IEN 116

J. Postel
ISI
August 1979

Obsoletes: 89, 61

INTERNET NAME SERVER

INTRODUCTION

This memo defines the procedure to access an Internet Name Server. Such a server provides the actual addresses of hosts in the internet when supplied with a host name. An Internet Name Server is a dynamic name-to-number translation service.

This server utilizes the User Datagram Protocol (UDP) [2], which in turn calls on the Internet Protocol (IP) [3].

NAME SYNTAX

It is strongly recommended that the use of host names in programs be consistent for both input and output across all hosts. To promote such consistency of the internet level, the following syntax is specified:

The SYNTAX of names as presented to the user and as entered by the user is:

! NET ! REST

where:

NET is a network name or number as defined in "Assigned Numbers" [1]

and

REST is a host name within that network expressed as a character string or as a number. When a number is used, it is expressed in decimal and is prefixed with a sharp sign (e.g., #1234).

Note that this syntax has minimal impact on the allowable character strings for host names within a network. The only restriction is that a REST string cannot begin with an exclamation point (!).

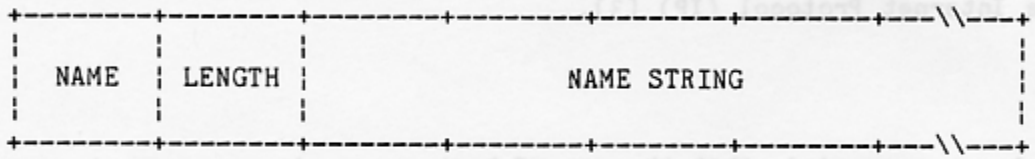
The !NET! may be omitted when specifying a host in the local network. That is "!" indicates the network portion of a name string.

BASIC NAME SERVER

To aid in the translation of names to internet addresses, several name server processes will be provided. The name server process will accept a name in the above form and will return a name, address pair.

The name server processes will have well-known addresses; addresses that are constant over long periods of time and published in documents such as "Assigned Numbers" [1].

A request sent to a name server is sent as a user datagram [2] with the following content:



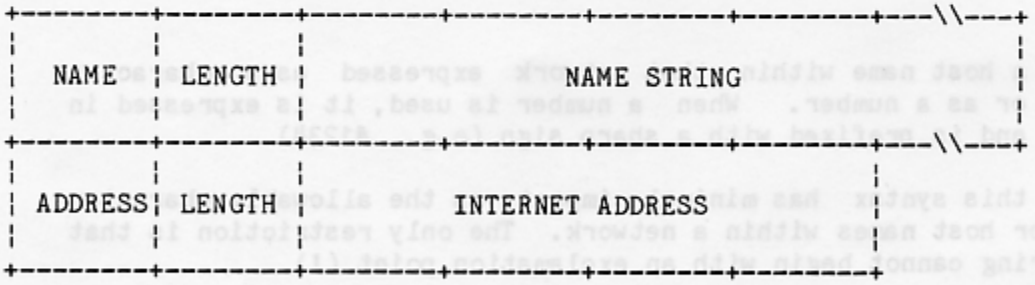
where:

NAME is a one octet code indicating that the following is a name,

LENGTH is a one octet count of the number of octets in the name string, and

NAME STRING is an ASCII character string of the form ! NET ! REST.

A reply to a successful translation is sent as a user datagram with the following content:



where:

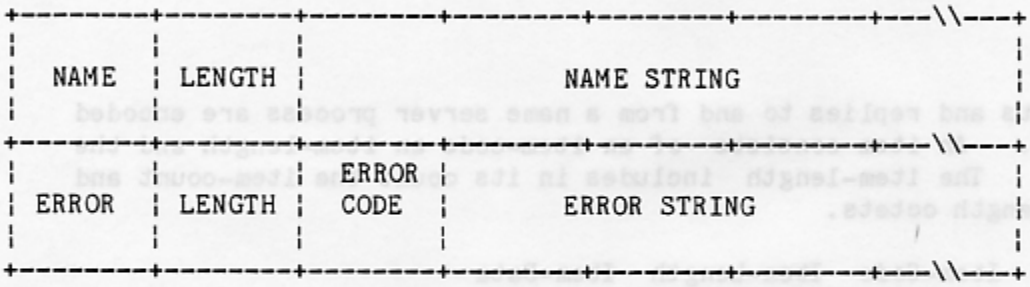
ADDRESS is a one octet code indicating that the following is an internet address,

LENGTH is a one octet count (=4) of the length of the internet address, and

INTERNET ADDRESS is the internet address.

Actually a particular name might map to several internet addresses, in this case the response would include a list of internet addresses.

When a name is not found, an error is reported via a user datagram as follows:



where:

ERROR CODE specifies the error.

ERROR STRING explains the error.

Error Codes

The following error codes are defined:

CODE	MEANING
0	Undetermined or undefined error
1	Name not found
2	Improper name syntax

Communication with a Name Server Process

Communication with a name server process is via user datagrams. User datagrams do not guarantee reliable communication. Thus, some requests or replies may be lost.

The name server process is a transaction oriented process; furthermore, the nature of the transactions allows them to be processed in any order and even to be duplicated. This allows the use of a very simple communication protocol.

If a request is made to the name server process and no response is received within a reasonable time, then the requester should make the request again. This recovers from communication errors which cause the loss of either the request or the reply.

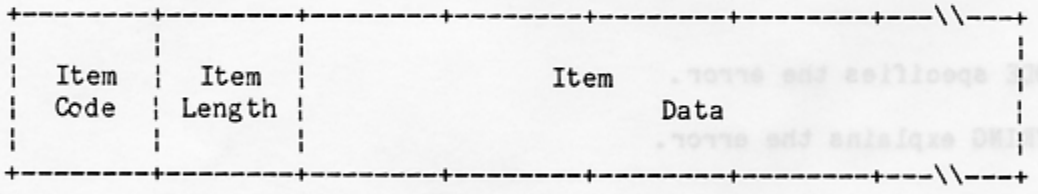
In order to use this simple strategy, care must be taken to allow replies to be properly matched with requests. The name server process does this by including in each reply a copy of the entire request.

The user datagram protocol does provide a checksum for the detection of errors.

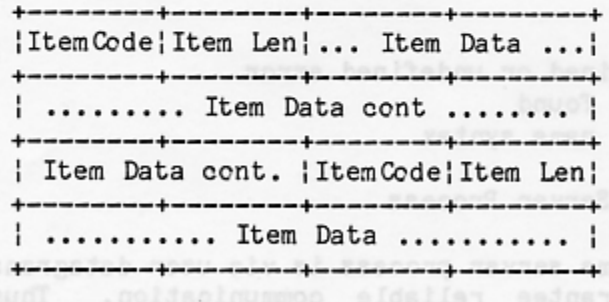
Format

The requests and replies to and from a name server process are encoded as "items". An item consists of an item-code an item-length and the item-data. The item-length includes in its count the item-count and the item-length octets.

Item := Item-Code Item-Length Item-Data



A request is typically one item, and a reply is typically two items.



Item Code Value Assignments:

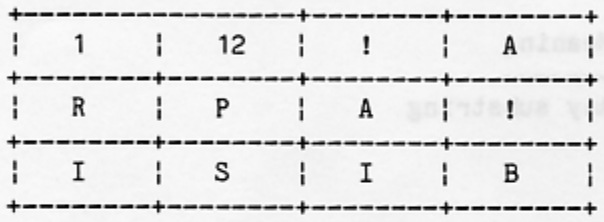
NAME = 1

ADDRESS = 2

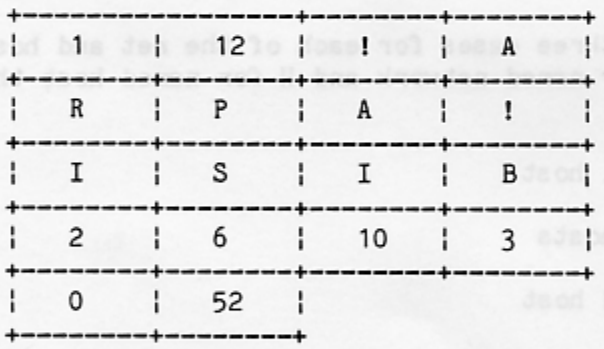
ERROR = 3

Example

a typical request:



and the reply:



EXTENDED NAME SERVER

Several extensions have been proposed [4], the following two are adopted: partially specified names, and a service field.

In the first extension partially specified names are allowed and are indicated by the use of "wild card" fields or characters.

Wild Card Field	Meaning
*	All
~	Local (Same as that of the requestor)

Wild Card Character	Meaning
*	Any substring

Examples:

!~!* all hosts on the net of the requestor.

!*!SRI* all hosts with names whose first three characters are SRI on all nets

In general, there are three cases for each of the net and host fields. Using the symbols N for named network and H for named host the 9 cases are:

!~!~ local net, local host

!~!* local net, all hosts

!~!H local net, named host

!*!~ all nets, local host

!*!* all nets, all hosts

!*!H all nets, named host

!N!~ named net, local host

!N!* named net, all hosts

!N!H named net, named host

When such a request is processed and the result is more than one name/address pair, the response is all the pairs.

Examples:

1)

request:

```
!ARPA!ISI*
```

response:

```
!ARPA!ISIA  10 1 0 22

!ARPA!ISIB  10 3 0 52

!ARPA!ISIC  10 2 0 22

!ARPA!ISID  10 3 0 22

!ARPA!ISIE  10 1 0 52
```

2)

request:

```
!~!SRI-R2D2
```

response:

```
!ARPA!SRI-R2D2    10 3 0 51

!SF-PR-1!SRI-R2D2  2 0 0 11
```

3)

request:

```
!*!ISIA
```

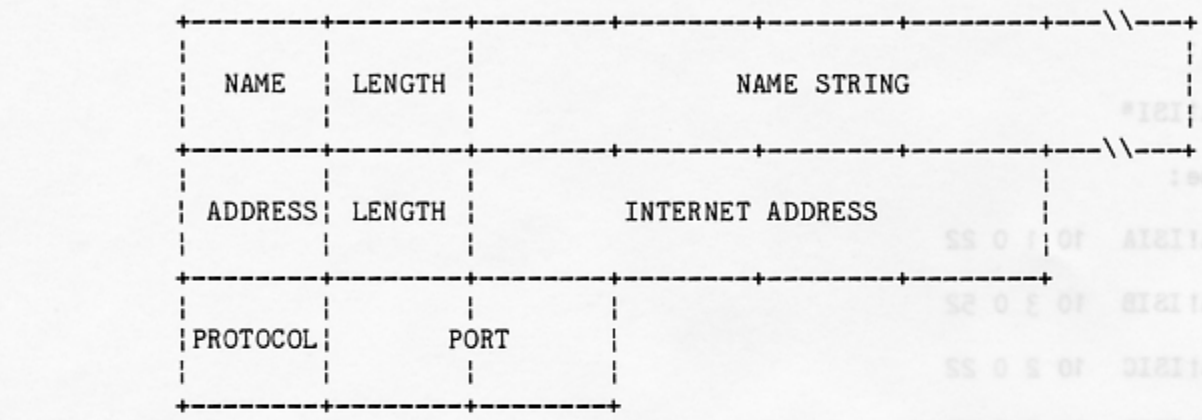
response:

```
!ARPA!ISIA  10 1 0 22
```

The second extension is that a third field may be appended to the name. This is the SERVICE field.

```
! NET ! HOST ! SERVICE
```

To reply to a request of this form the name server must provide the internet address (net and host), the protocol number, and the port number.



Examples:

1)

request:

```
!ARPA!ISIA!TELNET
```

response:

```
!ARPA!ISIA!TELNET  10 1 0 22 6 0 23
```

2)

request:

```
!ARPA!*!NAME-SERVER
```

response:

```
!ARPA!SRI-KL!NAME-SERVER  10 1 0 2 17 42
```

References

[1] J. Postel. "Assigned Numbers," IEN 117, USC/Information Sciences Institute, August 1979.

[2] J. Postel. "User Datagram Protocol," IEN 88, USC/Information Sciences Institute, May 1979.

[3] J. Postel. "Internet Protocol," IEN 111, USC/Information Sciences Institute, August 1979.

[4] J. Pickens, E. Feinler, and J. Mathis. "The NIC Name Server -- A Datagram Based Information Utility," Proceedings of the Fourth Berkeley Conference on Distributed Data Management and Computer Networks, pp. 275-283, August 1979.

Acknowledgments

John Pickens contributed the ideas for the Extended Name Server.