DNS

The DNS module implements the Domain Name Service client-side protocol, running on top of UDP. It provides an API for machine-name to IP address resolution. The module itself is machine independent and is configured through the *COMMAND* message.

Process Information

Prototype Name	dns
Link Order	does not matter
Process Name	“dns”

Configuration Commands

The DNS process is configured using two strings passed to it as *COMMAND* messages.

dns

dns *i1.i2.i3.i4*

The *dns* command specifies the IP address of the remote DNS server which will respond to queries from this machine. When the process receives the address of its server, it opens the UDP port and sets up a list of receive requests.

Example: “dns 138.15.103.81”

prefix

prefix *string*

The *prefix* command provides the domain prefix to be appended to simple names in order to generate fully-qualified domain names.

Example: “prefix nec-lab.com”

Process Operation

The module has only a main process. It accepts the *RESOLVE* message from the *NETINFO* messageset, and uses the *Standard* messageset internally to communicate with the UDP layer.

GETMBLK replies contain the responses to DNS resolve requests from the server. If the packet contents are a resolve reply, the corresponding request is matched in the list of outstanding queries and the results returned to the user, including any error code returned from the server. If the request was successful, the result is also added to the local DNS cache. DNS requests directed at this machine are ignored.

PUTMBLK messages are issued downwards containing DNS resolve requests. The replies are used used to update the request state, indicating that the packet has left the local machine.

RESOLVE messages are used by clients to request name-to-address translations. If the name string passed in to the process does not already contain a '.', the local prefix is added to the end of the name. The DNS process keeps a cache of known names, and values are returned from there if possible. Otherwise, a DNS resolve request is created and sent to the local server.

TIMEOUT messages are used to trigger re-transmissions of resolve requests when a reply has not been received. If a reply has not been received after multiple retransmission, a 'not found' error is passed back to the user.

Shared Library Macros and Routines

dns_resolve

int *dns_resolve*(
char **name*,
uint **where*)

The *dns_resolve* routine sets *where* to the IP address of the machine (interface) supplied in the string *name.* The routine returns zero if the name was successfully resolved, and a nono-zero error code otherwise. The routine formats and sends a *RESOLVE* message to the DNS process and waits for the reply, and so can only be used in a process context.