NetInfo MessageSet

The *NetInfo* messageset contains a set of messages used to pass information between the layers of the network stack. The reason this information is carried in messages, rather than through shared-library calls is that one or other endpoint of the message is an optional module, or a class or modules (for example an ethernet interface driver).

Message Definition

NETADDRESS

typedef struct
{

uchar	*mac*[6]	MAC-layer address
int	*events_generated*	flag for this driver

}**ROME_T_NETADDRESS;**

The *NETADDDRESS* message is sent to a interface driver to request its low-level MAC address (usually by the Ethernet ARP process), which is returned in the *mac* field. The *events_generated* field is set to *TRUE* if the driver generates events for this interface (for example PCMCIA card removal events). The message should be sent through a *FILE* interface, as the *dest_context* field in the message may be used by the driver to identify which of a number of interfaces is being requested

NIS_MATCH

typedef struct
{

char	**domain*	NIS domain name
char	**map*	NIS map name
char	**key*	key in map
char	**result*	output buffer pointer
int	*rlen*	length of returned data

}**ROME_T_NIS_MATCH;**

The *NIS_MATCH* message requests an NIS map lookup. The receiving process issues a lookup request for *key* in *map* in the NIS domain *domain.* If the *domain* is *NULL* or zero-length, the default configured domain is used. The result is returned in the *result* buffer and *rlen* is set to the number of valid bytes in the result. It is assumed that the supplied buffer is large enough to hold the data.

The *NIS* module further describes this processing, and the configuration of the NIS client. The *nis_match* routine in that module is the usual API for this message.

RESOLVE

```
typedef struct
{
        char        *name       domain name
        uint        ipaddr      returned IP address
}ROME_T_RESOLVE;
```

The *RESOLVE* message requests a name-to-IP address mapping from the Domain Name Service The *name* parameter shuld point to a machine name (either fully-qualified or a simple name within the local domain). The result is returned in the *ipaddr* field in the reply (if the replied errno is 0). The *dns_resolve* routine in the *DNS* module is the usual API for this message.