

```

/*
 * NeWS is a product of Sun Microsystems, Inc. and is provided for
 * unrestricted use provided that this legend is included on all tape
 * media and as a part of the software program in whole or part.  Users
 * may copy or modify NeWS without charge, but are not authorized to
 * license or distribute it to anyone else except as part of a product
 * or program developed by the user.
 *
 * NeWS IS PROVIDED AS IS WITH NO WARRANTIES OF ANY KIND INCLUDING THE
 * WARRANTIES OF DESIGN, MERCHANTIBILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE.
 *
 * NeWS is provided with no support and without any obligation on the
 * part of Sun Microsystems, Inc. to assist in its use, correction,
 * modification or enhancement.
 *
 * SUN MICROSYSTEMS, INC. SHALL HAVE NO LIABILITY WITH RESPECT TO THE
 * INFRINGEMENT OF COPYRIGHTS, TRADE SECRETS OR ANY PATENTS BY NeWS
 * OR ANY PART THEREOF.
 *
 * In no event will Sun Microsystems, Inc. be liable for any lost revenue
 * or profits or other special, indirect and consequential damages, even
 * if Sun has been advised of the possibility of such damages.
 *
 * Sun Microsystems, Inc.
 * 2550 Garcia Avenue
 * Mountain View, California 94043
 */

```

```

#ifndef lint
static char sccsid[] = "@(#)psh.c 1.4 87/03/13 Copyright 1985 Sun Micro";
#endif

```

```

/*
 * Copyright (c) 1985 by Sun Microsystems, Inc.
 */

```

```

/*-
    NeWS operating system command executer.  Used to execute a
    program with its input and output connected to the NeWS
    server. Based on psh.c.  Makes a connection to the server,
    sets standard in, out and error to be the connection, sends
    the string servicename (which defaults to programname, and can
    be set with the -s argument), followed by a newline, and then
    exec's the file given by the next arg, passing it the rest of
    the args.

```

psh.c, Sat Feb 14 09:48:11 1987

James Gosling,
Sun Microsystems

psexec.c, Tue Sep 15 22:52:51 EDT 1987

Don Hopkins,
University of Maryland

```

*/

```

```

#define MAXARGS 50

#include <stdio.h>
char      *programname;
char      *servicename;
char      *args[MAXARGS];
int        nargs;
int        input_file;

FILE       *PostScript,
           *PostScriptInput;

connect_and_exec()
{
    if (PostScript == 0 && ps_open_PostScript() == 0) {
        fprintf(stderr, "%s: Cannot connect to window server\n",
                programname);
        exit(0);
    }

    dup2(fileno(PostScriptInput), 0);
    dup2(0, 1);
    dup2(0, 2);
    fclose(PostScriptInput);
    fclose(PostScript);
    printf("%s\n", servicename);
    fflush(stdout);
    execv(args[0], &args[0]);
    perror("execv");
}

main(argc, argv)
char      **argv;
{
    servicename = programname = argv[0];
    while (--argc > 0)
        if ((++argv)[0][0] == '-' && nargs == 0)
            switch (argv[0][1]) {
                case 's':
                    if (--argc <= 0) {
                        fprintf(stderr, "%s: not enough arguments\n", programname);
                        exit(1);
                    }
                    servicename = (++argv)[0];
                    break;
                default:
                    fprintf(stderr, "%s: invalid switch '%s'\n",
                            programname, argv[0]);
                    exit(1);
            }
        else if (nargs < MAXARGS)
            args[nargs++] = argv[0];
        else {
            fprintf(stderr, "%s: too many arguments\n", programname);
            exit(1);
        }

    if (nargs == 0) {

```

```
        fprintf(stderr, "%s: not enough arguments\n", programname);
    }

    args[nargs] = NULL;
    connect_and_exec();
}
```