

```

*****
87937 Thu Jun 13 09:42:53 2019
new/usr/src/cmd/make/bin/main.cc
11235 make -C fails under GCC7
*****
_____unchanged_portion_omitted_____

2425 /*
2426  * Scan the argv for options and "=" type args and make them readonly.
2427  */
2428 static void
2429 enter_argv_values(int argc, char *argv[], ASCII_Dyn_Array *makeflags_and_macro)
2430 {
2431     register char    *cp;
2432     register int     i;
2433     int              length;
2434     register Name    name;
2435     int              opt_separator = argc;
2436     char             tmp_char;
2437     wchar_t          *tmp_wcs_buffer;
2438     register Name    value;
2439     Boolean          append = false;
2440     Property         macro;
2441     struct stat      statbuf;

2444     /* Read argv options and "=" type args and make them readonly. */
2445     makefile_type = reading_nothing;
2446     for (i = 1; i < argc; ++i) {
2447         append = false;
2448         if (argv[i] == NULL) {
2449             continue;
2450         } else if (((argv[i][0] == '-') && (argv[i][1] == '-')) ||
2451             ((argv[i][0] == (int) '-') &&
2452             (argv[i][1] == (int) '-') &&
2453             (argv[i][2] == (int) '-') &&
2454             (argv[i][3] == (int) '-'))) {
2455             argv[i] = NULL;
2456             opt_separator = i;
2457             continue;
2458         } else if ((i < opt_separator) && (argv[i][0] == (int) hyphen_ch
2459         switch (parse_command_option(argv[i][1])) {
2460         case 1: /* -f seen */
2461             ++i;
2462             continue;
2463         case 2: /* -c seen */
2464             if (argv[i+1] == NULL) {
2465                 fatal(gettext("No dmake rcfile argument
2466             })
2467             MBSTOWCS(wcs_buffer, "DMAKE_RCFILE");
2468             name = GETNAME(wcs_buffer, FIND_LENGTH);
2469             break;
2470         case 4: /* -g seen */
2471             if (argv[i+1] == NULL) {
2472                 fatal(gettext("No dmake group argument a
2473             })
2474             MBSTOWCS(wcs_buffer, "DMAKE_GROUP");
2475             name = GETNAME(wcs_buffer, FIND_LENGTH);
2476             break;
2477         case 8: /* -j seen */
2478             if (argv[i+1] == NULL) {
2479                 fatal(gettext("No dmake max jobs argumen
2480             })
2481             MBSTOWCS(wcs_buffer, "DMAKE_MAX_JOBS");
2482             name = GETNAME(wcs_buffer, FIND_LENGTH);
2483             break;

```

```

2484     case 16: /* -M seen */
2485         if (argv[i+1] == NULL) {
2486             fatal(gettext("No pmake machinesfile arg
2487         })
2488         MBSTOWCS(wcs_buffer, "PMAKE_MACHINESFILE");
2489         name = GETNAME(wcs_buffer, FIND_LENGTH);
2490         break;
2491     case 32: /* -m seen */
2492         if (argv[i+1] == NULL) {
2493             fatal(gettext("No dmake mode argument af
2494         })
2495         MBSTOWCS(wcs_buffer, "DMAKE_MODE");
2496         name = GETNAME(wcs_buffer, FIND_LENGTH);
2497         break;
2498     case 256: /* -K seen */
2499         if (argv[i+1] == NULL) {
2500             fatal(gettext("No makestate filename arg
2501         })
2502         MBSTOWCS(wcs_buffer, argv[i+1]);
2503         make_state = GETNAME(wcs_buffer, FIND_LENGTH);
2504         keep_state = true;
2505         argv[i] = NULL;
2506         argv[i+1] = NULL;
2507         continue;
2508     case 512: /* -o seen */
2509         if (argv[i+1] == NULL) {
2510             fatal(gettext("No dmake output dir argum
2511         })
2512         MBSTOWCS(wcs_buffer, "DMAKE_ODIR");
2513         name = GETNAME(wcs_buffer, FIND_LENGTH);
2514         break;
2515     case 1024: /* -x seen */
2516         if (argv[i+1] == NULL) {
2517             fatal(gettext("No argument after -x flag
2518         })
2519         length = strlen("SUN_MAKE_COMPAT_MODE=");
2520         if (strncmp(argv[i+1], "SUN_MAKE_COMPAT_MODE=",
2521             argv[i+1] == &argv[i+1][length];
2522             MBSTOWCS(wcs_buffer, "SUN_MAKE_COMPAT_MO
2523             name = GETNAME(wcs_buffer, FIND_LENGTH);
2524             dmake_compat_mode_specified = dmake_add_
2525             break;
2526         }
2527         length = strlen("DMAKE_OUTPUT_MODE=");
2528         if (strncmp(argv[i+1], "DMAKE_OUTPUT_MODE=", len
2529             argv[i+1] == &argv[i+1][length];
2530             MBSTOWCS(wcs_buffer, "DMAKE_OUTPUT_MODE"
2531             name = GETNAME(wcs_buffer, FIND_LENGTH);
2532             dmake_output_mode_specified = dmake_add_
2533         } else {
2534             warning(gettext("Unknown argument '%s' a
2535                 argv[i+1]);
2536             argv[i] = argv[i + 1] = NULL;
2537             continue;
2538         }
2539         break;
2540     case 2048: /* -C seen */
2541         if (argv[i + 1] == NULL) {
2542             fatal(gettext("No argument after -C flag
2543         })
2544         if (chdir(argv[i + 1]) != 0) {
2545             fatal(gettext("failed to change to direc
2546                 argv[i + 1], strerror(errno));
2547         }
2548         MBSTOWCS(wcs_buffer, "DMAKE_CDIR");
2549         name = GETNAME(wcs_buffer, FIND_LENGTH);

```

```

2550         path_reset = true;
2551         rebuild_arg0 = true;
2552         (void) get_current_path();
2553         break;
2554     default: /* Shouldn't reach here */
2555         argv[i] = NULL;
2556         continue;
2557     }
2558     argv[i] = NULL;
2559     if (i == (argc - 1)) {
2560         break;
2561     }
2562     if ((length = strlen(argv[i+1])) >= MAXPATHLEN) {
2563         tmp_wcs_buffer = ALLOC_WC(length + 1);
2564         (void) mbstowcs(tmp_wcs_buffer, argv[i+1], length);
2565         value = GETNAME(tmp_wcs_buffer, FIND_LENGTH);
2566         retmem(tmp_wcs_buffer);
2567     } else {
2568         MBSTOWCS(wcs_buffer, argv[i+1]);
2569         value = GETNAME(wcs_buffer, FIND_LENGTH);
2570     }
2571     argv[i+1] = NULL;
2572     } else if ((cp = strchr(argv[i], (int) equal_char)) != NULL) {
2573 /*
2574  * Combine all macro in dynamic array
2575  */
2576         if(*(cp-1) == (int) plus_char)
2577         {
2578             if(isspace(*(cp-2))) {
2579                 append = true;
2580                 cp--;
2581             }
2582         }
2583         if(!append)
2584             append_or_replace_macro_in_dyn_array(makeflags_a

2586         while (isspace(*(cp-1))) {
2587             cp--;
2588         }
2589         tmp_char = *cp;
2590         *cp = (int) nul_char;
2591         MBSTOWCS(wcs_buffer, argv[i]);
2592         *cp = tmp_char;
2593         name = GETNAME(wcs_buffer, wcslen(wcs_buffer));
2594         while (*cp != (int) equal_char) {
2595             cp++;
2596         }
2597         cp++;
2598         while (isspace(*cp) && (*cp != (int) nul_char)) {
2599             cp++;
2600         }
2601         if ((length = strlen(cp)) >= MAXPATHLEN) {
2602             tmp_wcs_buffer = ALLOC_WC(length + 1);
2603             (void) mbstowcs(tmp_wcs_buffer, cp, length + 1);
2604             value = GETNAME(tmp_wcs_buffer, FIND_LENGTH);
2605             retmem(tmp_wcs_buffer);
2606         } else {
2607             MBSTOWCS(wcs_buffer, cp);
2608             value = GETNAME(wcs_buffer, FIND_LENGTH);
2609         }
2610         argv[i] = NULL;
2611     } else {
2612         /* Illegal MAKEFLAGS argument */
2613         continue;
2614     }
2615     if(append) {

```

```

2616         setvar_append(name, value);
2617         append = false;
2618     } else {
2619         macro = maybe_append_prop(name, macro_prop);
2620         macro->body.macro.exported = true;
2621         SETVAR(name, value, false)->body.macro.read_only = true;
2622     }
2623 }
2624 }
_____unchanged_portion_omitted_

```