NAME

nsdispatch **-** name-service switch dispatcher routine

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sys/types.h>
#include <stdarg.h>
#include <nsswitch.h>

int
nsdispatch(*void *retval*, *const ns_dtab dtab[]*, *const char *database*, *const char *method_name*,
  const ns_src defaults[], *...*);

DESCRIPTION

The **nsdispatch**() function invokes the methods specified in *dtab* in the order given by nsswitch.conf(5) for the database *database* until a successful entry is found.

retval is passed to each method to modify as necessary, to pass back results to the caller of **nsdispatch**().

Each method has the function signature described by the typedef:

typedef int **(*nss_method)**(*void *retval*, *void *mdata*, *va_list *ap*);

dtab is an array of *ns_dtab* structures, which have the following format:

```
typedef struct _ns_dtab {
        const char      *src;
        nss_method       method;
        void            *mdata;
} ns_dtab;
```

The *dtab* array should consist of one entry for each source type that is implemented, with *src* as the name of the source, *method* as a function which handles that source, and *mdata* as a handle on arbitrary data to be passed to the method. The last entry in *dtab* should contain NULL values for *src*, *method*, and *mdata*.

Additionally, methods may be implemented in NSS modules, in which case they are selected using the *database* and *method_name* arguments along with the configured source. Modules must use source names different from the built-in ones.

defaults contains a list of default sources to try if nsswitch.conf(5) is missing or corrupted, or if there is no relevant entry for *database*. It is an array of *ns_src* structures, which have the following format:

```
typedef struct _ns_src {
        const char      *src;
        uint32_t    flags;
} ns_src;
```

The *defaults* array should consist of one entry for each source to be configured by default indicated by *src*, and *flags* set to the criterion desired (usually NS_SUCCESS; refer to *Method return values* for more information). The last entry in *defaults* should have *src* set to NULL and *flags* set to 0.

For convenience, a global variable defined as:

```
extern const ns_src __nsdefaultsrc[];
```

exists which contains a single default entry for the source 'files' that may be used by callers which do not require complicated default rules.

'...' are optional extra arguments, which are passed to the appropriate method as a variable argument list of the type *va_list*.

Valid source types

While there is support for arbitrary sources, the following #defines for commonly implemented sources are available:

#define	value
NSSRC_FILES	"files"
NSSRC_DB	"db"
NSSRC_DNS	"dns"
NSSRC_NIS	"nis"
NSSRC_COMPAT	"compat"

Refer to nsswitch.conf(5) for a complete description of what each source type is.

Method return values

The *nss_method* functions must return one of the following values depending upon status of the lookup:

Return value	Status code
NS_SUCCESS	

	success
NS_NOTFOUND	notfound
NS_UNAVAIL	unavail
NS_TRYAGAIN	tryagain
NS_RETURN	-none-

Refer to nsswitch.conf(5) for a complete description of each status code.

The **nsdispatch**() function returns the value of the method that caused the dispatcher to terminate, or NS_NOTFOUND otherwise.

NOTES

FreeBSD’s Standard C Library (libc, -lc) provides stubs for compatibility with NSS modules written for the GNU C Library **nsswitch** interface. However, these stubs only support the use of the "passwd" and "group" databases.

SEE ALSO

hesiod(3), stdarg(3), nsswitch.conf(5), yp(8)

HISTORY

The **nsdispatch**() function first appeared in FreeBSD 5.0. It was imported from the NetBSD Project, where it appeared first in NetBSD 1.4. Support for NSS modules first appeared in FreeBSD 5.1.

AUTHORS

Luke Mewburn <*lukem@netbsd.org*> wrote this freely-distributable name-service switch implementation, using ideas from the ULTRIX svc.conf(5) and Solaris nsswitch.conf(4) manual pages. The FreeBSD Project added the support for threads and NSS modules, and normalized the uses of **nsdispatch**() within the standard C library.