NIS

The NIS module implements the Network Information Service (which used to be called YP) client-side protocol, running on top of RPC. It provides an API for matching keyword in NIS maps according to the configurd services in the local environment. The module itself is machine independent and is configured through the *COMMAND* message.

Process Information

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

Configuration Command

The NIS process is configured using a string passed to it as a *COMMAND* message.

domain

domain *name servername*

The *domain* command specifies the NIS domain within which map queries should be resolved, and the name of the machine to which the queries should be sent. When the process receives the name of its server, it uses *DNS* to resolve the IP address and the *portmapper* function of *RPC* to obtain the UDP port of the remote service. It then opens an RPC_FILE data path to the remote service.

Example: “domain necusa hera.nec-lab.com”

Process Operation

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

NIS_MATCH messages are used by clients to request key lookups. The data from the message are converted into a remote procedure call which is sent to the configured server. The data returned in the rpc reply (if valid) are copied into the buffer supplied by the application before the reply is sent upwards.

TIMEOUT messages are used to trigger re-transmissions of match 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

nis_match

int *nis_match*(
    char **domain*,
    char **map*,
    char **key*,
    uchar **result*,
    int **rlen*)

The *nis_matcg* routine sets *result* to the contents of the entry matching *key* in the NIS map named *map* in the *domain* domain. If the *domain* value is *NULL* or a zero-length string, the default domain specified on the *domain* command is used. *rlen* is set to the length of the valid portion of the *result* array. The routine returns zero if the key was successfully matched, and a no-zero error code otherwise. The routine formats and sends a *NIS_MATCH* message to the NIS process and waits for the reply, and so can only be used in a process context.