

```

*****
101234 Sat Aug 1 08:08:29 2015
new/usr/src/cmd/sgs/rtld/common/util.c
6092 rtld should always provide assfail/assfail3
Reviewed by: Andy Stormont <astormont@racktopsystems.com>
*****
unchanged portion omitted

3075 #if      DEBUG
3075 /*
3076  * Provide assfail() for ASSERT() statements.  See <sys/debug.h> for further
3077  * details.
3078  */
3079 int
3080 assfail(const char *a, const char *f, int l)
3081 {
3082     (void) printf("assertion failed: %s, file: %s, line: %d\n", a, f, l);
3083     (void) _lwp_kill(_lwp_self(), SIGABRT);
3084     return (0);
3085 }

3087 void
3088 assfail3(const char *msg, uintmax_t a, const char *op, uintmax_t b,
3089          const char *f, int l)
3090 {
3091     (void) printf("assertion failed: %s (0x%llx %s 0x%llx), "
3092                 "file: %s, line: %d\n", msg, (unsigned long long)a, op,
3093                 (unsigned long long)b, f, l);
3092     (void) printf("assertion failed: %s (%llu %s %llu), "
3093                 "file: %s, line: %d\n", msg, a, op, b, f, l);
3094     (void) _lwp_kill(_lwp_self(), SIGABRT);
3095 }
3096 #endif

3097 /*
3098  * Exit.  If we arrive here with a non zero status it's because of a fatal
3099  * error condition (most commonly a relocation error).  If the application has
3100  * already had control, then the actual fatal error message will have been
3101  * recorded in the dlerror() message buffer.  Print the message before really
3102  * exiting.
3103  */
3104 void
3105 rtdlexit(Lm_list * lml, int status)
3106 {
3107     if (status) {
3108         if (rtld_flags & RT_FL_APPLIC) {
3109             /*
3110              * If the error buffer has been used, write out all
3111              * pending messages - lasterr is simply a pointer to
3112              * the last message in this buffer.  However, if the
3113              * buffer couldn't be created at all, lasterr points
3114              * to a constant error message string.
3115              */
3116             if (*errbuf) {
3117                 char *errptr = errbuf;
3118                 char *errend = errbuf + ERRSIZE;

3120                 while ((errptr < errend) && *errptr) {
3121                     size_t size = strlen(errptr);
3122                     (void) write(2, errptr, size);
3123                     (void) write(2, MSG_ORIG(MSG_STR_NL),
3124                                MSG_STR_NL_SIZE);
3125                     errptr += (size + 1);
3126                 }
3127             }
3128             if (lasterr && ((lasterr < errbuf) ||

```

```

3129                 (lasterr > (errbuf + ERRSIZE)))) {
3130                     (void) write(2, lasterr, strlen(lasterr));
3131                     (void) write(2, MSG_ORIG(MSG_STR_NL),
3132                                MSG_STR_NL_SIZE);
3133                 }
3134             }
3135             leave(lml, 0);
3136             (void) _lwp_kill(_lwp_self(), killsig);
3137         }
3138         _exit(status);
3139     }
unchanged portion omitted

```