

## NSI PROPRIETARY AND CONFIDENTIAL

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

# NSI Registry C API, Version 1.1.0

1.	INTRODUCTION.....	1
2.	RRP API EXAMPLE.....	2
3.	API COMMANDS.....	5
3.1.	RRPAddDomain .....	5
3.2.	RRPAddNameServer.....	6
3.3.	RRPCheckDomain .....	7
3.4.	RRPCheckNameServer .....	7
3.5.	RRPDeleteDomain .....	8
3.6.	RRPDeleteNameServer .....	8
3.7.	RRPDescribe .....	9
3.8.	RRPModifyDomain .....	9
3.9.	RRPModifyNameServer .....	11
3.10.	RRPRenewDomain .....	12
3.11.	RRPStartSession.....	13
3.12.	RRPEndSession.....	14
3.13.	RRPStatusDomain .....	14
3.14.	RRPStatusNameServer.....	15
3.15.	RRPTransferDomain.....	16

## 1. Introduction

This document provides a description of the Registry Registrar Protocol (RRP) C API. The description includes syntax, inputs, outputs, and returns. The document also provides an [example](#) for running an RRP session using this API. Refer to the *Registry Registrar Protocol, version 1.1.0* for command response codes.

Two new parameters have been added to the API functions [RRPAddDomain](#) and [RRPRenewDomain](#). For documentation on the Java RRP API, please refer to the `javadoc` directory, which contains a collection of HTML class descriptions.

[Document was last updated December 3, 1999.]

The following API commands are described in this document.

- [RRPAddDomain](#). This command registers a domain name.
- [RRPAddNameServer](#). This command registers a new nameserver.
- [RRPCheckDomain](#). This command checks the availability of a domain name.
- [RRPCheckNameServer](#). This command checks the availability of a nameserver.
- [RRPDeleteDomain](#). This command deletes the registration of a domain name.
- [RRPDeleteNameServer](#). This command deletes the registration of a nameserver.
- [RRPDescribe](#). This command extracts general information about RRP.

## NSI PROPRIETARY AND CONFIDENTIAL

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

- [RRPModifyDomain](#). This command updates the attributes of a registered domain name.
- [RRPModifyNameServer](#). This command updates the attributes of a registered nameserver.
- [RRPRenewDomain](#). This command renews the registration of a domain name.
- [RRPStartSession](#). This command creates an authenticated RRP session and allows a registrar to modify their logon password.
- [RRPEndSession](#). This command terminates an RRP session.
- [RRPStatusDomain](#). This command queries for the attributes of a domain name.
- [RRPStatusNameServer](#). This command queries for the attributes of a nameserver.
- [RRPTransferDomain](#). This command has two functions. A registrar can use the command to request the transfer of a domain from another registrar to itself. A registrar can use this function to approve or deny any requests previously issued for the transfer of a domain from the registrar to another registrar.

## 2. RRP API Example

The following section contains an example of a complete RRP session that registers a domain name and a nameserver using the RRP APIs described in the document.

```
/* Test driver for rrpAPI.c */

#include <stdio.h>

#include <string.h>

/* Include the RRP API Header file */
#include "../include/rrpAPI.h"

void main () {

/* The RRP API Returned Response Code */
RRPRESPONSE* response = NULL;

/* List of Domain Servers*/
RRPVECTOR* nameServers = NULL;

/* List of Nameserver IP Addresses */
```

**NSI PROPRIETARY AND CONFIDENTIAL**

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

```
RRPVECTOR* ipAddresses = NULL;

/* Connect to OTE environment */

if (RRPCreateConnection("RegyOTE.net", 648) < 0) {

    RRPPrintInternalErrorDescription();

    exit(1);

} /*Connection created, start an RRP session*/

response = RRPStartSession("steveid", "stevepw");

if (response == NULL) {

    RRPPrintInternalErrorDescription();

    exit(1);

}

else if (response-code != 200) {

    printf("Error creating RRP session: %s\n", response-
description);

    exit(1);

}

RRPFreeResponse(response);

printf("Session established successfully\n");

/* "12345thisisatest" with two nameserver*/

nameServers = RRPCreateVector();

RRPAddVectorElement(nameServers, "ns3.dn.net");

RRPAddVectorElement(nameServers, "ns4.dn.net");

response = RRPAddDomain("12345thisisatest.com",
nameServers);

RRPFreeVector(nameServers);

if (response == NULL) {

    RRPPrintInternalErrorDescription();
```

**NSI PROPRIETARY AND CONFIDENTIAL**

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

```
        exit(1);

    }

    else if (response-code != 200) {

        printf("Error registering domain name: %s\n",
            response-description);

        exit(1);

    }

    RRPFreeResponse(response);

    printf("Domain registered successfully\n");


    /* Register New Nameserver */

    ipAddresses = RRPCreateVector();

    RRPAddVectorElement(ipAddresses, "203.23.21.5");

    response = RRPAddNameServer("ns1.12345thisisatest.com",
        ipAddresses);

    RRPFreeVector(ipAddresses);

    if (response == NULL) {

        RRPPrintInternalErrorDescription();

        exit(1);

    }

    else if (response-code != 200) {

        printf("Error registering nameserver: %s\n", response-
            description);

        exit(1);

    }

    RRPFreeResponse(response);

    printf("Nameserver registered successfully\n");
```

**NSI PROPRIETARY AND CONFIDENTIAL**

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

```
/* End Session */

response = RRPEndSession();

if (response == NULL) {

    RRPPrintInternalErrorDescription();

    exit(1);

}

RRPFreeResponse(response);

printf("Closing connection\n");

exit(0);

} /* main() */
```

---

### 3. API Commands

#### 3.1. RRAddDomain

**Description:** This command enables a registrar to register a domain name (on behalf of a registrant) in the system. The request to register a domain name must contain the following data:

- Fully qualified domain name in the “DomainName” attribute
- Optional: One or more to a maximum of 13 fully qualified nameservers hosting the domain name in multiple instances of the “NameServer” attribute. The nameservers must already be registered in the Registry.

The request to renew a domain may contain the following optional parameter:

- An optional registration period for the domain to be added. The registration period  $n$  must be an integer that falls within the range of  $1 \leq n \leq 10$ .

The system will register the domain name for the number of years specified. If no registration period is provided, a default one year registration period will be used. If the domain name is successfully registered, the system will return the registration expiration date in the “RegistrationExpirationDate” attribute in the response.

**Syntax:**

RRPRESPONSE\*RRAddDomain

#### NSI PROPRIETARY AND CONFIDENTIAL

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

```
(char* domainName, RRPVECTOR* nameServers, ...)
```

#### Input:

`domainName` is a pointer to a character string defining a fully qualified domain name to register.

`nameServers` is a pointer to an RRP vector which defines nameservers that will be hosting this domain. NULL indicates that no nameservers host this domain.

`registrationPeriod` is an integer that defines the number of years that the domain will be registered for. It is an optional argument.

**Output:** None.

**Return:** A pointer to an RRPRESPONSE structure containing the components of the RRP response returned from the server. NULL is returned if an internal error occurs.

### 3.2. RRPAddNameServer

**Description:** This command enables a registrar to register a nameserver (on behalf of a registrant) in the system. The request to register a nameserver must contain the following data:

- Fully qualified host name of the nameserver in the “NameServer” attribute
- One or more IP addresses of the nameserver in multiple instances of the “IPAddress” attribute

The registrar must register the nameserver in the system before using it to host domain names. Further, the nameserver must be registered through the same registrar that is the current registrar of its parent domain name. The system will allow any registrar to use the nameserver to host domain names.

#### Syntax:

```
RRPRESPONSE* RRPAddNameServer (  
char* nameServer, RRPVECTOR* ipAddresses)
```

#### Input:

`nameServer` is a pointer to a character string defining a fully qualified host name of a nameserver to registrar.

`ipAddresses` is a pointer to an RRPVector structure defining a list of one or more IP addresses of the nameserver.

**Output:** None.

#### NSI PROPRIETARY AND CONFIDENTIAL

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

**Return:** A pointer to an RRPRESPONSE structure containing the components of the RRP response returned from the server. NULL is returned if an internal error occurs.

Refer to the the *Registry Registrar Protocol, version 1.1.0* for a list of applicable response codes.

### 3.3. RRPCheckDomain

**Description:** This command enables a registrar to check the availability of a domain name in the Registry. The request to check the availability of a domain name must contain the following data:

- Fully qualified domain name in the “DomainName” attribute

The system must respond positively or negatively about the availability of the domain name. A positive response does not guarantee domain name availability at registration time.

**Syntax:**

```
RRPRESPONSE* RRPCheckDomain
```

```
(char* domainName)
```

**Input:** domainName is a pointer to a character string defining a fully qualified domain name to check.

**Output:** None.

**Return:** A pointer to an RRPRESPONSE structure containing the components of the RRP response returned from the server. NULL is returned if an internal error occurs.

Refer to the *Registry/Registrar Protocol, version 1.1.0* for a list of applicable response codes.

### 3.4. RRPCheckNameServer

**Description:** This command enables a registrar to check the availability of a name server in the Registry. The request to check the availability of a name server must contain the following data:

- Fully qualified host name in the “NameServer” attribute

The system must respond positively or negatively about the availability of the domain name. A positive response does not guarantee the name server availability at registration time.

**Syntax:**

```
RRPRESPONSE* RRPCheckDomain
```

```
(char* nameServer)
```

#### NSI PROPRIETARY AND CONFIDENTIAL

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

**Input:** `nameserver` is a pointer to a character string defining a fully qualified name server to check.

**Output:** None.

**Return:** A pointer to an `RRRESPONSE` structure containing the components of the RRP response returned from the server. `NULL` is returned if an internal error occurs. If the name server is not available, then its IP Address will be returned in the `RRRESPONSE` structure as an attribute. The attribute key will be “`ipaddress`”.

Refer to the *Registry Registrar Protocol, version 1.1.0* for a list of applicable response codes.

### 3.5. RRPDeleteDomain

**Description:** This command enables a registrar to delete the registration of a domain name. The request to delete the registration of a domain name must contain the following data:

- Fully qualified domain name in the “`DomainName`” attribute

A request to delete a domain name may cause the deletion of all child nameservers of the domain name to be deleted.

**Syntax:**

```
RRRESPONSE* RRPDeleteDomain  
(char* domainName);
```

**Input:** `domainName` is a pointer to a character string defining a fully qualified domain name to delete.

**Output:** None.

**Return:** A pointer to an `RRRESPONSE` structure containing the components of the RRP response returned from the server. `NULL` is returned if an internal error occurs.

### 3.6. RRPDeleteNameServer

**Description:** This command enables a registrar to delete the registration of a nameserver. The request to delete a nameserver must contain the following data:

- Fully qualified host name of the nameserver in the “`NameServer`” attribute

The system does not allow the deletion of a nameserver if domain names are currently hosted on it.

**Syntax:**



#### NSI PROPRIETARY AND CONFIDENTIAL

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

RRPRESPONSE\* RRPDeleteNameServer

(char\* nameServer);

**Input:** nameServer is a pointer to a character string defining a fully qualified host name of the nameserver to delete.

**Output:** None.

**Return:** A pointer to an RRPRESPONSE structure containing the components of the RRP response returned from the server. NULL is returned if an internal error occurs.

Refer to the *Registry Registrar Protocol, version 1.1.0* for a list of applicable response codes.

### 3.7. RRPDescribe

**Description:** This command enables a registrar to get general information about RRP.

The request to get general information about RRP may contain the following command options:

- The “Target” option set to value “Protocol” to get the protocol version number

The system, by default, must return the protocol version number, irrespective of whether or not the request contains the “Target” option.

**Syntax:**

RRPRESPONSE\* RRPDescribe

(char\* target);

**Input:** target is a pointer to a character string defining an optional target indicating the type of information returned. If NULL, protocol version returns in the RRPRESPONSE structure.

**Output:** None.

**Return:** A pointer to an RRPRESPONSE structure containing the components of the RRP response returned from the server. NULL is returned if an internal error occurs.

Refer to the *Registry Registrar Protocol, version 1.1.0* for a list of applicable response codes.

### 3.8. RRPMModifyDomain

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

- Add an attribute value. The value added must be unique among the values of the

**NSI PROPRIETARY AND CONFIDENTIAL**

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

attribute. For a single-valued attribute, it replaces the current value.

- Remove an attribute value. The value removed must exist. Further, an attribute value cannot be removed if it is the only value of a required attribute.
- Replace an attribute value. The value replaced must exist and the replacing value must be unique among the values of the attribute.

The request to update a registered domain name must contain the following data:

- Fully qualified domain name in the “DomainName” attribute

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

- Update the nameservers of the domain name by setting one or more instances of the “Nameserver” attribute.
- Update the status of a domain by setting one or more instances of the “Status” attribute.

**Syntax:**

```
RRRESPONSE* RRPMModifyDomain
(char* domainName,
RRPVECTOR* addedNameServers,
RRPPROPERTIES* modifiedNameServers,
RRPVECTOR* deletedNameServers,
RRPVECTOR* addedStatuses,
RRPPROPERTIES* modifiedStatuses,
RRPVECTOR* deletedStatuses)
```

**Input:**

`domainName` is a pointer to a character string defining a fully qualified domain name to update.

`AddedNameServers` is an optional list of nameserver names to add to the domain. NULL indicates that there are no nameservers to add.

`ModifiedNameServers` is an optional list containing nameservers to replace. Each property 'key' indicates the old nameserver to replace, and the 'value' indicates the new nameserver with which to replace it. NULL indicates that there are no nameservers to replace.

`deletedNameServers` is an optional list of nameserver names to delete from the domain.

#### NSI PROPRIETARY AND CONFIDENTIAL

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

NULL indicates that there are no nameservers to delete.

AddedStatuses is an optional list of domain statuses to add to the domain. NULL indicates that there are no statuses to add.

modifiedStatuses is an optional list containing statuses to replace. Each property 'key' indicates the old status to replace, and the 'value' indicates the new status with which to replace it. NULL indicates that there are no statuses to replace.

DeletedStatuses is an optional list of domain statuses to delete from the domain. NULL indicates that there are no statuses to delete.

**Output:** None.

**Return:** A pointer to an RRPRESPONSE structure containing the components of the RRP response returned from the server. NULL is returned if an internal error occurs.

### 3.9. RRPMModifyNameServer

**Description:** This command updates the attributes of a registered nameserver. The command allows the following operations on an attribute value for both single-valued and multi-valued attributes:

- Add an attribute value. The value 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 removed must exist. Further, an attribute value cannot be removed if it is the only value of a required attribute.
- Replace an attribute value. The value replaced must exist and the replacing value must be unique among the values of the attribute.

The request to update a nameserver must contain the following data:

- Fully qualified host name of the nameserver in the "NameServer" attribute

The registrar can perform the following update operations on the nameserver:

- Update the Nameserver attribute of the nameserver.
- Update the IP addresses of the nameserver by setting one or more instances of the "IPAddress" attribute.

#### Syntax:

```
RRPRESPONSE* RRPMModifyNameServer  
  
(char* nameServer, char* newNameServer,
```

**NSI PROPRIETARY AND CONFIDENTIAL**

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

```
RRPVECTOR* addedIPAddresses,  
RRPPROPERTIES* modifiedIPAddresses,  
RRPVECTOR* deletedIPAddresses);
```

**Input:**

`nameServer` is a pointer to a character string defining the fully qualified host name of a nameserver to update.

`newNameServer` is a pointer to a character string defining an optional new name for nameserver. NULL indicates that the name of the nameserver will remain as is.

`AddedIPAddresses` is an optional list of IP addresses to add to nameserver. NULL indicates that there are no IP addresses to add.

`ModifiedIPAddresses` is an optional list containing IP addresses to replace. Each property 'key' indicates the old address to replace, and the 'value' indicates the new address with which to replace it. NULL indicates that there are no IP addresses to replace.

`DeletedIPAddresses` is an optional list of IP addresses to delete from nameserver. NULL indicates that there are no IP addresses to delete.

**Output:** None.

**Return:** A pointer to an RRPRESPONSE structure containing the components of the RRP response returned from the server. NULL is returned if an internal error occurs.

Refer to the the *Registry Registrar Protocol, version 1.1.0* for a list of applicable response codes.

### 3.10. RRPRenewDomain

**Description:** This command enables a registrar to re-register the domain name in the system. The request to renew a domain name must contain the following data:

- Fully qualified domain name in the "DomainName" attribute

The request to renew a domain may contain the following optional parameters. Please note that if either of these options are provided, then both are required.

- An optional re-registration period. The renewal period  $n$  must be an integer that falls within the range of  $1 \leq n \leq 10$ .
- An optional current expiration year. The expiration year must be a four-digit integer.

The system re-registers the domain name for specified number of years from year 'current expiration year'. System treats re-registration period and current expiration years as an optional

**NSI PROPRIETARY AND CONFIDENTIAL**

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

parameter pair, that is, if one is provided, the other one needs to be provided, also. If the renewal period and current expiration year are not provided, a default one year renewal period is used. If the domain name re-registration completes successfully, the system will return the new registration expiration date in the “RegistrationExpirationDate” attribute in the response.

**Syntax:**

RRPRESPONSE\* RRPRenewDomain

(char\* domainName, ...);

**Input:**

- domainName is a pointer to a character string defining the fully qualified domain name to renew.
- RenewRegistrationPeriod is an integer defining the number of years that the domain will be registered for.
- currentExpirationYear is an integer defining the current expiration year.

**Output:** None.

**Return:** A pointer to an RRPRESPONSE structure containing the components of the RRP response returned from the server. NULL is returned if an internal error occurs.

Refer to the *Registry Registrar Protocol, version 1.1.0* for a list of applicable response codes.

### 3.11. RRPSessionStart

**Description:** This command enables a registrar to create an RRP session and optionally modify their password.

The request to authenticate an RRP session must contain the following function parameters:

- ID of the registrar
- Password of the registrar

The request to authenticate an RRP session may contain the following optional parameter:

- A new password for the registrar (optional)

The registrar must send this request before any other request to the Registry. If the authentication fails, the registrar may re-send this request with corrected information. If the authentication fails a second time, the Registry closes the connection.

**Syntax:**

#### NSI PROPRIETARY AND CONFIDENTIAL

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

RRPRESPONSE\* RRPStartSession

```
(char* registrarID, char* registrarPassword, ...);
```

#### Input:

registrarID is a pointer to a character string defining the registrar's ID.

registrarPassword is a pointer to a character string defining the registrar's password.

... is a third optional pointer to a character string defining the registrar's new password to be set.

**Output:** None.

**Return:** A pointer to an RRPRESPONSE structure containing the components of the RRP response returned from the server. NULL is returned if an internal error occurs.

### 3.12. RRPEndSession

**Description:** This command closes an RRP connection.

#### Syntax:

RRPRESPONSE\* RRPEndSession

```
(void);
```

**Input:** None.

**Output:** None.

**Return:** A pointer to an RRPRESPONSE structure containing the components of the RRP response returned from the server. NULL is returned if an internal error occurs.

Refer to the the *Registry Registrar Protocol, version 1.1.0* for a list of applicable response codes.

### 3.13. RRPStatusDomain

**Description:** This command enables a registrar to query a domain name. The request to query a domain name must contain the following data:

- Fully qualified domain name in the "DomainName" attribute

The response from the system must contain the following data:

- Fully qualified domain name in the "DomainName" attribute
- Registration expiration date in the "RegistrationExpirationDate" attribute

#### NSI PROPRIETARY AND CONFIDENTIAL

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

- 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 “RegistrarTransferDate” attribute
- Current status of the domain name in the “Status” attribute
- Date the domain name was registered in the “CreatedDate” attribute
- ID of the registrar that registered the domain name in the “CreatedBy” attribute
- Date the domain name was last updated in the “UpdatedDate” attribute
- ID of the entity that last updated the domain name in the “UpdatedBy” attribute

#### Syntax:

```
RRRESPONSE* RRPStatusDomain  
  
(char* domainName);
```

**Input:** domainName is a pointer to a character string defining the fully qualified domain name to query.

**Output:** None.

**Return:** A pointer to an RRRRESPONSE structure containing the components of the RRP response returned from the server. NULL is returned if an internal error occurs.

### 3.14. RRPStatusNameServer

**Description:** This command enables a registrar to query a nameserver. The request to query a nameserver must contain the following data:

- Fully qualified name of the nameserver in the “NameServer” attribute

The response from the system must contain the following data:

- Fully qualified name of the nameserver in the “NameServer” attribute
- IP addresses of the nameserver in multiple instances of the “IPAddress” attribute
- ID of the current registrar of the nameserver in the “Registrar” attribute
- Date the nameserver was registered in the “CreatedDate” attribute
- ID of the entity that registered the nameserver in the “CreatedBy” attribute
- Date the nameserver was last updated in the “UpdatedDate” attribute

#### NSI PROPRIETARY AND CONFIDENTIAL

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

- ID of the entity that last updated the nameserver in the “UpdatedBy” attribute
- If the “ListDomains” option is set to “Yes”, a list of domain names hosted by the nameserver returns.

#### Syntax:

```
RRRESPONSE* RRPStatusNameServer
```

```
(char* nameServer);
```

**Input:** nameServer is a pointer to a character string defining the fully qualified host name of the nameserver to query.

**Output:** None.

**Return:** A pointer to an RRPRESPONSE structure containing the components of the RRP response returned from the server. NULL is returned if an internal error occurs.

Refer to the *Registry Registrar Protocol, version 1.1.0* for a list of applicable response codes.

### 3.15. RRPTTransferDomain

**Description:** This command enables a registrar to transfer a domain name from another registrar to itself. A registrar can also transfer a domain that it has registered to itself. The request to transfer a domain name must contain the following data:

- The “EntityName” attribute set to value “Domain”
- Fully qualified domain name in the “DomainName” attribute

Nameservers are implicitly transferred when their parent domain name is transferred.

If a registrar is requesting to transfer a domain from the current registrar to itself, the system will block any update or deletion requests to the transferred domain name and all nameservers transferred with it for a period of five calendar days. In addition, the system will notify both the old registrar and the new registrar by email of the transfer request. The registration expiration date of the transferred domain name is set to the transfer date plus two years.

The current registrar can also request a domain to be transferred to itself. This transfer request will be immediately executed with no email notifications sent to the requesting registrar.

#### Syntax:

```
RRRESPONSE RRPTTransferDomain
```

```
(char* domainName, char* approve);
```



**NSI PROPRIETARY AND CONFIDENTIAL**

This information includes trade secrets and confidential commercial and/or financial information belonging to Network Solutions, Inc. (NSI). It is exempt from disclosure under the Freedom of Information Act. Unauthorized disclosure and/or use of this information without the express written consent of Network Solutions, Inc. is prohibited and may result in criminal prosecution and penalties pursuant to 18 U.S.C. § 1905.

©Network Solutions, Inc. (Unpublished Work)

---

**Input:**

`domainName` is a pointer to a character string defining the domain name to be transferred.

`approve` is a pointer to a character string specifying whether the request for a domain transfer is approved or denied by the old registrar. If approving the request of another registrar to transfer one of your domains, pass “yes”. If denying the request of another registrar to transfer one of your domains, pass “no”. If this parameter is NULL, then the transfer command specifies a request to transfer the `domainName`.

**Output:** None.

**Return:** A pointer to an RRPRESPONSE structure containing the components of the RRP response returned from the server. NULL is returned if an internal error occurs.

Refer to the the *Registry Registrar Protocol, version 1.1.0* for a list of applicable response codes.