Subject: IDLRPC and widgets
Posted by kcjones on Fri, 11 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

I am trying to use the GUI tools of IDL to interact with a c program. The idea is that the c program reads the values from the GUI and changes its actions based upon the values read. I thought I ought to be able to do this with IDLRPC. My thoughts were to send commands to IDLRPC to cause it to read values from the GUI and set values in the Main or top level IDL context, then read back the variable values to the c program. The variable values may be changed by interactions with the GUI. I have a little IDL program (see below) that puts some buttons in a CW_BGroup widget and displays it. When I run this little pgm at the IDL command prompt with:

IDL> module

it displays the widget and I have an IDL command line prompt. I can enter commands at the prompt and they are acted upon. At the same time the widget is alive and functions.

BUT, when I fire up IDLRPC and sent the same command to it from a little c program (see below), the command is executed by IDL, displays the widget (and the widget acts as intented), but the IDL_RPCExecuteStr call eventually times out (about 2 minutes - I haven't messed with the time-out). But then again, if relatively shortly after starting the widget, I click 'done', the widget is destroyed (as programmed) and the c program completes promptly with no error indication.

So, IDLRPC seems to wait for the widget to go away, while interactive IDL doesn't care. The manual for IDL_RPCExecuteStr says the command "is executed just as if it had been entered from the IDL command line." Well, almost. How can I trick IDLRPC into not waiting, or is there some other way to skin this cat?

TIA

My platform is a Sun workstation running some flavor of Solaris, and I am using IDL 5.1.

Here's the little IDL widget pgm: ; module.pro simulator module menue widget ; ; 10 Sep 98 phw working

```
function module event, event
  common module, base
  case event.value of
; 'ttya0x01': junk = ttya0x01 (/Realize, Group_Leader=event.top)
;'ttya0x10': junk = ttya0x10 (/Realize, Group_Leader=event.top)
; 'ttya0x11': junk = ttya0x11 (/Realize, Group_Leader=event.top)
'done': begin
 base = 0
 widget_Control, event.top, /Destroy
 end
else: message, 'unknown action'
  endcase
  return, 0
end
pro module
  common module, base
  ; only (re)create menue if necessary!
  create_menue = (N_Elements(base) eq 0)
  if (not create_menue) then create_menue = (base eq 0)
  if (create_menue) then begin
buttons = [ 'ttya0x01', 'ttya0x10', 'ttya0x11', 'done' ]
base = widget_base (Title='Simulator Modules')
bgid = cw bgroup (base, buttons, /Row, $
 /Return_Name, Event_Funct='module_event')
widget control, base, /Realize
xmanager, 'module', base, /No_Block
  endif
end
and here's the c triggering program:
/* test_idlrpc.c test interface of c to idlrpc and a widget
* 11 Sep 98 phw
*/
#include <stdio.h>
#include <stdlib.h>
#include "idl_rpc.h"
/* local globals */
static CLIENT *pClient;
```

```
the assumption is that idlrpc has been started before
  getting here.
int main (int argc, char *argv[]) {
  char cmnd[] = "module";
  char *pgmName = argv[0];
  if ((pClient = IDL_RPCInit (0, NULL)) == NULL) {
fprintf (stderr, "%s: can't register with default RPC "
  "server on 'localhost'\n", pgmName);
exit (EXIT_FAILURE);
  /* issues 'module' command */
  if (IDL_RPCExecuteStr(pClient, cmnd)!= 1) {
fprintf (stderr, "%s: couldn't issue command '%s'\n", pgmName, cmnd);
exit (EXIT_FAILURE);
  /* close communication with RPC server
   * idlrpc is left active
   */
  if (IDL_RPCCleanup (pClient, 0) != 1) {
fprintf (stderr, "%s: error killing RPC client\n", pgmName);
exit (EXIT_FAILURE);
}
  exit (EXIT_SUCCESS);
}
```