a carriage-return linefeed sequence.

```
ResponseCode<space>responseDescription<crlf>
[attributeName:attributeValue<crlf>]
.<crlf>
```

4.3 Protocol Commands

Implementations of RRP commands MUST provide "all or nothing" success and failure operation. Failed command execution MUST leave the System in the same state it was in before the command was attempted and failed.

All RRP commands include features to provide idempotency. Command features that are not idempotent are explained fully as needed as part of the appropriate command description.

4.3.1 ADD

This command allows a registrar to register a domain name or a name server in the System.

4.3.1.1 Registering a Domain Name

The request to register a domain name MUST contain the following data:

- The "EntityName" parameter set to value "Domain".
- Fully qualified second level domain name in the "DomainName" parameter.

The request to register a domain name MAY contain 1 or more, and a maximum of 13, fully qualified name servers hosting the domain name in multiple instances of the "NameServer" parameter. The name servers MUST have already been registered in the registry. Implementations MAY allow specification of name servers associated with domains registered in other TLDs. For example, an implementation MAY allow use of ccTLD name servers for gTLD domain name registration.

The request to register a domain name MAY contain the initial registration period in years for the domain being registered in a single instance of the "Period" parameter. The System MUST provide a default initial registration period in years if the "Period" parameter is not provided. The acceptable year values for the "Period" parameter are implementation specific.

The System will register the domain name to the registrar for the period specified by the registrar. If the registrar does not specify a registration period, a System-specified default value MUST be used for the initial registration period. If the domain name is successfully registered, the System MUST return the registration expiration date in the "registration expiration date" attribute in the response.

Authorized User: All registrars MAY use the ADD command to register domain names.

Examples

A registrar registers a domain name without specifying name servers:

C:add<crlf>
C:EntityName:Domain<crlf>
C:DomainName:example.com<crlf>
C:-Period:10<crlf>
C:.<crlf>
S:200 Command completed successfully<crlf>
S:registration expiration date:2009-09-22 10:27:00.0<crlf>
S:status:ACTIVE<crlf>
S:.<crlf>

A registrar registers a domain name using previously-registered name servers:

```
C:add<crlf>
C:EntityName:Domain<crlf>
C:DomainName:example2.com<crlf>
C:-Period:10<crlf>
C:NameServer:ns1.example.com<crlf>
C:NameServer:ns2.example.com<crlf>
C:.<crlf>
S:200 Command completed successfully<crlf>
S:registration expiration date:2000-09-22 10:27:00.0<crlf>
S:status:ACTIVE<crlf>
S:.<crlf>
```

4.3.1.2 Registering a Name Server

The request to register a name server MUST contain the following data:

- The "EntityName" parameter set to value "NameServer".
- Fully qualified server name of the name server in the "NameServer" parameter.

If the name server being registered is the child of a registered domain name, the name server registration request MUST include one or more, and a maximum of 13, name server IP addresses in multiple instances of the "IPAddress" parameter. Name servers associated with domains registered in other TLDs SHOULD NOT be specified with IP addresses to reduce the possibility of duplicating DNS NS records for the name servers in multiple zone files.

The registrar MUST register the name server in the System before using it to host domain names. Further, the name server MUST be registered through the same registrar that is the current registrar of its parent domain name. The System MAY allow any registrar to use the name server to host domain names.

Authorized User: All registrars MAY use the ADD command to register name servers.

Examples

A registrar registers a new name server in an existing domain name:

```
C:add<crlf>
C:EntityName:NameServer<crlf>
C:NameServer:ns1.example.com<crlf>
C:IPAddress:198.41.1.11<crlf>
C:.<crlf>
S:200 Command completed successfully<crlf>
S:.<crlf>
```

4.3.2 CHECK

This command allows a registrar to determine if a domain name or name server has been registered in the System.

4.3.2.1 Domain Name Check

The request to determine if a domain name is registered MUST contain the following data:

- The "EntityName" parameter set to value "Domain".
  - Fully qualified second level domain name in the "DomainName" parameter.

The System MUST provide a positive or negative response to document domain name availability at the moment the command is executed.

Authorized User: All registrars MAY use the CHECK command to determine if a domain name has been registered or not.

Examples

A registrar checks the availability of a domain name in the System:

```
C:check<crlf>
C:EntityName:Domain<crlf>
C:DomainName:example.com<crlf>
C:.<crlf>
S:211 Domain name not available<crlf>
S:.<crlf>
```

4.3.2.2 Name Server Check

The request to determine if a name server is registered MUST contain the following data:

- The "EntityName" parameter set to value "NameServer".
- Fully qualified server name in the "NameServer" parameter.

The System MUST provide a positive or negative response to document name server availability at the moment the command is executed. If the name server has been registered, the System MUST return the IP address(es) of the name server.

Authorized User: All registrars MAY use the CHECK command to determine if a name server has been registered or not.

Examples

A registrar checks the availability of a server name in the System:

```
C:check<crlf>
C:EntityName:Nameserver<crlf>
C:Nameserver:ns1.example.com<crlf>
C:.<crlf>
S:213 Name server not available<crlf>
S:ipAddress:192.10.10.10<crlf>
S:.<crlf>
```

4.3.3 DEL

This command allows a registrar to delete (cancel the registration) of a domain name or delete a name server.

4.3.3.1 Deleting a Domain Name

The request to cancel the registration of a domain name MUST contain the following data:

- The "EntityName" parameter set to value "Domain".
- Fully qualified second level domain name in the "DomainName" parameter.

A request to delete a domain name SHOULD cause the deletion of all name servers that are children of the domain name being deleted. The name servers SHOULD be deleted if they are not actively hosting other domains. A domain MUST not be deleted if it has child name servers hosting other domains.

Authorized User: The current registrar of a domain name MAY use the DEL command to delete a domain name from the System.

Examples

A registrar deletes a domain name, implicitly deleting all name servers registered in the domain:

```
C:del<crlf>
C:EntityName:Domain<crlf>
C:DomainName:example.com<crlf>
C:.<crlf>
S:200 Command completed successfully<crlf>
S:.<crlf>
```

4.3.3.2 Deleting a Name Server

The request to delete a name server MUST contain the following data:

- The "EntityName" parameter set to value "NameServer".
- Fully qualified name of the name server in the "NameServer" parameter.

A name server MUST not be deleted if it is hosting domains. Deleting such domains or name servers is prohibited because their deletion WILL result in orphaning the hosted domains.

Authorized User: The current registrar of a name server MAY use the DEL command to delete a name server from the System.

Examples

A registrar deletes a name server that is not hosting domains:

```
C:del<crlf>
C:EntityName:NameServer<crlf>
C:NameServer:ns1.registrarA.com<crlf>
C:.<crlf>
S:200 Command completed successfully<crlf>
S:.<crlf>
```

A registrar tries to delete a name server that is hosting domains:

```
C:del<crlf>
C:EntityName:NameServer<crlf>
C:NameServer:ns1.registrarA.com<crlf>
C:.<crlf>
S:532 Domain names linked with name server<crlf>
S:.<crlf>
```

4.3.4 DESCRIBE

This command allows a registrar to obtain general information about an RRP implementation. The command MAY contain the following parameters:

- The "Target" parameter set to value "Protocol".

The implementation MUST return the protocol version number whether or not the request contains the "Target" parameter.

Authorized User: All registrars MAY use the DESCRIBE command.

Examples

A registrar obtains general information about an RRP implementation:

```
C:describe<crlf>
C:-Target:Protocol<crlf>
C:.<crlf>
S:200 Command completed successfully<crlf>
S:Protocol:RRP 1.1.0<crlf>
S:.<crlf>
```

4.3.5 MOD

This command allows a registrar to update a registered domain name or a name server. The command allows the following operations on an attribute value for both single-valued and multi-valued attributes:

- Add an attribute value. The value to be added MUST be unique among the values of the attribute. For a single-valued attribute, it replaces the current value.
- Remove an attribute value. The value to be removed MUST exist. Further, an attribute value cannot be removed if it is the only value of a required attribute.

Attribute values to be removed are identified by tagging with an "=" suffix.

4.3.5.1 Domain Modification

The request to modify a registered domain name MUST contain the following data:

- The "EntityName" parameter set to value "Domain".
- Fully qualified second level domain name in the "DomainName" parameter.

The registrar can perform the following update operations on the domain name:

- Update the name servers of the domain name by setting one or more instances of the "NameServer" parameter.
- Update the status of the domain name by setting one or more instances of the "Status" parameter. Valid values for the "Status" parameter are defined in Section 6.

Authorized User: The current registrar of a domain name MAY use the MOD command to modify the attributes of a domain name.

Examples

A registrar removes one name server (ns1) from a domain and adds a new name server (ns3) to the same domain:

```
C:mod<crlf>
C:EntityName:Domain<crlf>
C:DomainName:example.com<crlf>
C:NameServer:ns3.registrarA.com<crlf>
C:NameServer:ns1.registrarA.com=<crlf>
C:.<crlf>
S:200 Command completed successfully<crlf>
S:.<crlf>
```

4.3.5.2 Name Server Modification

The request to update a name server MUST contain the following data:

- The "EntityName" parameter set to value "NameServer".
- Fully qualified server name of the name server in the "NameServer" parameter.

The registrar can perform the following update operations on the name server:

- Update the "NameServer" attribute of the name server. This allows a registrar to change the name of a name server while preserving all existing associations.
- Update the IP addresses of the name server by setting one or more instances of the "IPAddress" parameter.

Authorized User: The current registrar of a name server MAY use the MOD command to modify the attributes of a domain name.

Examples

A registrar changes the name and IP address of a name server:

C:mod<crlf>
C:EntityName:NameServer<crlf>
C:NameServer:ns1.registrarA.com<crlf>
C:NewNameServer:ns2.registrarA.com<crlf>
C:IPAddress:198.42.1.11<crlf>
C:IPAddress:198.41.1.11=<crlf>
C:.<crlf>
S:200 Command completed successfully<crlf>
S:.<crlf>

4.3.6 QUIT

This command allows a registrar to close an RRP connection. A response MUST be sent before closing the connection.

Authorized User: All registrars MAY use the QUIT command.

Examples

A registrar ends an RRP session and closes an existing connection:

C:quit<crlf>
C:.<crlf>
S:220 Command completed successfully. Server closing connection<crlf>
S:.<crlf>

4.3.7 RENEW

This command allows a registrar to renew a domain name in the System. The request to renew a domain name MUST contain the following data:

- The "EntityName" parameter set to value "Domain".
- Fully qualified second level domain name in the "DomainName" parameter.

The request to renew a domain name MAY contain the renewal period in years for the domain being renewed in a single instance of a "Period" parameter and a single instance of a "CurrentExpirationYear" parameter. These parameters MUST appear together if either is specified, though the order in which the parameters appear is insignificant. The "Period" parameter identifies the number of years to be added to the registration. The "CurrentExpirationYear" parameter identifies the current expiration year, and is required to ensure that repeated attempts to retry this command do not result in multiple successful renewals. The System MUST provide a default number of renewal years if the "Period" and "CurrentExpirationYear" parameters are not provided. Repeated use of this command without the "Period" and "CurrentExpirationYear" parameters may result in repeated successful renewals since idempotency is not provided when these parameters are not used. The acceptable year values for the "Period" parameter are implementation specific subject to syntax restrictions.

The System renews the domain name for a period specified by the registrar. If the domain name renewal is completed successfully, the System MUST return the new registration expiration date in the "RegistrationExpirationDate" attribute in the response.

Authorized User: The current registrar of a domain name MAY use the RENEW command.

Examples

A registrar renews a domain name using a specified renewal period:

```
C:renew<crlf>
C:EntityName:Domain<crlf>
C:DomainName:example.com<crlf>
C:-Period:9<crlf>
C:-CurrentExpirationYear:2001<crlf>
C:.<crlf>
S:200 Command completed successfully<crlf>
S:registration expiration date:2010-09-22 10:27:00.0<crlf>
S:.<crlf>
```

4.3.8 SESSION

This command allows a registrar to establish an RRP session. A registrar can also use this command to change their password. The request to establish an RRP connection MUST contain the following command parameters:

- The "Id" parameter set to the registrar's System user ID.
- The "Password" parameter set to the registrar's current System password.

The request to establish an RRP session MAY contain a new password for the registrar in a single instance of the "NewPassword" parameter.

The registrar MUST send this command to the System before any other command. If the command fails due to invalid information (such as an invalid registrar ID or password), the registrar can resend this request with corrected information. If the command fails a second time, the System SHOULD close the connection.

Authorized User: All registrars MAY use the SESSION command.

Examples

A registrar establishes an RRP session:

```
C:session<crlf>
C:-Id:registrarA<crlf>
C:-Password:i-am-registrarA<crlf>
C:.<crlf>
S:200 Command completed successfully<crlf>
S:.<crlf>
```

4.3.9 STATUS

This command allows a registrar to determine the current status of a domain name or name server.

4.3.9.1 Domain Status

The request to query a domain name MUST contain the following data:

- The "EntityName" parameter set to value "Domain".
- Fully qualified second level domain name in the "DomainName" parameter.

The response from the System MAY contain the following data:

- Fully qualified server names of name servers hosting the domain name in multiple instances of the "nameserver" attribute.
- Registration expiration date in the "registration expiration date" attribute.
- ID of the current registrar of the domain name in the "registrar" attribute.
- Date the domain name was transferred by the current registrar in the "registrar transfer date" attribute.
- Current statuses of the domain name in multiple instances of the "status" attribute.
- Date the domain name was originally registered in the "created date" attribute.
- ID of the registrar that originally registered the domain name in the "created by" attribute.
- Date the domain name was last updated in the "updated date" attribute.
- ID of the entity (either a registrar or the registry) that last updated the domain name in the "updated by" attribute.

Authorized User: The current registrar of a domain name MAY use the STATUS command to view current domain name attributes.

Examples

The current registrar of a domain name queries the domain name:

C:status<crlf>
C:EntityName:Domain<crlf>
C:DomainName:example.com<crlf>
C:.<crlf>
S:200 Command completed successfully<crlf>
S:nameserver:ns2.registrarA.com<crlf>
S:nameserver:ns3.registrarA.com<crlf>
S:registration expiration date:2010-09-22 10:27:00.0<crlf>
S:registrar:registrarA<crlf>
S:registrar transfer date:1999-09-22 10:27:00.0<crlf>
S:status:ACTIVE<crlf>
S:created date:1998-09-22 10:27:00.0<crlf>
S:created by:registrarA<crlf>
S:updated date:2002-09-22 10:27:00.0<crlf>
S:updated by:registrarA<crlf>
S:.<crlf>

A registrar queries a domain name currently registered by another registrar:

C:status<crlf>
C:EntityName:Domain<crlf>
C:DomainName:example.com<crlf>
C:.<crlf>
S:531 Authorization failed<crlf>
S:.<crlf>

4.3.9.2 Name Server Status

The request to query a name server MUST contain the following data:

- The "EntityName" parameter set to value "NameServer".
- Fully qualified name of the name server in the "NameServer" parameter.

The response from the System MAY contain the following data:

- Fully qualified name of the name server in the "nameserver" attribute.
- IP addresses of the name server in multiple instances of the "ipaddress" attribute.
- ID of the current registrar of the name server in the "registrar" attribute.
- Date the name server was transferred by the current registrar in the "registrar transfer date" attribute.
- Date the name server was registered in the "created date" attribute.
- ID of the entity that registered the name server in the "created by" attribute.
- Date the name server was last updated in the "updated date" attribute.
- ID of the entity that last updated the name server in the "updated by" attribute.

Authorized User: The current registrar of a name server MAY use the STATUS command to view current domain name attributes.

Examples

The current registrar of a name server queries the name server:

C:status<crlf>
C:EntityName:NameServer<crlf>
C:NameServer:ns1.registrarA.com<crlf>
C:.<crlf>

```
S:200 Command completed successfully<crlf>
S:ipaddress:198.42.1.11<crlf>
S:registrar:registrarA<crlf>
S:registrar transfer date:1999-09-22 10:27:00.0<crlf>
S:CreatedDate:1998-09-22 10:27:00.0<crlf>
S:CreatedBy:registrarA<crlf>
S:UpdatedDate:2002-09-22 10:27:00.0<crlf>
S:UpdatedBy:registrarA<crlf>
S:.<crlf>
```

A registrar queries a name server that was registered by another registrar:

```
C:status<crlf>
C:EntityName:NameServer<crlf>
C:NameServer:ns1.registrarA.com<crlf>
C:.<crlf>
S:531 Authorization failed<crlf>
S:.<crlf>
```

4.3.10 TRANSFER

This command allows a registrar to request transfer of domain name sponsorship from a second registrar and to approve or reject transfer requests initiated by other registrars. The request to transfer a domain name MUST contain the following data:

- The "EntityName" parameter set to value "Domain".
- Fully qualified second level domain name in the "DomainName" parameter.

The identity of the requesting registrar is derived from the current active session. The identity of the current sponsoring registrar (the registrar who must approve or reject the transfer request) is known by the registry and does not need to be known by the requesting registrar in advance of issuing the transfer request.

The System MUST notify the potential losing registrar when a domain transfer request has been received using an out-of-band transport mechanism such as electronic mail and/or transaction reporting. The losing registrar SHOULD then explicitly approve or reject the transfer. A request to approve or reject a transfer request MUST contain a single instance of the "Approve" parameter with a value of "Yes" to approve the transfer or a value of "No" to reject the transfer. A server implementation MAY provide a default approval or rejection action to be taken if the losing registrar does not explicitly approve or reject the transfer request within a fixed amount of time. The criteria used by registrars to approve or deny

requested transfers are typically based on business policies that are beyond the scope of this document.

Approval of a transfer by the current sponsoring registrar results in a change of sponsorship to the original requesting registrar. Approval attempts by any other registrar MUST result in explicit failure of the attempted approval. Rejection of the transfer by the current sponsoring registrar results in an end to the transfer request with no change in sponsorship. Rejection attempts by any other registrar MUST result in explicit failure of the attempted rejection.

Name servers MUST be implicitly transferred when their parent domain name is transferred.

Authorized User: All registrars MAY use the TRANSFER command to request transfer of registration service authority to the requesting registrar. Only the current sponsoring registrar of a domain name may explicitly approve or reject a requested transfer. The registry MAY implicitly approve or reject requested transfers after a fixed amount of time.

Examples

A registrar requests transfer of a domain name from another registrar:

C:transfer<crlf>
C:EntityName:Domain<crlf>
C:DomainName:example.com<crlf>
C:.<crlf>
S:200 Command completed successfully<crlf>
S:.<crlf>

The original registrar approves the transfer request:

C:transfer<crlf>
C:-Approve:Yes<crlf>
C:EntityName:Domain<crlf>
C:DomainName:example.com<crlf>
C:.<crlf>
S:200 Command completed successfully<crlf>
S:.<crlf>

5. Response Codes

RRP commands may return a variety of response codes to signify normal completion or error conditions. This section documents all of the defined RRP response codes.

5.1 Response Code Summary

200 Command completed successfully
This is the normal response for successful completion of most RRP commands.

210 Domain name available
This is the normal response for successful completion of an RRP CHECK command for a domain name that is not currently registered.

211 Domain name not available
This is the normal response for successful completion of an RRP CHECK command for a domain name that is currently registered.

212 Name server available
This is the normal response for successful completion of an RRP CHECK command for a name server that is not currently registered.

213 Name server not available
This is the normal response for successful completion of an RRP CHECK command for a name server that is currently registered.

220 Command completed successfully. Server closing connection This is the normal response for successful completion of an RRP QUIT command. It may also be returned by other RRP commands if a transient situation is noted that requires closing the connection after successfully completing the RRP command.

420 Command failed due to server error. Server closing connection A transient server error has caused RRP command failure and session termination. A new session must be established before continued processing can be attempted.

421 Command failed due to server error. Client should try again A transient server error has caused RRP command failure. A subsequent retry may produce successful results.

500 Invalid command name
A client-specified RRP command name was not recognized as a valid RRP command name.

501 Invalid command option
A client-specified RRP command parameter was not recognized as a valid RRP command parameter.

502 Invalid entity value
The "value" of an entity name-value pair is invalid. Command blocks that require an "EntityName" parameter also require a value that specifies the entity name, and the provided value is invalid.

503 Invalid attribute name
A client-specified RRP command parameter was not recognized as a valid RRP command parameter.

504 Missing required attribute
A parameter required to execute the RRP command was not provided by the client. The command should be retried with all required parameters specified.

505 Invalid attribute value syntax
A supplied parameter value is syntactically incorrect. For example, a year value digit such as "5" may be required but the client provided a string of characters such as "five".

506 Invalid option value
A client-specified value for an RRP command parameter is out-of-bounds or otherwise not within acceptable System limits.

507 Invalid command format
The specified command does not resemble a well-formed RRP command. The command should be retried using the proper command structure and syntax.

508 Missing required entity
An entity required for command completion was not provided by the client. For example, the CHECK command requires specification of either a "Domain" entity or a "Nameserver" entity.

509 Missing command option
A command parameter that isn’t really optional (such as the registrar ID in a SESSION command) was not provided by the client. The command should be retried with all needed parameters.

520 Server closing connection. Client should try opening new connection; <why>
A timeout event has been detected, and the client’s session is being ended. The System SHOULD define timeout periods to begin a client

command, complete a client command, and for the duration of an open session. The reason for the timeout MUST be provided at the end of the response code string.

521 Too many sessions open. Server closing connection
A System-defined limit on the number of open connections has been exceeded, and it is impossible to establish a new session at the moment. It may be possible to establish a session by waiting for a few moments or by closing existing unused sessions.

530 Authentication failed
The client-supplied registrar identifier or password was not recognized by the System. A subsequent retry with valid values may produce successful results. Repeated authorization failures MAY result in termination of the TCP connection.

531 Authorization failed
Registrars may not view or alter data associated with either the registry or another registrar. This response code is typically returned when a registrar attempts to view or modify data belonging to either the registry or another registrar. A typical situation includes doing a STATUS command for a domain registered to another registrar.

532 Domain names linked with name server
The name server is hosting active domains. This error occurs when a registrar is trying to delete a server that is the name server for active domains. The registry MUST not allow the registrar to delete this server. All of the domain names using this server MUST be modified to use a different name server before the name server can be deleted.

533 Domain name has active name servers
The domain name has active name servers. The registrar is trying to delete a domain name that is a parent domain of an active name server, i.e., a server that is hosting active domains. All of the name servers within the domain MUST be removed from service before the domain can be deleted.

534 Domain name has not been flagged for transfer
The registrar is trying to approve or reject a domain name transfer for a domain name that is not pending transfer.

535 Restricted IP address
IANA identifies certain IP address ranges that are not valid for normal use. The registrar is trying to use an IP address that is in a restricted IP address range as identified by IANA.

536 Domain already flagged for transfer
The registrar tried to perform a transfer command for a domain name that is awaiting approval of an earlier transfer request.

540 Attribute value is not unique
A supplied attribute value is not unique. This occurs when the registrar is adding a domain name that already exists in the registry, a server that already exists in the registry, or an IP address that is already being used by another server in the registry. Another possibility occurs when performing domain modifications and the registrar is adding a server that is already in the list of servers for the domain name or setting a domain name to a status to which it is already set. The RRP STATUS command MAY be used to determine current domain name status before attempting to change the status. When modifying or adding a name server, the IP address of the name server might not be unique. The registry MUST not allow IP addresses to be used by more than one server.

541 Invalid attribute value
A supplied parameter value is invalid. Examples of invalid attribute values include an invalid IP address, an invalid domain name, an invalid server name, or an invalid renewal period.

542 Invalid old value for an attribute
A current attribute value to be modified is invalid. The registrar is trying to modify an attribute of a server or a domain name that does not exist in the registry.

543 Final or implicit attribute cannot be updated
The registrar is attempting to modify an attribute that is only modifiable by the registry. Registrars can not modify final or implicit attribute values.

544 Entity on hold
The attempted operation was rejected because the entity is on HOLD status. If the HOLD status was set by the registrar, the status can be changed using the MOD command and the requested command can be retried. If the HOLD status was set by the registry, the registrar must contact the registry to change the status before the command can be successful.

545 Entity reference not found
A required entity reference was not found. This occurs when the registrar tries to add a new name server and the parent domain of the name server does not exist in the registry. It also occurs when the user is trying to add a new name server to a domain name when the name server does not exist in the registry.

546 Credit limit exceeded
The registrar's credit limit has been exceeded. This is an implementation specific error that occurs when a potentially billable operation, such as adding a domain name, renewing a domain name, or transferring a domain name, is attempted and the registrar does not have sufficient financial standing with the registry to complete the operation.

547 Invalid command sequence
RRP commands are issued using a well-formed syntax that requires entry of command structures in particular sequences. This response code indicates that an ill-formed command was received and rejected.

548 Domain is not up for renewal
A RENEW command was attempted during a period in which the domain can not be renewed. Implementations MAY limit renewal periods to particular time frames, such as within 90 days of the domain's expiration. This response indicates that the RENEW command was received outside of the System-defined domain renewal period.

549 Command failed
A System error prevented successful completion of the requested RRP command. Retrying the command might produce success, but a repeated failure indicates a System error condition.

550 Parent domain not registered
The parent domain of a name server being registered is not registered. This occurs when the registrar tries to add a new name server and the parent domain for the server does not exist in the registry.

551 Parent domain status does not allow for operation
The status of the parent domain does not allow the requested operation. This occurs when a registrar tries to modify a server whose parent domain is flagged as LOCK or HOLD in the registry.

552 Domain status does not allow for operation
The status of the domain does not allow the requested operation. This occurs when a registrar tries to modify or delete a domain that is flagged as LOCK or HOLD in the registry.

553 Operation not allowed. Domain pending transfer
The status of the domain does not allow the requested operation. The registrar is attempting to delete a domain that is pending approval or denial of a transfer request.

554 Domain already registered
A registrar tried to register a domain name that has already been registered by the same registrar.

555 Domain already renewed
A registrar tried to renew a domain using the same parameters as specified for an earlier, successful renewal. This will commonly occur when executing the same RENEW command more than once.

556 Maximum registration period exceeded
A registrar tried to renew a domain registration, and the resulting new registration period exceeds the System-defined maximum registration period. If there is renewal time available with the System-defined maximum registration period it may be possible to retry the RENEW command with specified renewal period parameters.

5.2 Command-Response Correspondence

The session between the client and the server is intended to be an alternating dialogue. Each command issued by a client MUST be acted upon by the server, which MUST return a response code to document the success or failure of command execution. "Success" means that the command completed normal execution without error. "Failure" means that the System did not complete the command as requested. Failure may be due to either syntax, semantic, data, or System errors.

A complete list of response codes for each RRP command is listed below.

Command: ADD
Success: 200, 220
Failure: 420, 421, 500, 502, 503, 504, 505, 507, 508, 520, 531, 535, 540, 541, 545, 546, 547, 549, 550, 554

Command: CHECK
Success: 210, 211, 212, 213
Failure: 220, 420, 421, 500, 502, 503, 504, 505, 507, 508, 520, 541, 547, 549

Command: DEL
Success: 200, 220
Failure: 420, 421, 500, 502, 503, 504, 505, 507, 508, 520, 531, 532, 533, 541, 544, 545, 547, 549, 551, 552, 553

Command: DESCRIBE
Success: 200, 220
Failure: 420, 421, 500, 501, 506, 507, 509, 520, 547, 549

Command: MOD
Success: 200, 220
Failure: 420, 421, 500, 502, 503, 504, 505, 507, 508, 520, 531, 535, 540, 541, 542, 543, 544, 545, 547, 549, 550, 551, 552, 553

Command: QUIT
Success: 220
Failure: 420, 421, 500, 507, 520, 547, 549

Command: RENEW
Success: 200, 220
Failure: 420, 421, 500, 502, 503, 504, 505, 507, 508, 520, 531, 541, 545, 546, 547, 548, 549, 552, 553, 555, 556

Command: SESSION
Success: 200, 220
Failure: 420, 421, 500, 501, 506, 507, 508, 509, 520, 521, 530, 531, 547, 549

Command: STATUS
Success: 200, 220
Failure: 420, 421, 500, 501, 502, 503, 504, 505, 506, 507, 508, 520, 531, 541, 545, 547, 549

Command: TRANSFER
Success: 200, 220
Failure: 420, 421, 500, 501, 502, 503, 504, 505, 506, 507, 508, 520, 531, 534, 536, 541, 544, 545, 546, 547, 549, 552, 553

6. Domain Status Codes

The status of a domain can be viewed using the RRP STATUS command and modified using the RRP MOD command. Both the registry and the sponsoring registrar MAY view and change the status of a domain. The criteria for status changes are highly dependent on registry and registrar business models and are thus beyond the scope of this specification.

The domain’s status SHOULD have a direct bearing on whether or not the domain appears in the appropriate TLD zone file and whether or not the domain can be modified. A domain can have more than one assigned status, e.g., REGISTRAR-HOLD and REGISTRAR-LOCK. If a domain is in ACTIVE status, then the domain name can only be in this status. When a registrar sets a domain name to REGISTRAR-LOCK, the registry MUST automatically remove the ACTIVE status. When the registrar removes the REGISTRAR-LOCK and other domain statuses, the registry MUST automatically set the domain name status to ACTIVE.

6.1 Domain Status Code Description

ACTIVE: This is the default status of a domain at registration time. The registry sets the domain to this status. The domain is modifiable by the registrar. The domain can be renewed. The domain SHALL be included in the zone file when in this status if the domain has at least one associated name server.

REGISTRY-LOCK: The registry sets the domain to this status. The domain cannot be modified or deleted by the registrar. The registry MUST remove the REGISTRY-LOCK status for the registrar to modify the domain. The domain can be renewed. The domain SHALL be included in the zone file when in this status if the domain has at least one associated name server.

REGISTRY-HOLD: The registry sets the domain to this status. The domain cannot be modified or deleted by the registrar. The registry MUST remove the REGISTRY-HOLD status for the registrar to modify the domain. The domain can be renewed. The domain SHALL NOT be included in the zone file when in this status.

REGISTRAR-HOLD: The registrar of the domain sets the domain to this status. The domain can not be modified or deleted when in this status. The registrar MUST remove REGISTRAR-HOLD status to modify the domain. The domain can be renewed. The domain SHALL NOT be included in the zone file when in this status.

REGISTRAR-LOCK: The registrar of the domain sets the domain to this status. The domain cannot be modified or deleted when in this status. The registrar MUST remove REGISTRAR-LOCK status to modify the domain. The domain can be renewed. The domain SHALL be included in the zone file when in this status.

REGISTRY-DELETE-NOTIFY: A domain is set on this status if it has expired and has child name servers that are hosting other domains. Only the registry may set this status. The domain SHALL be included in the zone file when in this status if the domain has at least one associated name server.

7. Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in [ABNF].

```
; ABNF specification for Registry Registrar Protocol (RRP) v1.1.0
; Note that character string literals are case insensitive.
```

```
; Lexical tokens
space = %x20 ; " "
dot = %x2E ; "."
dash = %x2D ; "-"
underscore = %x5F ; "_"
colon = %x3A ; ":"
cr = %x0D ; ASCII carriage return
lf = %x0A ; ASCII linefeed
crlf = cr lf
alpha = %x41-5A / %x61-7A ; A-Z / a-z
digit = %x30-39 ; 0-9
dns-char = alpha / digit / dash
id-char = alpha / digit / underscore / dash
id-prefix = alpha / digit
id-word = id-prefix *id-char
printable-char = %x20-7E ; ASCII " " - "~"

; Start of basic grammar.
year = 4digit
month = 2digit
day = 2digit
ymd = year dash month dash day
hour = 2digit
minute = 2digit
second = 2digit
split-second = 1digit
hms = hour colon minute colon second dot split-second
time-stamp = ymd space hms
ip-address = 1*3digit dot 1*3digit dot 1*3digit dot 1*3digit
password = 4*16printable-char
option-name = 1*128id-word
option-tag = dash option-name
option-value = 1*128id-word
attribute-name = 1*128id-word
attribute-value = 1*128printable-char
attribute-line = attribute-name colon attribute-value crlf
response = 3digit space 1*printable-char crlf
version-number = "RRP" space 1*digit dot 1*digit dot 1*digit
label = id-prefix [*61dns-char id-prefix]
sldn = label dot label
servername = *(label dot) sldn
period = %x31-39 / (%x31-39 %x30-39) ; "1" - "9" or "10" - "99"
period-option = dash "Period" colon period crlf
yesno = "Yes" / "No"
domainstatus = "Active" / "Registry-Lock" / "Registry-Hold" /
               "Registrar-Lock" / "Registrar-Hold" /
               "Registry-Delete-Notify"
```

```
; RRP commands and responses.
rrp = add / check / delete / describe / mod / quit / renew /
      session / status / transfer

add = add-request add-response
check = check-request check-response
delete = del-request del-response
describe = describe-request describe-response
mod = mod-request mod-response
quit = quit-request quit-response
renew = renew-request renew-response
session = session-request session-response
status = status-request status-response
transfer = transfer-request transfer-response

; ADD command.
add-request = add-domain-request / add-nameserver-request
add-response = add-domain-response / add-nameserver-response
add-domain-request = "add" crlf
 "EntityName" colon "Domain" crlf
 "DomainName" colon sldn crlf
 [period-option]
 0*13("NameServer" colon servername crlf)
 dot crlf
add-nameserver-request = "add" crlf
 "EntityName" colon "NameServer" crlf
 "NameServer" colon servername crlf
 1*("IPAddress" colon ip-address crlf)
 dot crlf
add-domain-response = response
 "RegistrationExpirationDate" colon time-stamp crlf
 "status" colon domainstatus crlf
 dot crlf
add-nameserver-response = response
 dot crlf

; CHECK command.
check-request = check-domain-request / check-nameserver-request
check-response = check-domain-response / check-nameserver-response
check-domain-request = "check" crlf
 "EntityName" colon "Domain" crlf
 "DomainName" colon sldn crlf
 dot crlf
check-nameserver-request = "check" crlf
 "EntityName" colon "NameServer" crlf
 "NameServer" colon servername crlf
 dot crlf
check-domain-response = response
```

```
 dot crlf
check-nameserver-response = available-check-nameserver-response /
                            notavailable-check-nameserver-response
available-check-nameserver-response = response
 dot crlf
notavailable-check-nameserver-response = response
 1*("IPAddress" colon ip-address crlf)
 dot crlf

; DEL command.
del-request = del-domain-request / del-nameserver-request
del-response = response
 dot crlf
del-domain-request = "del" crlf
 "EntityName" colon "Domain" crlf
 "DomainName" colon sldn crlf
 dot crlf
del-nameserver-request = "del" crlf
 "EntityName" colon "NameServer" crlf
 "NameServer" colon servername crlf
 dot crlf

; DESCRIBE command.
describe-request = "describe" crlf
 [target-option]
 *(option-tag colon option-value crlf)
 dot crlf
describe-response = response
 "Protocol" colon version-number crlf
 *attribute-line
 dot crlf
target-option = dash "Target" colon "Protocol" crlf

; MOD command.
mod-request = mod-domain-request / mod-nameserver-request
mod-response = response
 *attribute-line
 dot crlf
mod-domain-request = "mod" crlf
 "EntityName" colon "Domain" crlf
 "DomainName" colon sldn crlf
 *(add-attribute-value-line /
 remove-attribute-value-line /
 replace-attribute-value-line)
```

```
 dot crlf
mod-nameserver-request = "mod" crlf
 "EntityName" colon "NameServer" crlf
 "NameServer" colon servername crlf
 ["NewNameServer" colon attribute-value crlf]
 *(add-attribute-value-line /
 remove-attribute-value-line /
 replace-attribute-value-line)
 dot crlf
add-attribute-value-line =
 attribute-name colon new-attribute-value
remove-attribute-value-line =
 attribute-name colon old-attribute-value "="
replace-attribute-value-line =
 attribute-name colon old-attribute-value "="
 new-attribute-value
old-attribute-value = attribute-value
new-attribute-value = attribute-value

; QUIT command.
quit-request = "quit" crlf
 dot crlf
quit-response = response
 dot crlf

; RENEW command.
renew-request = "renew" crlf
 "EntityName" colon "Domain" crlf
 "DomainName" colon sldn crlf
 [renew-period-option]
 dot crlf
expiration-year-option = dash "CurrentExpirationYear" colon year crlf
renew-period-option = period-option expiration-year-option /
                      expiration-year-option period-option
renew-response = response
 "RegistrationExpirationDate" colon time-stamp crlf
 dot crlf

; SESSION command.
session-request = "session" crlf
 registrar-id-option
 registrar-password-option
 [registrar-newpassword-option]
 dot crlf
session-response = response
 dot crlf
registrar-id-option = dash "Id" colon option-value crlf
registrar-password-option =
```

```
 dash "Password" colon password crlf
registrar-newpassword-option =
 dash "NewPassword" colon password crlf

; STATUS command.
status-request = status-domain-request /
                 status-nameserver-request
status-response = response
 *attribute-line
 dot crlf
status-domain-request = "status" crlf
 "EntityName" colon "Domain" crlf
 "DomainName" colon sldn crlf
 dot crlf
status-nameserver-request = "status" crlf
 "EntityName" colon "NameServer" crlf
 "NameServer" colon servername crlf
 dot crlf

; TRANSFER command.
transfer-request = "transfer" crlf
 [approve-option]
 "EntityName" colon "Domain" crlf
 "DomainName" colon sldn crlf
 dot crlf
transfer-response = response
 "RegistrationExpirationDate" colon time-stamp crlf
 dot crlf
approve-option = dash "Approve" colon yesno crlf

; End of grammar.
```

8. Internationalization

RRP is defined using 7-bit US-ASCII characters. Other character sets and character codes are not currently supported.

9. Known Issues

RRP was not designed to provide bulk data query features. The primary goal of the original protocol designers was to provide a fast, light weight transactional protocol that could be implemented with minimal need for database queries that would take a "long" time to complete or that would return a "large" amount of data. Implementers SHOULD consider developing offline reporting features to provide bulk data for registrar reporting in a fashion suitable for the given registry-registrar operating environment.

This version of RRP does contain a few limitations noted over the course of several months of operational experience with live domain name registrars. Later versions of this protocol or its successors should strive to resolve or address each of the following issues:

The DESCRIBE command should return information describing System-defined default implementation values.

Use of the RENEW command without the "CurrentExpirationYear" and "Period" parameters does not provide idempotency. Repeated execution of a RENEW command without these parameters can result in multiple successful RENEW commands, which may not be the desired action if a registrar is retrying a RENEW command due to network connectivity problems.

Time stamps returned by RRP do not include time zone identifiers and SHOULD be interpreted as local registry time.

The protocol does not provide features for a registrar to become aware of domain transfer requests and responses. Systems must rely on means outside of the protocol, such as electronic mail and/or registry-provided reports, to inform registrars of transfer requests and responses.

The protocol does not provide features for a registrar to determine all of the domains served by a name server. Systems must provide this information using a method outside of the protocol, such as through periodic extracts from a System database.

The protocol does not provide features to manage lame delegation of name servers. Any registrar may "use" name servers registered by another registrar. When a registrar tries to delete a domain or name server it is quite possible that name servers in the domain to be deleted or the name server to be deleted will be associated with other live domains, precluding immediate deletion. Systems must rely on means outside of the protocol to manage lame delegation of name servers.

The use of "=" within the MOD command to indicate a value to be removed is somewhat confusing. A more explicit means of identifying old and new attribute values within the protocol syntax could make this feature more obvious.

The CHECK command also returns name server IP addresses when returning positive confirmation of the registration of a name server. This extra information may be useful, but it is inconsistent with the limited function of the command. The command should return a positive or negative response and nothing more.

The formal protocol syntax described in this document requires a specific order for the elements of a command entity block and command options. The NSI Registry's server-side implementation of the protocol provides the additional flexibility of allowing order independent specification of options and entity block elements. Client-side implementers are strongly urged to observe the order of command elements as specified here to ensure compliance if the more restricted form is enforced in the future.

RRP does not return time stamps or transaction identifiers to track transactions. The NSI Registry provides registrars with daily and weekly reports that include time stamps in local registry time to document and synchronize data on a per-registrar basis.

10. Security Considerations

Misuse of the Registry Registrar Protocol can have catastrophic operational consequences for registrants, registrars, and registries. As such, all registrars must be authenticated prior to all interactions with the registry. In addition, all data exchanged between the registrar and the registry must be protected to avoid unintended disclosure of information.

11. IANA Considerations

IANA assigned TCP port 648 for RRP use in November 1998. No other action is required of IANA to support operation of this protocol.

IANA has reserved certain IPv4 address ranges as described in [ALLOCATION]. Implementers MUST ensure that name server IP addresses do not fall into one of the reserved address ranges to avoid operational DNS errors.

12. References

[ABNF] Crocker, D. (Editor) and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[ALLOCATION] Hubbard, K., Kosters, M., Conrad, D., Karrenberg, D. and J. Postel, "Internet Registry IP Allocation Guidelines", BCP 12, RFC 2050, November 1996.

[MUSTSHOULD] Bradner, S., "Key Words for Use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[SSL] A. Frier, P. Karlton, and P. Kocher, "The SSL 3.0 Protocol", Netscape Communications Corp., November 18, 1996.

[TLS] Dierks T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

13. Acknowledgments

Many people have contributed significantly to this document and the protocol it describes. Brad McMillen and Neeran Saraf deserve special mention as co-authors of an earlier internal protocol specification. Other content contributors to the earlier internal specification include Aristotle Balogh, Chris Bason, Mark Kosters, Jasdip Singh, and Yibing Wu. Finally, significant contributors to the review of this document include Steve Mahlstedt and Chris Smith.

14. Authors' Addresses

Scott Hollenbeck
Network Solutions, Inc. Registry
505 Huntmar Park Dr.
Herndon, VA 20170
USA

EMail: shollenb@netsol.com

Manoj Srivastava
Network Solutions, Inc. Registry
505 Huntmar Park Dr.
Herndon, VA 20170
USA

EMail: manojs@netsol.com

15. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.