EDOC092

EUROGAM PROJECT

NSF DATA ACQUISITION SYSTEM

Spectrum Access Server - C library

Edition 1.0
October 1991

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

Introduction

This document describes the C interface library for the Spectrum Access Server (SAS) forming part of the NSF data acquisition system.

The library implements the SAS specification described in the document *Spectrum Access Server* (EDOC078).

In order to use the library, programs should include the header file *sas.h* and should be linked with the libraries *libsas.a*.

Summary

The library consists of a number of C procedure calls, each one corresponding to a procedure of the Spectrum Access Server, as described in the *Spectrum Access Server* document. In addition there are miscellaneous routines for setting up and message printing.

The library must be given the port it is to use when establishing the connection. It does not call the server's portmapper to discover the port.

Description

Returns

The spectrum access server calls will return the value OK (0) or a positive value, if sucessfull, or ERROR (-1) if not. In the case of ERROR the procedure status (see EDC078) is copied to the global error status *errno*. A short message describing the error condition can be obtained by calling *sasPerror*. *sasGenerateErrorReport* may be called for a slightly more detailed explanation of the error.

Spectrum Access Server Calls

int sasNullProc ()

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.

int sasAuthorise (client, username, password, mode, capability)

CLIENT *client;

char *username;

char *password;

u_int mode;

char *capability;

This routine authorises the user specified by the string *username* to access the Spectrum Access Server with the read/write permissions given by *mode*.

The server generated capability is copied to the user supplied buffer pointed to by *Capability*. This will act as opaque access token for subsequent procedure requests. This routine must be used to start a session with the Spectrum Access Server.

Returns OK or ERROR.

Works for both VxWorks and UNIX. Will not work for more than 256 users.

int sasCreateSpectrum (client, capability, path, spectrum)

CLIENT *client;

char *capability;

char *path;

SPECTRUM *spectrum;

This routine creates a new spectrum with name *path* whose description is given by *spectrum*.

Returns OK or ERROR.

Works for both VxWorks and UNIX. (Tested with 1D spectra only)

Warning: Floating-point not implemented. Half-arrays not implemented. Values for *base* and *range* are not validated.

int sasDeleteSpectrum (client, capability, path)

CLIENT *client;

char *capability;

char *path;

This routine removes the spectrum file *path* from the system.

Returns OK or ERROR.

Works for both VxWorks and UNIX versions.

int sasLookupSpectrum (client, capability, path, spectrum)

CLIENT *client;

char *capability;

char *path;

SPECTRUM *spectrum;

This routine returns the header information for the spectrum *path*. The result is copied to the user supplied buffer *spectrum*, which will be large enough to hold the information.

Returns OK or ERROR.

Works for both VxWorks and UNIX versions.

int sasReadString (client, capability, path, stype, snumber, req_string)

CLIENT *client;

char *capability;

char *path;

u_int stype;

u_int snumber;

char *req_string;

This routine returns in the user-supplied buffer *req_string* the string held in the spectrum file *path* whose type and number are *stype* and *snumber* respectively..

Returns OK or ERROR.

int sasWriteString (client, capability, path, stype, snumber, data)

CLIENT *client;

char *capability;

char *path;

u_int stype;

u_int snumber;

char *data;

This routine stores the string in *data* whose type and number are *stype* and *snumber* in the spectrum file *path*.

Returns OK or ERROR.

int sasReadNames (client, capability, path, cookie, count, spnames)

CLIENT *client;

char *capability;

char *path;

u_int cookie;

u_int count;

SPECLIST **spnames;

This routine returns in *spnames* a list of all spectra and subdirectories for the path *path*. For an online server, this will be a list of all spectra known to that server.

Retuns OK or ERROR.

Names unsorted.

int sasReadSpectrum (client, capability, path, number, offset, count, base, range, size, type, spectrum, flag)

CLIENT *client;
char *capability;
char *path;
u_int number;
u_int offset;
u_int count;
BASE base;
RANGE range;
SIZE size;
u_int type;
SPECDATA *spectrum;
int *flag;

This routine returns the part of array *number* within the region defined by *base* and *range* of the spectrum *path*. An offset *offset* into this array may be given. The data is scaled as necessary to the size given by *size* and is retyped to the type given by *type*. The array is returned in the user-supplied buffer, which must be of sufficient size, pointed to by *spectrum*. In the event of an overflow or truncation error, *flag* will be set to —1. Under all other conditions, it will be set to zero.

Returns OK or ERROR.

Working for 1D spectra only. Floating-point not implemented.

int sasWriteSpectrum (client, capability, path, number, offset, base, range, type, data, flag)

CLIENT *client;
char *capability;
char *path;
u_int number;
u_int offset;
BASE base;
RANGE range;
u_int type;
char *data;
int *flag;

This routine stores the part of the spectrum *number* held in *data* in the spectrum file *path*. The data is assumed to be *range* in length and will be stored in the spectrum starting at *base*. In the event of an overflow or truncation error, *flag* will be set to —1. Under all other conditions, it will be set to zero.

Returns OK or ERROR.

NOTE: Working for 1D spectra only. Floating-point not implemented.

int sasChangeArrayOptions (client, capability, path, number, layout, type)

CLIENT *client;

char *capability;

char *path;

u_int number;

u_int layout;

u_int type;

u_int flag;

This routine changes the type and layout of a given spectrum array *number* for a given spectrum file *path*. The data held for that spectrum is destroyed. If the main spectrum is changed, the error spectrum is also destroyed.

Returns OK or ERROR.

Warning: Floating-point not implemented. Half-arrays not implemented.

int sasUnauthorise (client, capability)

CLIENT *client;

char *capability;

This routine deauthorises the user whose capability is *capability*. This routine should be used to finish a session with the Spectrum Access Server.

Returns OK or ERROR.

Works for both VxWorks and UNIX versions.

int sasGenerateErrorReport (client, capability, text)

CLIENT *client

char *capability;

char *text;

This routine copies a short message describing the last error associated with the user whose capability is *capability* into the user-supplied buffer *text*, which must be long enough to hold the string.

Returns OK or ERROR.

Works for both VxWorks and UNIX versions.

Miscellaneous Routines

CLIENT *sasOpenClient (host, port)

char *host;

int port;

This routine initialises the C library to use the spectrum server provided by the system identified by *host* and *port*. The routine returns a CLIENT pointer which should be supplied to subsequent SAS calls.

void sasCloseClient (client)

CLIENT *client;

This routine closes a connection to the Spectrum Server made by *sasOpenClient()*. This should be the last routine called.

char *sasErrMsg(num)

int num;

This routine returns a general message associated with the error number *num*.

char *sasPerror (string)

char *string;

This routine returns a general message associated with the error number in *errno*. The contents of *string* are printed on stderr along with the error number and a general error message associated with that number.