
Workgroup:	Network Working Group
Internet-Draft:	draft-kowalik-rpp-data-objects-00
Published:	11 September 2025
Intended Status:	Standards Track
Expires:	15 March 2026
Author:	P. Kowalik <i>DENIC</i>

Registry Provisioning Protocol (RPP) Data Objects

Abstract

This document defines abstract data objects for the Registry Provisioning Protocol (RPP). The definitions for domain name, contact, and host objects focus on the logical structure and constraints of their constituent data elements, independent of any specific data representation or media type. This document follows the architecture defined in [I-D.kowalik-rpp-architecture].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

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

This Internet-Draft will expire on 15 March 2026.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
1.1. Conventions and Terminology	3
2. Resource Definition Principles	3
2.1. Data Element Abstraction	3
2.2. Extensibility	3
2.3. Data Element Semantics	4
3. Common Data Types	4
3.1. Identifiers	4
3.2. Timestamps	4
3.3. Status Types	4
4. Component Objects	5
4.1. Period Object	5
4.2. Nameserver Object	5
4.3. Contact Association Object	6
4.4. Authorization Information Object	6
5. Domain Name Resource Object	7
5.1. Object Description	7
5.2. Data Elements	7
5.3. Operations	10
5.3.1. Create Operation	10
5.3.2. Read Operation	10
5.3.3. Delete Operation	11
5.3.4. Renew Operation	11
5.4. Object Description	12
5.5. Data Elements	12
5.6. Object Description	12
5.7. Data Elements	12

6. IANA Considerations	12
6.1. RPP Object Registry	12
6.1.1. Registration Policy	12
6.1.2. Registry Structure	12
6.1.3. Initial Registrations	13
7. Normative References	17
Author's Address	18

1. Introduction

The Registry Provisioning Protocol (RPP) requires a clear definition of its data objects. This document catalogues the fundamental resource objects managed through RPP: domains, contacts, and hosts.

In accordance with the RPP architecture [[I-D.kowalik-rpp-architecture](#)], the definitions herein are abstract. They specify the logical data elements, their meanings, and their constraints, rather than a specific representation format. This approach ensures that the core data model can be consistently implemented across different media types.

1.1. Conventions and Terminology

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.

2. Resource Definition Principles

2.1. Data Element Abstraction

Each resource is composed of logical data elements. A data element is a logical unit of information identified by a stable name, independent of its representation in any given media type. The definition for each element specifies its logical name, purpose, cardinality, data type, and constraints.

2.2. Extensibility

The set of data elements for a given resource object is extensible. New data elements MAY be defined and registered with IANA to support new features.

2.3. Data Element Semantics

The definition of each data element within an object consists of the following attributes:

- **Name:** A human-readable name for the data element.
- **Identifier:** A machine-readable, unique identifier for the element, using camelCase notation.
- **Cardinality:** Specifies the number of times an element may appear. The notation is as follows:
 - 1 for exactly one
 - 0-1 for zero or one
 - 0+ for zero or more
 - and 1+ for one or more
- **Data Type:** Defines the element's data structure, which can be a primitive type (e.g., String, Integer) or a reference to another component object.
- **Description:** Explains the purpose of the data element and any other relevant information.
- **Constraints:** Provides specific validation rules or limitations on top of the data type itself, such as value ranges.
- **Mutability:** Defines the lifecycle of the data element's value. It **MUST** be one of the following:
 - create-only: The element's value is provided during the object's creation and cannot be modified thereafter.
 - read-only: The element's value is managed by the server. It cannot be set or modified directly by the client, though it may change as a result of server-side operations.
 - read-write: The element's value can be set and modified by the client.

3. Common Data Types

This section defines primitive data types and structures that are re-used across multiple resource object definitions.

3.1. Identifiers

(Definition of a common type for identifiers, e.g., for contacts or hosts)

3.2. Timestamps

(Definition of a standardized timestamp format, e.g., ISO 8601)

3.3. Status Types

(Definition of a common structure for status flags on objects)

4. Component Objects

This section defines common component objects that are re-used in the definitions of top-level resource objects.

4.1. Period Object

- Name: Period Object
- Identifier: period
- Description: Represents a duration of time.
- Data Elements:
 - Value
 - Identifier: value
 - Cardinality: 1
 - Mutability: read-write
 - Data Type: Integer.
 - Description: The numeric value of the period.
 - Constraints: The value MUST be from 1 to 99, inclusive.
 - Unit
 - Identifier: unit
 - Cardinality: 1
 - Mutability: read-write
 - Data Type: String (Token).
 - Description: The unit of the period.
 - Constraints: The value MUST be one of: "y" (years) or "m" (months).

4.2. Nameserver Object

- Name: Nameserver Object
- Identifier: nameserver
- Description: Represents a single nameserver.
- Data Elements:
 - Host Name
 - Identifier: hostName
 - Cardinality: 1
 - Mutability: read-write
 - Data Type: String.
 - Description: The name of the host.
 - Constraints: The value MUST be a syntactically valid host name.

- IP Addresses
 - Identifier: ipAddresses
 - Cardinality: 0+
 - Mutability: read-write
 - Data Type: String.
 - Description: IP addresses associated with the host.
 - Constraints: Each value MUST be a syntactically valid IPv4 or IPv6 address.

4.3. Contact Association Object

TODO: find a better name for Identifier

TODO: decide if association with attributes shouldn't be something else

- Name: Contact Association Object
- Identifier: contactAssociation
- Description: Links a Contact resource to another resource, defining the role of the association.
- Data Elements:
 - Identifier
 - Identifier: identifier
 - Cardinality: 1
 - Mutability: read-write
 - Data Type: String (Client Identifier).
 - Description: The identifier of the associated contact object.
 - Constraints: The value MUST be a valid, server-known Client Identifier.
 - Type
 - Identifier: type
 - Cardinality: 1
 - Mutability: read-write
 - Data Type: String (Token).
 - Description: The role of the associated contact.
 - Constraints: For domain name associations, the value MUST be one of: "admin", "billing", or "tech".

4.4. Authorization Information Object

TODO: define it better

- Name: Authorization Information Object
- Identifier: authInfo

- **Description:** Contains information used to authorize operations on a resource object.
- **Data Elements:** The structure of this object is intentionally left abstract to accommodate various authentication mechanisms (e.g., password, token). The server defines the specific elements and constraints.

5. Domain Name Resource Object

5.1. Object Description

- **Name:** Domain Name Resource Object
- **Identifier:** domainName
- **Description:** A Domain Name resource object represents a domain name and contains the data required for its provisioning and management in the registry.

5.2. Data Elements

The following data elements are defined for the Domain Name resource object.

- **Name**
 - **Identifier:** name
 - **Cardinality:** 1
 - **Mutability:** create-only
 - **Data Type:** String.
 - **Description:** The fully qualified name of the domain object.
 - **Constraints:** The value **MUST** be a fully qualified domain name that conforms to the syntax described in [\[RFC1035\]](#).
- **Repository ID**
 - **Identifier:** repositoryId
 - **Cardinality:** 1
 - **Mutability:** read-only
 - **Data Type:** String.
 - **Description:** A server-assigned unique identifier for the object.
 - **Constraints:** (None)
- **Status**
 - **Identifier:** status
 - **Cardinality:** 0+
 - **Mutability:** read-only
 - **Data Type:** String (Token).
 - **Description:** The current status descriptors associated with the domain.

- Constraints: The value MUST be one of the status tokens defined in the IANA registry for domain statuses.
- Registrant
 - Identifier: registrant
 - Cardinality: 0-1
 - Mutability: read-write
 - Data Type: String (Client Identifier).
 - Description: The contact object associated with the domain as the registrant.
 - Constraints: The identifier MUST correspond to a valid Contact resource object known to the server.
- Contacts
 - Identifier: contacts
 - Cardinality: 0+
 - Mutability: read-write
 - Data Type: A collection of Contact Association Objects.
 - Description: A collection of other contact objects associated with the domain object.
 - Constraints: (None)
- Nameservers
 - Identifier: nameservers
 - Cardinality: 0-1
 - Mutability: read-write
 - Data Type: A collection of Nameserver Objects.
 - Description: A collection of nameservers associated with the domain.
 - Constraints: (None)
- Subordinate Hosts
 - Identifier: subordinateHosts
 - Cardinality: 0+
 - Mutability: read-only
 - Data Type: Collection of Strings.
 - Description: A collection of fully qualified names of subordinate host objects that exist under this domain.
 - Constraints: (None)
- Sponsoring Client ID
 - Identifier: sponsoringClientId
 - Cardinality: 1
 - Mutability: read-only

- Data Type: String (Client Identifier).
- Description: The identifier of the client that is the current sponsor of the domain object.
- Constraints: (None)
- Creating Client ID
 - Identifier: creatingClientId
 - Cardinality: 0-1
 - Mutability: read-only
 - Data Type: String (Client Identifier).
 - Description: The identifier of the client that created the domain object.
 - Constraints: (None)
- Creation Date
 - Identifier: creationDate
 - Cardinality: 0-1
 - Mutability: read-only
 - Data Type: Timestamp.
 - Description: The date and time of domain object creation.
 - Constraints: The value is set by the server and cannot be specified by the client.
- Updating Client ID
 - Identifier: updatingClientId
 - Cardinality: 0-1
 - Mutability: read-only
 - Data Type: String (Client Identifier).
 - Description: The identifier of the client that last updated the domain object.
 - Constraints: This element MUST NOT be present if the domain has never been modified.
- Update Date
 - Identifier: updateDate
 - Cardinality: 0-1
 - Mutability: read-only
 - Data Type: Timestamp.
 - Description: The date and time of the most recent domain object modification.
 - Constraints: This element MUST NOT be present if the domain object has never been modified.
- Expiry Date
 - Identifier: expiryDate
 - Cardinality: 0-1
 - Mutability: read-only

- Data Type: Timestamp.
- Description: The date and time identifying the end of the domain object's registration period.
- Constraints: The value is set by the server and cannot be specified by the client.
- Transfer Date
 - Identifier: transferDate
 - Cardinality: 0-1
 - Mutability: read-only
 - Data Type: Timestamp.
 - Description: The date and time of the most recent successful domain object transfer.
 - Constraints: This element MUST NOT be provided if the domain object has never been transferred.
- Authorization Information
 - Identifier: authInfo
 - Cardinality: 0-1
 - Mutability: read-write
 - Data Type: Authorization Information Object.
 - Description: Authorization information to be associated with the domain object.
 - Constraints: (None)

5.3. Operations

5.3.1. Create Operation

The Create operation allows a client to provision a new Domain Name resource. The operation accepts as input all create-only and read-write data elements defined for the Domain Name Resource Object.

In addition, the following transient data element is defined for this operation:

- Registration Period
 - Identifier: period
 - Cardinality: 0-1
 - Data Type: Period Object.
 - Description: The initial registration period for the domain name. This value is used by the server to calculate the initial expiryDate of the object. This element is not persisted as part of the object's state.

5.3.2. Read Operation

The Read operation allows a client to retrieve the data elements of a Domain Name resource. The server's response MAY vary depending on client authorization and server policy.

The following transient data elements are defined for this operation:

- Hosts Filter
 - Identifier: hostsFilter
 - Cardinality: 0-1
 - Data Type: String (Token).
 - Description: Controls which host information is returned with the object.
 - Constraints: The value MUST be one of "all", "del" (delegated), "sub" (subordinate), or "none". The default value is "all".
- Query Authorization Information
 - Identifier: queryAuthInfo
 - Cardinality: 0-1
 - Data Type: Authorization Information Object.
 - Description: Authorization information provided by the client to gain access to the full set of the object's data elements.

5.3.3. Delete Operation

The Delete operation allows a client to remove an existing Domain Name resource. The operation targets a specific resource object identified by its name.

The server SHOULD reject a delete request if subordinate host objects are associated with the domain name.

5.3.4. Renew Operation

The Renew operation allows a client to extend the validity period of an existing Domain Name resource. The operation targets a specific resource object identified by its name.

The following transient data elements are defined for this operation:

- Current Expiry Date
 - Identifier: currentExpiryDate
 - Cardinality: 1
 - Data Type: Timestamp
 - Description: The current expiry date of the domain name. The server MUST validate this against the object's current expiryDate to prevent unintended duplicate renewals.
- Renewal Period
 - Identifier: renewalPeriod
 - Cardinality: 0-1
 - Data Type: Period Object
 - Description: The duration to be added to the object's registration period. This value is used by the server to calculate the new expiryDate.

Contact Resource Object

5.4. Object Description

A Contact resource object represents the social information for an individual or organization associated with other objects.

5.5. Data Elements

(This section will list and define the abstract data elements for a contact.)

Host Resource Object

5.6. Object Description

A Host resource object represents a name server that provides DNS services for a domain name.

5.7. Data Elements

(This section will list and define the abstract data elements for a host.)

6. IANA Considerations

6.1. RPP Object Registry

This document establishes the "Registry Provisioning Protocol (RPP) Object Registry". This registry serves as a definitive, hierarchical catalogue of all resource objects, component objects, data elements, and operations used within RPP.

6.1.1. Registration Policy

The policy for adding new objects, data elements, or operations to this registry is "Specification Required" [[RFC8126](#)].

6.1.2. Registry Structure

The registry is organized as a collection of Object definitions. Each Object definition **MUST** include:

- A header containing the Object Identifier, Object Name, Object Type (Resource or Component), a brief description, and a reference to its defining specification.
- A "Data Elements" table listing all persisted data elements associated with the object. Each entry **MUST** specify the element's Identifier, Name, Cardinality, Mutability, Data Type, and description.
- If applicable, an "Operations" section. For each operation, the registry **MUST** provide:
 - The Operation's Name and a description.

- A "Parameters" table listing all data elements that are provided as input to the operation but are not persisted as part of the object's state. Each entry **MUST** specify the parameter's Identifier, Name, Cardinality, Data Type, and a description.

6.1.3. Initial Registrations

The initial contents of the RPP Object Registry are defined below.

Object: period

Object Name: Period Object

Object Type: Component

Description: Represents a duration of time.

Reference: [This-ID]

Data Elements

Element Identifier	Element Name	Card.	Mutability	Data Type	Description
value	Value	1	read-write	Integer	The numeric value of the period.
unit	Unit	1	read-write	Token	The unit of the period.

Table 1

Object: nameserver

Object Name: Nameserver Object

Object Type: Component

Description: Represents a single nameserver.

Reference: [This-ID]

Data Elements

Element Identifier	Element Name	Card.	Mutability	Data Type	Description
hostName	Host Name	1	read-write	String	The name of the host.
ipAddresses	IP Addresses	0+	read-write	String	IP addresses associated with the host.

Table 2

Object: contactAssociation

Object Name: Contact Association Object

Object Type: Component

Description: Links a Contact resource to another resource, defining its role.

Reference: [This-ID]

Data Elements

Element Identifier	Element Name	Card.	Mutability	Data Type	Description
identifier	Identifier	1	read-write	String (Client ID)	The identifier of the associated contact object.
type	Type	1	read-write	Token	The role of the associated contact.

Table 3

Object: authInfo

Object Name: Authorization Information Object

Object Type: Component

Description: Contains authorization credentials for an operation.

Reference: [This-ID]

Data Elements

Element Identifier	Element Name	Card.	Mutability	Data Type	Description
(n/a)	Authorization Info	(n/a)	read-write	Abstract Object	Authentication credentials.

Table 4

Object: domainName

Object Name: Domain Name Resource Object

Object Type: Resource

Description: Represents a domain name and its associated data.

Reference: [This-ID]

Data Elements

Identifier	Name	Card.	Mutability	Data Type	Description
name	Name	1	create-only	String	The fully qualified name of the domain object.
repositoryId	Repository ID	1	read-only	String	A server-assigned unique identifier for the object.
status	Status	0+	read-only	Token	The current status descriptors for the domain.
registrant	Registrant	0-1	read-write	String (Client ID)	The registrant contact ID.
contacts	Contacts	0+	read-write	Collection (contactAssociation)	Associated contact objects.
nameservers	Nameservers	0-1	read-write	Collection (nameserver)	Associated nameserver objects.
subordinateHosts	Subordinate Hosts	0+	read-only	Collection (String)	Subordinate host names.
sponsoringClientId	Sponsoring Client ID	1	read-only	String (Client ID)	The current sponsoring client ID.
creatingClientId	Creating Client ID	0-1	read-only	String (Client ID)	The client ID that created the object.

Identifier	Name	Card.	Mutability	Data Type	Description
creationDate	Creation Date	0-1	read-only	Timestamp	Creation timestamp.
updatingClientId	Updating Client ID	0-1	read-only	String (Client ID)	The client ID that last updated the object.
updateDate	Update Date	0-1	read-only	Timestamp	The timestamp of the last update.
expiryDate	Expiry Date	0-1	read-only	Timestamp	Expiry timestamp.
transferDate	Transfer Date	0-1	read-only	Timestamp	The timestamp of the last successful transfer.
authInfo	Authorization Info	0-1	read-write	authInfo	Authorization information for the object.

Table 5

Operations

Operation: Create

Description: Provisions a new Domain Name resource.

Parameters

Identifier	Name	Card.	Data Type	Description
period	Registration Period	0-1	period	The initial registration period for the domain name.

Table 6

Operation: Read

Description: Retrieves the data elements of a Domain Name resource.

Parameters

Identifier	Name	Card.	Data Type	Description
hostsFilter	Hosts Filter	0-1	Token	Controls which host information is returned.
queryAuthInfo	Query Authorization Information	0-1	authInfo	Credentials to authorize access to full object data.

Table 7

Operation: Delete

Description: Removes an existing Domain Name resource.

Parameters: (None)

Operation: Renew

Description: Extends the validity period of a Domain Name resource.

Parameters

Identifier	Name	Card.	Data Type	Description
currentExpiryDate	Current Expiry Date	1	Timestamp	The expected current expiry date, for validation.
renewalPeriod	Renewal Period	0-1	period	The duration to add to the registration period.

Table 8

Security Considerations

(This section will discuss security issues related to the data objects, such as data privacy, validation, and potential for misuse.)

7. Normative References

[I-D.kowalik-rpp-architecture] Kowalik, P. and M. Wullink, "RPP Architecture", Work in Progress, Internet-Draft, draft-kowalik-rpp-architecture-02, 29 June 2025, <<https://datatracker.ietf.org/doc/html/draft-kowalik-rpp-architecture-02>>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.

- [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/info/rfc2119>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [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/info/rfc8174>>.

Author's Address

Pawel Kowalik

DENIC

Email: pawel.kowalik@denic.de

URI: <https://denic.de/>