EDOC090

EUROGAM PROJECT

NSF DATA ACQUISITION SYSTEM

Message Handling System - C library

Edition 1.0
Sept 1991

NSF Software Systems Group
UK Science and Engineering Research Council
Daresbury Laboratory

Introduction

This document describes the C interface library for the Message Handling System forming part of the NSF data acquisition system.

The library implements the specification described in the document *Message Handling System* (EDOC089).

In order to use the library, UNIX programs should include the header file *msg.h* and should be linked with the library *libmsg.a*. Some of the functions are also available to VxWorks applications in which case the file *message.o* should be loaded on to the system.

Summary

The library consists of a number of C procedure calls, each one corresponding to a procedure of the Message Logger.

The library expects the servers to use the UDP ports specified in EDOC089. It does not call the server's portmapper to discover the port.

Description

Returns

The library calls will return the value OK (0) or a positive value, if successful, or ERROR (-1) if not. In the case of ERROR the procedure status is copied to the global error status *errno*. A short message describing the error condition can be obtained by calling *msgPerror*.

Message Handler System Calls

int msgNullProc ()

This routine does no work. It calls the server's null procedure. This is a procedure that is made available by all RPC servers to allow server testing and timing.

This routine is available for VxWorks applications.

int msgDefineMessage (id, class, level, source, body)

u_int id;

u_int class;

u_int level;

char *source;

char *body;

This routine sends an error report to the to the central message server.

Returns OK or ERROR.

This routine is available for VxWorks applications.

int msgSignOn (name, list)

char *name

MSG_LIST *list

This routine requests the central message server to forward messages to the system specified by *name*. If *name* is a null pointer or points to a null string then the client system is assumed. *list* is a null terminated linked list of message specifications defining which messages are to be passed on.

Returns OK or ERROR.

int msgSignOff (name)

char *name

This routine requests the central message server not to forward any more messages to the system specified by *name*.

Returns OK or ERROR.

Miscellaneous Routines

void msgInstall ()

This routine installs a server to receive messages forwarded by the central message server. It must be called before *msgSignOn*. The installed server prints any recieved messages to *stdout* in the same format as the message log file.

int msgInitialise (host)

char *host;

This routine initialises the C library to use the message logger provided by the system identified by *host*. If host is a NULL pointer the system will be initialised using the default host. This routine is called automatically by the other library calls so there should be no need to make an explicit call unless an unusual host is being used as the message logger.

This routine is available for VxWorks applications.

char *msgPerror (string)

char *string;

This routine produces a short error message on the standard error describing the error referred to by *errno*. If *string* is not a NULL pointer and does not point to a null string, the string it points to is printed, followed by a colon, followed by a space, followed by the message. If *string* is a NULL pointer or points to a null string, just the message is printed, followed by a NEWLINE. The message consists of the words "Message Logging System error " followed by *errno*, followed by a NEWLINE, followed by a description of the error. *Errno* is expected to be an error that was previously returned by one of the calls described above.

char *msgVersion ()

This routine returns the current library version string.

char *msgSeverity (level)

u_int level;

This routine returns a pointer to a text string corresponding to the severity level *level*.