This document contains information about the client side of the steering library. Please go through the FAQ first.

The library calls on the client side are :

G_InitGraphicsNode
G_GetParameterList
G_GetVariableList
G_TrackVariable
G_SteerParameter
G_TrackParameter
G_PostExitMessage
G_TerminateConnection

G_InitGraphicsNode

Initializes the data transfer protocol on the client side of the library.

SYNOPSIS

int G_InitGraphicsNode (Server, PortNumber, NodeNumber, MaxNode);

```
char * Server;
int PortNumber;
int * NodeNumber;
int * MaxNode;
char * Protocol;
```

DESCRIPTION

This function call initializes the client side of this library and requires the name of the server (for example, at Oak Ridge use *xps150.ccs.ornl.gov*)

The RPC mechanism works in the following way. Both the client and the server must send and receive on an unique port number. If the portmapper is working, then the port number is supplied by the port mapper on the server side. However the user may wish to supply the port number directly rather than have the portmapper determine it, (refer to FAQ for a complete explanation). Thus if the port number is 0, the port mapper takes over. Otherwise the communication takes place at the port number as specified by the user. The function then queries the server about the maximum number of nodes for this application and also supplies the graphics node number.

The transport protocol used is TCP (refer to FAQ to find out why this particular protocol is used).

This is an initialization function and should be called once.

PARAMETERS

*char * server*
The name of the server which in this context is the parallel machine. An example server name is *xps35.ccs.ornl.gov.*

int PortNumber
As explained above, if the user wants to let the port mapper handle the communication, then this parameter should be 0. Otherwise it should be a valid port number (as determined by rpcinfo or as reported by the corresponding compute program).

*int * NodeNumber*
The Graphics Node number as determined by the server is returned here if the function call succeeds.

*int * MaxNode*
The number of nodes participating in the application on the server side is returned here if the function call succeeds.

*char * Protocol*
This parameter will allow the type of networking protocol to be changed in the future.

RETURN VALUES

0 : If successfully Initialized
1 : Invalid Port Number
−1 : Unable to resolve host name
−2 : Remote Host Error. Connection Unestablished.
−3 : RPC call failed. Connection Established

FILES

client/client.c Source.

G_GetParameterList

SYNOPSIS

char ** G_GetParameterList (NumOfPar, ArrayOfType, ArraySizes,ArrayNode);

```
int * NumOfPar;
int * *ArrayOfType;
int * *ArraySizes;
int ** ArrayNodes;
```

DESCRIPTION

This is an initialization function and should be called once.

This function call determines which parameters are registered on the server side of the library. The client can change only the parameters which are named here. Parameters, in this context, are defined as variables which the client is allowed to change and send to the server (Steering). Variables are those which the client is not allowed to change at all but may read (Tracking).

Every parameter is identified by its name. The other associated information available with the parameter are its type and size. The type must be one of the valid types as listed in graphics.h Each of the parameters can only have a maximum size as determined by the constant MAX_VAR_SIZE in graphics.h.

PARAMETERS

$int * NumOfParameters$
On success the function returns the number of parameters registered on the server.

$int * *ArrayOfTypes$
For each of the parameters this array contains the type. The type must be one the valid types as listed in the graphics.h file.

$int * *ArrayOfSizes$
For each of the parameters, this array contains the size information.

$int * *ArrayNodes$
This returns the Node numbers associated with each of the above parameters.

RETURN VALUES

$NULL$: if the function call fails

$char * *$: An array of names of the parameters as reported by the server

FILES

client/client.c Source.

TIPS

The client side of the library should be initialized before this call can succeed.

This function allocates memory to the pointer provided by the user (Array Of Types, ArrayOfSizes, etc.) which needs to be freed up by an appropriate call to G_TerminateConnection.

This function call does NOT return the actual values of parameters, only the appropriate information to subsequently use the corresponding G_TrackParameter or G_SteerParameter calls.

SEE ALSO

G_InitGraphicsNode on server side
G_TrackParameter
G_SteerParameter
G_TerminateConnection
FAQ

G_GetVariableList
Get the variable information from the server side of the library.

SYNOPSIS
char ** G_GetVariablesList (NumOfVar, ArrayOfType, ArraySizes, ArrayNodes)
```
int * NumOfVar;
int * *ArrayOfType;
int * *ArraySizes;
int * *ArrayNodes;
```

DESCRIPTION
This is an initialization function and should be called once.

This function call determines the variables registered on the server side of the library. The client cannot change the variables which are named here, it can only query the value of these variables by a subsequent call to G_TrackVariables.

Every variable is identified by its name. The other associated information available with the parameter are its type and size. The type must be one of the valid types as listed in graphics.h Each of the parameters can only have a maximum size as determined by the constant MAX_VAR_SIZE in graphics.h.

PARAMETERS
$int * NumOfVar$
On success the function returns the number of variables registered on the server.

$int * *ArrayOfTypes$
For each of the variables this array contains the type. The type is one the valid types as listed in the graphics.h file.

$int * *ArrayOfSizes$
For each of the variables this array contains the size information.

$int * *ArrayNodes$
This array contains the compute node associated with each process.

RETURN VALUES
$NULL$: if the function call fails
$char * *$: An array of names of the variables as reported by the server.

FILES
client/client.c Source

TIPS
The client side of the library should be initialized before this call can succeed.

This function allocates memory to the pointer provided by the user. (Array Of Types, ArrayOfSizes, etc.) which needs to be freed up by an appropriate call to G_Terminate Connection.
This function call does NOT return the actual values of variables. It only determines the appropriate information for a subsequent call to G_TrackVariable.

SEE ALSO

G_InitGraphicsNode on server side
G_TrackVariable
G_TerminateConnection
FAQ

G_TrackVariable

SYNOPSIS

int G_TrackVariable (Data, NodeNumber, Name, type, start, end, step, tm, compression,threshold, nx , ny, nz);

```
void * Data,
int NodeNumber,
char * Name,
int type,
int start,
int end,
int step,
int *tm,
int compression,
double threshold,
int nx,
int ny,
int nz);
```

DESCRIPTION

This function call determines the value of the variable as determined by Name. A query is sent to the server to this effect.

PARAMETERS

*void * Data*
An array of return data values.

intNodeNumber
The NodeNumber to which the query is to be sent.

*char * Name*
The name of the variable which is to be queried.

int type
The type of the variable. This is one of the types as listed in graphics.h.

int start
The start index of the variable.

int step
The step value of the variable.

int end
The end index of the variable.

int tm
This pointer returns or sets the time index associated with the data set. For current applications, it can be ignored for all types except G_FILE_ types.

int compression
This parameter indicates whether wavelet compression is on or off.

double threshold
If compression is on, this value indicates the threshold value to be used for wavelet compression.

int nx ny nz
These parameters indicate the 3D grid size, required for wavelet compression.

RETURN VALUES

-1 : On Error
0 : On Success

FILES

client/client.c Source.

SEE ALSO

G_InitGraphicsNode on server side
G_GetVariableList
G_TerminateConnection
FAQ

G_SteerParameter

Selectively change the value of a particular parameter.

SYNOPSIS

int G_SteerParameter (Data, NodeNumber, Name, size, type)

```
void * Data,
int NodeNumber,
char * Name,
int size,
int type
```

DESCRIPTION

This function call changes the value of the parameter on the server side. The parameter is identified by its name, size and type.

FUNCTION PARAMETERS

void ∗ Data
An array of data values.

int NodeNumber
The NodeNumber to which the parameter data is to be sent.

char ∗ Name
The name of the parameter which is to be queried.

int type
The type of the parameter, which must be one of the types as listed in the graphics.h file.

int size
The size of the parameter.

RETURN VALUES

-1 : On Error
0 : On Success

FILES

client/client. Source

SEE ALSO

G_InitGraphicsNode on server side
G_GetParameterList
G_TrackParameter
G_TerminateConnection
FAQ

G_TrackParameter

Selectively inquire the value of a particular parameter.

SYNOPSIS

int G_TrackParameter (Data, NodeNumber, Name,type, start, step, end)

```
void * Data,
int NodeNumber,
char * Name,
int type,
int start,
int step,
int end
```

DESCRIPTION

This function call inquires the value of the parameter on the server side. The parameter is identified by its name, size and type.

FUNCTION PARAMETERS

*void * Data*

An array of data values.

intNodeNumber

The NodeNumber to which the query is to be sent.

*char * Name*

The name of the parameter which is to be queried.

int type

The type of the parameter.

int start

The start index of the parameter.

int step

The step value of the parameter.

int end

The end index of the parameter.

RETURN VALUES

-1 : On Error

0 : On Success

FILES

client/client.c Source

SEE ALSO

G_InitGraphicsNode on server side

G_GetParameterList

G_SteerParameter

G_TerminateConnection

FAQ

G_PostExitMessage

SYNOPSIS

Inform the server that this client wishes it to terminate as soon as possible. The server makes the ultimate decision on the timing of termination, but can defer that decision to any designated client.

void G_PostExitMessage();

DESCRIPTION

Inform the Server that it should quit as soon as possible. Returning from this routine does not mean that the server will be terminated. It just means that the GraphicsNode has posted a quit message to all the nodes and will quit as soon as all the compute nodes do so.

FUNCTION PARAMETERS

RETURN VALUES

FILES

client/client.c Source.

SEE ALSO

G_InitGraphicsNode on server side
FAQ

 G_TERMINATECONNECTION(1NX_MPI_RPC)

G_TerminateConnection

SYNOPSIS

Terminate the connection from the client side.

void G_TerminateConnection();

DESCRIPTION

Terminate the connection for the client side and free up any memory allocated during previous calls to G_GetVariableList or G_GetParameterList. This will not disturb the server from its processing.

FUNCTION PARAMETERS

RETURN VALUES

FILES

client/client.c Source.

SEE ALSO

G_InitGraphicsNode on server side
FAQ