Data flow for a simple plotting example with Web Tools

Client in any Web Browser

Plasma Current Plotting

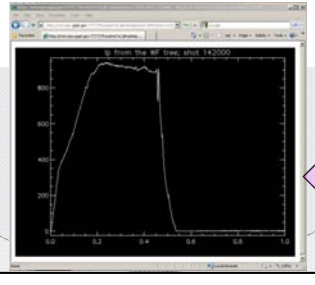
Tree: WF

Shot: 142000 (e.g., 142000)

Plot

HTML Code

```
<HTML>   <!-- Example of Web Page which uses IDL for plotting -->
   <HEAD> <TITLE> Simple IDL Calling Example </TITLE> </HEAD>
   <BODY>

      <H2> Plasma Current Plotting </H2>

      <!-- A Perl script is listening on port 7777 on nstxops: -->
      <FORM ACTION="http://nstxops.pppl.gov:7777/" METHOD=GET>
         <INPUT TYPE="hidden" NAME="RoutineToCall" VALUE="plotip">

         Tree: <SELECT NAME="tree" >
                  <OPTION VALUE="WF"> WF
                  <OPTION VALUE="EFIT"> EFIT
                  <OPTION VALUE="ENGINEERING"> ENGINEERING
               </SELECT> <P>

         Shot: <input TYPE="text" NAME="shot" SIZE=6>
                (e.g., 142000) <P>

          <INPUT TYPE="submit" VALUE="Plot">
      </FORM>

   </BODY>
</HTML>
```



Server written in Perl

```
#!/usr/bin/perl -w
# A simplified Perl script for serving HTML commands to IDL for plotting

use IO::Socket;
    # connect to a TCP/IP socket
$PORT = 7777;    # pick something not in use
$server = IO::Socket::INET->new(LocalPort => $PORT, Type => SOCK_STREAM,
                                Proto => 'tcp',Reuse => 1, Listen => 20)
                  or   die "Can't open port";
while ($client = $server->accept()) {
   $data=<$client>;
   ($cmd)=($data=~/\?([^ ]*?) /);       # clean up input string
   $cmd =~ s/\r/ /g;                    # replace CR with blank
   #  (...other command line cleanup & conversion to keyword pairs...)
   $strpos = index( $cmd, "RoutineToCall" );
   $rest = substr( $cmd, $strpos+15, 1000);
   $IDLroutine = substr( $rest, 0, index( $rest, "'" ) );

   if (!defined( $child_pid = fork() )) {    # fork process so no delays
      die "cannot fork";
   } elsif ($child_pid) {
   } else {
      $mypid = $$;
      $ENV{'IDL_DEVICE'} = 'Z'; # so don't try to connect to X

         # create a unique lock file name from PID
      $lock_file = "LockFile_$mypid";
      open ( LOCKFILE, "> $lock_file")  or die "can't open lockfile";
      close ( LOCKFILE )  or die "can't close lockfile";
      $pngfilename = 'mdsplots-'.$mypid.'.png';

         # open a pipe to IDL (need full path for CRON job)
      open IDL, "| idl" or die "Failed to open IDL";
      IDL->autoflush;          # so output gets there right away
      $idl_cmd="retall\n ".        # command for plotting
               " bigtextpng, filename='$pngfilename', port=$PORT\n".
               "$IDLroutine,$cmd\n tvlct,r,g,b,/get\n".
               " write_png,'$pngfilename', tvrd(),r,g,b\n".
               " spawn,'/bin/rm $lock_file'\n";
      print IDL $idl_cmd;

         # Loop to Check for IDL to delete lock file
      $count = 0;
      $sleepSec = 0.2;
      $giveUpSec = 600;          # give up after 10 minutes
      while (-e "$lock_file") {
         select(undef, undef, undef, $sleepSec);
         $count = $count+1;
            # if more than 10 minutes, kill the lock file
         if ( $count*$sleepSec > $giveUpSec ) {
            print "\n *** Giving up after $giveUpSec seconds waited \n";
            unlink $lock_file;  # force deletion lock file
         }
      };
      close IDL;
      open PNGOUT,$pngfilename;          # send PNG file back to browser
      $image=join("",<PNGOUT>);
      print $client "HTTP/1.0 200 Document follows\n";
      print $client "Content-type: image/png\n\n".$image;
      close (PNGOUT)                  or die "can't close PNGOUT";
      exit;      # exit child process
   }
  close $client;
};
```

IDL Code

```
; Example of a plot-producing IDL routine which
;    is called by a Perl script listening on a
;    TCP/IP socket to a HTML web tool.

pro plotip, tree=tree, shot=shot, _extra=_extra

if nwords(tree) eq 0 then tree='WF'
if nwords(shot) eq 0 then shot=LASTSHOT()
tree=STRUPCASE(tree)

if tree eq 'WF'   then signal = '\ip'        else  $
if tree eq 'EFIT' then signal = '\ipmeas'    else  $
if tree eq 'ENGINEERING' then signal = '\ip1'

MDSOPEN, tree, shot, status=status
if not status then return

data = MDSVALUE( signal )
time = MDSVALUE('dim_of('+signal+')')

PLOT, time, data>0, title='Ip from the '+tree+ $
      ' tree; shot '+ STRTRIM(shot,2), ystyle=3

end
```

PNG file