```
**********************************************************
   20916 Mon Oct  7 19:17:11 2013
new/usr/src/common/crypto/edonr/edonr.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * IDI,NTNU
 *
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://opensource.org/licenses/CDDL-1.0.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 *
 * Copyright (C) 2009, 2010, Jorn Amundsen <jorn.amundsen@ntnu.no>
 * Tweaked Edon-R implementation for SUPERCOP, based on NIST API.
 *
 * $Id: edonr.c 517 2013-02-17 20:34:39Z joern $
 */
/*
 * Portions copyright (c) 2013, Saso Kiselkov, All rights reserved
 */

/* determine where we can get bcopy/bzero declarations */
#ifdef  _KERNEL
#include <sys/systm.h>
#else
#include <strings.h>
#endif
#include <sys/edonr.h>
#include <sys/debug.h>

/* big endian support, provides no-op's if run on little endian hosts */
#include "edonr_byteorder.h"

#define hashState224(x) ((x)->pipe->p256)
#define hashState256(x) ((x)->pipe->p256)
#define hashState384(x) ((x)->pipe->p512)
#define hashState512(x) ((x)->pipe->p512)

/* shift and rotate shortcuts */
#define shl(x, n)       ((x) << n)
#define shr(x, n)       ((x) >> n)

#define rotl32(x, n)    (((x) << (n)) | ((x) >> (32 - (n))))
#define rotr32(x, n)    (((x) >> (n)) | ((x) << (32 - (n))))

#define rotl64(x, n)    (((x) << (n)) | ((x) >> (64 - (n))))
#define rotr64(x, n)    (((x) >> (n)) | ((x) << (64 - (n))))

#if !defined(__C99_RESTRICT)
#define restrict        /* restrict */
#endif
```



```
#define EDONR_VALID_HASHBITLEN(x) \
        ((x) == 512 || (x) == 384 || (x) == 256 || (x) == 224)

/* EdonR224 initial double chaining pipe */
static const uint32_t i224p2[16] = {
        0x00010203ul, 0x04050607ul, 0x08090a0bul, 0x0c0d0e0ful,
        0x10111213ul, 0x14151617ul, 0x18191a1bul, 0x1c1d1e1ful,
        0x20212223ul, 0x24252627ul, 0x28292a2bul, 0x2c2d2e2ful,
        0x30313233ul, 0x34353637ul, 0x38393a3bul, 0x3c3d3e3ful,
};

/* EdonR256 initial double chaining pipe */
static const uint32_t i256p2[16] = {
        0x40414243ul, 0x44454647ul, 0x48494a4bul, 0x4c4d4e4ful,
        0x50515253ul, 0x54555657ul, 0x58595a5bul, 0x5c5d5e5ful,
        0x60616263ul, 0x64656667ul, 0x68696a6bul, 0x6c6d6e6ful,
        0x70717273ul, 0x74757677ul, 0x78797a7bul, 0x7c7d7e7ful,
};

/* EdonR384 initial double chaining pipe */
static const uint64_t i384p2[16] = {
        0x0001020304050607ull, 0x08090a0b0c0d0e0full,
        0x1011121314151617ull, 0x18191a1b1c1d1e1full,
        0x2021222324252627ull, 0x28292a2b2c2d2e2full,
        0x3031323334353637ull, 0x38393a3b3c3d3e3full,
        0x4041424344454647ull, 0x48494a4b4c4d4e4full,
        0x5051525354555657ull, 0x58595a5b5c5d5e5full,
        0x6061626364656667ull, 0x68696a6b6c6d6e6full,
        0x7071727374757677ull, 0x78797a7b7c7d7e7full
};

/* EdonR512 initial double chaining pipe */
static const uint64_t i512p2[16] = {
        0x8081828384858687ull, 0x88898a8b8c8d8e8full,
        0x9091929394959697ull, 0x98999a9b9c9d9e9full,
        0xa0a1a2a3a4a5a6a7ull, 0xa8a9aaabacadaeafull,
        0xb0b1b2b3b4b5b6b7ull, 0xb8b9babbbcbdbebfull,
        0xc0c1c2c3c4c5c6c7ull, 0xc8c9cacbcccdcecfull,
        0xd0d1d2d3d4d5d6d7ull, 0xd8d9dadbdcdddedfull,
        0xe0e1e2e3e4e5e6e7ull, 0xe8e9eaebecedeeefull,
        0xf0f1f2f3f4f5f6f7ull, 0xf8f9fafbfcfdfeffull
};

/*
 * First Latin Square
 * 0   7   1   3   2   4   6   5
 * 4   1   7   6   3   0   5   2
 * 7   0   4   2   5   3   1   6
 * 1   4   0   5   6   2   7   3
 * 2   3   6   7   1   5   0   4
 * 5   2   3   1   7   6   4   0
 * 3   6   5   0   4   7   2   1
 * 6   5   2   4   0   1   3   7
 */
#define LS1_256(c, x0, x1, x2, x3, x4, x5, x6, x7)                      \
{                                                                       \
        uint32_t x04, x17, x23, x56, x07, x26;                          \
        x04 = x0+x4, x17 = x1+x7, x07 = x04+x17;                        \
        s0 = c + x07 + x2;                                              \
        s1 = rotl32(x07 + x3, 4);                                       \
        s2 = rotl32(x07 + x6, 8);                                       \
        x23 = x2 + x3;                                                  \
        s5 = rotl32(x04 + x23 + x5, 22);                                \
        x56 = x5 + x6;                                                  \
        s6 = rotl32(x17 + x56 + x0, 24);                                \
```

```
128         x26 = x23+x56;                                          \
129         s3 = rotl32(x26 + x7, 13);                              \
130         s4 = rotl32(x26 + x1, 17);                              \
131         s7 = rotl32(x26 + x4, 29);                              \
132 }

133 #define LS1_512(c, x0, x1, x2, x3, x4, x5, x6, x7)              \
134 {                                                               \
135         uint64_t x04, x17, x23, x56, x07, x26;                  \
136         x04 = x0+x4, x17 = x1+x7, x07 = x04+x17;                \
137         s0 = c + x07 + x2;                                      \
138         s1 = rotl64(x07 + x3, 5);                               \
139         s2 = rotl64(x07 + x6, 15);                              \
140         x23 = x2 + x3;                                          \
141         s5 = rotl64(x04 + x23 + x5, 40);                        \
142         x56 = x5 + x6;                                          \
143         s6 = rotl64(x17 + x56 + x0, 50);                        \
144         x26 = x23+x56;                                          \
145         s3 = rotl64(x26 + x7, 22);                              \
146         s4 = rotl64(x26 + x1, 31);                              \
147         s7 = rotl64(x26 + x4, 59);                              \
148 }

149 /*
150  * Second Orthogonal Latin Square
151  * 0   4   2   3   1   6   5   7
152  * 7   6   3   2   5   4   1   0
153  * 5   3   1   6   0   2   7   4
154  * 1   0   5   4   3   7   2   6
155  * 2   1   0   7   4   5   6   3
156  * 3   5   7   0   6   1   4   2
157  * 4   7   6   1   2   0   3   5
158  * 6   2   4   5   7   3   0   1
159  */
160 #define LS2_256(c, y0, y1, y2, y3, y4, y5, y6, y7)              \
161 {                                                               \
162         uint32_t y01, y25, y34, y67, y04, y05, y27, y37;        \
163         y01 = y0+y1, y25 = y2+y5, y05 = y01+y25;                \
164         t0  = ~c + y05 + y7;                                    \
165         t2 = rotl32(y05 + y3, 9);                               \
166         y34 = y3+y4, y04 = y01+y34;                             \
167         t1 = rotl32(y04 + y6, 5);                               \
168         t4 = rotl32(y04 + y5, 15);                              \
169         y67 = y6+y7, y37 = y34+y67;                             \
170         t3 = rotl32(y37 + y2, 11);                              \
171         t7 = rotl32(y37 + y0, 27);                              \
172         y27 = y25+y67;                                          \
173         t5 = rotl32(y27 + y4, 20);                              \
174         t6 = rotl32(y27 + y1, 25);                              \
175 }

176 #define LS2_512(c, y0, y1, y2, y3, y4, y5, y6, y7)              \
177 {                                                               \
178         uint64_t y01, y25, y34, y67, y04, y05, y27, y37;        \
179         y01 = y0+y1, y25 = y2+y5, y05 = y01+y25;                \
180         t0  = ~c + y05 + y7;                                    \
181         t2 = rotl64(y05 + y3, 19);                              \
182         y34 = y3+y4, y04 = y01+y34;                             \
183         t1 = rotl64(y04 + y6, 10);                              \
184         t4 = rotl64(y04 + y5, 36);                              \
185         y67 = y6+y7, y37 = y34+y67;                             \
186         t3 = rotl64(y37 + y2, 29);                              \
187         t7 = rotl64(y37 + y0, 55);                              \
188         y27 = y25+y67;                                          \
189         t5 = rotl64(y27 + y4, 44);                              \
190         t6 = rotl64(y27 + y1, 48);                              \
```



```
191 }

192 #define quasi_exform256(r0, r1, r2, r3, r4, r5, r6, r7)         \
193 {                                                               \
194         uint32_t s04, s17, s23, s56, t01, t25, t34, t67;        \
195         s04 = s0 ^ s4, t01 = t0 ^ t1;                           \
196         r0 = (s04 ^ s1) + (t01 ^ t5);                           \
197         t67 = t6 ^ t7;                                          \
198         r1 = (s04 ^ s7) + (t2 ^ t67);                           \
199         s23 = s2 ^ s3;                                          \
200         r7 = (s23 ^ s5) + (t4 ^ t67);                           \
201         t34 = t3 ^ t4;                                          \
202         r3 = (s23 ^ s4) + (t0 ^ t34);                           \
203         s56 = s5 ^ s6;                                          \
204         r5 = (s3 ^ s56) + (t34 ^ t6);                           \
205         t25 = t2 ^ t5;                                          \
206         r6 = (s2 ^ s56) + (t25 ^ t7);                           \
207         s17 = s1 ^ s7;                                          \
208         r4 = (s0 ^ s17) + (t1 ^ t25);                           \
209         r2 = (s17 ^ s6) + (t01 ^ t3);                           \
210 }

211 #define quasi_exform512(r0, r1, r2, r3, r4, r5, r6, r7)         \
212 {                                                               \
213         uint64_t s04, s17, s23, s56, t01, t25, t34, t67;        \
214         s04 = s0 ^ s4, t01 = t0 ^ t1;                           \
215         r0 = (s04 ^ s1) + (t01 ^ t5);                           \
216         t67 = t6 ^ t7;                                          \
217         r1 = (s04 ^ s7) + (t2 ^ t67);                           \
218         s23 = s2 ^ s3;                                          \
219         r7 = (s23 ^ s5) + (t4 ^ t67);                           \
220         t34 = t3 ^ t4;                                          \
221         r3 = (s23 ^ s4) + (t0 ^ t34);                           \
222         s56 = s5 ^ s6;                                          \
223         r5 = (s3 ^ s56) + (t34 ^ t6);                           \
224         t25 = t2 ^ t5;                                          \
225         r6 = (s2 ^ s56) + (t25 ^ t7);                           \
226         s17 = s1 ^ s7;                                          \
227         r4 = (s0 ^ s17) + (t1 ^ t25);                           \
228         r2 = (s17 ^ s6) + (t01 ^ t3);                           \
229 }

230 static size_t
231 Q256(size_t bitlen, const uint32_t *data, uint32_t *restrict p)
232 {
233         size_t bl;

234         for (bl = bitlen; bl >= EdonR256_BLOCK_BITSIZE;
235             bl -= EdonR256_BLOCK_BITSIZE, data += 16) {
236                 uint32_t s0, s1, s2, s3, s4, s5, s6, s7, t0, t1, t2, t3, t4,
237                     t5, t6, t7;
238                 uint32_t p0, p1, p2, p3, p4, p5, p6, p7, q0, q1, q2, q3, q4,
239                     q5, q6, q7;
240                 const uint32_t defix = 0xaaaaaaaa;
241 #if defined(MACHINE_IS_BIG_ENDIAN)
242                 uint32_t swp0, swp1, swp2, swp3, swp4, swp5, swp6, swp7, swp8,
243                     swp9, swp10, swp11, swp12, swp13, swp14, swp15;
244 #define d(j)    swp ## j
245 #define s32(j)  ld_swap32((uint32_t *)data + j, swp ## j)
246 #else
247 #define d(j)    data[j]
248 #endif

249                 /* First row of quasigroup e-transformations */
250 #if defined(MACHINE_IS_BIG_ENDIAN)
251                 s32(8);
```

```
260                 s32(9);
261                 s32(10);
262                 s32(11);
263                 s32(12);
264                 s32(13);
265                 s32(14);
266                 s32(15);
267 #endif
268                 LS1_256(defix, d(15), d(14), d(13), d(12), d(11), d(10), d(9),
269                     d(8));
270 #if defined(MACHINE_IS_BIG_ENDIAN)
271                 s32(0);
272                 s32(1);
273                 s32(2);
274                 s32(3);
275                 s32(4);
276                 s32(5);
277                 s32(6);
278                 s32(7);
279 #undef s32
280 #endif
281                 LS2_256(defix, d(0), d(1), d(2), d(3), d(4), d(5), d(6), d(7));
282                 quasi_exform256(p0, p1, p2, p3, p4, p5, p6, p7);
283
284                 LS1_256(defix, p0, p1, p2, p3, p4, p5, p6, p7);
285                 LS2_256(defix, d(8), d(9), d(10), d(11), d(12), d(13), d(14),
286                     d(15));
287                 quasi_exform256(q0, q1, q2, q3, q4, q5, q6, q7);
288
289                 /* Second row of quasigroup e-transformations */
290                 LS1_256(defix, p[8], p[9], p[10], p[11], p[12], p[13], p[14],
291                     p[15]);
292                 LS2_256(defix, p0, p1, p2, p3, p4, p5, p6, p7);
293                 quasi_exform256(p0, p1, p2, p3, p4, p5, p6, p7);
294
295                 LS1_256(defix, p0, p1, p2, p3, p4, p5, p6, p7);
296                 LS2_256(defix, q0, q1, q2, q3, q4, q5, q6, q7);
297                 quasi_exform256(q0, q1, q2, q3, q4, q5, q6, q7);
298
299                 /* Third row of quasigroup e-transformations */
300                 LS1_256(defix, p0, p1, p2, p3, p4, p5, p6, p7);
301                 LS2_256(defix, p[0], p[1], p[2], p[3], p[4], p[5], p[6], p[7]);
302                 quasi_exform256(p0, p1, p2, p3, p4, p5, p6, p7);
303
304                 LS1_256(defix, q0, q1, q2, q3, q4, q5, q6, q7);
305                 LS2_256(defix, p0, p1, p2, p3, p4, p5, p6, p7);
306                 quasi_exform256(q0, q1, q2, q3, q4, q5, q6, q7);
307
308                 /* Fourth row of quasigroup e-transformations */
309                 LS1_256(defix, d(7), d(6), d(5), d(4), d(3), d(2), d(1), d(0));
310                 LS2_256(defix, p0, p1, p2, p3, p4, p5, p6, p7);
311                 quasi_exform256(p0, p1, p2, p3, p4, p5, p6, p7);
312
313                 LS1_256(defix, p0, p1, p2, p3, p4, p5, p6, p7);
314                 LS2_256(defix, q0, q1, q2, q3, q4, q5, q6, q7);
315                 quasi_exform256(q0, q1, q2, q3, q4, q5, q6, q7);
316
317                 /* Edon-R tweak on the original SHA-3 Edon-R submission. */
318                 p[0] ^= d(8) ^ p0;
319                 p[1] ^= d(9) ^ p1;
320                 p[2] ^= d(10) ^ p2;
321                 p[3] ^= d(11) ^ p3;
322                 p[4] ^= d(12) ^ p4;
323                 p[5] ^= d(13) ^ p5;
324                 p[6] ^= d(14) ^ p6;
325                 p[7] ^= d(15) ^ p7;
```



```
326                 p[8] ^= d(0) ^ q0;
327                 p[9] ^= d(1) ^ q1;
328                 p[10] ^= d(2) ^ q2;
329                 p[11] ^= d(3) ^ q3;
330                 p[12] ^= d(4) ^ q4;
331                 p[13] ^= d(5) ^ q5;
332                 p[14] ^= d(6) ^ q6;
333                 p[15] ^= d(7) ^ q7;
334         }
335
336 #undef d
337         return (bitlen - bl);
338 }
339
340 #if defined(__IBMC__) && defined(_AIX) && defined(__64BIT__)
341 static inline size_t
342 #else
343 static size_t
344 #endif
345 Q512(size_t bitlen, const uint64_t *data, uint64_t *restrict p)
346 {
347         size_t bl;
348
349         for (bl = bitlen; bl >= EdonR512_BLOCK_BITSIZE;
350             bl -= EdonR512_BLOCK_BITSIZE, data += 16) {
351                 uint64_t s0, s1, s2, s3, s4, s5, s6, s7, t0, t1, t2, t3, t4,
352                     t5, t6, t7;
353                 uint64_t p0, p1, p2, p3, p4, p5, p6, p7, q0, q1, q2, q3, q4,
354                     q5, q6, q7;
355                 const uint64_t defix = 0xaaaaaaaaaaaaaaaaull;
356 #if defined(MACHINE_IS_BIG_ENDIAN)
357                 uint64_t swp0, swp1, swp2, swp3, swp4, swp5, swp6, swp7, swp8,
358                     swp9, swp10, swp11, swp12, swp13, swp14, swp15;
359 #define d(j)    swp##j
360 #define s64(j)  ld_swap64((uint64_t *)data+j, swp##j)
361 #else
362 #define d(j)    data[j]
363 #endif
364
365                 /* First row of quasigroup e-transformations */
366 #if defined(MACHINE_IS_BIG_ENDIAN)
367                 s64(8);
368                 s64(9);
369                 s64(10);
370                 s64(11);
371                 s64(12);
372                 s64(13);
373                 s64(14);
374                 s64(15);
375 #endif
376                 LS1_512(defix, d(15), d(14), d(13), d(12), d(11), d(10), d(9),
377                     d(8));
378 #if defined(MACHINE_IS_BIG_ENDIAN)
379                 s64(0);
380                 s64(1);
381                 s64(2);
382                 s64(3);
383                 s64(4);
384                 s64(5);
385                 s64(6);
386                 s64(7);
387 #undef s64
388 #endif
389                 LS2_512(defix, d(0), d(1), d(2), d(3), d(4), d(5), d(6), d(7));
390                 quasi_exform512(p0, p1, p2, p3, p4, p5, p6, p7);
```

```
392                 LS1_512(defix, p0, p1, p2, p3, p4, p5, p6, p7);
393                 LS2_512(defix, d(8), d(9), d(10), d(11), d(12), d(13), d(14),
394                     d(15));
395                 quasi_exform512(q0, q1, q2, q3, q4, q5, q6, q7);
396
397                 /* Second row of quasigroup e-transformations */
398                 LS1_512(defix, p[8], p[9], p[10], p[11], p[12], p[13], p[14],
399                     p[15]);
400                 LS2_512(defix, p0, p1, p2, p3, p4, p5, p6, p7);
401                 quasi_exform512(p0, p1, p2, p3, p4, p5, p6, p7);
402
403                 LS1_512(defix, p0, p1, p2, p3, p4, p5, p6, p7);
404                 LS2_512(defix, q0, q1, q2, q3, q4, q5, q6, q7);
405                 quasi_exform512(q0, q1, q2, q3, q4, q5, q6, q7);
406
407                 /* Third row of quasigroup e-transformations */
408                 LS1_512(defix, p0, p1, p2, p3, p4, p5, p6, p7);
409                 LS2_512(defix, p[0], p[1], p[2], p[3], p[4], p[5], p[6], p[7]);
410                 quasi_exform512(p0, p1, p2, p3, p4, p5, p6, p7);
411
412                 LS1_512(defix, q0, q1, q2, q3, q4, q5, q6, q7);
413                 LS2_512(defix, p0, p1, p2, p3, p4, p5, p6, p7);
414                 quasi_exform512(q0, q1, q2, q3, q4, q5, q6, q7);
415
416                 /* Fourth row of quasigroup e-transformations */
417                 LS1_512(defix, d(7), d(6), d(5), d(4), d(3), d(2), d(1), d(0));
418                 LS2_512(defix, p0, p1, p2, p3, p4, p5, p6, p7);
419                 quasi_exform512(p0, p1, p2, p3, p4, p5, p6, p7);
420
421                 LS1_512(defix, p0, p1, p2, p3, p4, p5, p6, p7);
422                 LS2_512(defix, q0, q1, q2, q3, q4, q5, q6, q7);
423                 quasi_exform512(q0, q1, q2, q3, q4, q5, q6, q7);
424
425                 /* Edon-R tweak on the original SHA-3 Edon-R submission. */
426                 p[0] ^= d(8) ^ p0;
427                 p[1] ^= d(9) ^ p1;
428                 p[2] ^= d(10) ^ p2;
429                 p[3] ^= d(11) ^ p3;
430                 p[4] ^= d(12) ^ p4;
431                 p[5] ^= d(13) ^ p5;
432                 p[6] ^= d(14) ^ p6;
433                 p[7] ^= d(15) ^ p7;
434                 p[8] ^= d(0) ^ q0;
435                 p[9] ^= d(1) ^ q1;
436                 p[10] ^= d(2) ^ q2;
437                 p[11] ^= d(3) ^ q3;
438                 p[12] ^= d(4) ^ q4;
439                 p[13] ^= d(5) ^ q5;
440                 p[14] ^= d(6) ^ q6;
441                 p[15] ^= d(7) ^ q7;
442         }
443
444 #undef d
445         return (bitlen - bl);
446 }
447
448 void
449 EdonRInit(EdonRState *state, size_t hashbitlen)
450 {
451         ASSERT(EDONR_VALID_HASHBITLEN(hashbitlen));
452         switch (hashbitlen) {
453         case 224:
454                 state->hashbitlen = 224;
455                 state->bits_processed = 0;
456                 state->unprocessed_bits = 0;
457                 bcopy(i224p2, hashState224(state)->DoublePipe,
```



```
458                     16 * sizeof (uint32_t));
459                 break;
460
461         case 256:
462                 state->hashbitlen = 256;
463                 state->bits_processed = 0;
464                 state->unprocessed_bits = 0;
465                 bcopy(i256p2, hashState256(state)->DoublePipe,
466                     16 * sizeof (uint32_t));
467                 break;
468
469         case 384:
470                 state->hashbitlen = 384;
471                 state->bits_processed = 0;
472                 state->unprocessed_bits = 0;
473                 bcopy(i384p2, hashState384(state)->DoublePipe,
474                     16 * sizeof (uint64_t));
475                 break;
476
477         case 512:
478                 state->hashbitlen = 512;
479                 state->bits_processed = 0;
480                 state->unprocessed_bits = 0;
481                 bcopy(i512p2, hashState224(state)->DoublePipe,
482                     16 * sizeof (uint64_t));
483                 break;
484         }
485 }
486
487
488 void
489 EdonRUpdate(EdonRState *state, const uint8_t *data, size_t databitlen)
490 {
491         uint32_t *data32;
492         uint64_t *data64;
493
494         size_t bits_processed;
495
496         ASSERT(EDONR_VALID_HASHBITLEN(state->hashbitlen));
497         switch (state->hashbitlen) {
498         case 224:
499         case 256:
500                 if (state->unprocessed_bits > 0) {
501                         ASSERT(state->unprocessed_bits + databitlen <=
502                             EdonR256_BLOCK_SIZE * 8);
503
504                         /* LastBytes = databitlen / 8 */
505                         /* LINTED E_DECLARATION_IN_CODE */
506                         int LastBytes = (int)databitlen >> 3;
507
508                         bcopy(data, hashState256(state)->LastPart
509                             + (state->unprocessed_bits >> 3), LastBytes);
510                         state->unprocessed_bits += (int)databitlen;
511                         databitlen = state->unprocessed_bits;
512                         /* LINTED E_BAD_PTR_CAST_ALIGN */
513                         data32 = (uint32_t *)hashState256(state)->LastPart;
514                 } else
515                         /* LINTED E_BAD_PTR_CAST_ALIGN */
516                         data32 = (uint32_t *)data;
517
518                 bits_processed = Q256(databitlen, data32,
519                     hashState256(state)->DoublePipe);
520                 state->bits_processed += bits_processed;
521                 databitlen -= bits_processed;
522                 state->unprocessed_bits = (int)databitlen;
523                 if (databitlen > 0) {
```

```
524                         /* LastBytes = Ceil(databitlen / 8) */
525                         int LastBytes =
526                             ((~(((-(int)databitlen) >> 3) & 0x01ff)) +
527                             1) & 0x01ff;
528
529                         data32 += bits_processed >> 5;  /* byte size update */
530                         bcopy(data32, hashState256(state)->LastPart, LastBytes);
531                 }
532                 break;
533
534         case 384:
535         case 512:
536                 if (state->unprocessed_bits > 0) {
537                         ASSERT(state->unprocessed_bits + databitlen <=
538                             EdonR512_BLOCK_SIZE * 8);
539                         /* LastBytes = databitlen / 8 */
540                         /* LINTED E_DECLARATION_IN_CODE */
541                         int LastBytes = (int)databitlen >> 3;
542
543                         bcopy(data, hashState512(state)->LastPart
544                             + (state->unprocessed_bits >> 3), LastBytes);
545                         state->unprocessed_bits += (int)databitlen;
546                         databitlen = state->unprocessed_bits;
547                         /* LINTED E_BAD_PTR_CAST_ALIGN */
548                         data64 = (uint64_t *)hashState512(state)->LastPart;
549                 } else
550                         /* LINTED E_BAD_PTR_CAST_ALIGN */
551                         data64 = (uint64_t *)data;
552
553                 bits_processed = Q512(databitlen, data64,
554                     hashState512(state)->DoublePipe);
555                 state->bits_processed += bits_processed;
556                 databitlen -= bits_processed;
557                 state->unprocessed_bits = (int)databitlen;
558                 if (databitlen > 0) {
559                         /* LastBytes = Ceil(databitlen / 8) */
560                         int LastBytes =
561                             ((~(((-(int)databitlen) >> 3) & 0x03ff)) +
562                             1) & 0x03ff;
563
564                         data64 += bits_processed >> 6;  /* byte size update */
565                         bcopy(data64, hashState512(state)->LastPart, LastBytes);
566                 }
567                 break;
568         }
569 }
570
571 void
572 EdonRFinal(EdonRState *state, uint8_t *hashval)
573 {
574         uint32_t *data32;
575         uint64_t *data64, num_bits;
576
577         size_t databitlen;
578         int LastByte, PadOnePosition;
579
580         num_bits = state->bits_processed + state->unprocessed_bits;
581         ASSERT(EDONR_VALID_HASHBITLEN(state->hashbitlen));
582         switch (state->hashbitlen) {
583         case 224:
584         case 256:
585                 LastByte = (int)state->unprocessed_bits >> 3;
586                 PadOnePosition = 7 - (state->unprocessed_bits & 0x07);
587                 hashState256(state)->LastPart[LastByte] =
588                     (hashState256(state)->LastPart[LastByte]
589                     & (0xff << (PadOnePosition + 1))) ^
```



```
590                     (0x01 << PadOnePosition);
591                 /* LINTED E_BAD_PTR_CAST_ALIGN */
592                 data64 = (uint64_t *)hashState256(state)->LastPart;
593
594                 if (state->unprocessed_bits < 448) {
595                         (void) memset((hashState256(state)->LastPart) +
596                             LastByte + 1, 0x00,
597                             EdonR256_BLOCK_SIZE - LastByte - 9);
598                         databitlen = EdonR256_BLOCK_SIZE * 8;
599 #if defined(MACHINE_IS_BIG_ENDIAN)
600                         st_swap64(num_bits, data64 + 7);
601 #else
602                         data64[7] = num_bits;
603 #endif
604                 } else {
605                         (void) memset((hashState256(state)->LastPart) +
606                             LastByte + 1, 0x00,
607                             EdonR256_BLOCK_SIZE * 2 - LastByte - 9);
608                         databitlen = EdonR256_BLOCK_SIZE * 16;
609 #if defined(MACHINE_IS_BIG_ENDIAN)
610                         st_swap64(num_bits, data64 + 15);
611 #else
612                         data64[15] = num_bits;
613 #endif
614                 }
615
616                 /* LINTED E_BAD_PTR_CAST_ALIGN */
617                 data32 = (uint32_t *)hashState256(state)->LastPart;
618                 state->bits_processed += Q256(databitlen, data32,
619                     hashState256(state)->DoublePipe);
620                 break;
621
622         case 384:
623         case 512:
624                 LastByte = (int)state->unprocessed_bits >> 3;
625                 PadOnePosition = 7 - (state->unprocessed_bits & 0x07);
626                 hashState512(state)->LastPart[LastByte] =
627                     (hashState512(state)->LastPart[LastByte]
628                     & (0xff << (PadOnePosition + 1))) ^
629                     (0x01 << PadOnePosition);
630                 /* LINTED E_BAD_PTR_CAST_ALIGN */
631                 data64 = (uint64_t *)hashState512(state)->LastPart;
632
633                 if (state->unprocessed_bits < 960) {
634                         (void) memset((hashState512(state)->LastPart) +
635                             LastByte + 1, 0x00,
636                             EdonR512_BLOCK_SIZE - LastByte - 9);
637                         databitlen = EdonR512_BLOCK_SIZE * 8;
638 #if defined(MACHINE_IS_BIG_ENDIAN)
639                         st_swap64(num_bits, data64 + 15);
640 #else
641                         data64[15] = num_bits;
642 #endif
643                 } else {
644                         (void) memset((hashState512(state)->LastPart) +
645                             LastByte + 1, 0x00,
646                             EdonR512_BLOCK_SIZE * 2 - LastByte - 9);
647                         databitlen = EdonR512_BLOCK_SIZE * 16;
648 #if defined(MACHINE_IS_BIG_ENDIAN)
649                         st_swap64(num_bits, data64 + 31);
650 #else
651                         data64[31] = num_bits;
652 #endif
653                 }
654
655                 state->bits_processed += Q512(databitlen, data64,
```

```
656                     hashState512(state)->DoublePipe);
657                 break;
658         }

660         switch (state->hashbitlen) {
661         case 224: {
662 #if defined(MACHINE_IS_BIG_ENDIAN)
663                 uint32_t *d32 = (uint32_t *)hashval;
664                 uint32_t *s32 = hashState224(state)->DoublePipe + 9;
665                 int j;

667                 for (j = 0; j < EdonR224_DIGEST_SIZE >> 2; j++)
668                         st_swap32(s32[j], d32 + j);
669 #else
670                 bcopy(hashState256(state)->DoublePipe + 9, hashval,
671                     EdonR224_DIGEST_SIZE);
672 #endif
673                 break;
674         }
675         case 256: {
676 #if defined(MACHINE_IS_BIG_ENDIAN)
677                 uint32_t *d32 = (uint32_t *)hashval;
678                 uint32_t *s32 = hashState224(state)->DoublePipe + 8;
679                 int j;

681                 for (j = 0; j < EdonR256_DIGEST_SIZE >> 2; j++)
682                         st_swap32(s32[j], d32 + j);
683 #else
684                 bcopy(hashState256(state)->DoublePipe + 8, hashval,
685                     EdonR256_DIGEST_SIZE);
686 #endif
687                 break;
688         }
689         case 384: {
690 #if defined(MACHINE_IS_BIG_ENDIAN)
691                 uint64_t *d64 = (uint64_t *)hashval;
692                 uint64_t *s64 = hashState384(state)->DoublePipe + 10;
693                 int j;

695                 for (j = 0; j < EdonR384_DIGEST_SIZE >> 3; j++)
696                         st_swap64(s64[j], d64 + j);
697 #else
698                 bcopy(hashState384(state)->DoublePipe + 10, hashval,
699                     EdonR384_DIGEST_SIZE);
700 #endif
701                 break;
702         }
703         case 512: {
704 #if defined(MACHINE_IS_BIG_ENDIAN)
705                 uint64_t *d64 = (uint64_t *)hashval;
706                 uint64_t *s64 = hashState512(state)->DoublePipe + 8;
707                 int j;

709                 for (j = 0; j < EdonR512_DIGEST_SIZE >> 3; j++)
710                         st_swap64(s64[j], d64 + j);
711 #else
712                 bcopy(hashState512(state)->DoublePipe + 8, hashval,
713                     EdonR512_DIGEST_SIZE);
714 #endif
715                 break;
716         }
717         }
718 }


721 void
```



```
722 EdonRHash(size_t hashbitlen, const uint8_t *data, size_t databitlen,
723     uint8_t *hashval)
724 {
725         EdonRState state;

727         EdonRInit(&state, hashbitlen);
728         EdonRUpdate(&state, data, databitlen);
729         EdonRFinal(&state, hashval);
730 }
```

```
**********************************************************
    7033 Mon Oct  7 19:17:11 2013
new/usr/src/common/crypto/edonr/edonr_byteorder.h
4185 New hash algorithm support
**********************************************************
```

```
/*
 * IDI,NTNU
 *
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://opensource.org/licenses/CDDL-1.0.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 *
 * Copyright (C) 2009, 2010, Jorn Amundsen <jorn.amundsen@ntnu.no>
 *
 * C header file to determine compile machine byte order. Take care when cross
 * compiling.
 *
 * $Id: byteorder.h 517 2013-02-17 20:34:39Z joern $
 */
/*
 * Portions copyright (c) 2013, Saso Kiselkov, All rights reserved
 */

#if !defined(BYTEORDER_INCLUDED)

#if defined(__linux)
#include <endian.h>
#else
#include <sys/param.h>
#endif

#if defined(__BYTE_ORDER)
#if (__BYTE_ORDER == __BIG_ENDIAN)
#define MACHINE_IS_BIG_ENDIAN
#elif (__BYTE_ORDER == __LITTLE_ENDIAN)
#define MACHINE_IS_LITTLE_ENDIAN
#endif
#elif defined(BYTE_ORDER)
#if (BYTE_ORDER == BIG_ENDIAN)
#define MACHINE_IS_BIG_ENDIAN
#elif (BYTE_ORDER == LITTLE_ENDIAN)
#define MACHINE_IS_LITTLE_ENDIAN
#endif
#endif /* __BYTE_ORDER || BYTE_ORDER */

#if !defined(MACHINE_IS_BIG_ENDIAN) && !defined(MACHINE_IS_LITTLE_ENDIAN)
#if defined(_BIG_ENDIAN) || defined(_MIPSEB)
#define MACHINE_IS_BIG_ENDIAN
#endif
#if defined(_LITTLE_ENDIAN) || defined(_MIPSEL)
#define MACHINE_IS_LITTLE_ENDIAN
```



```
#endif
#endif /* !MACHINE_IS_BIG_ENDIAN && !MACHINE_IS_LITTLE_ENDIAN */

#if !defined(MACHINE_IS_BIG_ENDIAN) && !defined(MACHINE_IS_LITTLE_ENDIAN)
#error unknown machine byte sex
#endif

#define BYTEORDER_INCLUDED

#if defined(MACHINE_IS_BIG_ENDIAN)
/*
 * Byte swapping macros for big endian architectures and compilers,
 * add as appropriate for other architectures and/or compilers.
 *
 *     ld_swap64(src,dst) : uint64_t dst = *(src)
 *     st_swap64(src,dst) : *(dst)       = uint64_t src
 */

#if defined(__PPC__) || defined(_ARCH_PPC)

#if defined(__64BIT__)
#if defined(_ARCH_PWR7)
#define aix_ld_swap64(s64, d64)\
        __asm__("ldbrx %0,0,%1" : "=r"(d64) : "r"(s64))
#define aix_st_swap64(s64, d64)\
        __asm__ volatile("stdbrx %1,0,%0" : : "r"(d64), "r"(s64))
#else
#define aix_ld_swap64(s64, d64)                                         \
{                                                                       \
        uint64_t *s4, h;                                                \
                                                                        \
        __asm__("addi %0,%3,4;lwbrx %1,0,%3;lwbrx %2,0,%0;rldimi %1,%2,32,0"\
                : "+r"(s4), "=r"(d64), "=r"(h) : "b"(s64));              \
}

#define aix_st_swap64(s64, d64)                                         \
{                                                                       \
        uint64_t *s4, h;                                                \
        h = (s64) >> 32;                                                \
        __asm__ volatile("addi %0,%3,4;stwbrx %1,0,%3;stwbrx %2,0,%0"   \
                : "+r"(s4) : "r"(s64), "r"(h), "b"(d64));               \
}
#endif /* 64BIT && PWR7 */
#else
#define aix_ld_swap64(s64, d64)                                         \
{                                                                       \
        uint32_t *s4, h, l;                                             \
        __asm__("addi %0,%3,4;lwbrx %1,0,%3;lwbrx %2,0,%0"              \
                : "+r"(s4), "=r"(l), "=r"(h) : "b"(s64));               \
        d64 = ((uint64_t)h<<32) | l;                                    \
}

#define aix_st_swap64(s64, d64)                                         \
{                                                                       \
        uint32_t *s4, h, l;                                             \
        l = (s64) & 0xffffffffful, h = (s64) >> 32;                     \
        __asm__ volatile("addi %0,%3,4;stwbrx %1,0,%3;stwbrx %2,0,%0"   \
                : "+r"(s4) : "r"(l), "r"(h), "b"(d64));                 \
}
#endif /* __64BIT__ */
#define aix_ld_swap32(s32, d32)\
        __asm__("lwbrx %0,0,%1" : "=r"(d32) : "r"(s32))
#define aix_st_swap32(s32, d32)\
        __asm__ volatile("stwbrx %1,0,%0" : : "r"(d32), "r"(s32))
#define ld_swap32(s, d) aix_ld_swap32(s, d)
#define st_swap32(s, d) aix_st_swap32(s, d)
```

```
#define ld_swap64(s, d) aix_ld_swap64(s, d)
#define st_swap64(s, d) aix_st_swap64(s, d)
#endif /* __PPC__ || _ARCH_PPC */

#if defined(__sparc)
#if !defined(__arch64__) && !defined(__sparcv8) && defined(__sparcv9)
#define __arch64__
#endif
#if defined(__GNUC__) || (defined(__SUNPRO_C) && __SUNPRO_C > 0x590)
/* need Sun Studio C 5.10 and above for GNU inline assembly */
#if defined(__arch64__)
#define sparc_ld_swap64(s64, d64)                                       \
        __asm__("ldxa [%1]0x88,%0" : "=r"(d64) : "r"(s64))
#define sparc_st_swap64(s64, d64)                                       \
        __asm__ volatile("stxa %0,[%1]0x88" : : "r"(s64), "r"(d64))
#define st_swap64(s, d) sparc_st_swap64(s, d)
#else
#define sparc_ld_swap64(s64, d64)                                       \
{                                                                       \
        uint32_t *s4, h, l;                                             \
        __asm__("add %3,4,%0\n\tlda [%3]0x88,%1\n\tlda [%0]0x88,%2"     \
                : "+r"(s4), "=r"(l), "=r"(h) : "r"(s64));               \
        d64 = ((uint64_t)h<<32) | l;                                    \
}
#define sparc_st_swap64(s64, d64)                                       \
{                                                                       \
        uint32_t *s4, h, l;                                             \
        l = (s64) & 0xffffffffful, h = (s64) >> 32;                     \
        __asm__ volatile("add %3,4,%0\n\tsta %1,[%3]0x88\n\tsta %2,[%0]0x88"\
                : "+r"(s4) : "r"(l), "r"(h), "r"(d64));                 \
}
#endif /* sparc64 */
#define sparc_ld_swap32(s32, d32)\
        __asm__("lda [%1]0x88,%0" : "=r"(d32) : "r"(s32))
#define sparc_st_swap32(s32, d32)\
        __asm__ volatile("sta %0,[%1]0x88" : : "r"(s32), "r"(d32))
#define ld_swap32(s, d) sparc_ld_swap32(s, d)
#define st_swap32(s, d) sparc_st_swap32(s, d)
#define ld_swap64(s, d) sparc_ld_swap64(s, d)
#define st_swap64(s, d) sparc_st_swap64(s, d)
#endif /* GCC || Sun Studio C > 5.9 */
#endif /* sparc */

/* GCC fallback */
#if ((__GNUC__ >= 4) || defined(__PGIC__)) && !defined(ld_swap32)
#define ld_swap32(s, d) (d = __builtin_bswap32(*(s)))
#define st_swap32(s, d) (*(d) = __builtin_bswap32(s))
#endif /* GCC4/PGIC && !swap32 */
#if ((__GNUC__ >= 4) || defined(__PGIC__)) && !defined(ld_swap64)
#define ld_swap64(s, d) (d = __builtin_bswap64(*(s)))
#define st_swap64(s, d) (*(d) = __builtin_bswap64(s))
#endif /* GCC4/PGIC && !swap64 */

/* generic fallback */
#if !defined(ld_swap32)
#define ld_swap32(s, d)                                                 \
        (d = (*(s) >> 24) | (*(s) >> 8 & 0xff00) |                      \
        (*(s) << 8 & 0xff0000) | (*(s) << 24))
#define st_swap32(s, d)                                                 \
        (*(d) = ((s) >> 24) | ((s) >> 8 & 0xff00) |                     \
        ((s) << 8 & 0xff0000) | ((s) << 24))
#endif
#if !defined(ld_swap64)
#define ld_swap64(s, d)                                                 \
        (d = (*(s) >> 56) | (*(s) >> 40 & 0xff00) |                     \
        (*(s) >> 24 & 0xff0000) | (*(s) >> 8 & 0xff000000) |            \
```



```
        (*(s) & 0xff000000) << 8 | (*(s) & 0xff0000) << 24 |            \
        (*(s) & 0xff00) << 40 | *(s) << 56)
#define st_swap64(s, d)                                                 \
        (*(d) = ((s) >> 56) | ((s) >> 40 & 0xff00) |                    \
        ((s) >> 24 & 0xff0000) | ((s) >> 8 & 0xff000000) |              \
        ((s) & 0xff000000) << 8 | ((s) & 0xff0000) << 24 |              \
        ((s) & 0xff00) << 40 | (s) << 56)
#endif

#endif /* MACHINE_IS_BIG_ENDIAN */


#if defined(MACHINE_IS_LITTLE_ENDIAN)
/* replace swaps with simple assignments on little endian systems */
#undef  ld_swap32
#undef  st_swap32
#define ld_swap32(s, d) (d = *(s))
#define st_swap32(s, d) (*(d) = s)
#undef  ld_swap64
#undef  st_swap64
#define ld_swap64(s, d) (d = *(s))
#define st_swap64(s, d) (*(d) = s)
#endif /* MACHINE_IS_LITTLE_ENDIAN */

#endif /* BYTEORDER_INCLUDED */
```

****
1761 Mon Oct 7 19:17:11 2013
new/usr/src/common/crypto/edonr/test/Makefile
4185 New hash algorithm support
****

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://opensource.org/licenses/CDDL-1.0.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2013 Saso Kiselkov.  All rights reserved.
#
#       This makefile drives the production of the edonr test suite.
#       To run the tests, simply type "make" or "make check".

include ../../../../Makefile.master

CFLAGS = -I${SRC}/uts/common -D_KERNEL -fno-builtin -O2 -W -Wall \
  -Wno-unknown-pragmas -std=c99
CPUMHZ:sh = psrinfo -v 0 | awk '/processor operates at/{print $6}'
GCC = $(GCC_ROOT)/bin/gcc

check : edonr_test32_check edonr_test64_check

all : edonr_test32 edonr_test64

edonr_test32 : edonr_test.c
        @ echo Building 32-bit test...
        @ $(GCC) $(CFLAGS) -o edonr_test32 edonr_test.c ../edonr.c

edonr_test64 : edonr_test.c
        @ echo Building 64-bit test...
        @ $(GCC) $(CFLAGS) -m64 -o edonr_test64 edonr_test.c ../edonr.c

edonr_test32_check : edonr_test32
        @ echo Running 32-bit test...
        @ ./edonr_test32 $(CPUMHZ)

edonr_test64_check : edonr_test64
        @ echo Running 64-bit test...
        @ ./edonr_test64 $(CPUMHZ)

clean :
        rm -f edonr_test32 edonr_test64
```

```
**********************************************************
    6542 Mon Oct  7 19:17:11 2013
new/usr/src/common/crypto/edonr/test/edonr_test.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://opensource.org/licenses/CDDL-1.0.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2013 Saso Kiselkov. All rights reserved.
 */

/*
 * This is just to keep the compiler happy about sys/time.h not declaring
 * gettimeofday due to -D_KERNEL (we can do this since we're actually
 * running in userspace, but we need -D_KERNEL for the remaining Edon-R code).
 */
#ifdef  _KERNEL
#undef  _KERNEL
#endif

#include <sys/edonr.h>
#include <stdlib.h>
#include <strings.h>
#include <stdio.h>
#include <sys/time.h>

/*
 * Test messages from:
 * http://csrc.nist.gov/groups/ST/toolkit/documents/Examples/SHA_All.pdf
 */
const char      *test_msg0 = "abc";
const char      *test_msg1 = "abcdbcdecdefdefgefghfghighijhijkijkljklmklmn"
        "lmnomnopnopq";
const char      *test_msg2 = "abcdefghbcdefghicdefghijdefghijkefghijklfghi"
        "jklmghijklmnhijklmnoijklmnopjklmnopqklmnopqrlmnopqrsmnopqrstnopqrstu";

/*
 * Test digests computed by hand. There's no formal standard or spec for edonr.
 */
const uint8_t   edonr_224_test_digests[][28] = {
        {
                // for test_msg0
                0x56, 0x63, 0xc4, 0x93, 0x95, 0x20, 0xfa, 0xf6,
                0x12, 0x31, 0x65, 0xa4, 0x66, 0xf2, 0x56, 0x01,
                0x95, 0x2e, 0xa9, 0xe4, 0x24, 0xdd, 0xc9, 0x6b,
                0xef, 0xd0, 0x40, 0x94
        },
```



```
        {
                // for test_msg1
                0xd0, 0x13, 0xe4, 0x87, 0x4d, 0x06, 0x8d, 0xca,
                0x4e, 0x14, 0xb9, 0x37, 0x2f, 0xce, 0x12, 0x20,
                0x60, 0xf8, 0x5c, 0x0a, 0xfd, 0x7a, 0x7d, 0x97,
                0x88, 0x2b, 0x05, 0x75
        }
        // no test vectorfor test_msg2
};

const uint8_t   edonr_256_test_digests[][32] = {
        {
                // for test_msg0
                0x54, 0xd7, 0x8b, 0x13, 0xc7, 0x4e, 0xda, 0x5a,
                0xed, 0xc2, 0x71, 0xcc, 0x88, 0x1f, 0xb2, 0x2f,
                0x83, 0x99, 0xaf, 0xd3, 0x04, 0x0b, 0x6a, 0x39,
                0x2d, 0x73, 0x94, 0x05, 0x50, 0x8d, 0xd8, 0x51
        },
        {
                // for test_msg1
                0x49, 0x2d, 0x0b, 0x19, 0xab, 0x1e, 0xde, 0x3a,
                0xea, 0x9b, 0xf2, 0x39, 0x3a, 0xb1, 0x21, 0xde,
                0x21, 0xf6, 0x80, 0x1f, 0xad, 0xbe, 0x8b, 0x07,
                0xc7, 0xfb, 0xe6, 0x99, 0x0e, 0x4d, 0x73, 0x63
        }
        // no test vectorfor test_msg2
};

const uint8_t   edonr_384_test_digests[][48] = {
        {
                // for test_msg0
                0x0e, 0x7c, 0xd7, 0x85, 0x78, 0x77, 0xe0, 0x89,
                0x5b, 0x1c, 0xdf, 0x49, 0xf4, 0x1d, 0x20, 0x9c,
                0x72, 0x7d, 0x2e, 0x57, 0x9b, 0x9b, 0x9a, 0xdc,
                0x60, 0x27, 0x97, 0x82, 0xb9, 0x90, 0x72, 0xec,
                0x7e, 0xce, 0xd3, 0x16, 0x5f, 0x47, 0x75, 0x48,
                0xfa, 0x60, 0x72, 0x7e, 0x01, 0xc7, 0x7c, 0xc6
        },
        {
                // no test vector for test_msg1
        },
        {
                // for test_msg2
                0xe2, 0x34, 0xa1, 0x02, 0x83, 0x76, 0xae, 0xe6,
                0x82, 0xd9, 0x38, 0x32, 0x0e, 0x00, 0x78, 0xd2,
                0x34, 0xdb, 0xb9, 0xbd, 0xf0, 0x08, 0xa8, 0x0f,
                0x63, 0x1c, 0x3d, 0x4a, 0xfd, 0x0a, 0xe9, 0x59,
                0xdc, 0xd4, 0xce, 0xcd, 0x8d, 0x67, 0x6c, 0xea,
                0xbb, 0x1a, 0x32, 0xed, 0x5c, 0x6b, 0xf1, 0x7f
        }
};

const uint8_t   edonr_512_test_digests[][64] = {
        {
                // for test_msg0
                0x1b, 0x14, 0xdb, 0x15, 0x5f, 0x1d, 0x40, 0x65,
                0x94, 0xb8, 0xce, 0xf7, 0x0a, 0x43, 0x62, 0xec,
                0x6b, 0x5d, 0xe6, 0xa5, 0xda, 0xf5, 0x0e, 0xc9,
                0x99, 0xe9, 0x87, 0xc1, 0x9d, 0x30, 0x49, 0xe2,
                0xde, 0x59, 0x77, 0xbb, 0x05, 0xb1, 0xbb, 0x22,
                0x00, 0x50, 0xa1, 0xea, 0x5b, 0x46, 0xa9, 0xf1,
                0x74, 0x0a, 0xca, 0xfb, 0xf6, 0xb4, 0x50, 0x32,
                0xad, 0xc9, 0x0c, 0x62, 0x83, 0x72, 0xc2, 0x2b
        },
        {
                // no test vector for test_msg1
```

```
128         },
129         {
130                 // for test_msg2
131                 0x53, 0x51, 0x07, 0x0d, 0xc5, 0x1c, 0x3b, 0x2b,
132                 0xac, 0xa5, 0xa6, 0x0d, 0x02, 0x52, 0xcc, 0xb4,
133                 0xe4, 0x92, 0x1a, 0x96, 0xfe, 0x5a, 0x69, 0xe7,
134                 0x6d, 0xad, 0x48, 0xfd, 0x21, 0xa0, 0x84, 0x5a,
135                 0xd5, 0x7f, 0x88, 0x0b, 0x3e, 0x4a, 0x90, 0x7b,
136                 0xc5, 0x03, 0x15, 0x18, 0x42, 0xbb, 0x94, 0x9e,
137                 0x1c, 0xba, 0x74, 0x39, 0xa6, 0x40, 0x9a, 0x34,
138                 0xb8, 0x43, 0x6c, 0xb4, 0x69, 0x21, 0x58, 0x3c
139         }
140 };

142 int
143 main(int argc, char *argv[])
144 {
145         boolean_t       failed = B_FALSE;
146         uint64_t        cpu_mhz = 0;

148         if (argc == 2)
149                 cpu_mhz = atoi(argv[1]);

151 #define EDONR_ALGO_TEST(_m, mode, testdigest)                           \
152         do {                                                            \
153                 EdonRState      ctx;                                    \
154                 uint8_t         digest[mode / 8];                       \
155                 EdonRInit(&ctx, mode);                                  \
156                 EdonRUpdate(&ctx, (const uint8_t *) _m, strlen(_m) * 8);\
157                 EdonRFinal(&ctx, digest);                               \
158                 printf("Edon-R-%-6sMessage: " #_m "\tResult: ", #mode); \
159                 if (bcmp(digest, testdigest, mode / 8) == 0) {          \
160                         printf("OK\n");                                 \
161                 } else {                                                \
162                         printf("FAILED!\n");                            \
163                         failed = B_TRUE;                                \
164                 }                                                       \
165         } while (0)

167 #define EDONR_PERF_TEST(mode)                                           \
168         do {                                                            \
169                 EdonRState      ctx;                                    \
170                 uint8_t         digest[mode / 8];                       \
171                 uint8_t         block[131072];                          \
172                 uint64_t        delta;                                  \
173                 double          cpb = 0;                                \
174                 int             i;                                      \
175                 struct timeval  start, end;                             \
176                 bzero(block, sizeof (block));                           \
177                 gettimeofday(&start, NULL);                             \
178                 EdonRInit(&ctx, mode);                                  \
179                 for (i = 0; i < 8192; i++)                              \
180                         EdonRUpdate(&ctx, block, sizeof (block) * 8);   \
181                 EdonRFinal(&ctx, digest);                               \
182                 gettimeofday(&end, NULL);                               \
183                 delta = (end.tv_sec * 1000000llu + end.tv_usec) -       \
184                     (start.tv_sec * 1000000llu + start.tv_usec);        \
185                 if (cpu_mhz != 0) {                                     \
186                         cpb = (cpu_mhz * 1e6 * ((double)delta /         \
187                             1000000)) / (8192 * 128 * 1024);            \
188                 }                                                       \
189                 printf("Edon-R-%-6s%llu us (%.02f CPB)\n", #mode, delta,\
190                     cpb);                                               \
191         } while (0)
```



```
194         printf("Running algorithm correctness tests:\n");
195         EDONR_ALGO_TEST(test_msg0, 224, edonr_224_test_digests[0]);
196         EDONR_ALGO_TEST(test_msg1, 224, edonr_224_test_digests[1]);
197         EDONR_ALGO_TEST(test_msg0, 256, edonr_256_test_digests[0]);
198         EDONR_ALGO_TEST(test_msg1, 256, edonr_256_test_digests[1]);
199         EDONR_ALGO_TEST(test_msg0, 384, edonr_384_test_digests[0]);
200         EDONR_ALGO_TEST(test_msg2, 384, edonr_384_test_digests[2]);
201         EDONR_ALGO_TEST(test_msg0, 512, edonr_512_test_digests[0]);
202         EDONR_ALGO_TEST(test_msg2, 512, edonr_512_test_digests[2]);
203         if (failed)
204                 return (1);

206         printf("Running performance tests (hashing 1024 MiB of data):\n");
207         EDONR_PERF_TEST(256);
208         EDONR_PERF_TEST(512);

210         return (0);
211 }
```

```
**********************************************************
   33078 Mon Oct  7 19:17:11 2013
new/usr/src/common/crypto/sha2/sha2.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2013 Saso Kiselkov.  All rights reserved.
 */

/*
 * The basic framework for this code came from the reference
 * implementation for MD5.  That implementation is Copyright (C)
 * 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.
 *
 * License to copy and use this software is granted provided that it
 * is identified as the "RSA Data Security, Inc. MD5 Message-Digest
 * Algorithm" in all material mentioning or referencing this software
 * or this function.
 *
 * License is also granted to make and use derivative works provided
 * that such works are identified as "derived from the RSA Data
 * Security, Inc. MD5 Message-Digest Algorithm" in all material
 * mentioning or referencing the derived work.
 *
 * RSA Data Security, Inc. makes no representations concerning either
 * the merchantability of this software or the suitability of this
 * software for any particular purpose. It is provided "as is"
 * without express or implied warranty of any kind.
 *
 * These notices must be retained in any copies of any part of this
 * documentation and/or software.
 *
 * NOTE: Cleaned-up and optimized, version of SHA2, based on the FIPS 180-2
 * standard, available at
 * http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf
 * Not as fast as one would like -- further optimizations are encouraged
 * and appreciated.
 */

#ifndef _KERNEL
#include <stdint.h>
#include <strings.h>
#include <stdlib.h>
#include <errno.h>
#endif /* _KERNEL */

#include <sys/types.h>
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/sysmacros.h>
#define _SHA2_IMPL
#include <sys/sha2.h>
#include <sys/sha2_consts.h>

#ifdef _KERNEL
#include <sys/cmn_err.h>

#else
#pragma weak SHA256Update = SHA2Update
#pragma weak SHA384Update = SHA2Update
#pragma weak SHA512Update = SHA2Update
```



```
#pragma weak SHA256Final = SHA2Final
#pragma weak SHA384Final = SHA2Final
#pragma weak SHA512Final = SHA2Final

#endif  /* _KERNEL */

#ifdef _LITTLE_ENDIAN
#include <sys/byteorder.h>
#define HAVE_HTONL
#endif

static void Encode(uint8_t *, uint32_t *, size_t);
static void Encode64(uint8_t *, uint64_t *, size_t);

#if     defined(__amd64)
#define SHA512Transform(ctx, in) SHA512TransformBlocks((ctx), (in), 1)
#define SHA256Transform(ctx, in) SHA256TransformBlocks((ctx), (in), 1)

void SHA512TransformBlocks(SHA2_CTX *ctx, const void *in, size_t num);
void SHA256TransformBlocks(SHA2_CTX *ctx, const void *in, size_t num);

#else
static void SHA256Transform(SHA2_CTX *, const uint8_t *);
static void SHA512Transform(SHA2_CTX *, const uint8_t *);
#endif  /* __amd64 */

static uint8_t PADDING[128] = { 0x80, /* all zeros */ };

/* Ch and Maj are the basic SHA2 functions. */
#define Ch(b, c, d)     (((b) & (c)) ^ ((~b) & (d)))
#define Maj(b, c, d)    (((b) & (c)) ^ ((b) & (d)) ^ ((c) & (d)))

/* Rotates x right n bits. */
#define ROTR(x, n)      \
        (((x) >> (n)) | ((x) << ((sizeof (x) * NBBY)-(n))))

/* Shift x right n bits */
#define SHR(x, n)       ((x) >> (n))

/* SHA256 Functions */
#define BIGSIGMA0_256(x)        (ROTR((x), 2) ^ ROTR((x), 13) ^ ROTR((x), 22))
#define BIGSIGMA1_256(x)        (ROTR((x), 6) ^ ROTR((x), 11) ^ ROTR((x), 25))
#define SIGMA0_256(x)           (ROTR((x), 7) ^ ROTR((x), 18) ^ SHR((x), 3))
#define SIGMA1_256(x)           (ROTR((x), 17) ^ ROTR((x), 19) ^ SHR((x), 10))

#define SHA256ROUND(a, b, c, d, e, f, g, h, i, w)                       \
        T1 = h + BIGSIGMA1_256(e) + Ch(e, f, g) + SHA256_CONST(i) + w;  \
        d += T1;                                                        \
        T2 = BIGSIGMA0_256(a) + Maj(a, b, c);                           \
        h = T1 + T2

/* SHA384/512 Functions */
#define BIGSIGMA0(x)    (ROTR((x), 28) ^ ROTR((x), 34) ^ ROTR((x), 39))
#define BIGSIGMA1(x)    (ROTR((x), 14) ^ ROTR((x), 18) ^ ROTR((x), 41))
#define SIGMA0(x)       (ROTR((x), 1) ^ ROTR((x), 8) ^ SHR((x), 7))
#define SIGMA1(x)       (ROTR((x), 19) ^ ROTR((x), 61) ^ SHR((x), 6))
#define SHA512ROUND(a, b, c, d, e, f, g, h, i, w)                       \
        T1 = h + BIGSIGMA1(e) + Ch(e, f, g) + SHA512_CONST(i) + w;      \
        d += T1;                                                        \
        T2 = BIGSIGMA0(a) + Maj(a, b, c);                               \
        h = T1 + T2

/*
 * sparc optimization:
 *
 * on the sparc, we can load big endian 32-bit data easily.  note that
```

```
 128  * special care must be taken to ensure the address is 32-bit aligned.
 129  * in the interest of speed, we don't check to make sure, since
 130  * careful programming can guarantee this for us.
 131  */

 133 #if     defined(_BIG_ENDIAN)
 134 #define LOAD_BIG_32(addr)       (*(uint32_t *)(addr))
 135 #define LOAD_BIG_64(addr)       (*(uint64_t *)(addr))

 137 #elif   defined(HAVE_HTONL)
 138 #define LOAD_BIG_32(addr) htonl(*((uint32_t *)(addr)))
 139 #define LOAD_BIG_64(addr) htonll(*((uint64_t *)(addr)))

 141 #else
 142 /* little endian -- will work on big endian, but slowly */
 143 #define LOAD_BIG_32(addr)       \
 144         (((addr)[0] << 24) | ((addr)[1] << 16) | ((addr)[2] << 8) | (addr)[3])
 145 #define LOAD_BIG_64(addr)       \
 146         (((uint64_t)(addr)[0] << 56) | ((uint64_t)(addr)[1] << 48) |    \
 147             ((uint64_t)(addr)[2] << 40) | ((uint64_t)(addr)[3] << 32) | \
 148             ((uint64_t)(addr)[4] << 24) | ((uint64_t)(addr)[5] << 16) | \
 149             ((uint64_t)(addr)[6] << 8) | (uint64_t)(addr)[7])
 150 #endif  /* _BIG_ENDIAN */


 153 #if     !defined(__amd64)
 154 /* SHA256 Transform */

 156 static void
 157 SHA256Transform(SHA2_CTX *ctx, const uint8_t *blk)
 158 {
 159         uint32_t a = ctx->state.s32[0];
 160         uint32_t b = ctx->state.s32[1];
 161         uint32_t c = ctx->state.s32[2];
 162         uint32_t d = ctx->state.s32[3];
 163         uint32_t e = ctx->state.s32[4];
 164         uint32_t f = ctx->state.s32[5];
 165         uint32_t g = ctx->state.s32[6];
 166         uint32_t h = ctx->state.s32[7];

 168         uint32_t w0, w1, w2, w3, w4, w5, w6, w7;
 169         uint32_t w8, w9, w10, w11, w12, w13, w14, w15;
 170         uint32_t T1, T2;

 172 #if     defined(__sparc)
 173         static const uint32_t sha256_consts[] = {
 174                 SHA256_CONST_0, SHA256_CONST_1, SHA256_CONST_2,
 175                 SHA256_CONST_3, SHA256_CONST_4, SHA256_CONST_5,
 176                 SHA256_CONST_6, SHA256_CONST_7, SHA256_CONST_8,
 177                 SHA256_CONST_9, SHA256_CONST_10, SHA256_CONST_11,
 178                 SHA256_CONST_12, SHA256_CONST_13, SHA256_CONST_14,
 179                 SHA256_CONST_15, SHA256_CONST_16, SHA256_CONST_17,
 180                 SHA256_CONST_18, SHA256_CONST_19, SHA256_CONST_20,
 181                 SHA256_CONST_21, SHA256_CONST_22, SHA256_CONST_23,
 182                 SHA256_CONST_24, SHA256_CONST_25, SHA256_CONST_26,
 183                 SHA256_CONST_27, SHA256_CONST_28, SHA256_CONST_29,
 184                 SHA256_CONST_30, SHA256_CONST_31, SHA256_CONST_32,
 185                 SHA256_CONST_33, SHA256_CONST_34, SHA256_CONST_35,
 186                 SHA256_CONST_36, SHA256_CONST_37, SHA256_CONST_38,
 187                 SHA256_CONST_39, SHA256_CONST_40, SHA256_CONST_41,
 188                 SHA256_CONST_42, SHA256_CONST_43, SHA256_CONST_44,
 189                 SHA256_CONST_45, SHA256_CONST_46, SHA256_CONST_47,
 190                 SHA256_CONST_48, SHA256_CONST_49, SHA256_CONST_50,
 191                 SHA256_CONST_51, SHA256_CONST_52, SHA256_CONST_53,
 192                 SHA256_CONST_54, SHA256_CONST_55, SHA256_CONST_56,
 193                 SHA256_CONST_57, SHA256_CONST_58, SHA256_CONST_59,
```



```
 194                 SHA256_CONST_60, SHA256_CONST_61, SHA256_CONST_62,
 195                 SHA256_CONST_63
 196         };
 197 #endif  /* __sparc */

 199         if ((uintptr_t)blk & 0x3) {             /* not 4-byte aligned? */
 200                 bcopy(blk, ctx->buf_un.buf32,  sizeof (ctx->buf_un.buf32));
 201                 blk = (uint8_t *)ctx->buf_un.buf32;
 202         }

 204         /* LINTED E_BAD_PTR_CAST_ALIGN */
 205         w0 =  LOAD_BIG_32(blk + 4 * 0);
 206         SHA256ROUND(a, b, c, d, e, f, g, h, 0, w0);
 207         /* LINTED E_BAD_PTR_CAST_ALIGN */
 208         w1 =  LOAD_BIG_32(blk + 4 * 1);
 209         SHA256ROUND(h, a, b, c, d, e, f, g, 1, w1);
 210         /* LINTED E_BAD_PTR_CAST_ALIGN */
 211         w2 =  LOAD_BIG_32(blk + 4 * 2);
 212         SHA256ROUND(g, h, a, b, c, d, e, f, 2, w2);
 213         /* LINTED E_BAD_PTR_CAST_ALIGN */
 214         w3 =  LOAD_BIG_32(blk + 4 * 3);
 215         SHA256ROUND(f, g, h, a, b, c, d, e, 3, w3);
 216         /* LINTED E_BAD_PTR_CAST_ALIGN */
 217         w4 =  LOAD_BIG_32(blk + 4 * 4);
 218         SHA256ROUND(e, f, g, h, a, b, c, d, 4, w4);
 219         /* LINTED E_BAD_PTR_CAST_ALIGN */
 220         w5 =  LOAD_BIG_32(blk + 4 * 5);
 221         SHA256ROUND(d, e, f, g, h, a, b, c, 5, w5);
 222         /* LINTED E_BAD_PTR_CAST_ALIGN */
 223         w6 =  LOAD_BIG_32(blk + 4 * 6);
 224         SHA256ROUND(c, d, e, f, g, h, a, b, 6, w6);
 225         /* LINTED E_BAD_PTR_CAST_ALIGN */
 226         w7 =  LOAD_BIG_32(blk + 4 * 7);
 227         SHA256ROUND(b, c, d, e, f, g, h, a, 7, w7);
 228         /* LINTED E_BAD_PTR_CAST_ALIGN */
 229         w8 =  LOAD_BIG_32(blk + 4 * 8);
 230         SHA256ROUND(a, b, c, d, e, f, g, h, 8, w8);
 231         /* LINTED E_BAD_PTR_CAST_ALIGN */
 232         w9 =  LOAD_BIG_32(blk + 4 * 9);
 233         SHA256ROUND(h, a, b, c, d, e, f, g, 9, w9);
 234         /* LINTED E_BAD_PTR_CAST_ALIGN */
 235         w10 =  LOAD_BIG_32(blk + 4 * 10);
 236         SHA256ROUND(g, h, a, b, c, d, e, f, 10, w10);
 237         /* LINTED E_BAD_PTR_CAST_ALIGN */
 238         w11 =  LOAD_BIG_32(blk + 4 * 11);
 239         SHA256ROUND(f, g, h, a, b, c, d, e, 11, w11);
 240         /* LINTED E_BAD_PTR_CAST_ALIGN */
 241         w12 =  LOAD_BIG_32(blk + 4 * 12);
 242         SHA256ROUND(e, f, g, h, a, b, c, d, 12, w12);
 243         /* LINTED E_BAD_PTR_CAST_ALIGN */
 244         w13 =  LOAD_BIG_32(blk + 4 * 13);
 245         SHA256ROUND(d, e, f, g, h, a, b, c, 13, w13);
 246         /* LINTED E_BAD_PTR_CAST_ALIGN */
 247         w14 =  LOAD_BIG_32(blk + 4 * 14);
 248         SHA256ROUND(c, d, e, f, g, h, a, b, 14, w14);
 249         /* LINTED E_BAD_PTR_CAST_ALIGN */
 250         w15 =  LOAD_BIG_32(blk + 4 * 15);
 251         SHA256ROUND(b, c, d, e, f, g, h, a, 15, w15);

 253         w0 = SIGMA1_256(w14) + w9 + SIGMA0_256(w1) + w0;
 254         SHA256ROUND(a, b, c, d, e, f, g, h, 16, w0);
 255         w1 = SIGMA1_256(w15) + w10 + SIGMA0_256(w2) + w1;
 256         SHA256ROUND(h, a, b, c, d, e, f, g, 17, w1);
 257         w2 = SIGMA1_256(w0) + w11 + SIGMA0_256(w3) + w2;
 258         SHA256ROUND(g, h, a, b, c, d, e, f, 18, w2);
 259         w3 = SIGMA1_256(w1) + w12 + SIGMA0_256(w4) + w3;
```

```
260         SHA256ROUND(f, g, h, a, b, c, d, e, 19, w3);
261         w4 = SIGMA1_256(w2) + w13 + SIGMA0_256(w5) + w4;
262         SHA256ROUND(e, f, g, h, a, b, c, d, 20, w4);
263         w5 = SIGMA1_256(w3) + w14 + SIGMA0_256(w6) + w5;
264         SHA256ROUND(d, e, f, g, h, a, b, c, 21, w5);
265         w6 = SIGMA1_256(w4) + w15 + SIGMA0_256(w7) + w6;
266         SHA256ROUND(c, d, e, f, g, h, a, b, 22, w6);
267         w7 = SIGMA1_256(w5) + w0 + SIGMA0_256(w8) + w7;
268         SHA256ROUND(b, c, d, e, f, g, h, a, 23, w7);
269         w8 = SIGMA1_256(w6) + w1 + SIGMA0_256(w9) + w8;
270         SHA256ROUND(a, b, c, d, e, f, g, h, 24, w8);
271         w9 = SIGMA1_256(w7) + w2 + SIGMA0_256(w10) + w9;
272         SHA256ROUND(h, a, b, c, d, e, f, g, 25, w9);
273         w10 = SIGMA1_256(w8) + w3 + SIGMA0_256(w11) + w10;
274         SHA256ROUND(g, h, a, b, c, d, e, f, 26, w10);
275         w11 = SIGMA1_256(w9) + w4 + SIGMA0_256(w12) + w11;
276         SHA256ROUND(f, g, h, a, b, c, d, e, 27, w11);
277         w12 = SIGMA1_256(w10) + w5 + SIGMA0_256(w13) + w12;
278         SHA256ROUND(e, f, g, h, a, b, c, d, 28, w12);
279         w13 = SIGMA1_256(w11) + w6 + SIGMA0_256(w14) + w13;
280         SHA256ROUND(d, e, f, g, h, a, b, c, 29, w13);
281         w14 = SIGMA1_256(w12) + w7 + SIGMA0_256(w15) + w14;
282         SHA256ROUND(c, d, e, f, g, h, a, b, 30, w14);
283         w15 = SIGMA1_256(w13) + w8 + SIGMA0_256(w0) + w15;
284         SHA256ROUND(b, c, d, e, f, g, h, a, 31, w15);

286         w0 = SIGMA1_256(w14) + w9 + SIGMA0_256(w1) + w0;
287         SHA256ROUND(a, b, c, d, e, f, g, h, 32, w0);
288         w1 = SIGMA1_256(w15) + w10 + SIGMA0_256(w2) + w1;
289         SHA256ROUND(h, a, b, c, d, e, f, g, 33, w1);
290         w2 = SIGMA1_256(w0) + w11 + SIGMA0_256(w3) + w2;
291         SHA256ROUND(g, h, a, b, c, d, e, f, 34, w2);
292         w3 = SIGMA1_256(w1) + w12 + SIGMA0_256(w4) + w3;
293         SHA256ROUND(f, g, h, a, b, c, d, e, 35, w3);
294         w4 = SIGMA1_256(w2) + w13 + SIGMA0_256(w5) + w4;
295         SHA256ROUND(e, f, g, h, a, b, c, d, 36, w4);
296         w5 = SIGMA1_256(w3) + w14 + SIGMA0_256(w6) + w5;
297         SHA256ROUND(d, e, f, g, h, a, b, c, 37, w5);
298         w6 = SIGMA1_256(w4) + w15 + SIGMA0_256(w7) + w6;
299         SHA256ROUND(c, d, e, f, g, h, a, b, 38, w6);
300         w7 = SIGMA1_256(w5) + w0 + SIGMA0_256(w8) + w7;
301         SHA256ROUND(b, c, d, e, f, g, h, a, 39, w7);
302         w8 = SIGMA1_256(w6) + w1 + SIGMA0_256(w9) + w8;
303         SHA256ROUND(a, b, c, d, e, f, g, h, 40, w8);
304         w9 = SIGMA1_256(w7) + w2 + SIGMA0_256(w10) + w9;
305         SHA256ROUND(h, a, b, c, d, e, f, g, 41, w9);
306         w10 = SIGMA1_256(w8) + w3 + SIGMA0_256(w11) + w10;
307         SHA256ROUND(g, h, a, b, c, d, e, f, 42, w10);
308         w11 = SIGMA1_256(w9) + w4 + SIGMA0_256(w12) + w11;
309         SHA256ROUND(f, g, h, a, b, c, d, e, 43, w11);
310         w12 = SIGMA1_256(w10) + w5 + SIGMA0_256(w13) + w12;
311         SHA256ROUND(e, f, g, h, a, b, c, d, 44, w12);
312         w13 = SIGMA1_256(w11) + w6 + SIGMA0_256(w14) + w13;
313         SHA256ROUND(d, e, f, g, h, a, b, c, 45, w13);
314         w14 = SIGMA1_256(w12) + w7 + SIGMA0_256(w15) + w14;
315         SHA256ROUND(c, d, e, f, g, h, a, b, 46, w14);
316         w15 = SIGMA1_256(w13) + w8 + SIGMA0_256(w0) + w15;
317         SHA256ROUND(b, c, d, e, f, g, h, a, 47, w15);

319         w0 = SIGMA1_256(w14) + w9 + SIGMA0_256(w1) + w0;
320         SHA256ROUND(a, b, c, d, e, f, g, h, 48, w0);
321         w1 = SIGMA1_256(w15) + w10 + SIGMA0_256(w2) + w1;
322         SHA256ROUND(h, a, b, c, d, e, f, g, 49, w1);
323         w2 = SIGMA1_256(w0) + w11 + SIGMA0_256(w3) + w2;
324         SHA256ROUND(g, h, a, b, c, d, e, f, 50, w2);
325         w3 = SIGMA1_256(w1) + w12 + SIGMA0_256(w4) + w3;
```



```
326         SHA256ROUND(f, g, h, a, b, c, d, e, 51, w3);
327         w4 = SIGMA1_256(w2) + w13 + SIGMA0_256(w5) + w4;
328         SHA256ROUND(e, f, g, h, a, b, c, d, 52, w4);
329         w5 = SIGMA1_256(w3) + w14 + SIGMA0_256(w6) + w5;
330         SHA256ROUND(d, e, f, g, h, a, b, c, 53, w5);
331         w6 = SIGMA1_256(w4) + w15 + SIGMA0_256(w7) + w6;
332         SHA256ROUND(c, d, e, f, g, h, a, b, 54, w6);
333         w7 = SIGMA1_256(w5) + w0 + SIGMA0_256(w8) + w7;
334         SHA256ROUND(b, c, d, e, f, g, h, a, 55, w7);
335         w8 = SIGMA1_256(w6) + w1 + SIGMA0_256(w9) + w8;
336         SHA256ROUND(a, b, c, d, e, f, g, h, 56, w8);
337         w9 = SIGMA1_256(w7) + w2 + SIGMA0_256(w10) + w9;
338         SHA256ROUND(h, a, b, c, d, e, f, g, 57, w9);
339         w10 = SIGMA1_256(w8) + w3 + SIGMA0_256(w11) + w10;
340         SHA256ROUND(g, h, a, b, c, d, e, f, 58, w10);
341         w11 = SIGMA1_256(w9) + w4 + SIGMA0_256(w12) + w11;
342         SHA256ROUND(f, g, h, a, b, c, d, e, 59, w11);
343         w12 = SIGMA1_256(w10) + w5 + SIGMA0_256(w13) + w12;
344         SHA256ROUND(e, f, g, h, a, b, c, d, 60, w12);
345         w13 = SIGMA1_256(w11) + w6 + SIGMA0_256(w14) + w13;
346         SHA256ROUND(d, e, f, g, h, a, b, c, 61, w13);
347         w14 = SIGMA1_256(w12) + w7 + SIGMA0_256(w15) + w14;
348         SHA256ROUND(c, d, e, f, g, h, a, b, 62, w14);
349         w15 = SIGMA1_256(w13) + w8 + SIGMA0_256(w0) + w15;
350         SHA256ROUND(b, c, d, e, f, g, h, a, 63, w15);

352         ctx->state.s32[0] += a;
353         ctx->state.s32[1] += b;
354         ctx->state.s32[2] += c;
355         ctx->state.s32[3] += d;
356         ctx->state.s32[4] += e;
357         ctx->state.s32[5] += f;
358         ctx->state.s32[6] += g;
359         ctx->state.s32[7] += h;
360 }
```

_____***unchanged_portion_omitted_***

```
680 void
681 SHA2Init(uint64_t mech, SHA2_CTX *ctx)
682 {

684         switch (mech) {
685         case SHA256_MECH_INFO_TYPE:
686         case SHA256_HMAC_MECH_INFO_TYPE:
687         case SHA256_HMAC_GEN_MECH_INFO_TYPE:
688                 ctx->state.s32[0] = 0x6a09e667U;
689                 ctx->state.s32[1] = 0xbb67ae85U;
690                 ctx->state.s32[2] = 0x3c6ef372U;
691                 ctx->state.s32[3] = 0xa54ff53aU;
692                 ctx->state.s32[4] = 0x510e527fU;
693                 ctx->state.s32[5] = 0x9b05688cU;
694                 ctx->state.s32[6] = 0x1f83d9abU;
695                 ctx->state.s32[7] = 0x5be0cd19U;
696                 break;
697         case SHA384_MECH_INFO_TYPE:
698         case SHA384_HMAC_MECH_INFO_TYPE:
699         case SHA384_HMAC_GEN_MECH_INFO_TYPE:
700                 ctx->state.s64[0] = 0xcbbb9d5dc1059ed8ULL;
701                 ctx->state.s64[1] = 0x629a292a367cd507ULL;
702                 ctx->state.s64[2] = 0x9159015a3070dd17ULL;
703                 ctx->state.s64[3] = 0x152fecd8f70e5939ULL;
704                 ctx->state.s64[4] = 0x67332667ffc00b31ULL;
705                 ctx->state.s64[5] = 0x8eb44a8768581511ULL;
706                 ctx->state.s64[6] = 0xdb0c2e0d64f98fa7ULL;
707                 ctx->state.s64[7] = 0x47b5481dbefa4fa4ULL;
```

```
                break;
        case SHA512_MECH_INFO_TYPE:
        case SHA512_HMAC_MECH_INFO_TYPE:
        case SHA512_HMAC_GEN_MECH_INFO_TYPE:
                ctx->state.s64[0] = 0x6a09e667f3bcc908ULL;
                ctx->state.s64[1] = 0xbb67ae8584caa73bULL;
                ctx->state.s64[2] = 0x3c6ef372fe94f82bULL;
                ctx->state.s64[3] = 0xa54ff53a5f1d36f1ULL;
                ctx->state.s64[4] = 0x510e527fade682d1ULL;
                ctx->state.s64[5] = 0x9b05688c2b3e6c1fULL;
                ctx->state.s64[6] = 0x1f83d9abfb41bd6bULL;
                ctx->state.s64[7] = 0x5be0cd19137e2179ULL;
                break;
        case SHA512_224_MECH_INFO_TYPE:
                ctx->state.s64[0] = 0x8C3D37C819544DA2ULL;
                ctx->state.s64[1] = 0x73E1996689DCD4D6ULL;
                ctx->state.s64[2] = 0x1DFAB7AE32FF9C82ULL;
                ctx->state.s64[3] = 0x679DD514582F9FCFULL;
                ctx->state.s64[4] = 0x0F6D2B697BD44DA8ULL;
                ctx->state.s64[5] = 0x77E36F7304C48942ULL;
                ctx->state.s64[6] = 0x3F9D85A86A1D36C8ULL;
                ctx->state.s64[7] = 0x1112E6AD91D692A1ULL;
                break;
        case SHA512_256_MECH_INFO_TYPE:
                ctx->state.s64[0] = 0x22312194FC2BF72CULL;
                ctx->state.s64[1] = 0x9F555FA3C84C64C2ULL;
                ctx->state.s64[2] = 0x2393B86B6F53B151ULL;
                ctx->state.s64[3] = 0x963877195940EABDULL;
                ctx->state.s64[4] = 0x96283EE2A88EFFE3ULL;
                ctx->state.s64[5] = 0xBE5E1E2553863992ULL;
                ctx->state.s64[6] = 0x2B0199FC2C85B8AAULL;
                ctx->state.s64[7] = 0x0EB72DDC81C52CA2ULL;
                break;
#ifdef _KERNEL
        default:
                cmn_err(CE_PANIC,
                    "sha2_init: failed to find a supported algorithm: 0x%x",
                    (uint32_t)mech);

#endif /* _KERNEL */
        }

        ctx->algotype = (uint32_t)mech;
        ctx->count.c64[0] = ctx->count.c64[1] = 0;
}
```

*_____unchanged_portion_omitted_*

```

/*
 * SHA2Final()
 *
 * purpose: ends an sha2 digest operation, finalizing the message digest and
 *          zeroing the context.
 *   input: uchar_t *   : a buffer to store the digest
 *                      : The function actually uses void* because many
 *                      : callers pass things other than uchar_t here.
 *          SHA2_CTX *  : the context to finalize, save, and zero
 *  output: void
 */

void
SHA2Final(void *digest, SHA2_CTX *ctx)
{
        uint8_t         bitcount_be[sizeof (ctx->count.c32)];
        uint8_t         bitcount_be64[sizeof (ctx->count.c64)];
        uint32_t        index;
```



```
        uint32_t        algotype = ctx->algotype;

        if (algotype <= SHA256_HMAC_GEN_MECH_INFO_TYPE) {
                index  = (ctx->count.c32[1] >> 3) & 0x3f;
                Encode(bitcount_be, ctx->count.c32, sizeof (bitcount_be));
                SHA2Update(ctx, PADDING, ((index < 56) ? 56 : 120) - index);
                SHA2Update(ctx, bitcount_be, sizeof (bitcount_be));
                Encode(digest, ctx->state.s32, sizeof (ctx->state.s32));

        } else {
                index  = (ctx->count.c64[1] >> 3) & 0x7f;
                Encode64(bitcount_be64, ctx->count.c64,
                    sizeof (bitcount_be64));
                SHA2Update(ctx, PADDING, ((index < 112) ? 112 : 240) - index);
                SHA2Update(ctx, bitcount_be64, sizeof (bitcount_be64));
                if (algotype <= SHA384_HMAC_GEN_MECH_INFO_TYPE) {
                        ctx->state.s64[6] = ctx->state.s64[7] = 0;
                        Encode64(digest, ctx->state.s64,
                            sizeof (uint64_t) * 6);
                } else if (algotype == SHA512_224_MECH_INFO_TYPE) {
                        uint8_t last[sizeof (uint64_t)];
                        /*
                         * Since SHA-512/224 doesn't align well to 64-bit
                         * boundaries, we must do the encode in three steps:
                         * 1) encode the three 64-bit words that fit neatly
                         * 2) encode the last 64-bit word to a temp buffer
                         * 3) chop out the lower 32-bits from the temp buffer
                         *    and append them to the digest
                         */
                        Encode64(digest, ctx->state.s64, sizeof (uint64_t) * 3);
                        Encode64(last, &ctx->state.s64[3], sizeof (uint64_t));
                        bcopy(last, (uint8_t *)digest + 24, 4);
                } else if (algotype == SHA512_256_MECH_INFO_TYPE) {
                        Encode64(digest, ctx->state.s64, sizeof (uint64_t) * 4);
                } else {
                } else
                        Encode64(digest, ctx->state.s64,
                            sizeof (ctx->state.s64));
                }
        }

        /* zeroize sensitive information */
        bzero(ctx, sizeof (*ctx));
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    2300 Mon Oct  7 19:17:12 2013
new/usr/src/common/crypto/sha2/test/Makefile
4185 New hash algorithm support
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://opensource.org/licenses/CDDL-1.0.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2013 Saso Kiselkov.  All rights reserved.
#
#       This makefile drives the production of the sha2 test suite.
#       To run the tests, simply type "make" or "make check".

include ../../../../Makefile.master

CFLAGS = -I${SRC}/uts/common -D_KERNEL -fno-builtin -O2 -W -Wall \
    -Wno-unknown-pragmas -Wno-unused -std=c99
CPUMHZ:sh = psrinfo -v 0 | awk '/processor operates at/{print $6}'
GCC = $(GCC_ROOT)/bin/gcc

PERL = perl
AS = /opt/onbld/bin/i386/aw
ASSFLAGS = -P -xarch=amd64 -P -Ui386 -U__i386 -D_KERNEL -D_SYSCALL32 \
    -D_SYSCALL32_IMPL -D_ELF64  -D_DDI_STRICT -Dsun -D__sun -D__SVR4 \
    -DDEBUG   -D_ASM -D__STDC__=0  -I../../../../uts/intel \
    -I../../../../uts/common
COMPILE.s = $(AS) $(ASSFLAGS)

check : sha2_test32_check sha2_test64_check
all : sha2_test32 sha2_test64

sha2_test32 : sha2_test.c
        @ echo Building 32-bit test...
        @ $(GCC) $(CFLAGS) -o sha2_test32 sha2_test.c ../sha2.c

sha2_test64 : sha2_test.c sha256-x86_64.o sha512-x86_64.o ../sha2.c
        @ echo Building 64-bit test...
        @ $(GCC) $(CFLAGS) -m64 -o $@ sha2_test.c sha256-x86_64.o \
            sha512-x86_64.o ../sha2.c

sha2_test32_check : sha2_test32
        @ echo Running 32-bit test...
        @ ./sha2_test32 $(CPUMHZ)

sha2_test64_check : sha2_test64
        @ echo Running 64-bit test...
        @ ./sha2_test64 $(CPUMHZ)

%.o: %.s
```



```
        @ $(COMPILE.s) -o $@ ${@F:.o=.s}

sha512-x86_64.s: ../amd64/sha512-x86_64.pl
        @ $(PERL) $? $@

sha256-x86_64.s: ../amd64/sha512-x86_64.pl
        @ $(PERL) $? $@

clean :
        rm -f sha2_test32 sha2_test64 *.o *.s
```

```
**********************************************************
    7537 Mon Oct  7 19:17:12 2013
new/usr/src/common/crypto/sha2/test/sha2_test.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://opensource.org/licenses/CDDL-1.0.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2013 Saso Kiselkov. All rights reserved.
 */

/*
 * This is just to keep the compiler happy about sys/time.h not declaring
 * gettimeofday due to -D_KERNEL (we can do this since we're actually
 * running in userspace, but we need -D_KERNEL for the remaining SHA2 code).
 */
#ifdef  _KERNEL
#undef  _KERNEL
#endif

#include <sys/varargs.h>
#include <stdlib.h>
#include <strings.h>
#include <stdio.h>
#include <sys/time.h>
#define _SHA2_IMPL
#include <sys/sha2.h>

/*
 * Test messages from:
 * http://csrc.nist.gov/groups/ST/toolkit/documents/Examples/SHA_All.pdf
 */

const char      *test_msg0 = "abc";
const char      *test_msg1 = "abcdbcdecdefdefgefghfghighijhijkijkljklmklmn"
        "lmnomnopnopq";
const char      *test_msg2 = "abcdefghbcdefghicdefghijdefghijkefghijklfghi"
        "jklmghijklmnhijklmnoijklmnopjklmnopqklmnopqrlmnopqrsmnopqrstnopqrstu";

/*
 * Test digests from:
 * http://csrc.nist.gov/groups/ST/toolkit/documents/Examples/SHA_All.pdf
 */
const uint8_t   sha256_test_digests[][32] = {
        {
                // for test_msg0
                0xBA, 0x78, 0x16, 0xBF, 0x8F, 0x01, 0xCF, 0xEA,
```



```
                0x41, 0x41, 0x40, 0xDE, 0x5D, 0xAE, 0x22, 0x23,
                0xB0, 0x03, 0x61, 0xA3, 0x96, 0x17, 0x7A, 0x9C,
                0xB4, 0x10, 0xFF, 0x61, 0xF2, 0x00, 0x15, 0xAD
        },
        {
                // for test_msg1
                0x24, 0x8D, 0x6A, 0x61, 0xD2, 0x06, 0x38, 0xB8,
                0xE5, 0xC0, 0x26, 0x93, 0x0C, 0x3E, 0x60, 0x39,
                0xA3, 0x3C, 0xE4, 0x59, 0x64, 0xFF, 0x21, 0x67,
                0xF6, 0xEC, 0xED, 0xD4, 0x19, 0xDB, 0x06, 0xC1
        },
        // no test vector for test_msg2
};

const uint8_t   sha384_test_digests[][48] = {
        {
                // for test_msg0
                0xCB, 0x00, 0x75, 0x3F, 0x45, 0xA3, 0x5E, 0x8B,
                0xB5, 0xA0, 0x3D, 0x69, 0x9A, 0xC6, 0x50, 0x07,
                0x27, 0x2C, 0x32, 0xAB, 0x0E, 0xDE, 0xD1, 0x63,
                0x1A, 0x8B, 0x60, 0x5A, 0x43, 0xFF, 0x5B, 0xED,
                0x80, 0x86, 0x07, 0x2B, 0xA1, 0xE7, 0xCC, 0x23,
                0x58, 0xBA, 0xEC, 0xA1, 0x34, 0xC8, 0x25, 0xA7
        },
        {
                // no test vector for test_msg1
        },
        {
                // for test_msg2
                0x09, 0x33, 0x0C, 0x33, 0xF7, 0x11, 0x47, 0xE8,
                0x3D, 0x19, 0x2F, 0xC7, 0x82, 0xCD, 0x1B, 0x47,
                0x53, 0x11, 0x1B, 0x17, 0x3B, 0x3B, 0x05, 0xD2,
                0x2F, 0xA0, 0x80, 0x86, 0xE3, 0xB0, 0xF7, 0x12,
                0xFC, 0xC7, 0xC7, 0x1A, 0x55, 0x7E, 0x2D, 0xB9,
                0x66, 0xC3, 0xE9, 0xFA, 0x91, 0x74, 0x60, 0x39
        }
};

const uint8_t   sha512_test_digests[][64] = {
        {
                // for test_msg0
                0xDD, 0xAF, 0x35, 0xA1, 0x93, 0x61, 0x7A, 0xBA,
                0xCC, 0x41, 0x73, 0x49, 0xAE, 0x20, 0x41, 0x31,
                0x12, 0xE6, 0xFA, 0x4E, 0x89, 0xA9, 0x7E, 0xA2,
                0x0A, 0x9E, 0xEE, 0xE6, 0x4B, 0x55, 0xD3, 0x9A,
                0x21, 0x92, 0x99, 0x2A, 0x27, 0x4F, 0xC1, 0xA8,
                0x36, 0xBA, 0x3C, 0x23, 0xA3, 0xFE, 0xEB, 0xBD,
                0x45, 0x4D, 0x44, 0x23, 0x64, 0x3C, 0xE8, 0x0E,
                0x2A, 0x9A, 0xC9, 0x4F, 0xA5, 0x4C, 0xA4, 0x9F
        },
        {
                // no test vector for test_msg1
        },
        {
                // for test_msg2
                0x8E, 0x95, 0x9B, 0x75, 0xDA, 0xE3, 0x13, 0xDA,
                0x8C, 0xF4, 0xF7, 0x28, 0x14, 0xFC, 0x14, 0x3F,
                0x8F, 0x77, 0x79, 0xC6, 0xEB, 0x9F, 0x7F, 0xA1,
                0x72, 0x99, 0xAE, 0xAD, 0xB6, 0x88, 0x90, 0x18,
                0x50, 0x1D, 0x28, 0x9E, 0x49, 0x00, 0xF7, 0xE4,
                0x33, 0x1B, 0x99, 0xDE, 0xC4, 0xB5, 0x43, 0x3A,
                0xC7, 0xD3, 0x29, 0xEE, 0xB6, 0xDD, 0x26, 0x54,
                0x5E, 0x96, 0xE5, 0x5B, 0x87, 0x4B, 0xE9, 0x09
        }
};
```

```
128 const uint8_t   sha512_224_test_digests[][28] = {
129         {
130                 // for test_msg0
131                 0x46, 0x34, 0x27, 0x0F, 0x70, 0x7B, 0x6A, 0x54,
132                 0xDA, 0xAE, 0x75, 0x30, 0x46, 0x08, 0x42, 0xE2,
133                 0x0E, 0x37, 0xED, 0x26, 0x5C, 0xEE, 0xE9, 0xA4,
134                 0x3E, 0x89, 0x24, 0xAA
135         },
136         {
137                 // no test vector for test_msg1
138         },
139         {
140                 // for test_msg2
141                 0x23, 0xFE, 0xC5, 0xBB, 0x94, 0xD6, 0x0B, 0x23,
142                 0x30, 0x81, 0x92, 0x64, 0x0B, 0x0C, 0x45, 0x33,
143                 0x35, 0xD6, 0x64, 0x73, 0x4F, 0xE4, 0x0E, 0x72,
144                 0x68, 0x67, 0x4A, 0xF9
145         }
146 };

148 const uint8_t   sha512_256_test_digests[][32] = {
149         {
150                 // for test_msg0
151                 0x53, 0x04, 0x8E, 0x26, 0x81, 0x94, 0x1E, 0xF9,
152                 0x9B, 0x2E, 0x29, 0xB7, 0x6B, 0x4C, 0x7D, 0xAB,
153                 0xE4, 0xC2, 0xD0, 0xC6, 0x34, 0xFC, 0x6D, 0x46,
154                 0xE0, 0xE2, 0xF1, 0x31, 0x07, 0xE7, 0xAF, 0x23
155         },
156         {
157                 // no test vector for test_msg1
158         },
159         {
160                 // for test_msg2
161                 0x39, 0x28, 0xE1, 0x84, 0xFB, 0x86, 0x90, 0xF8,
162                 0x40, 0xDA, 0x39, 0x88, 0x12, 0x1D, 0x31, 0xBE,
163                 0x65, 0xCB, 0x9D, 0x3E, 0xF8, 0x3E, 0xE6, 0x14,
164                 0x6F, 0xEA, 0xC8, 0x61, 0xE1, 0x9B, 0x56, 0x3A
165         }
166 };

168 /*
169  * Local reimplementation of cmn_err, since it's used in sha2.c.
170  */
171 /*ARGSUSED*/
172 void
173 cmn_err(int level, char *format, ...)
174 {
175         va_list ap;
176         va_start(ap, format);
177         vfprintf(stderr, format, ap);
178         va_end(ap);
179 }

181 int
182 main(int argc, char *argv[])
183 {
184         boolean_t       failed = B_FALSE;
185         uint64_t        cpu_mhz = 0;

187         if (argc == 2)
188                 cpu_mhz = atoi(argv[1]);

190 #define SHA2_ALGO_TEST(_m, mode, diglen, testdigest)                    \
191         do {                                                            \
192                 SHA2_CTX                ctx;                            \
193                 uint8_t                 digest[diglen / 8];             \
```



```
194                 SHA2Init(SHA ## mode ## _MECH_INFO_TYPE, &ctx);         \
195                 SHA2Update(&ctx, _m, strlen(_m));                       \
196                 SHA2Final(digest, &ctx);                                \
197                 printf("SHA%-9sMessage: " #_m "\tResult: ", #mode);     \
198                 if (bcmp(digest, testdigest, diglen / 8) == 0) {        \
199                         printf("OK\n");                                 \
200                 } else {                                                \
201                         printf("FAILED!\n");                            \
202                         failed = B_TRUE;                                \
203                 }                                                       \
204         } while (0)

206 #define SHA2_PERF_TEST(mode, diglen)                                    \
207         do {                                                            \
208                 SHA2_CTX        ctx;                                    \
209                 uint8_t         digest[diglen / 8];                     \
210                 uint8_t         block[131072];                          \
211                 uint64_t        delta;                                  \
212                 double          cpb = 0;                                \
213                 int             i;                                      \
214                 struct timeval  start, end;                             \
215                 bzero(block, sizeof (block));                           \
216                 gettimeofday(&start, NULL);                             \
217                 SHA2Init(SHA ## mode ## _MECH_INFO_TYPE, &ctx);         \
218                 for (i = 0; i < 8192; i++)                              \
219                         SHA2Update(&ctx, block, sizeof (block));        \
220                 SHA2Final(digest, &ctx);                                \
221                 gettimeofday(&end, NULL);                               \
222                 delta = (end.tv_sec * 1000000llu + end.tv_usec) -       \
223                     (start.tv_sec * 1000000llu + start.tv_usec);        \
224                 if (cpu_mhz != 0) {                                     \
225                         cpb = (cpu_mhz * 1e6 * ((double)delta /         \
226                             1000000)) / (8192 * 128 * 1024);            \
227                 }                                                       \
228                 printf("SHA%-9s%llu us (%.02f CPB)\n", #mode, delta, cpb);\
229         } while (0)

231         printf("Running algorithm correctness tests:\n");
232         SHA2_ALGO_TEST(test_msg0, 256, 256, sha256_test_digests[0]);
233         SHA2_ALGO_TEST(test_msg1, 256, 256, sha256_test_digests[1]);
234         SHA2_ALGO_TEST(test_msg0, 384, 384, sha384_test_digests[0]);
235         SHA2_ALGO_TEST(test_msg2, 384, 384, sha384_test_digests[2]);
236         SHA2_ALGO_TEST(test_msg0, 512, 512, sha512_test_digests[0]);
237         SHA2_ALGO_TEST(test_msg2, 512, 512, sha512_test_digests[2]);
238         SHA2_ALGO_TEST(test_msg0, 512_224, 224, sha512_224_test_digests[0]);
239         SHA2_ALGO_TEST(test_msg2, 512_224, 224, sha512_224_test_digests[2]);
240         SHA2_ALGO_TEST(test_msg0, 512_256, 256, sha512_256_test_digests[0]);
241         SHA2_ALGO_TEST(test_msg2, 512_256, 256, sha512_256_test_digests[2]);

243         if (failed)
244                 return (1);

246         printf("Running performance tests (hashing 1024 MiB of data):\n");
247         SHA2_PERF_TEST(256, 256);
248         SHA2_PERF_TEST(512, 512);

250         return (0);
251 }
```

****
148 Mon Oct 7 19:17:12 2013
new/usr/src/common/crypto/skein/THIRDPARTYLICENSE
4185 New hash algorithm support
****

```
1 Implementation of the Skein hash function.
2 Source code author: Doug Whiting, 2008.
3 This algorithm and source code is released to the public domain.
```

```
**********************************************************
      53 Mon Oct  7 19:17:12 2013
new/usr/src/common/crypto/skein/THIRDPARTYLICENSE.descrip
4185 New hash algorithm support
**********************************************************
```

```
   1 LICENSE TERMS OF SKEIN HASH ALGORITHM IMPLEMENTATION
```

```
**********************************************************
   28815 Mon Oct  7 19:17:12 2013
new/usr/src/common/crypto/skein/skein.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * Implementation of the Skein hash function.
 * Source code author: Doug Whiting, 2008.
 * This algorithm and source code is released to the public domain.
 */
/* Copyright 2013 Doug Whiting. This code is released to the public domain. */

#define SKEIN_PORT_CODE         /* instantiate any code in skein_port.h */

#include <sys/types.h>
#include <sys/note.h>
#include <sys/skein.h>          /* get the Skein API definitions   */
#include "skein_impl.h"         /* get internal definitions */
#include "skein_iv.h"           /* get precomputed IVs */

/* External function to process blkCnt (nonzero) full block(s) of data. */
void Skein_256_Process_Block(Skein_256_Ctxt_t *ctx, const uint8_t *blkPtr,
    size_t blkCnt, size_t byteCntAdd);
void Skein_512_Process_Block(Skein_512_Ctxt_t *ctx, const uint8_t *blkPtr,
    size_t blkCnt, size_t byteCntAdd);
void Skein1024_Process_Block(Skein1024_Ctxt_t *ctx, const uint8_t *blkPtr,
    size_t blkCnt, size_t byteCntAdd);

/* 256-bit Skein */
/* init the context for a straight hashing operation  */
int
Skein_256_Init(Skein_256_Ctxt_t *ctx, size_t hashBitLen)
{
        union {
                uint8_t b[SKEIN_256_STATE_BYTES];
                uint64_t w[SKEIN_256_STATE_WORDS];
        } cfg;                  /* config block */

        Skein_Assert(hashBitLen > 0, SKEIN_BAD_HASHLEN);
        ctx->h.hashBitLen = hashBitLen; /* output hash bit count */

        switch (hashBitLen) {   /* use pre-computed values, where available */
#ifndef SKEIN_NO_PRECOMP
        case 256:
                bcopy(SKEIN_256_IV_256, ctx->X, sizeof (ctx->X));
                break;
        case 224:
                bcopy(SKEIN_256_IV_224, ctx->X, sizeof (ctx->X));
                break;
        case 160:
                bcopy(SKEIN_256_IV_160, ctx->X, sizeof (ctx->X));
                break;
        case 128:
                bcopy(SKEIN_256_IV_128, ctx->X, sizeof (ctx->X));
                break;
#endif
        default:
                /* here if there is no precomputed IV value available */
                /*
                 * build/process the config block, type == CONFIG (could be
                 * precomputed)
                 */
                /* set tweaks: T0=0; T1=CFG | FINAL */
                Skein_Start_New_Type(ctx, CFG_FINAL);

                /* set the schema, version */
```



```
                cfg.w[0] = Skein_Swap64(SKEIN_SCHEMA_VER);
                /* hash result length in bits */
                cfg.w[1] = Skein_Swap64(hashBitLen);
                cfg.w[2] = Skein_Swap64(SKEIN_CFG_TREE_INFO_SEQUENTIAL);
                /* zero pad config block */
                bzero(&cfg.w[3], sizeof (cfg) - 3 * sizeof (cfg.w[0]));

                /* compute the initial chaining values from config block */
                /* zero the chaining variables */
                bzero(ctx->X, sizeof (ctx->X));
                Skein_256_Process_Block(ctx, cfg.b, 1, SKEIN_CFG_STR_LEN);
                break;
        }
        /*
         * The chaining vars ctx->X are now initialized for the given
         * hashBitLen.
         * Set up to process the data message portion of the hash (default)
         */
        Skein_Start_New_Type(ctx, MSG); /* T0=0, T1= MSG type */

        return (SKEIN_SUCCESS);
}

/* init the context for a MAC and/or tree hash operation */
/*
 * [identical to Skein_256_Init() when keyBytes == 0 &&
 * treeInfo == SKEIN_CFG_TREE_INFO_SEQUENTIAL]
 */
int
Skein_256_InitExt(Skein_256_Ctxt_t *ctx, size_t hashBitLen, uint64_t treeInfo,
    const uint8_t *key, size_t keyBytes)
{
        union {
                uint8_t b[SKEIN_256_STATE_BYTES];
                uint64_t w[SKEIN_256_STATE_WORDS];
        } cfg;                  /* config block */

        Skein_Assert(hashBitLen > 0, SKEIN_BAD_HASHLEN);
        Skein_Assert(keyBytes == 0 || key != NULL, SKEIN_FAIL);

        /* compute the initial chaining values ctx->X[], based on key */
        if (keyBytes == 0) {    /* is there a key? */
                /* no key: use all zeroes as key for config block */
                bzero(ctx->X, sizeof (ctx->X));
        } else {                /* here to pre-process a key */

                Skein_assert(sizeof (cfg.b) >= sizeof (ctx->X));
                /* do a mini-Init right here */
                /* set output hash bit count = state size */
                ctx->h.hashBitLen = 8 * sizeof (ctx->X);
                /* set tweaks: T0 = 0; T1 = KEY type */
                Skein_Start_New_Type(ctx, KEY);
                /* zero the initial chaining variables */
                bzero(ctx->X, sizeof (ctx->X));
                /* hash the key */
                (void) Skein_256_Update(ctx, key, keyBytes);
                /* put result into cfg.b[] */
                (void) Skein_256_Final_Pad(ctx, cfg.b);
                /* copy over into ctx->X[] */
                bcopy(cfg.b, ctx->X, sizeof (cfg.b));
#if     SKEIN_NEED_SWAP
                {
                        uint_t i;
                        /* convert key bytes to context words */
                        for (i = 0; i < SKEIN_256_STATE_WORDS; i++)
                                ctx->X[i] = Skein_Swap64(ctx->X[i]);
```

```
128                 }
129 #endif
130         }
131         /*
132          * build/process the config block, type == CONFIG (could be
133          * precomputed for each key)
134          */
135         ctx->h.hashBitLen = hashBitLen; /* output hash bit count */
136         Skein_Start_New_Type(ctx, CFG_FINAL);
137 
138         bzero(&cfg.w, sizeof (cfg.w));  /* pre-pad cfg.w[] with zeroes */
139         cfg.w[0] = Skein_Swap64(SKEIN_SCHEMA_VER);
140         cfg.w[1] = Skein_Swap64(hashBitLen);    /* hash result length in bits */
141         /* tree hash config info (or SKEIN_CFG_TREE_INFO_SEQUENTIAL) */
142         cfg.w[2] = Skein_Swap64(treeInfo);
143 
144         Skein_Show_Key(256, &ctx->h, key, keyBytes);
145 
146         /* compute the initial chaining values from config block */
147         Skein_256_Process_Block(ctx, cfg.b, 1, SKEIN_CFG_STR_LEN);
148 
149         /* The chaining vars ctx->X are now initialized */
150         /* Set up to process the data message portion of the hash (default) */
151         ctx->h.bCnt = 0;        /* buffer b[] starts out empty */
152         Skein_Start_New_Type(ctx, MSG);
153 
154         return (SKEIN_SUCCESS);
155 }
156 
157 /* process the input bytes */
158 int
159 Skein_256_Update(Skein_256_Ctxt_t *ctx, const uint8_t *msg, size_t msgByteCnt)
160 {
161         size_t n;
162 
163         /* catch uninitialized context */
164         Skein_Assert(ctx->h.bCnt <= SKEIN_256_BLOCK_BYTES, SKEIN_FAIL);
165 
166         /* process full blocks, if any */
167         if (msgByteCnt + ctx->h.bCnt > SKEIN_256_BLOCK_BYTES) {
168                 /* finish up any buffered message data */
169                 if (ctx->h.bCnt) {
170                         /* # bytes free in buffer b[] */
171                         n = SKEIN_256_BLOCK_BYTES - ctx->h.bCnt;
172                         if (n) {
173                                 /* check on our logic here */
174                                 Skein_assert(n < msgByteCnt);
175                                 bcopy(msg, &ctx->b[ctx->h.bCnt], n);
176                                 msgByteCnt -= n;
177                                 msg += n;
178                                 ctx->h.bCnt += n;
179                         }
180                         Skein_assert(ctx->h.bCnt == SKEIN_256_BLOCK_BYTES);
181                         Skein_256_Process_Block(ctx, ctx->b, 1,
182                             SKEIN_256_BLOCK_BYTES);
183                         ctx->h.bCnt = 0;
184                 }
185                 /*
186                  * now process any remaining full blocks, directly from input
187                  * message data
188                  */
189                 if (msgByteCnt > SKEIN_256_BLOCK_BYTES) {
190                         /* number of full blocks to process */
191                         n = (msgByteCnt - 1) / SKEIN_256_BLOCK_BYTES;
192                         Skein_256_Process_Block(ctx, msg, n,
193                             SKEIN_256_BLOCK_BYTES);
```



```
194                         msgByteCnt -= n * SKEIN_256_BLOCK_BYTES;
195                         msg += n * SKEIN_256_BLOCK_BYTES;
196                 }
197                 Skein_assert(ctx->h.bCnt == 0);
198         }
199 
200         /* copy any remaining source message data bytes into b[] */
201         if (msgByteCnt) {
202                 Skein_assert(msgByteCnt + ctx->h.bCnt <= SKEIN_256_BLOCK_BYTES);
203                 bcopy(msg, &ctx->b[ctx->h.bCnt], msgByteCnt);
204                 ctx->h.bCnt += msgByteCnt;
205         }
206 
207         return (SKEIN_SUCCESS);
208 }
209 
210 /* finalize the hash computation and output the result */
211 int
212 Skein_256_Final(Skein_256_Ctxt_t *ctx, uint8_t *hashVal)
213 {
214         size_t i, n, byteCnt;
215         uint64_t X[SKEIN_256_STATE_WORDS];
216 
217         /* catch uninitialized context */
218         Skein_Assert(ctx->h.bCnt <= SKEIN_256_BLOCK_BYTES, SKEIN_FAIL);
219 
220         ctx->h.T[1] |= SKEIN_T1_FLAG_FINAL;     /* tag as the final block */
221         /* zero pad b[] if necessary */
222         if (ctx->h.bCnt < SKEIN_256_BLOCK_BYTES)
223                 bzero(&ctx->b[ctx->h.bCnt],
224                     SKEIN_256_BLOCK_BYTES - ctx->h.bCnt);
225 
226         /* process the final block */
227         Skein_256_Process_Block(ctx, ctx->b, 1, ctx->h.bCnt);
228 
229         /* now output the result */
230         /* total number of output bytes */
231         byteCnt = (ctx->h.hashBitLen + 7) >> 3;
232 
233         /* run Threefish in "counter mode" to generate output */
234         /* zero out b[], so it can hold the counter */
235         bzero(ctx->b, sizeof (ctx->b));
236         /* keep a local copy of counter mode "key" */
237         bcopy(ctx->X, X, sizeof (X));
238         for (i = 0; i * SKEIN_256_BLOCK_BYTES < byteCnt; i++) {
239                 /* build the counter block */
240                 uint64_t tmp = Skein_Swap64((uint64_t)i);
241                 bcopy(&tmp, ctx->b, sizeof (tmp));
242                 Skein_Start_New_Type(ctx, OUT_FINAL);
243                 /* run "counter mode" */
244                 Skein_256_Process_Block(ctx, ctx->b, 1, sizeof (uint64_t));
245                 /* number of output bytes left to go */
246                 n = byteCnt - i * SKEIN_256_BLOCK_BYTES;
247                 if (n >= SKEIN_256_BLOCK_BYTES)
248                         n = SKEIN_256_BLOCK_BYTES;
249                 Skein_Put64_LSB_First(hashVal + i * SKEIN_256_BLOCK_BYTES,
250                     ctx->X, n); /* "output" the ctr mode bytes */
251                 Skein_Show_Final(256, &ctx->h, n,
252                     hashVal + i * SKEIN_256_BLOCK_BYTES);
253                 /* restore the counter mode key for next time */
254                 bcopy(X, ctx->X, sizeof (X));
255         }
256         return (SKEIN_SUCCESS);
257 }
258 
259 /* 512-bit Skein */
```

```
/* init the context for a straight hashing operation  */
int
Skein_512_Init(Skein_512_Ctxt_t *ctx, size_t hashBitLen)
{
        union {
                uint8_t b[SKEIN_512_STATE_BYTES];
                uint64_t w[SKEIN_512_STATE_WORDS];
        } cfg;                  /* config block */

        Skein_Assert(hashBitLen > 0, SKEIN_BAD_HASHLEN);
        ctx->h.hashBitLen = hashBitLen; /* output hash bit count */

        switch (hashBitLen) {   /* use pre-computed values, where available */
#ifndef SKEIN_NO_PRECOMP
        case 512:
                bcopy(SKEIN_512_IV_512, ctx->X, sizeof (ctx->X));
                break;
        case 384:
                bcopy(SKEIN_512_IV_384, ctx->X, sizeof (ctx->X));
                break;
        case 256:
                bcopy(SKEIN_512_IV_256, ctx->X, sizeof (ctx->X));
                break;
        case 224:
                bcopy(SKEIN_512_IV_224, ctx->X, sizeof (ctx->X));
                break;
#endif
        default:
                /*
                 * here if there is no precomputed IV value available
                 * build/process the config block, type == CONFIG (could be
                 * precomputed)
                 */
                /* set tweaks: T0=0; T1=CFG | FINAL */
                Skein_Start_New_Type(ctx, CFG_FINAL);

                /* set the schema, version */
                cfg.w[0] = Skein_Swap64(SKEIN_SCHEMA_VER);
                /* hash result length in bits */
                cfg.w[1] = Skein_Swap64(hashBitLen);
                cfg.w[2] = Skein_Swap64(SKEIN_CFG_TREE_INFO_SEQUENTIAL);
                /* zero pad config block */
                bzero(&cfg.w[3], sizeof (cfg) - 3 * sizeof (cfg.w[0]));

                /* compute the initial chaining values from config block */
                /* zero the chaining variables */
                bzero(ctx->X, sizeof (ctx->X));
                Skein_512_Process_Block(ctx, cfg.b, 1, SKEIN_CFG_STR_LEN);
                break;
        }

        /*
         * The chaining vars ctx->X are now initialized for the given
         * hashBitLen. Set up to process the data message portion of the
         * hash (default)
         */
        Skein_Start_New_Type(ctx, MSG); /* T0=0, T1= MSG type */

        return (SKEIN_SUCCESS);
}

/* init the context for a MAC and/or tree hash operation */
/*
 * [identical to Skein_512_Init() when keyBytes == 0 &&
 * treeInfo == SKEIN_CFG_TREE_INFO_SEQUENTIAL]
```



```
 */
int
Skein_512_InitExt(Skein_512_Ctxt_t *ctx, size_t hashBitLen, uint64_t treeInfo,
    const uint8_t *key, size_t keyBytes)
{
        union {
                uint8_t b[SKEIN_512_STATE_BYTES];
                uint64_t w[SKEIN_512_STATE_WORDS];
        } cfg;                  /* config block */

        Skein_Assert(hashBitLen > 0, SKEIN_BAD_HASHLEN);
        Skein_Assert(keyBytes == 0 || key != NULL, SKEIN_FAIL);

        /* compute the initial chaining values ctx->X[], based on key */
        if (keyBytes == 0) {    /* is there a key? */
                /* no key: use all zeroes as key for config block */
                bzero(ctx->X, sizeof (ctx->X));
        } else {                /* here to pre-process a key */

                Skein_assert(sizeof (cfg.b) >= sizeof (ctx->X));
                /* do a mini-Init right here */
                /* set output hash bit count = state size */
                ctx->h.hashBitLen = 8 * sizeof (ctx->X);
                /* set tweaks: T0 = 0; T1 = KEY type */
                Skein_Start_New_Type(ctx, KEY);
                /* zero the initial chaining variables */
                bzero(ctx->X, sizeof (ctx->X));
                (void) Skein_512_Update(ctx, key, keyBytes); /* hash the key */
                /* put result into cfg.b[] */
                (void) Skein_512_Final_Pad(ctx, cfg.b);
                /* copy over into ctx->X[] */
                bcopy(cfg.b, ctx->X, sizeof (cfg.b));
#if     SKEIN_NEED_SWAP
                {
                        uint_t i;
                        /* convert key bytes to context words */
                        for (i = 0; i < SKEIN_512_STATE_WORDS; i++)
                                ctx->X[i] = Skein_Swap64(ctx->X[i]);
                }
#endif
        }
        /*
         * build/process the config block, type == CONFIG (could be
         * precomputed for each key)
         */
        ctx->h.hashBitLen = hashBitLen; /* output hash bit count */
        Skein_Start_New_Type(ctx, CFG_FINAL);

        bzero(&cfg.w, sizeof (cfg.w));  /* pre-pad cfg.w[] with zeroes */
        cfg.w[0] = Skein_Swap64(SKEIN_SCHEMA_VER);
        cfg.w[1] = Skein_Swap64(hashBitLen);    /* hash result length in bits */
        /* tree hash config info (or SKEIN_CFG_TREE_INFO_SEQUENTIAL) */
        cfg.w[2] = Skein_Swap64(treeInfo);

        Skein_Show_Key(512, &ctx->h, key, keyBytes);

        /* compute the initial chaining values from config block */
        Skein_512_Process_Block(ctx, cfg.b, 1, SKEIN_CFG_STR_LEN);

        /* The chaining vars ctx->X are now initialized */
        /* Set up to process the data message portion of the hash (default) */
        ctx->h.bCnt = 0;        /* buffer b[] starts out empty */
        Skein_Start_New_Type(ctx, MSG);

        return (SKEIN_SUCCESS);
}
```

```
393 /* process the input bytes */
394 int
395 Skein_512_Update(Skein_512_Ctxt_t *ctx, const uint8_t *msg, size_t msgByteCnt)
396 {
397         size_t n;

399         /* catch uninitialized context */
400         Skein_Assert(ctx->h.bCnt <= SKEIN_512_BLOCK_BYTES, SKEIN_FAIL);

402         /* process full blocks, if any */
403         if (msgByteCnt + ctx->h.bCnt > SKEIN_512_BLOCK_BYTES) {
404                 /* finish up any buffered message data */
405                 if (ctx->h.bCnt) {
406                         /* # bytes free in buffer b[] */
407                         n = SKEIN_512_BLOCK_BYTES - ctx->h.bCnt;
408                         if (n) {
409                                 /* check on our logic here */
410                                 Skein_assert(n < msgByteCnt);
411                                 bcopy(msg, &ctx->b[ctx->h.bCnt], n);
412                                 msgByteCnt -= n;
413                                 msg += n;
414                                 ctx->h.bCnt += n;
415                         }
416                         Skein_assert(ctx->h.bCnt == SKEIN_512_BLOCK_BYTES);
417                         Skein_512_Process_Block(ctx, ctx->b, 1,
418                             SKEIN_512_BLOCK_BYTES);
419                         ctx->h.bCnt = 0;
420                 }
421                 /*
422                  * now process any remaining full blocks, directly from input
423                  * message data
424                  */
425                 if (msgByteCnt > SKEIN_512_BLOCK_BYTES) {
426                         /* number of full blocks to process */
427                         n = (msgByteCnt - 1) / SKEIN_512_BLOCK_BYTES;
428                         Skein_512_Process_Block(ctx, msg, n,
429                             SKEIN_512_BLOCK_BYTES);
430                         msgByteCnt -= n * SKEIN_512_BLOCK_BYTES;
431                         msg += n * SKEIN_512_BLOCK_BYTES;
432                 }
433                 Skein_assert(ctx->h.bCnt == 0);
434         }

436         /* copy any remaining source message data bytes into b[] */
437         if (msgByteCnt) {
438                 Skein_assert(msgByteCnt + ctx->h.bCnt <= SKEIN_512_BLOCK_BYTES);
439                 bcopy(msg, &ctx->b[ctx->h.bCnt], msgByteCnt);
440                 ctx->h.bCnt += msgByteCnt;
441         }

443         return (SKEIN_SUCCESS);
444 }

446 /* finalize the hash computation and output the result */
447 int
448 Skein_512_Final(Skein_512_Ctxt_t *ctx, uint8_t *hashVal)
449 {
450         size_t i, n, byteCnt;
451         uint64_t X[SKEIN_512_STATE_WORDS];

453         /* catch uninitialized context */
454         Skein_Assert(ctx->h.bCnt <= SKEIN_512_BLOCK_BYTES, SKEIN_FAIL);

456         ctx->h.T[1] |= SKEIN_T1_FLAG_FINAL;     /* tag as the final block */
457         /* zero pad b[] if necessary */
```



```
458         if (ctx->h.bCnt < SKEIN_512_BLOCK_BYTES)
459                 bzero(&ctx->b[ctx->h.bCnt],
460                     SKEIN_512_BLOCK_BYTES - ctx->h.bCnt);

462         /* process the final block */
463         Skein_512_Process_Block(ctx, ctx->b, 1, ctx->h.bCnt);

465         /* now output the result */
466         /* total number of output bytes */
467         byteCnt = (ctx->h.hashBitLen + 7) >> 3;

469         /* run Threefish in "counter mode" to generate output */
470         /* zero out b[], so it can hold the counter */
471         bzero(ctx->b, sizeof (ctx->b));
472         /* keep a local copy of counter mode "key" */
473         bcopy(ctx->X, X, sizeof (X));
474         for (i = 0; i * SKEIN_512_BLOCK_BYTES < byteCnt; i++) {
475                 /* build the counter block */
476                 uint64_t tmp = Skein_Swap64((uint64_t)i);
477                 bcopy(&tmp, ctx->b, sizeof (tmp));
478                 Skein_Start_New_Type(ctx, OUT_FINAL);
479                 /* run "counter mode" */
480                 Skein_512_Process_Block(ctx, ctx->b, 1, sizeof (uint64_t));
481                 /* number of output bytes left to go */
482                 n = byteCnt - i * SKEIN_512_BLOCK_BYTES;
483                 if (n >= SKEIN_512_BLOCK_BYTES)
484                         n = SKEIN_512_BLOCK_BYTES;
485                 Skein_Put64_LSB_First(hashVal + i * SKEIN_512_BLOCK_BYTES,
486                     ctx->X, n); /* "output" the ctr mode bytes */
487                 Skein_Show_Final(512, &ctx->h, n,
488                     hashVal + i * SKEIN_512_BLOCK_BYTES);
489                 /* restore the counter mode key for next time */
490                 bcopy(X, ctx->X, sizeof (X));
491         }
492         return (SKEIN_SUCCESS);
493 }

495 /* 1024-bit Skein */

497 /* init the context for a straight hashing operation  */
498 int
499 Skein1024_Init(Skein1024_Ctxt_t *ctx, size_t hashBitLen)
500 {
501         union {
502                 uint8_t b[SKEIN1024_STATE_BYTES];
503                 uint64_t w[SKEIN1024_STATE_WORDS];
504         } cfg;                  /* config block */

506         Skein_Assert(hashBitLen > 0, SKEIN_BAD_HASHLEN);
507         ctx->h.hashBitLen = hashBitLen; /* output hash bit count */

509         switch (hashBitLen) {   /* use pre-computed values, where available */
510 #ifndef SKEIN_NO_PRECOMP
511         case 512:
512                 bcopy(SKEIN1024_IV_512, ctx->X, sizeof (ctx->X));
513                 break;
514         case 384:
515                 bcopy(SKEIN1024_IV_384, ctx->X, sizeof (ctx->X));
516                 break;
517         case 1024:
518                 bcopy(SKEIN1024_IV_1024, ctx->X, sizeof (ctx->X));
519                 break;
520 #endif
521         default:
522                 /* here if there is no precomputed IV value available */
523                 /*
```

```
524                  * build/process the config block, type == CONFIG (could be
525                  * precomputed)
526                  */
527                 /* set tweaks: T0=0; T1=CFG | FINAL */
528                 Skein_Start_New_Type(ctx, CFG_FINAL);
529
530                 /* set the schema, version */
531                 cfg.w[0] = Skein_Swap64(SKEIN_SCHEMA_VER);
532                 /* hash result length in bits */
533                 cfg.w[1] = Skein_Swap64(hashBitLen);
534                 cfg.w[2] = Skein_Swap64(SKEIN_CFG_TREE_INFO_SEQUENTIAL);
535                 /* zero pad config block */
536                 bzero(&cfg.w[3], sizeof (cfg) - 3 * sizeof (cfg.w[0]));
537
538                 /* compute the initial chaining values from config block */
539                 /* zero the chaining variables */
540                 bzero(ctx->X, sizeof (ctx->X));
541                 Skein1024_Process_Block(ctx, cfg.b, 1, SKEIN_CFG_STR_LEN);
542                 break;
543         }
544
545         /*
546          * The chaining vars ctx->X are now initialized for the given
547          * hashBitLen. Set up to process the data message portion of the hash
548          * (default)
549          */
550         Skein_Start_New_Type(ctx, MSG); /* T0=0, T1= MSG type */
551
552         return (SKEIN_SUCCESS);
553 }
554
555 /* init the context for a MAC and/or tree hash operation */
556 /*
557  * [identical to Skein1024_Init() when keyBytes == 0 &&
558  * treeInfo == SKEIN_CFG_TREE_INFO_SEQUENTIAL]
559  */
560 int
561 Skein1024_InitExt(Skein1024_Ctxt_t *ctx, size_t hashBitLen, uint64_t treeInfo,
562     const uint8_t *key, size_t keyBytes)
563 {
564         union {
565                 uint8_t b[SKEIN1024_STATE_BYTES];
566                 uint64_t w[SKEIN1024_STATE_WORDS];
567         } cfg;                  /* config block */
568
569         Skein_Assert(hashBitLen > 0, SKEIN_BAD_HASHLEN);
570         Skein_Assert(keyBytes == 0 || key != NULL, SKEIN_FAIL);
571
572         /* compute the initial chaining values ctx->X[], based on key */
573         if (keyBytes == 0) {    /* is there a key? */
574                 /* no key: use all zeroes as key for config block */
575                 bzero(ctx->X, sizeof (ctx->X));
576         } else {                /* here to pre-process a key */
577                 Skein_assert(sizeof (cfg.b) >= sizeof (ctx->X));
578                 /* do a mini-Init right here */
579                 /* set output hash bit count = state size */
580                 ctx->h.hashBitLen = 8 * sizeof (ctx->X);
581                 /* set tweaks: T0 = 0; T1 = KEY type */
582                 Skein_Start_New_Type(ctx, KEY);
583                 /* zero the initial chaining variables */
584                 bzero(ctx->X, sizeof (ctx->X));
585                 (void) Skein1024_Update(ctx, key, keyBytes); /* hash the key */
586                 /* put result into cfg.b[] */
587                 (void) Skein1024_Final_Pad(ctx, cfg.b);
588                 /* copy over into ctx->X[] */
589                 bcopy(cfg.b, ctx->X, sizeof (cfg.b));
```



```
590 #if     SKEIN_NEED_SWAP
591                 {
592                         uint_t i;
593                         /* convert key bytes to context words */
594                         for (i = 0; i < SKEIN1024_STATE_WORDS; i++)
595                                 ctx->X[i] = Skein_Swap64(ctx->X[i]);
596                 }
597 #endif
598         }
599         /*
600          * build/process the config block, type == CONFIG (could be
601          * precomputed for each key)
602          */
603         ctx->h.hashBitLen = hashBitLen; /* output hash bit count */
604         Skein_Start_New_Type(ctx, CFG_FINAL);
605
606         bzero(&cfg.w, sizeof (cfg.w));  /* pre-pad cfg.w[] with zeroes */
607         cfg.w[0] = Skein_Swap64(SKEIN_SCHEMA_VER);
608         /* hash result length in bits */
609         cfg.w[1] = Skein_Swap64(hashBitLen);
610         /* tree hash config info (or SKEIN_CFG_TREE_INFO_SEQUENTIAL) */
611         cfg.w[2] = Skein_Swap64(treeInfo);
612
613         Skein_Show_Key(1024, &ctx->h, key, keyBytes);
614
615         /* compute the initial chaining values from config block */
616         Skein1024_Process_Block(ctx, cfg.b, 1, SKEIN_CFG_STR_LEN);
617
618         /* The chaining vars ctx->X are now initialized */
619         /* Set up to process the data message portion of the hash (default) */
620         ctx->h.bCnt = 0;        /* buffer b[] starts out empty */
621         Skein_Start_New_Type(ctx, MSG);
622
623         return (SKEIN_SUCCESS);
624 }
625
626 /* process the input bytes */
627 int
628 Skein1024_Update(Skein1024_Ctxt_t *ctx, const uint8_t *msg, size_t msgByteCnt)
629 {
630         size_t n;
631
632         /* catch uninitialized context */
633         Skein_Assert(ctx->h.bCnt <= SKEIN1024_BLOCK_BYTES, SKEIN_FAIL);
634
635         /* process full blocks, if any */
636         if (msgByteCnt + ctx->h.bCnt > SKEIN1024_BLOCK_BYTES) {
637                 /* finish up any buffered message data */
638                 if (ctx->h.bCnt) {
639                         /* # bytes free in buffer b[] */
640                         n = SKEIN1024_BLOCK_BYTES - ctx->h.bCnt;
641                         if (n) {
642                                 /* check on our logic here */
643                                 Skein_assert(n < msgByteCnt);
644                                 bcopy(msg, &ctx->b[ctx->h.bCnt], n);
645                                 msgByteCnt -= n;
646                                 msg += n;
647                                 ctx->h.bCnt += n;
648                         }
649                         Skein_assert(ctx->h.bCnt == SKEIN1024_BLOCK_BYTES);
650                         Skein1024_Process_Block(ctx, ctx->b, 1,
651                             SKEIN1024_BLOCK_BYTES);
652                         ctx->h.bCnt = 0;
653                 }
654                 /*
655                  * now process any remaining full blocks, directly from
```

```
656                  * input message data
657                  */
658                 if (msgByteCnt > SKEIN1024_BLOCK_BYTES) {
659                         /* number of full blocks to process */
660                         n = (msgByteCnt - 1) / SKEIN1024_BLOCK_BYTES;
661                         Skein1024_Process_Block(ctx, msg, n,
662                             SKEIN1024_BLOCK_BYTES);
663                         msgByteCnt -= n * SKEIN1024_BLOCK_BYTES;
664                         msg += n * SKEIN1024_BLOCK_BYTES;
665                 }
666                 Skein_assert(ctx->h.bCnt == 0);
667         }
668
669         /* copy any remaining source message data bytes into b[] */
670         if (msgByteCnt) {
671                 Skein_assert(msgByteCnt + ctx->h.bCnt <= SKEIN1024_BLOCK_BYTES);
672                 bcopy(msg, &ctx->b[ctx->h.bCnt], msgByteCnt);
673                 ctx->h.bCnt += msgByteCnt;
674         }
675
676         return (SKEIN_SUCCESS);
677 }
678
679 /* finalize the hash computation and output the result */
680 int
681 Skein1024_Final(Skein1024_Ctxt_t *ctx, uint8_t *hashVal)
682 {
683         size_t i, n, byteCnt;
684         uint64_t X[SKEIN1024_STATE_WORDS];
685
686         /* catch uninitialized context */
687         Skein_Assert(ctx->h.bCnt <= SKEIN1024_BLOCK_BYTES, SKEIN_FAIL);
688
689         ctx->h.T[1] |= SKEIN_T1_FLAG_FINAL;     /* tag as the final block */
690         /* zero pad b[] if necessary */
691         if (ctx->h.bCnt < SKEIN1024_BLOCK_BYTES)
692                 bzero(&ctx->b[ctx->h.bCnt],
693                     SKEIN1024_BLOCK_BYTES - ctx->h.bCnt);
694
695         /* process the final block */
696         Skein1024_Process_Block(ctx, ctx->b, 1, ctx->h.bCnt);
697
698         /* now output the result */
699         /* total number of output bytes */
700         byteCnt = (ctx->h.hashBitLen + 7) >> 3;
701
702         /* run Threefish in "counter mode" to generate output */
703         /* zero out b[], so it can hold the counter */
704         bzero(ctx->b, sizeof (ctx->b));
705         /* keep a local copy of counter mode "key" */
706         bcopy(ctx->X, X, sizeof (X));
707         for (i = 0; i * SKEIN1024_BLOCK_BYTES < byteCnt; i++) {
708                 /* build the counter block */
709                 uint64_t tmp = Skein_Swap64((uint64_t)i);
710                 bcopy(&tmp, ctx->b, sizeof (tmp));
711                 Skein_Start_New_Type(ctx, OUT_FINAL);
712                 /* run "counter mode" */
713                 Skein1024_Process_Block(ctx, ctx->b, 1, sizeof (uint64_t));
714                 /* number of output bytes left to go */
715                 n = byteCnt - i * SKEIN1024_BLOCK_BYTES;
716                 if (n >= SKEIN1024_BLOCK_BYTES)
717                         n = SKEIN1024_BLOCK_BYTES;
718                 Skein_Put64_LSB_First(hashVal + i * SKEIN1024_BLOCK_BYTES,
719                     ctx->X, n); /* "output" the ctr mode bytes */
720                 Skein_Show_Final(1024, &ctx->h, n,
721                     hashVal + i * SKEIN1024_BLOCK_BYTES);
```



```
722                 /* restore the counter mode key for next time */
723                 bcopy(X, ctx->X, sizeof (X));
724         }
725         return (SKEIN_SUCCESS);
726 }
727
728 /* Functions to support MAC/tree hashing */
729 /* (this code is identical for Optimized and Reference versions) */
730
731 /* finalize the hash computation and output the block, no OUTPUT stage */
732 int
733 Skein_256_Final_Pad(Skein_256_Ctxt_t *ctx, uint8_t *hashVal)
734 {
735         /* catch uninitialized context */
736         Skein_Assert(ctx->h.bCnt <= SKEIN_256_BLOCK_BYTES, SKEIN_FAIL);
737
738         ctx->h.T[1] |= SKEIN_T1_FLAG_FINAL;     /* tag as the final block */
739         /* zero pad b[] if necessary */
740         if (ctx->h.bCnt < SKEIN_256_BLOCK_BYTES)
741                 bzero(&ctx->b[ctx->h.bCnt],
742                     SKEIN_256_BLOCK_BYTES - ctx->h.bCnt);
743         /* process the final block */
744         Skein_256_Process_Block(ctx, ctx->b, 1, ctx->h.bCnt);
745
746         /* "output" the state bytes */
747         Skein_Put64_LSB_First(hashVal, ctx->X, SKEIN_256_BLOCK_BYTES);
748
749         return (SKEIN_SUCCESS);
750 }
751
752 /* finalize the hash computation and output the block, no OUTPUT stage */
753 int
754 Skein_512_Final_Pad(Skein_512_Ctxt_t *ctx, uint8_t *hashVal)
755 {
756         /* catch uninitialized context */
757         Skein_Assert(ctx->h.bCnt <= SKEIN_512_BLOCK_BYTES, SKEIN_FAIL);
758
759         ctx->h.T[1] |= SKEIN_T1_FLAG_FINAL;     /* tag as the final block */
760         /* zero pad b[] if necessary */
761         if (ctx->h.bCnt < SKEIN_512_BLOCK_BYTES)
762                 bzero(&ctx->b[ctx->h.bCnt],
763                     SKEIN_512_BLOCK_BYTES - ctx->h.bCnt);
764         /* process the final block */
765         Skein_512_Process_Block(ctx, ctx->b, 1, ctx->h.bCnt);
766
767         /* "output" the state bytes */
768         Skein_Put64_LSB_First(hashVal, ctx->X, SKEIN_512_BLOCK_BYTES);
769
770         return (SKEIN_SUCCESS);
771 }
772
773 /* finalize the hash computation and output the block, no OUTPUT stage */
774 int
775 Skein1024_Final_Pad(Skein1024_Ctxt_t *ctx, uint8_t *hashVal)
776 {
777         /* catch uninitialized context */
778         Skein_Assert(ctx->h.bCnt <= SKEIN1024_BLOCK_BYTES, SKEIN_FAIL);
779
780         /* tag as the final block */
781         ctx->h.T[1] |= SKEIN_T1_FLAG_FINAL;
782         /* zero pad b[] if necessary */
783         if (ctx->h.bCnt < SKEIN1024_BLOCK_BYTES)
784                 bzero(&ctx->b[ctx->h.bCnt],
785                     SKEIN1024_BLOCK_BYTES - ctx->h.bCnt);
786         /* process the final block */
787         Skein1024_Process_Block(ctx, ctx->b, 1, ctx->h.bCnt);
```

```
789         /* "output" the state bytes */
790         Skein_Put64_LSB_First(hashVal, ctx->X, SKEIN1024_BLOCK_BYTES);
791
792         return (SKEIN_SUCCESS);
793 }
794
795 #if     SKEIN_TREE_HASH
796 /* just do the OUTPUT stage */
797 int
798 Skein_256_Output(Skein_256_Ctxt_t *ctx, uint8_t *hashVal)
799 {
800         size_t i, n, byteCnt;
801         uint64_t X[SKEIN_256_STATE_WORDS];
802
803         /* catch uninitialized context */
804         Skein_Assert(ctx->h.bCnt <= SKEIN_256_BLOCK_BYTES, SKEIN_FAIL);
805
806         /* now output the result */
807         /* total number of output bytes */
808         byteCnt = (ctx->h.hashBitLen + 7) >> 3;
809
810         /* run Threefish in "counter mode" to generate output */
811         /* zero out b[], so it can hold the counter */
812         bzero(ctx->b, sizeof (ctx->b));
813         /* keep a local copy of counter mode "key" */
814         bcopy(ctx->X, X, sizeof (X));
815         for (i = 0; i * SKEIN_256_BLOCK_BYTES < byteCnt; i++) {
816                 /* build the counter block */
817                 uint64_t tmp = Skein_Swap64((uint64_t)i);
818                 bcopy(&tmp, ctx->b, sizeof (tmp));
819                 Skein_Start_New_Type(ctx, OUT_FINAL);
820                 /* run "counter mode" */
821                 Skein_256_Process_Block(ctx, ctx->b, 1, sizeof (uint64_t));
822                 /* number of output bytes left to go */
823                 n = byteCnt - i * SKEIN_256_BLOCK_BYTES;
824                 if (n >= SKEIN_256_BLOCK_BYTES)
825                         n = SKEIN_256_BLOCK_BYTES;
826                 Skein_Put64_LSB_First(hashVal + i * SKEIN_256_BLOCK_BYTES,
827                     ctx->X, n); /* "output" the ctr mode bytes */
828                 Skein_Show_Final(256, &ctx->h, n,
829                     hashVal + i * SKEIN_256_BLOCK_BYTES);
830                 /* restore the counter mode key for next time */
831                 bcopy(X, ctx->X, sizeof (X));
832         }
833         return (SKEIN_SUCCESS);
834 }
835
836 /* just do the OUTPUT stage */
837 int
838 Skein_512_Output(Skein_512_Ctxt_t *ctx, uint8_t *hashVal)
839 {
840         size_t i, n, byteCnt;
841         uint64_t X[SKEIN_512_STATE_WORDS];
842
843         /* catch uninitialized context */
844         Skein_Assert(ctx->h.bCnt <= SKEIN_512_BLOCK_BYTES, SKEIN_FAIL);
845
846         /* now output the result */
847         /* total number of output bytes */
848         byteCnt = (ctx->h.hashBitLen + 7) >> 3;
849
850         /* run Threefish in "counter mode" to generate output */
851         /* zero out b[], so it can hold the counter */
852         bzero(ctx->b, sizeof (ctx->b));
853         /* keep a local copy of counter mode "key" */
```



```
854         bcopy(ctx->X, X, sizeof (X));
855         for (i = 0; i * SKEIN_512_BLOCK_BYTES < byteCnt; i++) {
856                 /* build the counter block */
857                 uint64_t tmp = Skein_Swap64((uint64_t)i);
858                 bcopy(&tmp, ctx->b, sizeof (tmp));
859                 Skein_Start_New_Type(ctx, OUT_FINAL);
860                 /* run "counter mode" */
861                 Skein_512_Process_Block(ctx, ctx->b, 1, sizeof (uint64_t));
862                 /* number of output bytes left to go */
863                 n = byteCnt - i * SKEIN_512_BLOCK_BYTES;
864                 if (n >= SKEIN_512_BLOCK_BYTES)
865                         n = SKEIN_512_BLOCK_BYTES;
866                 Skein_Put64_LSB_First(hashVal + i * SKEIN_512_BLOCK_BYTES,
867                     ctx->X, n); /* "output" the ctr mode bytes */
868                 Skein_Show_Final(256, &ctx->h, n,
869                     hashVal + i * SKEIN_512_BLOCK_BYTES);
870                 /* restore the counter mode key for next time */
871                 bcopy(X, ctx->X, sizeof (X));
872         }
873         return (SKEIN_SUCCESS);
874 }
875
876 /* just do the OUTPUT stage */
877 int
878 Skein1024_Output(Skein1024_Ctxt_t *ctx, uint8_t *hashVal)
879 {
880         size_t i, n, byteCnt;
881         uint64_t X[SKEIN1024_STATE_WORDS];
882
883         /* catch uninitialized context */
884         Skein_Assert(ctx->h.bCnt <= SKEIN1024_BLOCK_BYTES, SKEIN_FAIL);
885
886         /* now output the result */
887         /* total number of output bytes */
888         byteCnt = (ctx->h.hashBitLen + 7) >> 3;
889
890         /* run Threefish in "counter mode" to generate output */
891         /* zero out b[], so it can hold the counter */
892         bzero(ctx->b, sizeof (ctx->b));
893         /* keep a local copy of counter mode "key" */
894         bcopy(ctx->X, X, sizeof (X));
895         for (i = 0; i * SKEIN1024_BLOCK_BYTES < byteCnt; i++) {
896                 /* build the counter block */
897                 uint64_t tmp = Skein_Swap64((uint64_t)i);
898                 bcopy(&tmp, ctx->b, sizeof (tmp));
899                 Skein_Start_New_Type(ctx, OUT_FINAL);
900                 /* run "counter mode" */
901                 Skein1024_Process_Block(ctx, ctx->b, 1, sizeof (uint64_t));
902                 /* number of output bytes left to go */
903                 n = byteCnt - i * SKEIN1024_BLOCK_BYTES;
904                 if (n >= SKEIN1024_BLOCK_BYTES)
905                         n = SKEIN1024_BLOCK_BYTES;
906                 Skein_Put64_LSB_First(hashVal + i * SKEIN1024_BLOCK_BYTES,
907                     ctx->X, n); /* "output" the ctr mode bytes */
908                 Skein_Show_Final(256, &ctx->h, n,
909                     hashVal + i * SKEIN1024_BLOCK_BYTES);
910                 /* restore the counter mode key for next time */
911                 bcopy(X, ctx->X, sizeof (X));
912         }
913         return (SKEIN_SUCCESS);
914 }
915 #endif
```

```
**********************************************************
   21391 Mon Oct  7 19:17:12 2013
new/usr/src/common/crypto/skein/skein_block.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * Implementation of the Skein block functions.
 * Source code author: Doug Whiting, 2008.
 * This algorithm and source code is released to the public domain.
 * Compile-time switches:
 *  SKEIN_USE_ASM  -- set bits (256/512/1024) to select which
 *                    versions use ASM code for block processing
 *                    [default: use C for all block sizes]
 */
/* Copyright 2013 Doug Whiting. This code is released to the public domain. */

#include <sys/skein.h>
#include "skein_impl.h"

#ifndef SKEIN_USE_ASM
#define SKEIN_USE_ASM   (0)     /* default is all C code (no ASM) */
#endif

#ifndef SKEIN_LOOP
#define SKEIN_LOOP 001          /* default: unroll 256 and 512, but not 1024 */
#endif

/* some useful definitions for code here */
#define BLK_BITS        (WCNT*64)
#define KW_TWK_BASE     (0)
#define KW_KEY_BASE     (3)
#define ks              (kw + KW_KEY_BASE)
#define ts              (kw + KW_TWK_BASE)

/* no debugging in Illumos version */
#define DebugSaveTweak(ctx)

/* Skein_256 */
#if     !(SKEIN_USE_ASM & 256)
void
Skein_256_Process_Block(Skein_256_Ctxt_t *ctx, const uint8_t *blkPtr,
    size_t blkCnt, size_t byteCntAdd)
{                               /* do it in C */
        enum {
                WCNT = SKEIN_256_STATE_WORDS
        };
#undef  RCNT
#define RCNT  (SKEIN_256_ROUNDS_TOTAL / 8)

#ifdef  SKEIN_LOOP              /* configure how much to unroll the loop */
#define SKEIN_UNROLL_256 (((SKEIN_LOOP) / 100) % 10)
#else
#define SKEIN_UNROLL_256 (0)
#endif

#if     SKEIN_UNROLL_256
#if     (RCNT % SKEIN_UNROLL_256)
#error "Invalid SKEIN_UNROLL_256"       /* sanity check on unroll count */
#endif
        size_t r;
        /* key schedule words : chaining vars + tweak + "rotation" */
        uint64_t kw[WCNT + 4 + RCNT * 2];
#else
        uint64_t kw[WCNT + 4];  /* key schedule words : chaining vars + tweak */
#endif
        /* local copy of context vars, for speed */
```



```
        uint64_t X0, X1, X2, X3;
        uint64_t w[WCNT];               /* local copy of input block */
#ifdef  SKEIN_DEBUG
        /* use for debugging (help compiler put Xn in registers) */
        const uint64_t *Xptr[4];
        Xptr[0] = &X0;
        Xptr[1] = &X1;
        Xptr[2] = &X2;
        Xptr[3] = &X3;
#endif
        Skein_assert(blkCnt != 0);      /* never call with blkCnt == 0! */
        ts[0] = ctx->h.T[0];
        ts[1] = ctx->h.T[1];
        do {
                /*
                 * this implementation only supports 2**64 input bytes
                 * (no carry out here)
                 */
                ts[0] += byteCntAdd;    /* update processed length */

                /* precompute the key schedule for this block */
                ks[0] = ctx->X[0];
                ks[1] = ctx->X[1];
                ks[2] = ctx->X[2];
                ks[3] = ctx->X[3];
                ks[4] = ks[0] ^ ks[1] ^ ks[2] ^ ks[3] ^ SKEIN_KS_PARITY;

                ts[2] = ts[0] ^ ts[1];

                /* get input block in little-endian format */
                Skein_Get64_LSB_First(w, blkPtr, WCNT);
                DebugSaveTweak(ctx);
                Skein_Show_Block(BLK_BITS, &ctx->h, ctx->X, blkPtr, w, ks, ts);

                X0 = w[0] + ks[0];      /* do the first full key injection */
                X1 = w[1] + ks[1] + ts[0];
                X2 = w[2] + ks[2] + ts[1];
                X3 = w[3] + ks[3];

                Skein_Show_R_Ptr(BLK_BITS, &ctx->h, SKEIN_RND_KEY_INITIAL,
                    Xptr);      /* show starting state values */

                blkPtr += SKEIN_256_BLOCK_BYTES;

                /* run the rounds */

#define Round256(p0, p1, p2, p3, ROT, rNum)                             \
    X##p0 += X##p1; X##p1 = RotL_64(X##p1, ROT##_0); X##p1 ^= X##p0; \
    X##p2 += X##p3; X##p3 = RotL_64(X##p3, ROT##_1); X##p3 ^= X##p2; \

#if     SKEIN_UNROLL_256 == 0
#define R256(p0, p1, p2, p3, ROT, rNum)         /* fully unrolled */    \
    Round256(p0, p1, p2, p3, ROT, rNum)                                 \
    Skein_Show_R_Ptr(BLK_BITS, &ctx->h, rNum, Xptr);

#define I256(R)                                                         \
    X0 += ks[((R) + 1) % 5];    /* inject the key schedule value */     \
    X1 += ks[((R) + 2) % 5] + ts[((R) + 1) % 3];                        \
    X2 += ks[((R) + 3) % 5] + ts[((R) + 2) % 3];                        \
    X3 += ks[((R) + 4) % 5] + (R) + 1;                                  \
    Skein_Show_R_Ptr(BLK_BITS, &ctx->h, SKEIN_RND_KEY_INJECT, Xptr);
#else                           /* looping version */
#define R256(p0, p1, p2, p3, ROT, rNum)                                 \
    Round256(p0, p1, p2, p3, ROT, rNum)                                 \
    Skein_Show_R_Ptr(BLK_BITS, &ctx->h, 4 * (r - 1) + rNum, Xptr);
```

```
#define I256(R)                                                 \
        X0 += ks[r + (R) + 0];  /* inject the key schedule value */     \
        X1 += ks[r + (R) + 1] + ts[r + (R) + 0];                        \
        X2 += ks[r + (R) + 2] + ts[r + (R) + 1];                        \
        X3 += ks[r + (R) + 3] + r + (R);                                \
        ks[r + (R) + 4] = ks[r + (R) - 1];   /* rotate key schedule */  \
    ts[r + (R) + 2] = ts[r + (R) - 1];                                  \
    Skein_Show_R_Ptr(BLK_BITS, &ctx->h, SKEIN_RND_KEY_INJECT, Xptr);

                /* loop thru it */
                for (r = 1; r < 2 * RCNT; r += 2 * SKEIN_UNROLL_256)
#endif
                {
#define R256_8_rounds(R)                        \
        R256(0, 1, 2, 3, R_256_0, 8 * (R) + 1);  \
        R256(0, 3, 2, 1, R_256_1, 8 * (R) + 2);  \
        R256(0, 1, 2, 3, R_256_2, 8 * (R) + 3);  \
        R256(0, 3, 2, 1, R_256_3, 8 * (R) + 4);  \
        I256(2 * (R));                          \
        R256(0, 1, 2, 3, R_256_4, 8 * (R) + 5);  \
        R256(0, 3, 2, 1, R_256_5, 8 * (R) + 6);  \
        R256(0, 1, 2, 3, R_256_6, 8 * (R) + 7);  \
        R256(0, 3, 2, 1, R_256_7, 8 * (R) + 8);  \
        I256(2 * (R) + 1);

                        R256_8_rounds(0);

#define R256_Unroll_R(NN) \
        ((SKEIN_UNROLL_256 == 0 && SKEIN_256_ROUNDS_TOTAL / 8 > (NN)) || \
        (SKEIN_UNROLL_256 > (NN)))

#if     R256_Unroll_R(1)
                        R256_8_rounds(1);
#endif
#if     R256_Unroll_R(2)
                        R256_8_rounds(2);
#endif
#if     R256_Unroll_R(3)
                        R256_8_rounds(3);
#endif
#if     R256_Unroll_R(4)
                        R256_8_rounds(4);
#endif
#if     R256_Unroll_R(5)
                        R256_8_rounds(5);
#endif
#if     R256_Unroll_R(6)
                        R256_8_rounds(6);
#endif
#if     R256_Unroll_R(7)
                        R256_8_rounds(7);
#endif
#if     R256_Unroll_R(8)
                        R256_8_rounds(8);
#endif
#if     R256_Unroll_R(9)
                        R256_8_rounds(9);
#endif
#if     R256_Unroll_R(10)
                        R256_8_rounds(10);
#endif
#if     R256_Unroll_R(11)
                        R256_8_rounds(11);
#endif
#if     R256_Unroll_R(12)
                        R256_8_rounds(12);
```



```
#endif
#if     R256_Unroll_R(13)
                        R256_8_rounds(13);
#endif
#if     R256_Unroll_R(14)
                        R256_8_rounds(14);
#endif
#if     (SKEIN_UNROLL_256 > 14)
#error  "need more unrolling in Skein_256_Process_Block"
#endif
                }
                /*
                 * do the final "feedforward" xor, update context chaining vars
                 */
                ctx->X[0] = X0 ^ w[0];
                ctx->X[1] = X1 ^ w[1];
                ctx->X[2] = X2 ^ w[2];
                ctx->X[3] = X3 ^ w[3];

                Skein_Show_Round(BLK_BITS, &ctx->h, SKEIN_RND_FEED_FWD, ctx->X);

                ts[1] &= ~SKEIN_T1_FLAG_FIRST;
        }
        while (--blkCnt);
        ctx->h.T[0] = ts[0];
        ctx->h.T[1] = ts[1];
}

#if     defined(SKEIN_CODE_SIZE) || defined(SKEIN_PERF)
size_t
Skein_256_Process_Block_CodeSize(void)
{
        return ((uint8_t *)Skein_256_Process_Block_CodeSize) -
            ((uint8_t *)Skein_256_Process_Block);
}

uint_t
Skein_256_Unroll_Cnt(void)
{
        return (SKEIN_UNROLL_256);
}
#endif
#endif

/* Skein_512 */
#if     !(SKEIN_USE_ASM & 512)
void
Skein_512_Process_Block(Skein_512_Ctxt_t *ctx, const uint8_t *blkPtr,
    size_t blkCnt, size_t byteCntAdd)
{                               /* do it in C */
        enum {
                WCNT = SKEIN_512_STATE_WORDS
        };
#undef  RCNT
#define RCNT  (SKEIN_512_ROUNDS_TOTAL / 8)

#ifdef  SKEIN_LOOP              /* configure how much to unroll the loop */
#define SKEIN_UNROLL_512 (((SKEIN_LOOP) / 10) % 10)
#else
#define SKEIN_UNROLL_512 (0)
#endif

#if     SKEIN_UNROLL_512
#if     (RCNT % SKEIN_UNROLL_512)
#error "Invalid SKEIN_UNROLL_512"       /* sanity check on unroll count */
#endif
```

```
260         size_t r;
261         /* key schedule words : chaining vars + tweak + "rotation" */
262         uint64_t kw[WCNT + 4 + RCNT * 2];
263 #else
264         uint64_t kw[WCNT + 4];  /* key schedule words : chaining vars + tweak */
265 #endif
266         /* local copy of vars, for speed */
267         uint64_t X0, X1, X2, X3, X4, X5, X6, X7;
268         uint64_t w[WCNT];               /* local copy of input block */
269 #ifdef  SKEIN_DEBUG
270         /* use for debugging (help compiler put Xn in registers) */
271         const uint64_t *Xptr[8];
272         Xptr[0] = &X0;
273         Xptr[1] = &X1;
274         Xptr[2] = &X2;
275         Xptr[3] = &X3;
276         Xptr[4] = &X4;
277         Xptr[5] = &X5;
278         Xptr[6] = &X6;
279         Xptr[7] = &X7;
280 #endif

282         Skein_assert(blkCnt != 0);      /* never call with blkCnt == 0! */
283         ts[0] = ctx->h.T[0];
284         ts[1] = ctx->h.T[1];
285         do {
286                 /*
287                  * this implementation only supports 2**64 input bytes
288                  * (no carry out here)
289                  */
290                 ts[0] += byteCntAdd;    /* update processed length */

292                 /* precompute the key schedule for this block */
293                 ks[0] = ctx->X[0];
294                 ks[1] = ctx->X[1];
295                 ks[2] = ctx->X[2];
296                 ks[3] = ctx->X[3];
297                 ks[4] = ctx->X[4];
298                 ks[5] = ctx->X[5];
299                 ks[6] = ctx->X[6];
300                 ks[7] = ctx->X[7];
301                 ks[8] = ks[0] ^ ks[1] ^ ks[2] ^ ks[3] ^
302                     ks[4] ^ ks[5] ^ ks[6] ^ ks[7] ^ SKEIN_KS_PARITY;

304                 ts[2] = ts[0] ^ ts[1];

306                 /* get input block in little-endian format */
307                 Skein_Get64_LSB_First(w, blkPtr, WCNT);
308                 DebugSaveTweak(ctx);
309                 Skein_Show_Block(BLK_BITS, &ctx->h, ctx->X, blkPtr, w, ks, ts);

311                 X0 = w[0] + ks[0];      /* do the first full key injection */
312                 X1 = w[1] + ks[1];
313                 X2 = w[2] + ks[2];
314                 X3 = w[3] + ks[3];
315                 X4 = w[4] + ks[4];
316                 X5 = w[5] + ks[5] + ts[0];
317                 X6 = w[6] + ks[6] + ts[1];
318                 X7 = w[7] + ks[7];

320                 blkPtr += SKEIN_512_BLOCK_BYTES;

322                 Skein_Show_R_Ptr(BLK_BITS, &ctx->h, SKEIN_RND_KEY_INITIAL,
323                     Xptr);
324                 /* run the rounds */
325 #define Round512(p0, p1, p2, p3, p4, p5, p6, p7, ROT, rNum)             \
```



```
326         X##p0 += X##p1; X##p1 = RotL_64(X##p1, ROT##_0); X##p1 ^= X##p0;\
327         X##p2 += X##p3; X##p3 = RotL_64(X##p3, ROT##_1); X##p3 ^= X##p2;\
328         X##p4 += X##p5; X##p5 = RotL_64(X##p5, ROT##_2); X##p5 ^= X##p4;\
329         X##p6 += X##p7; X##p7 = RotL_64(X##p7, ROT##_3); X##p7 ^= X##p6;

331 #if     SKEIN_UNROLL_512 == 0
332 #define R512(p0, p1, p2, p3, p4, p5, p6, p7, ROT, rNum) /* unrolled */  \
333         Round512(p0, p1, p2, p3, p4, p5, p6, p7, ROT, rNum)             \
334         Skein_Show_R_Ptr(BLK_BITS, &ctx->h, rNum, Xptr);

336 #define I512(R)                                                         \
337         X0 += ks[((R) + 1) % 9];        /* inject the key schedule value */\
338         X1 += ks[((R) + 2) % 9];                                        \
339         X2 += ks[((R) + 3) % 9];                                        \
340         X3 += ks[((R) + 4) % 9];                                        \
341         X4 += ks[((R) + 5) % 9];                                        \
342         X5 += ks[((R) + 6) % 9] + ts[((R) + 1) % 3];                    \
343         X6 += ks[((R) + 7) % 9] + ts[((R) + 2) % 3];                    \
344         X7 += ks[((R) + 8) % 9] + (R) + 1;                              \
345         Skein_Show_R_Ptr(BLK_BITS, &ctx->h, SKEIN_RND_KEY_INJECT, Xptr);
346 #else                           /* looping version */
347 #define R512(p0, p1, p2, p3, p4, p5, p6, p7, ROT, rNum)                 \
348         Round512(p0, p1, p2, p3, p4, p5, p6, p7, ROT, rNum)             \
349         Skein_Show_R_Ptr(BLK_BITS, &ctx->h, 4 * (r - 1) + rNum, Xptr);

351 #define I512(R)                                                         \
352         X0 += ks[r + (R) + 0];  /* inject the key schedule value */     \
353         X1 += ks[r + (R) + 1];                                          \
354         X2 += ks[r + (R) + 2];                                          \
355         X3 += ks[r + (R) + 3];                                          \
356         X4 += ks[r + (R) + 4];                                          \
357         X5 += ks[r + (R) + 5] + ts[r + (R) + 0];                        \
358         X6 += ks[r + (R) + 6] + ts[r + (R) + 1];                        \
359         X7 += ks[r + (R) + 7] + r + (R);                                \
360         ks[r + (R)+8] = ks[r + (R) - 1];        /* rotate key schedule */\
361         ts[r + (R)+2] = ts[r + (R) - 1];                                \
362         Skein_Show_R_Ptr(BLK_BITS, &ctx->h, SKEIN_RND_KEY_INJECT, Xptr);

364                 /* loop thru it */
365                 for (r = 1; r < 2 * RCNT; r += 2 * SKEIN_UNROLL_512)
366 #endif                          /* end of looped code definitions */
367                 {
368 #define R512_8_rounds(R)        /* do 8 full rounds */                  \
369         R512(0, 1, 2, 3, 4, 5, 6, 7, R_512_0, 8 * (R) + 1);             \
370         R512(2, 1, 4, 7, 6, 5, 0, 3, R_512_1, 8 * (R) + 2);             \
371         R512(4, 1, 6, 3, 0, 5, 2, 7, R_512_2, 8 * (R) + 3);             \
372         R512(6, 1, 0, 7, 2, 5, 4, 3, R_512_3, 8 * (R) + 4);             \
373         I512(2 * (R));                                                  \
374         R512(0, 1, 2, 3, 4, 5, 6, 7, R_512_4, 8 * (R) + 5);             \
375         R512(2, 1, 4, 7, 6, 5, 0, 3, R_512_5, 8 * (R) + 6);             \
376         R512(4, 1, 6, 3, 0, 5, 2, 7, R_512_6, 8 * (R) + 7);             \
377         R512(6, 1, 0, 7, 2, 5, 4, 3, R_512_7, 8 * (R) + 8);             \
378         I512(2*(R) + 1);                /* and key injection */

380                         R512_8_rounds(0);

382 #define R512_Unroll_R(NN) \
383         ((SKEIN_UNROLL_512 == 0 && SKEIN_512_ROUNDS_TOTAL / 8 > (NN)) || \
384         (SKEIN_UNROLL_512 > (NN)))

386 #if     R512_Unroll_R(1)
387                         R512_8_rounds(1);
388 #endif
389 #if     R512_Unroll_R(2)
390                         R512_8_rounds(2);
391 #endif
```

```
#if     R512_Unroll_R(3)
                        R512_8_rounds(3);
#endif
#if     R512_Unroll_R(4)
                        R512_8_rounds(4);
#endif
#if     R512_Unroll_R(5)
                        R512_8_rounds(5);
#endif
#if     R512_Unroll_R(6)
                        R512_8_rounds(6);
#endif
#if     R512_Unroll_R(7)
                        R512_8_rounds(7);
#endif
#if     R512_Unroll_R(8)
                        R512_8_rounds(8);
#endif
#if     R512_Unroll_R(9)
                        R512_8_rounds(9);
#endif
#if     R512_Unroll_R(10)
                        R512_8_rounds(10);
#endif
#if     R512_Unroll_R(11)
                        R512_8_rounds(11);
#endif
#if     R512_Unroll_R(12)
                        R512_8_rounds(12);
#endif
#if     R512_Unroll_R(13)
                        R512_8_rounds(13);
#endif
#if     R512_Unroll_R(14)
                        R512_8_rounds(14);
#endif
#if     (SKEIN_UNROLL_512 > 14)
#error "need more unrolling in Skein_512_Process_Block"
#endif
                }

                /*
                 * do the final "feedforward" xor, update context chaining vars
                 */
                ctx->X[0] = X0 ^ w[0];
                ctx->X[1] = X1 ^ w[1];
                ctx->X[2] = X2 ^ w[2];
                ctx->X[3] = X3 ^ w[3];
                ctx->X[4] = X4 ^ w[4];
                ctx->X[5] = X5 ^ w[5];
                ctx->X[6] = X6 ^ w[6];
                ctx->X[7] = X7 ^ w[7];
                Skein_Show_Round(BLK_BITS, &ctx->h, SKEIN_RND_FEED_FWD, ctx->X);

                ts[1] &= ~SKEIN_T1_FLAG_FIRST;
        }
        while (--blkCnt);
        ctx->h.T[0] = ts[0];
        ctx->h.T[1] = ts[1];
}

#if     defined(SKEIN_CODE_SIZE) || defined(SKEIN_PERF)
size_t
Skein_512_Process_Block_CodeSize(void)
{
        return ((uint8_t *)Skein_512_Process_Block_CodeSize) -
```



```
            ((uint8_t *)Skein_512_Process_Block);
}

uint_t
Skein_512_Unroll_Cnt(void)
{
        return (SKEIN_UNROLL_512);
}
#endif
#endif

/*  Skein1024 */
#if     !(SKEIN_USE_ASM & 1024)
void
Skein1024_Process_Block(Skein1024_Ctxt_t *ctx, const uint8_t *blkPtr,
    size_t blkCnt, size_t byteCntAdd)
{
        /* do it in C, always looping (unrolled is bigger AND slower!) */
        enum {
                WCNT = SKEIN1024_STATE_WORDS
        };
#undef  RCNT
#define RCNT  (SKEIN1024_ROUNDS_TOTAL/8)

#ifdef  SKEIN_LOOP              /* configure how much to unroll the loop */
#define SKEIN_UNROLL_1024 ((SKEIN_LOOP)%10)
#else
#define SKEIN_UNROLL_1024 (0)
#endif

#if     (SKEIN_UNROLL_1024 != 0)
#if     (RCNT % SKEIN_UNROLL_1024)
#error "Invalid SKEIN_UNROLL_1024"      /* sanity check on unroll count */
#endif
        size_t r;
        /* key schedule words : chaining vars + tweak + "rotation" */
        uint64_t kw[WCNT + 4 + RCNT * 2];
#else
        uint64_t kw[WCNT + 4];  /* key schedule words : chaining vars + tweak */
#endif

        /* local copy of vars, for speed */
        uint64_t X00, X01, X02, X03, X04, X05, X06, X07, X08, X09, X10, X11,
            X12, X13, X14, X15;
        uint64_t w[WCNT];               /* local copy of input block */
#ifdef  SKEIN_DEBUG
        /* use for debugging (help compiler put Xn in registers) */
        const uint64_t *Xptr[16];
        Xptr[0] = &X00;
        Xptr[1] = &X01;
        Xptr[2] = &X02;
        Xptr[3] = &X03;
        Xptr[4] = &X04;
        Xptr[5] = &X05;
        Xptr[6] = &X06;
        Xptr[7] = &X07;
        Xptr[8] = &X08;
        Xptr[9] = &X09;
        Xptr[10] = &X10;
        Xptr[11] = &X11;
        Xptr[12] = &X12;
        Xptr[13] = &X13;
        Xptr[14] = &X14;
        Xptr[15] = &X15;
#endif
```

```
524         Skein_assert(blkCnt != 0);      /* never call with blkCnt == 0! */
525         ts[0] = ctx->h.T[0];
526         ts[1] = ctx->h.T[1];
527         do {
528                 /*
529                  * this implementation only supports 2**64 input bytes
530                  * (no carry out here)
531                  */
532                 ts[0] += byteCntAdd;    /* update processed length */

534                 /* precompute the key schedule for this block */
535                 ks[0] = ctx->X[0];
536                 ks[1] = ctx->X[1];
537                 ks[2] = ctx->X[2];
538                 ks[3] = ctx->X[3];
539                 ks[4] = ctx->X[4];
540                 ks[5] = ctx->X[5];
541                 ks[6] = ctx->X[6];
542                 ks[7] = ctx->X[7];
543                 ks[8] = ctx->X[8];
544                 ks[9] = ctx->X[9];
545                 ks[10] = ctx->X[10];
546                 ks[11] = ctx->X[11];
547                 ks[12] = ctx->X[12];
548                 ks[13] = ctx->X[13];
549                 ks[14] = ctx->X[14];
550                 ks[15] = ctx->X[15];
551                 ks[16] = ks[0] ^ ks[1] ^ ks[2] ^ ks[3] ^
552                     ks[4] ^ ks[5] ^ ks[6] ^ ks[7] ^
553                     ks[8] ^ ks[9] ^ ks[10] ^ ks[11] ^
554                     ks[12] ^ ks[13] ^ ks[14] ^ ks[15] ^ SKEIN_KS_PARITY;

556                 ts[2] = ts[0] ^ ts[1];

558                 /* get input block in little-endian format */
559                 Skein_Get64_LSB_First(w, blkPtr, WCNT);
560                 DebugSaveTweak(ctx);
561                 Skein_Show_Block(BLK_BITS, &ctx->h, ctx->X, blkPtr, w, ks, ts);

563                 X00 = w[0] + ks[0];     /* do the first full key injection */
564                 X01 = w[1] + ks[1];
565                 X02 = w[2] + ks[2];
566                 X03 = w[3] + ks[3];
567                 X04 = w[4] + ks[4];
568                 X05 = w[5] + ks[5];
569                 X06 = w[6] + ks[6];
570                 X07 = w[7] + ks[7];
571                 X08 = w[8] + ks[8];
572                 X09 = w[9] + ks[9];
573                 X10 = w[10] + ks[10];
574                 X11 = w[11] + ks[11];
575                 X12 = w[12] + ks[12];
576                 X13 = w[13] + ks[13] + ts[0];
577                 X14 = w[14] + ks[14] + ts[1];
578                 X15 = w[15] + ks[15];

580                 Skein_Show_R_Ptr(BLK_BITS, &ctx->h, SKEIN_RND_KEY_INITIAL,
581                     Xptr);

583 #define Round1024(p0, p1, p2, p3, p4, p5, p6, p7, p8, p9, pA, pB, pC,   \
584         pD, pE, pF, ROT, rNum)                                          \
585         X##p0 += X##p1; X##p1 = RotL_64(X##p1, ROT##_0); X##p1 ^= X##p0;\
586         X##p2 += X##p3; X##p3 = RotL_64(X##p3, ROT##_1); X##p3 ^= X##p2;\
587         X##p4 += X##p5; X##p5 = RotL_64(X##p5, ROT##_2); X##p5 ^= X##p4;\
588         X##p6 += X##p7; X##p7 = RotL_64(X##p7, ROT##_3); X##p7 ^= X##p6;\
589         X##p8 += X##p9; X##p9 = RotL_64(X##p9, ROT##_4); X##p9 ^= X##p8;\
```



```
590         X##pA += X##pB; X##pB = RotL_64(X##pB, ROT##_5); X##pB ^= X##pA;\
591         X##pC += X##pD; X##pD = RotL_64(X##pD, ROT##_6); X##pD ^= X##pC;\
592         X##pE += X##pF; X##pF = RotL_64(X##pF, ROT##_7); X##pF ^= X##pE;

594 #if     SKEIN_UNROLL_1024 == 0
595 #define R1024(p0, p1, p2, p3, p4, p5, p6, p7, p8, p9, pA, pB, pC, pD,   \
596         pE, pF, ROT, rn)                                                \
597         Round1024(p0, p1, p2, p3, p4, p5, p6, p7, p8, p9, pA, pB, pC,   \
598         pD, pE, pF, ROT, rn)                                            \
599         Skein_Show_R_Ptr(BLK_BITS, &ctx->h, rn, Xptr);

601 #define I1024(R)                                                        \
602         X00 += ks[((R) + 1) % 17];      /* inject the key schedule value */\
603         X01 += ks[((R) + 2) % 17];                                      \
604         X02 += ks[((R) + 3) % 17];                                      \
605         X03 += ks[((R) + 4) % 17];                                      \
606         X04 += ks[((R) + 5) % 17];                                      \
607         X05 += ks[((R) + 6) % 17];                                      \
608         X06 += ks[((R) + 7) % 17];                                      \
609         X07 += ks[((R) + 8) % 17];                                      \
610         X08 += ks[((R) + 9) % 17];                                      \
611         X09 += ks[((R) + 10) % 17];                                     \
612         X10 += ks[((R) + 11) % 17];                                     \
613         X11 += ks[((R) + 12) % 17];                                     \
614         X12 += ks[((R) + 13) % 17];                                     \
615         X13 += ks[((R) + 14) % 17] + ts[((R) + 1) % 3];                 \
616         X14 += ks[((R) + 15) % 17] + ts[((R) + 2) % 3];                 \
617         X15 += ks[((R) + 16) % 17] + (R) +1;                            \
618         Skein_Show_R_Ptr(BLK_BITS, &ctx->h, SKEIN_RND_KEY_INJECT, Xptr);
619 #else                           /* looping version */
620 #define R1024(p0, p1, p2, p3, p4, p5, p6, p7, p8, p9, pA, pB, pC, pD,   \
621         pE, pF, ROT, rn)                                                \
622         Round1024(p0, p1, p2, p3, p4, p5, p6, p7, p8, p9, pA, pB, pC,   \
623         pD, pE, pF, ROT, rn)                                            \
624         Skein_Show_R_Ptr(BLK_BITS, &ctx->h, 4 * (r - 1) + rn, Xptr);

626 #define I1024(R)                                                        \
627         X00 += ks[r + (R) + 0]; /* inject the key schedule value */     \
628         X01 += ks[r + (R) + 1];                                         \
629         X02 += ks[r + (R) + 2];                                         \
630         X03 += ks[r + (R) + 3];                                         \
631         X04 += ks[r + (R) + 4];                                         \
632         X05 += ks[r + (R) + 5];                                         \
633         X06 += ks[r + (R) + 6];                                         \
634         X07 += ks[r + (R) + 7];                                         \
635         X08 += ks[r + (R) + 8];                                         \
636         X09 += ks[r + (R) + 9];                                         \
637         X10 += ks[r + (R) + 10];                                        \
638         X11 += ks[r + (R) + 11];                                        \
639         X12 += ks[r + (R) + 12];                                        \
640         X13 += ks[r + (R) + 13] + ts[r + (R) + 0];                      \
641         X14 += ks[r + (R) + 14] + ts[r + (R) + 1];                      \
642         X15 += ks[r + (R) + 15] +  r + (R);                             \
643         ks[r + (R) + 16] = ks[r + (R) - 1];     /* rotate key schedule */\
644         ts[r + (R) + 2] = ts[r + (R) - 1];                              \
645         Skein_Show_R_Ptr(BLK_BITS, &ctx->h, SKEIN_RND_KEY_INJECT, Xptr);

647                 /* loop thru it */
648                 for (r = 1; r <= 2 * RCNT; r += 2 * SKEIN_UNROLL_1024)
649 #endif
650                 {
651 #define R1024_8_rounds(R)       /* do 8 full rounds */                  \
652         R1024(00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13,   \
653             14, 15, R1024_0, 8 * (R) + 1);                              \
654         R1024(00, 09, 02, 13, 06, 11, 04, 15, 10, 07, 12, 03, 14, 05,   \
655             08, 01, R1024_1, 8 * (R) + 2);                              \
```

```
656         R1024(00, 07, 02, 05, 04, 03, 06, 01, 12, 15, 14, 13, 08, 11,   \
657             10, 09, R1024_2, 8 * (R) + 3);                               \
658         R1024(00, 15, 02, 11, 06, 13, 04, 09, 14, 01, 08, 05, 10, 03,   \
659             12, 07, R1024_3, 8 * (R) + 4);                               \
660         I1024(2 * (R));                                                  \
661         R1024(00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13,   \
662             14, 15, R1024_4, 8 * (R) + 5);                               \
663         R1024(00, 09, 02, 13, 06, 11, 04, 15, 10, 07, 12, 03, 14, 05,   \
664             08, 01, R1024_5, 8 * (R) + 6);                               \
665         R1024(00, 07, 02, 05, 04, 03, 06, 01, 12, 15, 14, 13, 08, 11,   \
666             10, 09, R1024_6, 8 * (R) + 7);                               \
667         R1024(00, 15, 02, 11, 06, 13, 04, 09, 14, 01, 08, 05, 10, 03,   \
668             12, 07, R1024_7, 8 * (R) + 8);                               \
669         I1024(2 * (R) + 1);

671                         R1024_8_rounds(0);

673 #define R1024_Unroll_R(NN)                                               \
674         ((SKEIN_UNROLL_1024 == 0 && SKEIN1024_ROUNDS_TOTAL/8 > (NN)) || \
675         (SKEIN_UNROLL_1024 > (NN)))

677 #if     R1024_Unroll_R(1)
678                         R1024_8_rounds(1);
679 #endif
680 #if     R1024_Unroll_R(2)
681                         R1024_8_rounds(2);
682 #endif
683 #if     R1024_Unroll_R(3)
684                         R1024_8_rounds(3);
685 #endif
686 #if     R1024_Unroll_R(4)
687                         R1024_8_rounds(4);
688 #endif
689 #if     R1024_Unroll_R(5)
690                         R1024_8_rounds(5);
691 #endif
692 #if     R1024_Unroll_R(6)
693                         R1024_8_rounds(6);
694 #endif
695 #if     R1024_Unroll_R(7)
696                         R1024_8_rounds(7);
697 #endif
698 #if     R1024_Unroll_R(8)
699                         R1024_8_rounds(8);
700 #endif
701 #if     R1024_Unroll_R(9)
702                         R1024_8_rounds(9);
703 #endif
704 #if     R1024_Unroll_R(10)
705                         R1024_8_rounds(10);
706 #endif
707 #if     R1024_Unroll_R(11)
708                         R1024_8_rounds(11);
709 #endif
710 #if     R1024_Unroll_R(12)
711                         R1024_8_rounds(12);
712 #endif
713 #if     R1024_Unroll_R(13)
714                         R1024_8_rounds(13);
715 #endif
716 #if     R1024_Unroll_R(14)
717                         R1024_8_rounds(14);
718 #endif
719 #if     (SKEIN_UNROLL_1024 > 14)
720 #error  "need more unrolling in Skein_1024_Process_Block"
721 #endif
```



```
722                 }
723                 /*
724                  * do the final "feedforward" xor, update context chaining vars
725                  */

727                 ctx->X[0] = X00 ^ w[0];
728                 ctx->X[1] = X01 ^ w[1];
729                 ctx->X[2] = X02 ^ w[2];
730                 ctx->X[3] = X03 ^ w[3];
731                 ctx->X[4] = X04 ^ w[4];
732                 ctx->X[5] = X05 ^ w[5];
733                 ctx->X[6] = X06 ^ w[6];
734                 ctx->X[7] = X07 ^ w[7];
735                 ctx->X[8] = X08 ^ w[8];
736                 ctx->X[9] = X09 ^ w[9];
737                 ctx->X[10] = X10 ^ w[10];
738                 ctx->X[11] = X11 ^ w[11];
739                 ctx->X[12] = X12 ^ w[12];
740                 ctx->X[13] = X13 ^ w[13];
741                 ctx->X[14] = X14 ^ w[14];
742                 ctx->X[15] = X15 ^ w[15];

744                 Skein_Show_Round(BLK_BITS, &ctx->h, SKEIN_RND_FEED_FWD, ctx->X);

746                 ts[1] &= ~SKEIN_T1_FLAG_FIRST;
747                 blkPtr += SKEIN1024_BLOCK_BYTES;
748         } while (--blkCnt);
749         ctx->h.T[0] = ts[0];
750         ctx->h.T[1] = ts[1];
751 }

753 #if     defined(SKEIN_CODE_SIZE) || defined(SKEIN_PERF)
754 size_t
755 Skein1024_Process_Block_CodeSize(void)
756 {
757         return ((uint8_t *)Skein1024_Process_Block_CodeSize) -
758             ((uint8_t *)Skein1024_Process_Block);
759 }

761 uint_t
762 Skein1024_Unroll_Cnt(void)
763 {
764         return (SKEIN_UNROLL_1024);
765 }
766 #endif
767 #endif
```

```
**********************************************************
   10185 Mon Oct  7 19:17:12 2013
new/usr/src/common/crypto/skein/skein_impl.h
4185 New hash algorithm support
**********************************************************
```

```
/*
 * Internal definitions for Skein hashing.
 * Source code author: Doug Whiting, 2008.
 * This algorithm and source code is released to the public domain.
 *
 * The following compile-time switches may be defined to control some
 * tradeoffs between speed, code size, error checking, and security.
 *
 * The "default" note explains what happens when the switch is not defined.
 *
 *  SKEIN_DEBUG            -- make callouts from inside Skein code
 *                            to examine/display intermediate values.
 *                            [default: no callouts (no overhead)]
 *
 *  SKEIN_ERR_CHECK        -- how error checking is handled inside Skein
 *                            code. If not defined, most error checking
 *                            is disabled (for performance). Otherwise,
 *                            the switch value is interpreted as:
 *                                0: use assert()      to flag errors
 *                                1: return SKEIN_FAIL to flag errors
 */
/* Copyright 2013 Doug Whiting. This code is released to the public domain. */

#ifndef _SKEIN_IMPL_H_
#define _SKEIN_IMPL_H_

#include <sys/skein.h>
#include "skein_port.h"

/* determine where we can get bcopy/bzero declarations */
#ifdef  _KERNEL
#include <sys/systm.h>
#else
#include <strings.h>
#endif

/*
 * "Internal" Skein definitions
 *    -- not needed for sequential hashing API, but will be
 *           helpful for other uses of Skein (e.g., tree hash mode).
 *    -- included here so that they can be shared between
 *           reference and optimized code.
 */

/* tweak word T[1]: bit field starting positions */
/* offset 64 because it's the second word  */
#define SKEIN_T1_BIT(BIT)       ((BIT) - 64)

/* bits 112..118: level in hash tree */
#define SKEIN_T1_POS_TREE_LVL   SKEIN_T1_BIT(112)
/* bit  119: partial final input byte */
#define SKEIN_T1_POS_BIT_PAD    SKEIN_T1_BIT(119)
/* bits 120..125: type field */
#define SKEIN_T1_POS_BLK_TYPE   SKEIN_T1_BIT(120)
/* bits 126: first block flag */
#define SKEIN_T1_POS_FIRST      SKEIN_T1_BIT(126)
/* bit  127: final block flag */
#define SKEIN_T1_POS_FINAL      SKEIN_T1_BIT(127)

/* tweak word T[1]: flag bit definition(s) */
#define SKEIN_T1_FLAG_FIRST     (((uint64_t)1) << SKEIN_T1_POS_FIRST)
```



```
#define SKEIN_T1_FLAG_FINAL     (((uint64_t)1) << SKEIN_T1_POS_FINAL)
#define SKEIN_T1_FLAG_BIT_PAD   (((uint64_t)1) << SKEIN_T1_POS_BIT_PAD)

/* tweak word T[1]: tree level bit field mask */
#define SKEIN_T1_TREE_LVL_MASK  (((uint64_t)0x7F) << SKEIN_T1_POS_TREE_LVL)
#define SKEIN_T1_TREE_LEVEL(n)  (((uint64_t)(n)) << SKEIN_T1_POS_TREE_LVL)

/* tweak word T[1]: block type field */
#define SKEIN_BLK_TYPE_KEY      (0)     /* key, for MAC and KDF */
#define SKEIN_BLK_TYPE_CFG      (4)     /* configuration block */
#define SKEIN_BLK_TYPE_PERS     (8)     /* personalization string */
#define SKEIN_BLK_TYPE_PK       (12)    /* public key (for signature hashing) */
#define SKEIN_BLK_TYPE_KDF      (16)    /* key identifier for KDF */
#define SKEIN_BLK_TYPE_NONCE    (20)    /* nonce for PRNG */
#define SKEIN_BLK_TYPE_MSG      (48)    /* message processing */
#define SKEIN_BLK_TYPE_OUT      (63)    /* output stage */
#define SKEIN_BLK_TYPE_MASK     (63)    /* bit field mask */

#define SKEIN_T1_BLK_TYPE(T)    \
        (((uint64_t)(SKEIN_BLK_TYPE_##T)) << SKEIN_T1_POS_BLK_TYPE)
/* key, for MAC and KDF */
#define SKEIN_T1_BLK_TYPE_KEY   SKEIN_T1_BLK_TYPE(KEY)
/* configuration block */
#define SKEIN_T1_BLK_TYPE_CFG   SKEIN_T1_BLK_TYPE(CFG)
/* personalization string */
#define SKEIN_T1_BLK_TYPE_PERS  SKEIN_T1_BLK_TYPE(PERS)
/* public key (for digital signature hashing) */
#define SKEIN_T1_BLK_TYPE_PK    SKEIN_T1_BLK_TYPE(PK)
/* key identifier for KDF */
#define SKEIN_T1_BLK_TYPE_KDF   SKEIN_T1_BLK_TYPE(KDF)
/* nonce for PRNG */
#define SKEIN_T1_BLK_TYPE_NONCE SKEIN_T1_BLK_TYPE(NONCE)
/* message processing */
#define SKEIN_T1_BLK_TYPE_MSG   SKEIN_T1_BLK_TYPE(MSG)
/* output stage */
#define SKEIN_T1_BLK_TYPE_OUT   SKEIN_T1_BLK_TYPE(OUT)
/* field bit mask */
#define SKEIN_T1_BLK_TYPE_MASK  SKEIN_T1_BLK_TYPE(MASK)

#define SKEIN_T1_BLK_TYPE_CFG_FINAL     \
        (SKEIN_T1_BLK_TYPE_CFG | SKEIN_T1_FLAG_FINAL)
#define SKEIN_T1_BLK_TYPE_OUT_FINAL     \
        (SKEIN_T1_BLK_TYPE_OUT | SKEIN_T1_FLAG_FINAL)

#define SKEIN_VERSION           (1)

#ifndef SKEIN_ID_STRING_LE      /* allow compile-time personalization */
#define SKEIN_ID_STRING_LE      (0x33414853)    /* "SHA3" (little-endian) */
#endif

#define SKEIN_MK_64(hi32, lo32) ((lo32) + (((uint64_t)(hi32)) << 32))
#define SKEIN_SCHEMA_VER        SKEIN_MK_64(SKEIN_VERSION, SKEIN_ID_STRING_LE)
#define SKEIN_KS_PARITY         SKEIN_MK_64(0x1BD11BDA, 0xA9FC1A22)

#define SKEIN_CFG_STR_LEN       (4*8)

/* bit field definitions in config block treeInfo word */
#define SKEIN_CFG_TREE_LEAF_SIZE_POS    (0)
#define SKEIN_CFG_TREE_NODE_SIZE_POS    (8)
#define SKEIN_CFG_TREE_MAX_LEVEL_POS    (16)

#define SKEIN_CFG_TREE_LEAF_SIZE_MSK    \
        (((uint64_t)0xFF) << SKEIN_CFG_TREE_LEAF_SIZE_POS)
#define SKEIN_CFG_TREE_NODE_SIZE_MSK    \
        (((uint64_t)0xFF) << SKEIN_CFG_TREE_NODE_SIZE_POS)
#define SKEIN_CFG_TREE_MAX_LEVEL_MSK    \
```

```
128         (((uint64_t)0xFF) << SKEIN_CFG_TREE_MAX_LEVEL_POS)

130 #define SKEIN_CFG_TREE_INFO(leaf, node, maxLvl)                 \
131         ((((uint64_t)(leaf)) << SKEIN_CFG_TREE_LEAF_SIZE_POS) | \
132         (((uint64_t)(node)) << SKEIN_CFG_TREE_NODE_SIZE_POS) |  \
133         (((uint64_t)(maxLvl)) << SKEIN_CFG_TREE_MAX_LEVEL_POS))

135 /* use as treeInfo in InitExt() call for sequential processing */
136 #define SKEIN_CFG_TREE_INFO_SEQUENTIAL  SKEIN_CFG_TREE_INFO(0, 0, 0)

138 /*
139  * Skein macros for getting/setting tweak words, etc.
140  * These are useful for partial input bytes, hash tree init/update, etc.
141  */
142 #define Skein_Get_Tweak(ctxPtr, TWK_NUM)        ((ctxPtr)->h.T[TWK_NUM])
143 #define Skein_Set_Tweak(ctxPtr, TWK_NUM, tVal)          \
144         do {                                            \
145                 (ctxPtr)->h.T[TWK_NUM] = (tVal);        \
146                 _NOTE(CONSTCOND)                        \
147         } while (0)

149 #define Skein_Get_T0(ctxPtr)            Skein_Get_Tweak(ctxPtr, 0)
150 #define Skein_Get_T1(ctxPtr)            Skein_Get_Tweak(ctxPtr, 1)
151 #define Skein_Set_T0(ctxPtr, T0)        Skein_Set_Tweak(ctxPtr, 0, T0)
152 #define Skein_Set_T1(ctxPtr, T1)        Skein_Set_Tweak(ctxPtr, 1, T1)

154 /* set both tweak words at once */
155 #define Skein_Set_T0_T1(ctxPtr, T0, T1)         \
156         do {                                    \
157                 Skein_Set_T0(ctxPtr, (T0));     \
158                 Skein_Set_T1(ctxPtr, (T1));     \
159                 _NOTE(CONSTCOND)                \
160         } while (0)

162 #define Skein_Set_Type(ctxPtr, BLK_TYPE)        \
163         Skein_Set_T1(ctxPtr, SKEIN_T1_BLK_TYPE_##BLK_TYPE)

165 /*
166  * set up for starting with a new type: h.T[0]=0; h.T[1] = NEW_TYPE; h.bCnt=0;
167  */
168 #define Skein_Start_New_Type(ctxPtr, BLK_TYPE)                          \
169         do {                                                            \
170                 Skein_Set_T0_T1(ctxPtr, 0, SKEIN_T1_FLAG_FIRST |        \
171                     SKEIN_T1_BLK_TYPE_ ## BLK_TYPE);                    \
172                 (ctxPtr)->h.bCnt = 0;   \
173                 _NOTE(CONSTCOND)                                        \
174         } while (0)

176 #define Skein_Clear_First_Flag(hdr)                                     \
177         do {                                                            \
178                 (hdr).T[1] &= ~SKEIN_T1_FLAG_FIRST;                     \
179                 _NOTE(CONSTCOND)                                        \
180         } while (0)
181 #define Skein_Set_Bit_Pad_Flag(hdr)                                     \
182         do {                                                            \
183                 (hdr).T[1] |=  SKEIN_T1_FLAG_BIT_PAD;                   \
184                 _NOTE(CONSTCOND)                                        \
185         } while (0)

187 #define Skein_Set_Tree_Level(hdr, height)                               \
188         do {                                                            \
189                 (hdr).T[1] |= SKEIN_T1_TREE_LEVEL(height);              \
190                 _NOTE(CONSTCOND)                                        \
191         } while (0)

193 /*
```



```
194  * "Internal" Skein definitions for debugging and error checking
195  * Note: in Illumos we always disable debugging features.
196  */
197 #define Skein_Show_Block(bits, ctx, X, blkPtr, wPtr, ksEvenPtr, ksOddPtr)
198 #define Skein_Show_Round(bits, ctx, r, X)
199 #define Skein_Show_R_Ptr(bits, ctx, r, X_ptr)
200 #define Skein_Show_Final(bits, ctx, cnt, outPtr)
201 #define Skein_Show_Key(bits, ctx, key, keyBytes)

203 /* run-time checks (e.g., bad params, uninitialized context)? */
204 #ifndef SKEIN_ERR_CHECK
205 /* default: ignore all Asserts, for performance */
206 #define Skein_Assert(x, retCode)
207 #define Skein_assert(x)
208 #elif   defined(SKEIN_ASSERT)
209 #include <sys/debug.h>
210 #define Skein_Assert(x, retCode)        ASSERT(x)
211 #define Skein_assert(x)                 ASSERT(x)
212 #else
213 #include <sys/debug.h>
214 /*  caller error */
215 #define Skein_Assert(x, retCode)                \
216         do {                                    \
217                 if (!(x))                       \
218                         return (retCode);       \
219                 _NOTE(CONSTCOND)                \
220         } while (0)
221 /* internal error */
222 #define Skein_assert(x) ASSERT(x)
223 #endif

225 /*
226  * Skein block function constants (shared across Ref and Opt code)
227  */
228 enum {
229         /* Skein_256 round rotation constants */
230         R_256_0_0 = 14, R_256_0_1 = 16,
231         R_256_1_0 = 52, R_256_1_1 = 57,
232         R_256_2_0 = 23, R_256_2_1 = 40,
233         R_256_3_0 = 5, R_256_3_1 = 37,
234         R_256_4_0 = 25, R_256_4_1 = 33,
235         R_256_5_0 = 46, R_256_5_1 = 12,
236         R_256_6_0 = 58, R_256_6_1 = 22,
237         R_256_7_0 = 32, R_256_7_1 = 32,

239         /* Skein_512 round rotation constants */
240         R_512_0_0 = 46, R_512_0_1 = 36, R_512_0_2 = 19, R_512_0_3 = 37,
241         R_512_1_0 = 33, R_512_1_1 = 27, R_512_1_2 = 14, R_512_1_3 = 42,
242         R_512_2_0 = 17, R_512_2_1 = 49, R_512_2_2 = 36, R_512_2_3 = 39,
243         R_512_3_0 = 44, R_512_3_1 = 9, R_512_3_2 = 54, R_512_3_3 = 56,
244         R_512_4_0 = 39, R_512_4_1 = 30, R_512_4_2 = 34, R_512_4_3 = 24,
245         R_512_5_0 = 13, R_512_5_1 = 50, R_512_5_2 = 10, R_512_5_3 = 17,
246         R_512_6_0 = 25, R_512_6_1 = 29, R_512_6_2 = 39, R_512_6_3 = 43,
247         R_512_7_0 = 8, R_512_7_1 = 35, R_512_7_2 = 56, R_512_7_3 = 22,

249         /* Skein1024 round rotation constants */
250         R1024_0_0 = 24, R1024_0_1 = 13, R1024_0_2 = 8, R1024_0_3 =
251             47, R1024_0_4 = 8, R1024_0_5 = 17, R1024_0_6 = 22, R1024_0_7 = 37,
252         R1024_1_0 = 38, R1024_1_1 = 19, R1024_1_2 = 10, R1024_1_3 =
253             55, R1024_1_4 = 49, R1024_1_5 = 18, R1024_1_6 = 23, R1024_1_7 = 52,
254         R1024_2_0 = 33, R1024_2_1 = 4, R1024_2_2 = 51, R1024_2_3 =
255             13, R1024_2_4 = 34, R1024_2_5 = 41, R1024_2_6 = 59, R1024_2_7 = 17,
256         R1024_3_0 = 5, R1024_3_1 = 20, R1024_3_2 = 48, R1024_3_3 =
257             41, R1024_3_4 = 47, R1024_3_5 = 28, R1024_3_6 = 16, R1024_3_7 = 25,
258         R1024_4_0 = 41, R1024_4_1 = 9, R1024_4_2 = 37, R1024_4_3 =
259             31, R1024_4_4 = 12, R1024_4_5 = 47, R1024_4_6 = 44, R1024_4_7 = 30,
```

```
260         R1024_5_0 = 16, R1024_5_1 = 34, R1024_5_2 = 56, R1024_5_3 =
261             51, R1024_5_4 = 4, R1024_5_5 = 53, R1024_5_6 = 42, R1024_5_7 = 41,
262         R1024_6_0 = 31, R1024_6_1 = 44, R1024_6_2 = 47, R1024_6_3 =
263             46, R1024_6_4 = 19, R1024_6_5 = 42, R1024_6_6 = 44, R1024_6_7 = 25,
264         R1024_7_0 = 9, R1024_7_1 = 48, R1024_7_2 = 35, R1024_7_3 =
265             52, R1024_7_4 = 23, R1024_7_5 = 31, R1024_7_6 = 37, R1024_7_7 = 20
266 };

268 /* number of rounds for the different block sizes */
269 #define SKEIN_256_ROUNDS_TOTAL  (72)
270 #define SKEIN_512_ROUNDS_TOTAL  (72)
271 #define SKEIN1024_ROUNDS_TOTAL  (80)

273 #endif  /* _SKEIN_IMPL_H_ */
```

```
**********************************************************
    5482 Mon Oct  7 19:17:12 2013
new/usr/src/common/crypto/skein/skein_iv.h
4185 New hash algorithm support
**********************************************************
```

```
/*
 * Pre-computed Skein IVs
 *
 * NOTE: these values are not "magic" constants, but
 * are generated using the Threefish block function.
 * They are pre-computed here only for speed; i.e., to
 * avoid the need for a Threefish call during Init().
 *
 * The IV for any fixed hash length may be pre-computed.
 * Only the most common values are included here.
 */
/* Copyright 2013 Doug Whiting. This code is released to the public domain. */
/*
 * Illumos implementation note: these constants are for Skein v1.3 as per:
 * http://www.skein-hash.info/sites/default/files/skein1.3.pdf
 */

#ifndef _SKEIN_IV_H_
#define _SKEIN_IV_H_

#include <sys/skein.h>          /* get Skein macros and types */

#define MK_64 SKEIN_MK_64

/* blkSize =  256 bits. hashSize =  128 bits */
const uint64_t SKEIN_256_IV_128[] = {
        MK_64(0xE1111906, 0x964D7260),
        MK_64(0x883DAAA7, 0x7C8D811C),
        MK_64(0x10080DF4, 0x91960F7A),
        MK_64(0xCCF7DDE5, 0xB45BC1C2)
};

/* blkSize =  256 bits. hashSize =  160 bits */
const uint64_t SKEIN_256_IV_160[] = {
        MK_64(0x14202314, 0x72825E98),
        MK_64(0x2AC4E9A2, 0x5A77E590),
        MK_64(0xD47A5856, 0x8838D63E),
        MK_64(0x2DD2E496, 0x8586AB7D)
};

/* blkSize =  256 bits. hashSize =  224 bits */
const uint64_t SKEIN_256_IV_224[] = {
        MK_64(0xC6098A8C, 0x9AE5EA0B),
        MK_64(0x876D5686, 0x08C5191C),
        MK_64(0x99CB88D7, 0xD7F53884),
        MK_64(0x384BDDB1, 0xAEDDB5DE)
};

/* blkSize =  256 bits. hashSize =  256 bits */
const uint64_t SKEIN_256_IV_256[] = {
        MK_64(0xFC9DA860, 0xD048B449),
        MK_64(0x2FCA6647, 0x9FA7D833),
        MK_64(0xB33BC389, 0x6656840F),
        MK_64(0x6A54E920, 0xFDE8DA69)
};

/* blkSize =  512 bits. hashSize =  128 bits */
const uint64_t SKEIN_512_IV_128[] = {
        MK_64(0xA8BC7BF3, 0x6FBF9F52),
        MK_64(0x1E9872CE, 0xBD1AF0AA),
        MK_64(0x309B1790, 0xB32190D3),
```



```
        MK_64(0xBCFBB854, 0x3F94805C),
        MK_64(0x0DA61BCD, 0x6E31B11B),
        MK_64(0x1A18EBEA, 0xD46A32E3),
        MK_64(0xA2CC5B18, 0xCE84AA82),
        MK_64(0x6982AB28, 0x9D46982D)
};

/* blkSize =  512 bits. hashSize =  160 bits */
const uint64_t SKEIN_512_IV_160[] = {
        MK_64(0x28B81A2A, 0xE013BD91),
        MK_64(0xC2F11668, 0xB5BDF78F),
        MK_64(0x1760D8F3, 0xF6A56F12),
        MK_64(0x4FB74758, 0x8239904F),
        MK_64(0x21EDE07F, 0x7EAF5056),
        MK_64(0xD908922E, 0x63ED70B8),
        MK_64(0xB8EC76FF, 0xECCB52FA),
        MK_64(0x01A47BB8, 0xA3F27A6E)
};

/* blkSize =  512 bits. hashSize =  224 bits */
const uint64_t SKEIN_512_IV_224[] = {
        MK_64(0xCCD06162, 0x48677224),
        MK_64(0xCBA65CF3, 0xA92339EF),
        MK_64(0x8CCD69D6, 0x52FF4B64),
        MK_64(0x398AED7B, 0x3AB890B4),
        MK_64(0x0F59D1B1, 0x457D2BD0),
        MK_64(0x6776FE65, 0x75D4EB3D),
        MK_64(0x99FBC70E, 0x997413E9),
        MK_64(0x9E2CFCCF, 0xE1C41EF7)
};

/* blkSize =  512 bits. hashSize =  256 bits */
const uint64_t SKEIN_512_IV_256[] = {
        MK_64(0xCCD044A1, 0x2FDB3E13),
        MK_64(0xE8359030, 0x1A79A9EB),
        MK_64(0x55AEA061, 0x4F816E6F),
        MK_64(0x2A2767A4, 0xAE9B94DB),
        MK_64(0xEC06025E, 0x74DD7683),
        MK_64(0xE7A436CD, 0xC4746251),
        MK_64(0xC36FBAF9, 0x393AD185),
        MK_64(0x3EEDBA18, 0x33EDFC13)
};

/* blkSize =  512 bits. hashSize =  384 bits */
const uint64_t SKEIN_512_IV_384[] = {
        MK_64(0xA3F6C6BF, 0x3A75EF5F),
        MK_64(0xB0FEF9CC, 0xFD84FAA4),
        MK_64(0x9D77DD66, 0x3D770CFE),
        MK_64(0xD798CBF3, 0xB468FDDA),
        MK_64(0x1BC4A666, 0x8A0E4465),
        MK_64(0x7ED7D434, 0xE5807407),
        MK_64(0x548FC1AC, 0xD4EC44D6),
        MK_64(0x266E1754, 0x6AA18FF8)
};

/* blkSize =  512 bits. hashSize =  512 bits */
const uint64_t SKEIN_512_IV_512[] = {
        MK_64(0x4903ADFF, 0x749C51CE),
        MK_64(0x0D95DE39, 0x9746DF03),
        MK_64(0x8FD19341, 0x27C79BCE),
        MK_64(0x9A255629, 0xFF352CB1),
        MK_64(0x5DB62599, 0xDF6CA7B0),
        MK_64(0xEABE394C, 0xA9D5C3F4),
        MK_64(0x991112C7, 0x1A75B523),
        MK_64(0xAE18A40B, 0x660FCC33)
};
```

```
129 /* blkSize = 1024 bits. hashSize =  384 bits */
130 const uint64_t SKEIN1024_IV_384[] = {
131         MK_64(0x5102B6B8, 0xC1894A35),
132         MK_64(0xFEEBC9E3, 0xFE8AF11A),
133         MK_64(0x0C807F06, 0xE32BED71),
134         MK_64(0x60C13A52, 0xB41A91F6),
135         MK_64(0x9716D35D, 0xD4917C38),
136         MK_64(0xE780DF12, 0x6FD31D3A),
137         MK_64(0x797846B6, 0xC898303A),
138         MK_64(0xB172C2A8, 0xB3572A3B),
139         MK_64(0xC9BC8203, 0xA6104A6C),
140         MK_64(0x65909338, 0xD75624F4),
141         MK_64(0x94BCC568, 0x4B3F81A0),
142         MK_64(0x3EBBF51E, 0x10ECFD46),
143         MK_64(0x2DF50F0B, 0xEEB08542),
144         MK_64(0x3B5A6530, 0x0DBC6516),
145         MK_64(0x484B9CD2, 0x167BBCE1),
146         MK_64(0x2D136947, 0xD4CBAFEA)
147 };

149 /* blkSize = 1024 bits. hashSize =  512 bits */
150 const uint64_t SKEIN1024_IV_512[] = {
151         MK_64(0xCAEC0E5D, 0x7C1B1B18),
152         MK_64(0xA01B0E04, 0x5F03E802),
153         MK_64(0x33840451, 0xED912885),
154         MK_64(0x374AFB04, 0xEAEC2E1C),
155         MK_64(0xDF25A0E2, 0x813581F7),
156         MK_64(0xE4004093, 0x8B12F9D2),
157         MK_64(0xA662D539, 0xC2ED39B6),
158         MK_64(0xFA8B85CF, 0x45D8C75A),
159         MK_64(0x8316ED8E, 0x29EDE796),
160         MK_64(0x053289C0, 0x2E9F91B8),
161         MK_64(0xC3F8EF1D, 0x6D518B73),
162         MK_64(0xBDCEC3C4, 0xD5EF332E),
163         MK_64(0x549A7E52, 0x22974487),
164         MK_64(0x67070872, 0x5B749816),
165         MK_64(0xB9CD28FB, 0xF0581BD1),
166         MK_64(0x0E2940B8, 0x15804974)
167 };

169 /* blkSize = 1024 bits. hashSize = 1024 bits */
170 const uint64_t SKEIN1024_IV_1024[] = {
171         MK_64(0xD593DA07, 0x41E72355),
172         MK_64(0x15B5E511, 0xAC73E00C),
173         MK_64(0x5180E5AE, 0xBAF2C4F0),
174         MK_64(0x03BD41D3, 0xFCBCAFAF),
175         MK_64(0x1CAEC6FD, 0x1983A898),
176         MK_64(0x6E510B8B, 0xCDD0589F),
177         MK_64(0x77E2BDFD, 0xC6394ADA),
178         MK_64(0xC11E1DB5, 0x24DCB0A3),
179         MK_64(0xD6D14AF9, 0xC6329AB5),
180         MK_64(0x6A9B0BFC, 0x6EB67E0D),
181         MK_64(0x9243C60D, 0xCCFF1332),
182         MK_64(0x1A1F1DDE, 0x743F02D4),
183         MK_64(0x0996753C, 0x10ED0BB8),
184         MK_64(0x6572DD22, 0xF2B4969A),
185         MK_64(0x61FD3062, 0xD00A579A),
186         MK_64(0x1DE0536E, 0x8682E539)
187 };

189 #endif  /* _SKEIN_IV_H_ */
```

```
**********************************************************
    3828 Mon Oct  7 19:17:12 2013
new/usr/src/common/crypto/skein/skein_port.h
4185 New hash algorithm support
**********************************************************
```

```
/*
 * Platform-specific definitions for Skein hash function.
 *
 * Source code author: Doug Whiting, 2008.
 *
 * This algorithm and source code is released to the public domain.
 *
 * Many thanks to Brian Gladman for his portable header files.
 *
 * To port Skein to an "unsupported" platform, change the definitions
 * in this file appropriately.
 */
/* Copyright 2013 Doug Whiting. This code is released to the public domain. */

#ifndef _SKEIN_PORT_H_
#define _SKEIN_PORT_H_

#include <sys/types.h>  /* get integer type definitions */
#include <sys/systm.h>  /* for bcopy() */

#ifndef RotL_64
#define RotL_64(x, N)   (((x) << (N)) | ((x) >> (64 - (N))))
#endif

/*
 * Skein is "natively" little-endian (unlike SHA-xxx), for optimal
 * performance on x86 CPUs. The Skein code requires the following
 * definitions for dealing with endianness:
 *
 *    SKEIN_NEED_SWAP:  0 for little-endian, 1 for big-endian
 *    Skein_Put64_LSB_First
 *    Skein_Get64_LSB_First
 *    Skein_Swap64
 *
 * If SKEIN_NEED_SWAP is defined at compile time, it is used here
 * along with the portable versions of Put64/Get64/Swap64, which
 * are slow in general.
 *
 * Otherwise, an "auto-detect" of endianness is attempted below.
 * If the default handling doesn't work well, the user may insert
 * platform-specific code instead (e.g., for big-endian CPUs).
 *
 */
#ifndef SKEIN_NEED_SWAP         /* compile-time "override" for endianness? */

#include <sys/isa_defs.h>       /* get endianness selection */

#define PLATFORM_MUST_ALIGN     _ALIGNMENT_REQUIRED
#if     defined(_BIG_ENDIAN)
/* here for big-endian CPUs */
#define SKEIN_NEED_SWAP   (1)
#else
/* here for x86 and x86-64 CPUs (and other detected little-endian CPUs) */
#define SKEIN_NEED_SWAP   (0)
#if     PLATFORM_MUST_ALIGN == 0        /* ok to use "fast" versions? */
#define Skein_Put64_LSB_First(dst08, src64, bCnt) bcopy(src64, dst08, bCnt)
#define Skein_Get64_LSB_First(dst64, src08, wCnt) \
        bcopy(src08, dst64, 8 * (wCnt))
#endif
#endif
```



```
#endif                          /* ifndef SKEIN_NEED_SWAP */

/*
 * Provide any definitions still needed.
 */
#ifndef Skein_Swap64    /* swap for big-endian, nop for little-endian */
#if     SKEIN_NEED_SWAP
#define Skein_Swap64(w64)                                       \
        (((((uint64_t)(w64)) & 0xFF) << 56) |           \
        ((((uint64_t)(w64)) >> 8) & 0xFF) << 48) |      \
        ((((uint64_t)(w64)) >> 16) & 0xFF) << 40) |     \
        ((((uint64_t)(w64)) >> 24) & 0xFF) << 32) |     \
        ((((uint64_t)(w64)) >> 32) & 0xFF) << 24) |     \
        ((((uint64_t)(w64)) >> 40) & 0xFF) << 16) |     \
        ((((uint64_t)(w64)) >> 48) & 0xFF) << 8) |      \
        ((((uint64_t)(w64)) >> 56) & 0xFF)))
#else
#define Skein_Swap64(w64)  (w64)
#endif
#endif                          /* ifndef Skein_Swap64 */

#ifndef Skein_Put64_LSB_First
void
Skein_Put64_LSB_First(uint8_t *dst, const uint64_t *src, size_t bCnt)
#ifdef  SKEIN_PORT_CODE         /* instantiate the function code here? */
{
        /*
         * this version is fully portable (big-endian or little-endian),
         * but slow
         */
        size_t n;

        for (n = 0; n < bCnt; n++)
                dst[n] = (uint8_t)(src[n >> 3] >> (8 * (n & 7)));
}
#else
;                               /* output only the function prototype */
#endif
#endif                          /* ifndef Skein_Put64_LSB_First */

#ifndef Skein_Get64_LSB_First
void
Skein_Get64_LSB_First(uint64_t *dst, const uint8_t *src, size_t wCnt)
#ifdef  SKEIN_PORT_CODE         /* instantiate the function code here? */
{
        /*
         * this version is fully portable (big-endian or little-endian),
         * but slow
         */
        size_t n;

        for (n = 0; n < 8 * wCnt; n += 8)
                dst[n / 8] = (((uint64_t)src[n])) +
                    (((uint64_t)src[n + 1]) << 8) +
                    (((uint64_t)src[n + 2]) << 16) +
                    (((uint64_t)src[n + 3]) << 24) +
                    (((uint64_t)src[n + 4]) << 32) +
                    (((uint64_t)src[n + 5]) << 40) +
                    (((uint64_t)src[n + 6]) << 48) +
                    (((uint64_t)src[n + 7]) << 56);
}
#else
;                               /* output only the function prototype */
#endif
#endif                          /* ifndef Skein_Get64_LSB_First */
```

```
128 #endif  /* _SKEIN_PORT_H_ */
```

```
**********************************************************
    1809 Mon Oct  7 19:17:12 2013
new/usr/src/common/crypto/skein/test/Makefile
4185 New hash algorithm support
**********************************************************
```

```
  1 #
  2 # CDDL HEADER START
  3 #
  4 # The contents of this file are subject to the terms of the
  5 # Common Development and Distribution License (the "License").
  6 # You may not use this file except in compliance with the License.
  7 #
  8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
  9 # or http://opensource.org/licenses/CDDL-1.0.
 10 # See the License for the specific language governing permissions
 11 # and limitations under the License.
 12 #
 13 # When distributing Covered Code, include this CDDL HEADER in each
 14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 15 # If applicable, add the following below this CDDL HEADER, with the
 16 # fields enclosed by brackets "[]" replaced with your own identifying
 17 # information: Portions Copyright [yyyy] [name of copyright owner]
 18 #
 19 # CDDL HEADER END
 20 #
 21 # Copyright 2013 Saso Kiselkov.  All rights reserved.
 22 #
 23 #       This makefile drives the production of the skein test suite.
 24 #       To run the tests, simply type "make" or "make check".

 26 include ../../../../Makefile.master

 28 CFLAGS = -I${SRC}/uts/common -D_KERNEL -fno-builtin -O2 -W -Wall \
 29   -Wno-unknown-pragmas -std=c99
 30 CPUMHZ:sh = psrinfo -v 0 | awk '/processor operates at/{print $6}'
 31 GCC = $(GCC_ROOT)/bin/gcc

 33 check : skein_test32_check skein_test64_check

 35 all : skein_test32 skein_test64

 37 skein_test32 : skein_test.c
 38         @ echo Building 32-bit test...
 39         @ $(GCC) $(CFLAGS) -o skein_test32 skein_test.c ../skein.c \
 40             ../skein_block.c

 42 skein_test64 : skein_test.c
 43         @ echo Building 64-bit test...
 44         @ $(GCC) $(CFLAGS) -m64 -o skein_test64 skein_test.c ../skein.c \
 45             ../skein_block.c

 47 skein_test32_check : skein_test32
 48         @ echo Running 32-bit test...
 49         @ ./skein_test32 $(CPUMHZ)

 51 skein_test64_check : skein_test64
 52         @ echo Running 64-bit test...
 53         @ ./skein_test64 $(CPUMHZ)

 55 clean :
 56         rm -f skein_test32 skein_test64
```

```
**********************************************************
   11863 Mon Oct  7 19:17:12 2013
new/usr/src/common/crypto/skein/test/skein_test.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://opensource.org/licenses/CDDL-1.0.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2013 Saso Kiselkov. All rights reserved.
 */

/*
 * This is just to keep the compiler happy about sys/time.h not declaring
 * gettimeofday due to -D_KERNEL (we can do this since we're actually
 * running in userspace, but we need -D_KERNEL for the remaining Skein code).
 */
#ifdef  _KERNEL
#undef  _KERNEL
#endif

#include <sys/skein.h>
#include <stdlib.h>
#include <strings.h>
#include <stdio.h>
#include <sys/time.h>

/*
 * Skein test suite using values from the Skein V1.3 specification found at:
 * http://www.skein-hash.info/sites/default/files/skein1.3.pdf
 */

/*
 * Test messages from the Skein spec, Appendix C.
 */
const uint8_t   test_msg0[] = {
        0xFF
};

const uint8_t   test_msg1[] = {
        0xFF, 0xFE, 0xFD, 0xFC, 0xFB, 0xFA, 0xF9, 0xF8,
        0xF7, 0xF6, 0xF5, 0xF4, 0xF3, 0xF2, 0xF1, 0xF0,
        0xEF, 0xEE, 0xED, 0xEC, 0xEB, 0xEA, 0xE9, 0xE8,
        0xE7, 0xE6, 0xE5, 0xE4, 0xE3, 0xE2, 0xE1, 0xE0
};

const uint8_t   test_msg2[] = {
        0xFF, 0xFE, 0xFD, 0xFC, 0xFB, 0xFA, 0xF9, 0xF8,
```



```
        0xF7, 0xF6, 0xF5, 0xF4, 0xF3, 0xF2, 0xF1, 0xF0,
        0xEF, 0xEE, 0xED, 0xEC, 0xEB, 0xEA, 0xE9, 0xE8,
        0xE7, 0xE6, 0xE5, 0xE4, 0xE3, 0xE2, 0xE1, 0xE0,
        0xDF, 0xDE, 0xDD, 0xDC, 0xDB, 0xDA, 0xD9, 0xD8,
        0xD7, 0xD6, 0xD5, 0xD4, 0xD3, 0xD2, 0xD1, 0xD0,
        0xCF, 0xCE, 0xCD, 0xCC, 0xCB, 0xCA, 0xC9, 0xC8,
        0xC7, 0xC6, 0xC5, 0xC4, 0xC3, 0xC2, 0xC1, 0xC0
};

const uint8_t   test_msg3[] = {
        0xFF, 0xFE, 0xFD, 0xFC, 0xFB, 0xFA, 0xF9, 0xF8,
        0xF7, 0xF6, 0xF5, 0xF4, 0xF3, 0xF2, 0xF1, 0xF0,
        0xEF, 0xEE, 0xED, 0xEC, 0xEB, 0xEA, 0xE9, 0xE8,
        0xE7, 0xE6, 0xE5, 0xE4, 0xE3, 0xE2, 0xE1, 0xE0,
        0xDF, 0xDE, 0xDD, 0xDC, 0xDB, 0xDA, 0xD9, 0xD8,
        0xD7, 0xD6, 0xD5, 0xD4, 0xD3, 0xD2, 0xD1, 0xD0,
        0xCF, 0xCE, 0xCD, 0xCC, 0xCB, 0xCA, 0xC9, 0xC8,
        0xC7, 0xC6, 0xC5, 0xC4, 0xC3, 0xC2, 0xC1, 0xC0,
        0xBF, 0xBE, 0xBD, 0xBC, 0xBB, 0xBA, 0xB9, 0xB8,
        0xB7, 0xB6, 0xB5, 0xB4, 0xB3, 0xB2, 0xB1, 0xB0,
        0xAF, 0xAE, 0xAD, 0xAC, 0xAB, 0xAA, 0xA9, 0xA8,
        0xA7, 0xA6, 0xA5, 0xA4, 0xA3, 0xA2, 0xA1, 0xA0,
        0x9F, 0x9E, 0x9D, 0x9C, 0x9B, 0x9A, 0x99, 0x98,
        0x97, 0x96, 0x95, 0x94, 0x93, 0x92, 0x91, 0x90,
        0x8F, 0x8E, 0x8D, 0x8C, 0x8B, 0x8A, 0x89, 0x88,
        0x87, 0x86, 0x85, 0x84, 0x83, 0x82, 0x81, 0x80
};

const uint8_t   test_msg4[] = {
        0xFF, 0xFE, 0xFD, 0xFC, 0xFB, 0xFA, 0xF9, 0xF8,
        0xF7, 0xF6, 0xF5, 0xF4, 0xF3, 0xF2, 0xF1, 0xF0,
        0xEF, 0xEE, 0xED, 0xEC, 0xEB, 0xEA, 0xE9, 0xE8,
        0xE7, 0xE6, 0xE5, 0xE4, 0xE3, 0xE2, 0xE1, 0xE0,
        0xDF, 0xDE, 0xDD, 0xDC, 0xDB, 0xDA, 0xD9, 0xD8,
        0xD7, 0xD6, 0xD5, 0xD4, 0xD3, 0xD2, 0xD1, 0xD0,
        0xCF, 0xCE, 0xCD, 0xCC, 0xCB, 0xCA, 0xC9, 0xC8,
        0xC7, 0xC6, 0xC5, 0xC4, 0xC3, 0xC2, 0xC1, 0xC0,
        0xBF, 0xBE, 0xBD, 0xBC, 0xBB, 0xBA, 0xB9, 0xB8,
        0xB7, 0xB6, 0xB5, 0xB4, 0xB3, 0xB2, 0xB1, 0xB0,
        0xAF, 0xAE, 0xAD, 0xAC, 0xAB, 0xAA, 0xA9, 0xA8,
        0xA7, 0xA6, 0xA5, 0xA4, 0xA3, 0xA2, 0xA1, 0xA0,
        0x9F, 0x9E, 0x9D, 0x9C, 0x9B, 0x9A, 0x99, 0x98,
        0x97, 0x96, 0x95, 0x94, 0x93, 0x92, 0x91, 0x90,
        0x8F, 0x8E, 0x8D, 0x8C, 0x8B, 0x8A, 0x89, 0x88,
        0x87, 0x86, 0x85, 0x84, 0x83, 0x82, 0x81, 0x80,
        0x7F, 0x7E, 0x7D, 0x7C, 0x7B, 0x7A, 0x79, 0x78,
        0x77, 0x76, 0x75, 0x74, 0x73, 0x72, 0x71, 0x70,
        0x6F, 0x6E, 0x6D, 0x6C, 0x6B, 0x6A, 0x69, 0x68,
        0x67, 0x66, 0x65, 0x64, 0x63, 0x62, 0x61, 0x60,
        0x5F, 0x5E, 0x5D, 0x5C, 0x5B, 0x5A, 0x59, 0x58,
        0x57, 0x56, 0x55, 0x54, 0x53, 0x52, 0x51, 0x50,
        0x4F, 0x4E, 0x4D, 0x4C, 0x4B, 0x4A, 0x49, 0x48,
        0x47, 0x46, 0x45, 0x44, 0x43, 0x42, 0x41, 0x40,
        0x3F, 0x3E, 0x3D, 0x3C, 0x3B, 0x3A, 0x39, 0x38,
        0x37, 0x36, 0x35, 0x34, 0x33, 0x32, 0x31, 0x30,
        0x2F, 0x2E, 0x2D, 0x2C, 0x2B, 0x2A, 0x29, 0x28,
        0x27, 0x26, 0x25, 0x24, 0x23, 0x22, 0x21, 0x20,
        0x1F, 0x1E, 0x1D, 0x1C, 0x1B, 0x1A, 0x19, 0x18,
        0x17, 0x16, 0x15, 0x14, 0x13, 0x12, 0x11, 0x10,
        0x0F, 0x0E, 0x0D, 0x0C, 0x0B, 0x0A, 0x09, 0x08,
        0x07, 0x06, 0x05, 0x04, 0x03, 0x02, 0x01, 0x00
};

/*
 * Test digests from the Skein spec, Appendix C.
 */
```

```
128 const uint8_t   skein_256_test_digests[][32] = {
129         {
130                 // for test_msg0
131                 0x0B, 0x98, 0xDC, 0xD1, 0x98, 0xEA, 0x0E, 0x50,
132                 0xA7, 0xA2, 0x44, 0xC4, 0x44, 0xE2, 0x5C, 0x23,
133                 0xDA, 0x30, 0xC1, 0x0F, 0xC9, 0xA1, 0xF2, 0x70,
134                 0xA6, 0x63, 0x7F, 0x1F, 0x34, 0xE6, 0x7E, 0xD2
135         },
136         {
137                 // for test_msg1
138                 0x8D, 0x0F, 0xA4, 0xEF, 0x77, 0x7F, 0xD7, 0x59,
139                 0xDF, 0xD4, 0x04, 0x4E, 0x6F, 0x6A, 0x5A, 0xC3,
140                 0xC7, 0x74, 0xAE, 0xC9, 0x43, 0xDC, 0xFC, 0x07,
141                 0x92, 0x7B, 0x72, 0x3B, 0x5D, 0xBF, 0x40, 0x8B
142         },
143         {
144                 // for test_msg2
145                 0xDF, 0x28, 0xE9, 0x16, 0x63, 0x0D, 0x0B, 0x44,
146                 0xC4, 0xA8, 0x49, 0xDC, 0x9A, 0x02, 0xF0, 0x7A,
147                 0x07, 0xCB, 0x30, 0xF7, 0x32, 0x31, 0x82, 0x56,
148                 0xB1, 0x5D, 0x86, 0x5A, 0xC4, 0xAE, 0x16, 0x2F
149         }
150         // no test digests for test_msg3 and test_msg4
151 };

153 const uint8_t   skein_512_test_digests[][64] = {
154         {
155                 // for test_msg0
156                 0x71, 0xB7, 0xBC, 0xE6, 0xFE, 0x64, 0x52, 0x22,
157                 0x7B, 0x9C, 0xED, 0x60, 0x14, 0x24, 0x9E, 0x5B,
158                 0xF9, 0xA9, 0x75, 0x4C, 0x3A, 0xD6, 0x18, 0xCC,
159                 0xC4, 0xE0, 0xAA, 0xE1, 0x6B, 0x31, 0x6C, 0xC8,
160                 0xCA, 0x69, 0x8D, 0x86, 0x43, 0x07, 0xED, 0x3E,
161                 0x80, 0xB6, 0xEF, 0x15, 0x70, 0x81, 0x2A, 0xC5,
162                 0x27, 0x2D, 0xC4, 0x09, 0xB5, 0xA0, 0x12, 0xDF,
163                 0x2A, 0x57, 0x91, 0x02, 0xF3, 0x40, 0x61, 0x7A
164         },
165         {
166                 // no test vector for test_msg1
167         },
168         {
169                 // for test_msg2
170                 0x45, 0x86, 0x3B, 0xA3, 0xBE, 0x0C, 0x4D, 0xFC,
171                 0x27, 0xE7, 0x5D, 0x35, 0x84, 0x96, 0xF4, 0xAC,
172                 0x9A, 0x73, 0x6A, 0x50, 0x5D, 0x93, 0x13, 0xB4,
173                 0x2B, 0x2F, 0x5E, 0xAD, 0xA7, 0x9F, 0xC1, 0x7F,
174                 0x63, 0x86, 0x1E, 0x94, 0x7A, 0xFB, 0x1D, 0x05,
175                 0x6A, 0xA1, 0x99, 0x57, 0x5A, 0xD3, 0xF8, 0xC9,
176                 0xA3, 0xCC, 0x17, 0x80, 0xB5, 0xE5, 0xFA, 0x4C,
177                 0xAE, 0x05, 0x0E, 0x98, 0x98, 0x76, 0x62, 0x5B
178         },
179         {
180                 // for test_msg3
181                 0x91, 0xCC, 0xA5, 0x10, 0xC2, 0x63, 0xC4, 0xDD,
182                 0xD0, 0x10, 0x53, 0x0A, 0x33, 0x07, 0x33, 0x09,
183                 0x62, 0x86, 0x31, 0xF3, 0x08, 0x74, 0x7E, 0x1B,
184                 0xCB, 0xAA, 0x90, 0xE4, 0x51, 0xCA, 0xB9, 0x2E,
185                 0x51, 0x88, 0x08, 0x7A, 0xF4, 0x18, 0x87, 0x73,
186                 0xA3, 0x32, 0x30, 0x3E, 0x66, 0x67, 0xA7, 0xA2,
187                 0x10, 0x85, 0x6F, 0x74, 0x21, 0x39, 0x00, 0x00,
188                 0x71, 0xF4, 0x8E, 0x8B, 0xA2, 0xA5, 0xAD, 0xB7
189         }
190         // no test digests for test_msg4
191 };

193 const uint8_t   skein_1024_test_digests[][128] = {
```



```
194         {
195                 // for test_msg0
196                 0xE6, 0x2C, 0x05, 0x80, 0x2E, 0xA0, 0x15, 0x24,
197                 0x07, 0xCD, 0xD8, 0x78, 0x7F, 0xDA, 0x9E, 0x35,
198                 0x70, 0x3D, 0xE8, 0x62, 0xA4, 0xFB, 0xC1, 0x19,
199                 0xCF, 0xF8, 0x59, 0x0A, 0xFE, 0x79, 0x25, 0x0B,
200                 0xCC, 0xC8, 0xB3, 0xFA, 0xF1, 0xBD, 0x24, 0x22,
201                 0xAB, 0x5C, 0x0D, 0x26, 0x3F, 0xB2, 0xF8, 0xAF,
202                 0xB3, 0xF7, 0x96, 0xF0, 0x48, 0x00, 0x03, 0x81,
203                 0x53, 0x1B, 0x6F, 0x00, 0xD8, 0x51, 0x61, 0xBC,
204                 0x0F, 0xFF, 0x4B, 0xEF, 0x24, 0x86, 0xB1, 0xEB,
205                 0xCD, 0x37, 0x73, 0xFA, 0xBF, 0x50, 0xAD, 0x4A,
206                 0xD5, 0x63, 0x9A, 0xF9, 0x04, 0x0E, 0x3F, 0x29,
207                 0xC6, 0xC9, 0x31, 0x30, 0x1B, 0xF7, 0x98, 0x32,
208                 0xE9, 0xDA, 0x09, 0x85, 0x7E, 0x83, 0x1E, 0x82,
209                 0xEF, 0x8B, 0x46, 0x91, 0xC2, 0x35, 0x65, 0x65,
210                 0x15, 0xD4, 0x37, 0xD2, 0xBD, 0xA3, 0x3B, 0xCE,
211                 0xC0, 0x01, 0xC6, 0x7F, 0xFD, 0xE1, 0x5B, 0xA8
212         },
213         {
214                 // no test vector for test_msg1
215         },
216         {
217                 // no test vector for test_msg2
218         },
219         {
220                 // for test_msg3
221                 0x1F, 0x3E, 0x02, 0xC4, 0x6F, 0xB8, 0x0A, 0x3F,
222                 0xCD, 0x2D, 0xFB, 0xBC, 0x7C, 0x17, 0x38, 0x00,
223                 0xB4, 0x0C, 0x60, 0xC2, 0x35, 0x4A, 0xF5, 0x51,
224                 0x18, 0x9E, 0xBF, 0x43, 0x3C, 0x3D, 0x85, 0xF9,
225                 0xFF, 0x18, 0x03, 0xE6, 0xD9, 0x20, 0x49, 0x31,
226                 0x79, 0xED, 0x7A, 0xE7, 0xFC, 0xE6, 0x9C, 0x35,
227                 0x81, 0xA5, 0xA2, 0xF8, 0x2D, 0x3E, 0x0C, 0x7A,
228                 0x29, 0x55, 0x74, 0xD0, 0xCD, 0x7D, 0x21, 0x7C,
229                 0x48, 0x4D, 0x2F, 0x63, 0x13, 0xD5, 0x9A, 0x77,
230                 0x18, 0xEA, 0xD0, 0x7D, 0x07, 0x29, 0xC2, 0x48,
231                 0x51, 0xD7, 0xE7, 0xD2, 0x49, 0x1B, 0x90, 0x2D,
232                 0x48, 0x91, 0x94, 0xE6, 0xB7, 0xD3, 0x69, 0xDB,
233                 0x0A, 0xB7, 0xAA, 0x10, 0x6F, 0x0E, 0xE0, 0xA3,
234                 0x9A, 0x42, 0xEF, 0xC5, 0x4F, 0x18, 0xD9, 0x37,
235                 0x76, 0x08, 0x09, 0x85, 0xF9, 0x07, 0x57, 0x4F,
236                 0x99, 0x5E, 0xC6, 0xA3, 0x71, 0x53, 0xA5, 0x78
237         },
238         {
239                 // for test_msg4
240                 0x84, 0x2A, 0x53, 0xC9, 0x9C, 0x12, 0xB0, 0xCF,
241                 0x80, 0xCF, 0x69, 0x49, 0x1B, 0xE5, 0xE2, 0xF7,
242                 0x51, 0x5D, 0xE8, 0x73, 0x3B, 0x6E, 0xA9, 0x42,
243                 0x2D, 0xFD, 0x67, 0x66, 0x65, 0xB5, 0xFA, 0x42,
244                 0xFF, 0xB3, 0xA9, 0xC4, 0x8C, 0x21, 0x77, 0x77,
245                 0x95, 0x08, 0x48, 0xCE, 0xCD, 0xB4, 0x8F, 0x64,
246                 0x0F, 0x81, 0xFB, 0x92, 0xBE, 0xF6, 0xF8, 0x8F,
247                 0x7A, 0x85, 0xC1, 0xF7, 0xCD, 0x14, 0x46, 0xC9,
248                 0x16, 0x1C, 0x0A, 0xFE, 0x8F, 0x25, 0xAE, 0x44,
249                 0x4F, 0x40, 0xD3, 0x68, 0x00, 0x81, 0xC3, 0x5A,
250                 0xA4, 0x3F, 0x64, 0x0F, 0xD5, 0xFA, 0x3C, 0x3C,
251                 0x03, 0x0B, 0xCC, 0x06, 0xAB, 0xAC, 0x01, 0xD0,
252                 0x98, 0xBC, 0xC9, 0x84, 0xEB, 0xD8, 0x32, 0x27,
253                 0x12, 0x92, 0x1E, 0x00, 0xB1, 0xBA, 0x07, 0xD6,
254                 0xD0, 0x1F, 0x26, 0x90, 0x70, 0x50, 0x25, 0x5E,
255                 0xF2, 0xC8, 0xE2, 0x4F, 0x71, 0x6C, 0x52, 0xA5
256         }
257 };

259 int
```

```
260 main(int argc, char *argv[])
261 {
262         boolean_t       failed = B_FALSE;
263         uint64_t        cpu_mhz = 0;

265         if (argc == 2)
266                 cpu_mhz = atoi(argv[1]);

268 #define SKEIN_ALGO_TEST(_m, mode, diglen, testdigest)                   \
269         do {                                                            \
270                 Skein ## mode ## _Ctxt_t        ctx;                    \
271                 uint8_t                         digest[diglen / 8];     \
272                 Skein ## mode ## _Init(&ctx, diglen);                   \
273                 Skein ## mode ## _Update(&ctx, _m, sizeof (_m));        \
274                 Skein ## mode ## _Final(&ctx, digest);                  \
275                 printf("Skein" #mode "/" #diglen "\tMessage: " #_m      \
276                     "\tResult: ");                                      \
277                 if (bcmp(digest, testdigest, diglen / 8) == 0) {        \
278                         printf("OK\n");                                 \
279                 } else {                                                \
280                         printf("FAILED!\n");                            \
281                         failed = B_TRUE;                                \
282                 }                                                       \
283         } while (0)

285 #define SKEIN_PERF_TEST(mode, diglen)                                   \
286         do {                                                            \
287                 Skein ## mode ## _Ctxt_t ctx;                           \
288                 uint8_t         digest[diglen / 8];                     \
289                 uint8_t         block[131072];                          \
290                 uint64_t        delta;                                  \
291                 double          cpb = 0;                                \
292                 int             i;                                      \
293                 struct timeval  start, end;                             \
294                 bzero(block, sizeof (block));                           \
295                 gettimeofday(&start, NULL);                             \
296                 Skein ## mode ## _Init(&ctx, diglen);                   \
297                 for (i = 0; i < 8192; i++) {                            \
298                         Skein ## mode ## _Update(&ctx, block,           \
299                             sizeof (block));                            \
300                 }                                                       \
301                 Skein ## mode ## _Final(&ctx, digest);                  \
302                 gettimeofday(&end, NULL);                               \
303                 delta = (end.tv_sec * 1000000llu + end.tv_usec) -       \
304                     (start.tv_sec * 1000000llu + start.tv_usec);        \
305                 if (cpu_mhz != 0) {                                     \
306                         cpb = (cpu_mhz * 1e6 * ((double)delta /         \
307                             1000000)) / (8192 * 128 * 1024);            \
308                 }                                                       \
309                 printf("Skein" #mode "/" #diglen "\t%llu us "           \
310                     "(%.02f CPB)\n", delta, cpb);                       \
311         } while (0)

313         printf("Running algorithm correctness tests:\n");
314         SKEIN_ALGO_TEST(test_msg0, _256, 256, skein_256_test_digests[0]);
315         SKEIN_ALGO_TEST(test_msg1, _256, 256, skein_256_test_digests[1]);
316         SKEIN_ALGO_TEST(test_msg2, _256, 256, skein_256_test_digests[2]);
317         SKEIN_ALGO_TEST(test_msg0, _512, 512, skein_512_test_digests[0]);
318         SKEIN_ALGO_TEST(test_msg2, _512, 512, skein_512_test_digests[2]);
319         SKEIN_ALGO_TEST(test_msg3, _512, 512, skein_512_test_digests[3]);
320         SKEIN_ALGO_TEST(test_msg0, 1024, 1024, skein_1024_test_digests[0]);
321         SKEIN_ALGO_TEST(test_msg3, 1024, 1024, skein_1024_test_digests[3]);
322         SKEIN_ALGO_TEST(test_msg4, 1024, 1024, skein_1024_test_digests[4]);
323         if (failed)
324                 return (1);
```



```
326         printf("Running performance tests (hashing 1024 MiB of data):\n");
327         SKEIN_PERF_TEST(_256, 256);
328         SKEIN_PERF_TEST(_512, 512);
329         SKEIN_PERF_TEST(1024, 1024);

331         return (0);
332 }
```

****
5033 Mon Oct 7 19:17:12 2013
new/usr/src/common/zfs/zfeature_common.c
4185 New hash algorithm support
****
_____***unchanged_portion_omitted_***

```
 152 void
 153 zpool_feature_init(void)
 154 {
 155         zfeature_register(SPA_FEATURE_ASYNC_DESTROY,
 156             "com.delphix:async_destroy", "async_destroy",
 157             "Destroy filesystems asynchronously.", B_TRUE, B_FALSE, NULL);
 158         zfeature_register(SPA_FEATURE_EMPTY_BPOBJ,
 159             "com.delphix:empty_bpobj", "empty_bpobj",
 160             "Snapshots use less space.", B_TRUE, B_FALSE, NULL);
 161         zfeature_register(SPA_FEATURE_LZ4_COMPRESS,
 162             "org.illumos:lz4_compress", "lz4_compress",
 163             "LZ4 compression algorithm support.", B_FALSE, B_FALSE, NULL);
 164         zfeature_register(SPA_FEATURE_MULTI_VDEV_CRASH_DUMP,
 165             "com.joyent:multi_vdev_crash_dump", "multi_vdev_crash_dump",
 166             "Crash dumps to multiple vdev pools.", B_FALSE, B_FALSE, NULL);
 167         zfeature_register(SPA_FEATURE_SPACEMAP_HISTOGRAM,
 168             "com.delphix:spacemap_histogram", "spacemap_histogram",
 169             "Spacemaps maintain space histograms.", B_TRUE, B_FALSE, NULL);
 170         zfeature_register(SPA_FEATURE_SHA512,
 171             "org.illumos:sha512", "sha512",
 172             "SHA-512/256 truncated hash algorithm.", B_FALSE, B_FALSE, NULL);
 173         zfeature_register(SPA_FEATURE_SKEIN,
 174             "org.illumos:skein", "skein",
 175             "Skein hash algorithm.", B_FALSE, B_FALSE, NULL);
 176         zfeature_register(SPA_FEATURE_EDONR,
 177             "org.illumos:edonr", "edonr",
 178             "Edon-R hash algorithm.", B_FALSE, B_FALSE, NULL);
 179 }
```

_____***unchanged_portion_omitted_***

****
2365 Mon Oct 7 19:17:13 2013
new/usr/src/common/zfs/zfeature_common.h
4185 New hash algorithm support
****
_____***unchanged_portion_omitted_***

```
  50 typedef int (zfeature_func_t)(zfeature_info_t *fi, void *arg);

  52 #define ZFS_FEATURE_DEBUG

  54 enum spa_feature {
  55         SPA_FEATURE_ASYNC_DESTROY,
  56         SPA_FEATURE_EMPTY_BPOBJ,
  57         SPA_FEATURE_LZ4_COMPRESS,
  58         SPA_FEATURE_MULTI_VDEV_CRASH_DUMP,
  59         SPA_FEATURE_SPACEMAP_HISTOGRAM,
  60         SPA_FEATURE_SHA512,
  61         SPA_FEATURE_SKEIN,
  62         SPA_FEATURE_EDONR,
  63         SPA_FEATURES
  64 } spa_feature_t;
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    7065 Mon Oct  7 19:17:13 2013
new/usr/src/common/zfs/zfs_fletcher.c
4185 New hash algorithm support
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2013 Saso Kiselkov. All rights reserved.
 */

/*
 * Fletcher Checksums
 * ------------------
 *
 * ZFS's 2nd and 4th order Fletcher checksums are defined by the following
 * recurrence relations:
 *
 *      a  = a    + f
 *       i    i-1    i-1
 *
 *      b  = b    + a
 *       i    i-1    i
 *
 *      c  = c    + b           (fletcher-4 only)
 *       i    i-1    i
 *
 *      d  = d    + c           (fletcher-4 only)
 *       i    i-1    i
 *
 * Where
 *      a_0 = b_0 = c_0 = d_0 = 0
 * and
 *      f_0 .. f_(n-1) are the input data.
 *
 * Using standard techniques, these translate into the following series:
 *
 *           __n_                          __n_
 *           \   |                         \   |
 *      a  =  >     f               b  =  >     i * f
 *       n   /___|   n - i            n   /___|       n - i
 *           i = 1                         i = 1
 *
 *
```



```
 *           __n_                          __n_
 *           \   |  i*(i+1)                 \   |  i*(i+1)*(i+2)
 *      c  = >     ------- f           d  = >     ------------- f
 *       n   /___|     2     n - i       n   /___|        6        n - i
 *           i = 1                         i = 1
 *
 * For fletcher-2, the f_is are 64-bit, and [ab]_i are 64-bit accumulators.
 * Since the additions are done mod (2^64), errors in the high bits may not
 * be noticed.  For this reason, fletcher-2 is deprecated.
 *
 * For fletcher-4, the f_is are 32-bit, and [abcd]_i are 64-bit accumulators.
 * A conservative estimate of how big the buffer can get before we overflow
 * can be estimated using f_i = 0xffffffff for all i:
 *
 * % bc
 *  f=2^32-1;d=0; for (i = 1; d<2^64; i++) { d += f*i*(i+1)*(i+2)/6 }; (i-1)*4
 * 2264
 *  quit
 * %
 *
 * So blocks of up to 2k will not overflow.  Our largest block size is
 * 128k, which has 32k 4-byte words, so we can compute the largest possible
 * accumulators, then divide by 2^64 to figure the max amount of overflow:
 *
 * % bc
 *  a=b=c=d=0; f=2^32-1; for (i=1; i<=32*1024; i++) { a+=f; b+=a; c+=b; d+=c }
 *  a/2^64;b/2^64;c/2^64;d/2^64
 * 0
 * 0
 * 1365
 * 11186858
 *  quit
 * %
 *
 * So a and b cannot overflow.  To make sure each bit of input has some
 * effect on the contents of c and d, we can look at what the factors of
 * the coefficients in the equations for c_n and d_n are.  The number of 2s
 * in the factors determines the lowest set bit in the multiplier.  Running
 * through the cases for n*(n+1)/2 reveals that the highest power of 2 is
 * 2^14, and for n*(n+1)*(n+2)/6 it is 2^15.  So while some data may overflow
 * the 64-bit accumulators, every bit of every f_i effects every accumulator,
 * even for 128k blocks.
 *
 * If we wanted to make a stronger version of fletcher4 (fletcher4c?),
 * we could do our calculations mod (2^32 - 1) by adding in the carries
 * periodically, and store the number of carries in the top 32-bits.
 *
 * --------------------
 * Checksum Performance
 * --------------------
 *
 * There are two interesting components to checksum performance: cached and
 * uncached performance.  With cached data, fletcher-2 is about four times
 * faster than fletcher-4.  With uncached data, the performance difference is
 * negligible, since the cost of a cache fill dominates the processing time.
 * Even though fletcher-4 is slower than fletcher-2, it is still a pretty
 * efficient pass over the data.
 *
 * In normal operation, the data which is being checksummed is in a buffer
 * which has been filled either by:
 *
 *      1. a compression step, which will be mostly cached, or
 *      2. a bcopy() or copyin(), which will be uncached (because the
 *         copy is cache-bypassing).
 *
 * For both cached and uncached data, both fletcher checksums are much faster
```

```
 * than sha-256, and slower than 'off', which doesn't touch the data at all.
 */

#include <sys/types.h>
#include <sys/sysmacros.h>
#include <sys/byteorder.h>
#include <sys/zio.h>
#include <sys/spa.h>

/*ARGSUSED*/
void
fletcher_2_native(const void *buf, uint64_t size, const zio_cksum_salt_t *salt,
    const void *ctx_template, zio_cksum_t *zcp)
fletcher_2_native(const void *buf, uint64_t size, zio_cksum_t *zcp)
{
        const uint64_t *ip = buf;
        const uint64_t *ipend = ip + (size / sizeof (uint64_t));
        uint64_t a0, b0, a1, b1;

        for (a0 = b0 = a1 = b1 = 0; ip < ipend; ip += 2) {
                a0 += ip[0];
                a1 += ip[1];
                b0 += a0;
                b1 += a1;
        }

        ZIO_SET_CHECKSUM(zcp, a0, a1, b0, b1);
}

/*ARGSUSED*/
void
fletcher_2_byteswap(const void *buf, uint64_t size,
    const zio_cksum_salt_t *salt, const void *ctx_template, zio_cksum_t *zcp)
fletcher_2_byteswap(const void *buf, uint64_t size, zio_cksum_t *zcp)
{
        const uint64_t *ip = buf;
        const uint64_t *ipend = ip + (size / sizeof (uint64_t));
        uint64_t a0, b0, a1, b1;

        for (a0 = b0 = a1 = b1 = 0; ip < ipend; ip += 2) {
                a0 += BSWAP_64(ip[0]);
                a1 += BSWAP_64(ip[1]);
                b0 += a0;
                b1 += a1;
        }

        ZIO_SET_CHECKSUM(zcp, a0, a1, b0, b1);
}

/*ARGSUSED*/
void
fletcher_4_native(const void *buf, uint64_t size, const zio_cksum_salt_t *salt,
    const void *ctx_template, zio_cksum_t *zcp)
fletcher_4_native(const void *buf, uint64_t size, zio_cksum_t *zcp)
{
        const uint32_t *ip = buf;
        const uint32_t *ipend = ip + (size / sizeof (uint32_t));
        uint64_t a, b, c, d;

        for (a = b = c = d = 0; ip < ipend; ip++) {
                a += ip[0];
                b += a;
                c += b;
                d += c;
        }
```



```
        ZIO_SET_CHECKSUM(zcp, a, b, c, d);
}

/*ARGSUSED*/
void
fletcher_4_byteswap(const void *buf, uint64_t size,
    const zio_cksum_salt_t *salt, const void *ctx_template, zio_cksum_t *zcp)
fletcher_4_byteswap(const void *buf, uint64_t size, zio_cksum_t *zcp)
{
        const uint32_t *ip = buf;
        const uint32_t *ipend = ip + (size / sizeof (uint32_t));
        uint64_t a, b, c, d;

        for (a = b = c = d = 0; ip < ipend; ip++) {
                a += BSWAP_32(ip[0]);
                b += a;
                c += b;
                d += c;
        }

        ZIO_SET_CHECKSUM(zcp, a, b, c, d);
}
______unchanged_portion_omitted_
```

```
**********************************************************
    1812 Mon Oct  7 19:17:13 2013
new/usr/src/common/zfs/zfs_fletcher.h
4185 New hash algorithm support
**********************************************************
```

```
  1 /*
  2  * CDDL HEADER START
  3  *
  4  * The contents of this file are subject to the terms of the
  5  * Common Development and Distribution License (the "License").
  6  * You may not use this file except in compliance with the License.
  7  *
  8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
  9  * or http://www.opensolaris.org/os/licensing.
 10  * See the License for the specific language governing permissions
 11  * and limitations under the License.
 12  *
 13  * When distributing Covered Code, include this CDDL HEADER in each
 14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 15  * If applicable, add the following below this CDDL HEADER, with the
 16  * fields enclosed by brackets "[]" replaced with your own identifying
 17  * information: Portions Copyright [yyyy] [name of copyright owner]
 18  *
 19  * CDDL HEADER END
 20  */
 21 /*
 22  * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 23  * Use is subject to license terms.
 24  */
 25 /*
 26  * Copyright 2013 Saso Kiselkov. All rights reserved.
 27  */

 29 #ifndef _ZFS_FLETCHER_H
 30 #define _ZFS_FLETCHER_H

 32 #include <sys/types.h>
 33 #include <sys/spa.h>

 35 #ifdef  __cplusplus
 36 extern "C" {
 37 #endif

 39 /*
 40  * fletcher checksum functions
 41  */

 43 void fletcher_2_native(const void *, uint64_t, const zio_cksum_salt_t *,
 44     const void *, zio_cksum_t *);
 45 void fletcher_2_byteswap(const void *, uint64_t, const zio_cksum_salt_t *,
 46     const void *, zio_cksum_t *);
 47 void fletcher_4_native(const void *, uint64_t, const zio_cksum_salt_t *,
 48     const void *, zio_cksum_t *);
 49 void fletcher_4_byteswap(const void *, uint64_t, const zio_cksum_salt_t *,
 50     const void *, zio_cksum_t *);
 40 void fletcher_2_native(const void *, uint64_t, zio_cksum_t *);
 41 void fletcher_2_byteswap(const void *, uint64_t, zio_cksum_t *);
 42 void fletcher_4_native(const void *, uint64_t, zio_cksum_t *);
 43 void fletcher_4_byteswap(const void *, uint64_t, zio_cksum_t *);
 51 void fletcher_4_incremental_native(const void *, uint64_t,
 52     zio_cksum_t *);
 53 void fletcher_4_incremental_byteswap(const void *, uint64_t,
 54     zio_cksum_t *);

 56 #ifdef  __cplusplus
 57 }
```

_____*unchanged_portion_omitted_*

```
**********************************************************
   20779 Mon Oct  7 19:17:13 2013
new/usr/src/common/zfs/zfs_prop.c
4185 New hash algorithm support
**********************************************************
______unchanged_portion_omitted_

  64 void
  65 zfs_prop_init(void)
  66 {
  67         static zprop_index_t checksum_table[] = {
  68                 { "on",         ZIO_CHECKSUM_ON },
  69                 { "off",        ZIO_CHECKSUM_OFF },
  70                 { "fletcher2",  ZIO_CHECKSUM_FLETCHER_2 },
  71                 { "fletcher4",  ZIO_CHECKSUM_FLETCHER_4 },
  72                 { "sha256",     ZIO_CHECKSUM_SHA256 },
  73                 { "noparity",   ZIO_CHECKSUM_NOPARITY },
  74                 { "sha512",     ZIO_CHECKSUM_SHA512 },
  75                 { "skein",      ZIO_CHECKSUM_SKEIN },
  76                 { "edonr",      ZIO_CHECKSUM_EDONR },
  77                 { NULL }
  78         };
  79
  80         static zprop_index_t dedup_table[] = {
  81                 { "on",         ZIO_CHECKSUM_ON },
  82                 { "off",        ZIO_CHECKSUM_OFF },
  83                 { "verify",     ZIO_CHECKSUM_ON | ZIO_CHECKSUM_VERIFY },
  84                 { "sha256",     ZIO_CHECKSUM_SHA256 },
  85                 { "sha256,verify",
  86                                 ZIO_CHECKSUM_SHA256 | ZIO_CHECKSUM_VERIFY },
  87                 { "sha512",     ZIO_CHECKSUM_SHA512 },
  88                 { "sha512,verify",
  89                                 ZIO_CHECKSUM_SHA512 | ZIO_CHECKSUM_VERIFY },
  90                 { "skein",      ZIO_CHECKSUM_SKEIN },
  91                 { "skein,verify",
  92                                 ZIO_CHECKSUM_SKEIN | ZIO_CHECKSUM_VERIFY },
  93                 { "edonr",      ZIO_CHECKSUM_EDONR },
  94                 { "edonr,verify",
  95                                 ZIO_CHECKSUM_EDONR | ZIO_CHECKSUM_VERIFY },
  96                 { NULL }
  97         };
  98
  99         static zprop_index_t compress_table[] = {
 100                 { "on",         ZIO_COMPRESS_ON },
 101                 { "off",        ZIO_COMPRESS_OFF },
 102                 { "lzjb",       ZIO_COMPRESS_LZJB },
 103                 { "gzip",       ZIO_COMPRESS_GZIP_6 },  /* gzip default */
 104                 { "gzip-1",     ZIO_COMPRESS_GZIP_1 },
 105                 { "gzip-2",     ZIO_COMPRESS_GZIP_2 },
 106                 { "gzip-3",     ZIO_COMPRESS_GZIP_3 },
 107                 { "gzip-4",     ZIO_COMPRESS_GZIP_4 },
 108                 { "gzip-5",     ZIO_COMPRESS_GZIP_5 },
 109                 { "gzip-6",     ZIO_COMPRESS_GZIP_6 },
 110                 { "gzip-7",     ZIO_COMPRESS_GZIP_7 },
 111                 { "gzip-8",     ZIO_COMPRESS_GZIP_8 },
 112                 { "gzip-9",     ZIO_COMPRESS_GZIP_9 },
 113                 { "zle",        ZIO_COMPRESS_ZLE },
 114                 { "lz4",        ZIO_COMPRESS_LZ4 },
 115                 { NULL }
 116         };
 117
 118         static zprop_index_t snapdir_table[] = {
 119                 { "hidden",     ZFS_SNAPDIR_HIDDEN },
 120                 { "visible",    ZFS_SNAPDIR_VISIBLE },
 121                 { NULL }
 122         };
```



```
 124         static zprop_index_t acl_mode_table[] = {
 125                 { "discard",    ZFS_ACL_DISCARD },
 126                 { "groupmask",  ZFS_ACL_GROUPMASK },
 127                 { "passthrough", ZFS_ACL_PASSTHROUGH },
 128                 { "restricted", ZFS_ACL_RESTRICTED },
 129                 { NULL }
 130         };
 131
 132         static zprop_index_t acl_inherit_table[] = {
 133                 { "discard",    ZFS_ACL_DISCARD },
 134                 { "noallow",    ZFS_ACL_NOALLOW },
 135                 { "restricted", ZFS_ACL_RESTRICTED },
 136                 { "passthrough", ZFS_ACL_PASSTHROUGH },
 137                 { "secure",     ZFS_ACL_RESTRICTED }, /* bkwrd compatability */
 138                 { "passthrough-x", ZFS_ACL_PASSTHROUGH_X },
 139                 { NULL }
 140         };
 141
 142         static zprop_index_t case_table[] = {
 143                 { "sensitive",          ZFS_CASE_SENSITIVE },
 144                 { "insensitive",        ZFS_CASE_INSENSITIVE },
 145                 { "mixed",              ZFS_CASE_MIXED },
 146                 { NULL }
 147         };
 148
 149         static zprop_index_t copies_table[] = {
 150                 { "1",          1 },
 151                 { "2",          2 },
 152                 { "3",          3 },
 153                 { NULL }
 154         };
 155
 156         /*
 157          * Use the unique flags we have to send to u8_strcmp() and/or
 158          * u8_textprep() to represent the various normalization property
 159          * values.
 160          */
 161         static zprop_index_t normalize_table[] = {
 162                 { "none",       0 },
 163                 { "formD",      U8_TEXTPREP_NFD },
 164                 { "formKC",     U8_TEXTPREP_NFKC },
 165                 { "formC",      U8_TEXTPREP_NFC },
 166                 { "formKD",     U8_TEXTPREP_NFKD },
 167                 { NULL }
 168         };
 169
 170         static zprop_index_t version_table[] = {
 171                 { "1",          1 },
 172                 { "2",          2 },
 173                 { "3",          3 },
 174                 { "4",          4 },
 175                 { "5",          5 },
 176                 { "current",    ZPL_VERSION },
 177                 { NULL }
 178         };
 179
 180         static zprop_index_t boolean_table[] = {
 181                 { "off",        0 },
 182                 { "on",         1 },
 183                 { NULL }
 184         };
 185
 186         static zprop_index_t logbias_table[] = {
 187                 { "latency",    ZFS_LOGBIAS_LATENCY },
 188                 { "throughput", ZFS_LOGBIAS_THROUGHPUT },
```

```
189                 { NULL }
190         };

192         static zprop_index_t canmount_table[] = {
193                 { "off",        ZFS_CANMOUNT_OFF },
194                 { "on",         ZFS_CANMOUNT_ON },
195                 { "noauto",     ZFS_CANMOUNT_NOAUTO },
196                 { NULL }
197         };

199         static zprop_index_t cache_table[] = {
200                 { "none",       ZFS_CACHE_NONE },
201                 { "metadata",   ZFS_CACHE_METADATA },
202                 { "all",        ZFS_CACHE_ALL },
203                 { NULL }
204         };

206         static zprop_index_t sync_table[] = {
207                 { "standard",   ZFS_SYNC_STANDARD },
208                 { "always",     ZFS_SYNC_ALWAYS },
209                 { "disabled",   ZFS_SYNC_DISABLED },
210                 { NULL }
211         };

213         /* inherit index properties */
214         zprop_register_index(ZFS_PROP_SYNC, "sync", ZFS_SYNC_STANDARD,
215             PROP_INHERIT, ZFS_TYPE_FILESYSTEM | ZFS_TYPE_VOLUME,
216             "standard | always | disabled", "SYNC",
217             sync_table);
218         zprop_register_index(ZFS_PROP_CHECKSUM, "checksum",
219             ZIO_CHECKSUM_DEFAULT, PROP_INHERIT, ZFS_TYPE_FILESYSTEM |
220             ZFS_TYPE_VOLUME,
221             "on | off | fletcher2 | fletcher4 | sha256 | sha512 | "
222             "skein | edonr", "CHECKSUM", checksum_table);
209             "on | off | fletcher2 | fletcher4 | sha256", "CHECKSUM",
210             checksum_table);
223         zprop_register_index(ZFS_PROP_DEDUP, "dedup", ZIO_CHECKSUM_OFF,
224             PROP_INHERIT, ZFS_TYPE_FILESYSTEM | ZFS_TYPE_VOLUME,
225             "on | off | verify | sha256[,verify], sha512[,verify], "
226             "skein[,verify], edonr[,verify]", "DEDUP", dedup_table);
213             "on | off | verify | sha256[,verify]", "DEDUP",
214             dedup_table);
227         zprop_register_index(ZFS_PROP_COMPRESSION, "compression",
228             ZIO_COMPRESS_DEFAULT, PROP_INHERIT,
229             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_VOLUME,
230             "on | off | lzjb | gzip | gzip-[1-9] | zle | lz4",
231             "COMPRESS", compress_table);
232         zprop_register_index(ZFS_PROP_SNAPDIR, "snapdir", ZFS_SNAPDIR_HIDDEN,
233             PROP_INHERIT, ZFS_TYPE_FILESYSTEM,
234             "hidden | visible", "SNAPDIR", snapdir_table);
235         zprop_register_index(ZFS_PROP_ACLMODE, "aclmode", ZFS_ACL_DISCARD,
236             PROP_INHERIT, ZFS_TYPE_FILESYSTEM,
237             "discard | groupmask | passthrough | restricted", "ACLMODE",
238             acl_mode_table);
239         zprop_register_index(ZFS_PROP_ACLINHERIT, "aclinherit",
240             ZFS_ACL_RESTRICTED, PROP_INHERIT, ZFS_TYPE_FILESYSTEM,
241             "discard | noallow | restricted | passthrough | passthrough-x",
242             "ACLINHERIT", acl_inherit_table);
243         zprop_register_index(ZFS_PROP_COPIES, "copies", 1, PROP_INHERIT,
244             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_VOLUME,
245             "1 | 2 | 3", "COPIES", copies_table);
246         zprop_register_index(ZFS_PROP_PRIMARYCACHE, "primarycache",
247             ZFS_CACHE_ALL, PROP_INHERIT,
248             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_SNAPSHOT | ZFS_TYPE_VOLUME,
249             "all | none | metadata", "PRIMARYCACHE", cache_table);
250         zprop_register_index(ZFS_PROP_SECONDARYCACHE, "secondarycache",
```



```
251             ZFS_CACHE_ALL, PROP_INHERIT,
252             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_SNAPSHOT | ZFS_TYPE_VOLUME,
253             "all | none | metadata", "SECONDARYCACHE", cache_table);
254         zprop_register_index(ZFS_PROP_LOGBIAS, "logbias", ZFS_LOGBIAS_LATENCY,
255             PROP_INHERIT, ZFS_TYPE_FILESYSTEM | ZFS_TYPE_VOLUME,
256             "latency | throughput", "LOGBIAS", logbias_table);

258         /* inherit index (boolean) properties */
259         zprop_register_index(ZFS_PROP_ATIME, "atime", 1, PROP_INHERIT,
260             ZFS_TYPE_FILESYSTEM, "on | off", "ATIME", boolean_table);
261         zprop_register_index(ZFS_PROP_DEVICES, "devices", 1, PROP_INHERIT,
262             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_SNAPSHOT, "on | off", "DEVICES",
263             boolean_table);
264         zprop_register_index(ZFS_PROP_EXEC, "exec", 1, PROP_INHERIT,
265             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_SNAPSHOT, "on | off", "EXEC",
266             boolean_table);
267         zprop_register_index(ZFS_PROP_SETUID, "setuid", 1, PROP_INHERIT,
268             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_SNAPSHOT, "on | off", "SETUID",
269             boolean_table);
270         zprop_register_index(ZFS_PROP_READONLY, "readonly", 0, PROP_INHERIT,
271             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_VOLUME, "on | off", "RDONLY",
272             boolean_table);
273         zprop_register_index(ZFS_PROP_ZONED, "zoned", 0, PROP_INHERIT,
274             ZFS_TYPE_FILESYSTEM, "on | off", "ZONED", boolean_table);
275         zprop_register_index(ZFS_PROP_XATTR, "xattr", 1, PROP_INHERIT,
276             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_SNAPSHOT, "on | off", "XATTR",
277             boolean_table);
278         zprop_register_index(ZFS_PROP_VSCAN, "vscan", 0, PROP_INHERIT,
279             ZFS_TYPE_FILESYSTEM, "on | off", "VSCAN",
280             boolean_table);
281         zprop_register_index(ZFS_PROP_NBMAND, "nbmand", 0, PROP_INHERIT,
282             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_SNAPSHOT, "on | off", "NBMAND",
283             boolean_table);

285         /* default index properties */
286         zprop_register_index(ZFS_PROP_VERSION, "version", 0, PROP_DEFAULT,
287             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_SNAPSHOT,
288             "1 | 2 | 3 | 4 | 5 | current", "VERSION", version_table);
289         zprop_register_index(ZFS_PROP_CANMOUNT, "canmount", ZFS_CANMOUNT_ON,
290             PROP_DEFAULT, ZFS_TYPE_FILESYSTEM, "on | off | noauto",
291             "CANMOUNT", canmount_table);

293         /* readonly index (boolean) properties */
294         zprop_register_index(ZFS_PROP_MOUNTED, "mounted", 0, PROP_READONLY,
295             ZFS_TYPE_FILESYSTEM, "yes | no", "MOUNTED", boolean_table);
296         zprop_register_index(ZFS_PROP_DEFER_DESTROY, "defer_destroy", 0,
297             PROP_READONLY, ZFS_TYPE_SNAPSHOT, "yes | no", "DEFER_DESTROY",
298             boolean_table);

300         /* set once index properties */
301         zprop_register_index(ZFS_PROP_NORMALIZE, "normalization", 0,
302             PROP_ONETIME, ZFS_TYPE_FILESYSTEM | ZFS_TYPE_SNAPSHOT,
303             "none | formC | formD | formKC | formKD", "NORMALIZATION",
304             normalize_table);
305         zprop_register_index(ZFS_PROP_CASE, "casesensitivity",
306             ZFS_CASE_SENSITIVE, PROP_ONETIME, ZFS_TYPE_FILESYSTEM |
307             ZFS_TYPE_SNAPSHOT,
308             "sensitive | insensitive | mixed", "CASE", case_table);

310         /* set once index (boolean) properties */
311         zprop_register_index(ZFS_PROP_UTF8ONLY, "utf8only", 0, PROP_ONETIME,
312             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_SNAPSHOT,
313             "on | off", "UTF8ONLY", boolean_table);

315         /* string properties */
316         zprop_register_string(ZFS_PROP_ORIGIN, "origin", NULL, PROP_READONLY,
```

```
317             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_VOLUME, "<snapshot>", "ORIGIN");
318         zprop_register_string(ZFS_PROP_CLONES, "clones", NULL, PROP_READONLY,
319             ZFS_TYPE_SNAPSHOT, "<dataset>[,...]", "CLONES");
320         zprop_register_string(ZFS_PROP_MOUNTPOINT, "mountpoint", "/",
321             PROP_INHERIT, ZFS_TYPE_FILESYSTEM, "<path> | legacy | none",
322             "MOUNTPOINT");
323         zprop_register_string(ZFS_PROP_SHARENFS, "sharenfs", "off",
324             PROP_INHERIT, ZFS_TYPE_FILESYSTEM, "on | off | share(1M) options",
325             "SHARENFS");
326         zprop_register_string(ZFS_PROP_TYPE, "type", NULL, PROP_READONLY,
327             ZFS_TYPE_DATASET, "filesystem | volume | snapshot", "TYPE");
328         zprop_register_string(ZFS_PROP_SHARESMB, "sharesmb", "off",
329             PROP_INHERIT, ZFS_TYPE_FILESYSTEM,
330             "on | off | sharemgr(1M) options", "SHARESMB");
331         zprop_register_string(ZFS_PROP_MLSLABEL, "mlslabel",
332             ZFS_MLSLABEL_DEFAULT, PROP_INHERIT, ZFS_TYPE_DATASET,
333             "<sensitivity label>", "MLSLABEL");

335         /* readonly number properties */
336         zprop_register_number(ZFS_PROP_USED, "used", 0, PROP_READONLY,
337             ZFS_TYPE_DATASET, "<size>", "USED");
338         zprop_register_number(ZFS_PROP_AVAILABLE, "available", 0, PROP_READONLY,
339             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_VOLUME, "<size>", "AVAIL");
340         zprop_register_number(ZFS_PROP_REFERENCED, "referenced", 0,
341             PROP_READONLY, ZFS_TYPE_DATASET, "<size>", "REFER");
342         zprop_register_number(ZFS_PROP_COMPRESSRATIO, "compressratio", 0,
343             PROP_READONLY, ZFS_TYPE_DATASET,
344             "<1.00x or higher if compressed>", "RATIO");
345         zprop_register_number(ZFS_PROP_REFRATIO, "refcompressratio", 0,
346             PROP_READONLY, ZFS_TYPE_DATASET,
347             "<1.00x or higher if compressed>", "REFRATIO");
348         zprop_register_number(ZFS_PROP_VOLBLOCKSIZE, "volblocksize",
349             ZVOL_DEFAULT_BLOCKSIZE, PROP_ONETIME,
350             ZFS_TYPE_VOLUME, "512 to 128k, power of 2", "VOLBLOCK");
351         zprop_register_number(ZFS_PROP_USEDSNAP, "usedbysnapshots", 0,
352             PROP_READONLY, ZFS_TYPE_FILESYSTEM | ZFS_TYPE_VOLUME, "<size>",
353             "USEDSNAP");
354         zprop_register_number(ZFS_PROP_USEDDS, "usedbydataset", 0,
355             PROP_READONLY, ZFS_TYPE_FILESYSTEM | ZFS_TYPE_VOLUME, "<size>",
356             "USEDDS");
357         zprop_register_number(ZFS_PROP_USEDCHILD, "usedbychildren", 0,
358             PROP_READONLY, ZFS_TYPE_FILESYSTEM | ZFS_TYPE_VOLUME, "<size>",
359             "USEDCHILD");
360         zprop_register_number(ZFS_PROP_USEDREFRESERV, "usedbyrefreservation", 0,
361             PROP_READONLY,
362             ZFS_TYPE_FILESYSTEM | ZFS_TYPE_VOLUME, "<size>", "USEDREFRESERV");
363         zprop_register_number(ZFS_PROP_USERREFS, "userrefs", 0, PROP_READONLY,
364             ZFS_TYPE_SNAPSHOT, "<count>", "USERREFS");
365         zprop_register_number(ZFS_PROP_WRITTEN, "written", 0, PROP_READONLY,
366             ZFS_TYPE_DATASET, "<size>", "WRITTEN");
367         zprop_register_number(ZFS_PROP_LOGICALUSED, "logicalused", 0,
368             PROP_READONLY, ZFS_TYPE_DATASET, "<size>", "LUSED");
369         zprop_register_number(ZFS_PROP_LOGICALREFERENCED, "logicalreferenced",
370             0, PROP_READONLY, ZFS_TYPE_DATASET, "<size>", "LREFER");

372         /* default number properties */
373         zprop_register_number(ZFS_PROP_QUOTA, "quota", 0, PROP_DEFAULT,
374             ZFS_TYPE_FILESYSTEM, "<size> | none", "QUOTA");
375         zprop_register_number(ZFS_PROP_RESERVATION, "reservation", 0,
376             PROP_DEFAULT, ZFS_TYPE_FILESYSTEM | ZFS_TYPE_VOLUME,
377             "<size> | none", "RESERV");
378         zprop_register_number(ZFS_PROP_VOLSIZE, "volsize", 0, PROP_DEFAULT,
379             ZFS_TYPE_VOLUME, "<size>", "VOLSIZE");
380         zprop_register_number(ZFS_PROP_REFQUOTA, "refquota", 0, PROP_DEFAULT,
381             ZFS_TYPE_FILESYSTEM, "<size> | none", "REFQUOTA");
382         zprop_register_number(ZFS_PROP_REFRESERVATION, "refreservation", 0,
```



```
383             PROP_DEFAULT, ZFS_TYPE_FILESYSTEM | ZFS_TYPE_VOLUME,
384             "<size> | none", "REFRESERV");

386         /* inherit number properties */
387         zprop_register_number(ZFS_PROP_RECORDSIZE, "recordsize",
388             SPA_MAXBLOCKSIZE, PROP_INHERIT,
389             ZFS_TYPE_FILESYSTEM, "512 to 128k, power of 2", "RECSIZE");

391         /* hidden properties */
392         zprop_register_hidden(ZFS_PROP_CREATETXG, "createtxg", PROP_TYPE_NUMBER,
393             PROP_READONLY, ZFS_TYPE_DATASET, "CREATETXG");
394         zprop_register_hidden(ZFS_PROP_NUMCLONES, "numclones", PROP_TYPE_NUMBER,
395             PROP_READONLY, ZFS_TYPE_SNAPSHOT, "NUMCLONES");
396         zprop_register_hidden(ZFS_PROP_NAME, "name", PROP_TYPE_STRING,
397             PROP_READONLY, ZFS_TYPE_DATASET, "NAME");
398         zprop_register_hidden(ZFS_PROP_ISCSIOPTIONS, "iscsioptions",
399             PROP_TYPE_STRING, PROP_INHERIT, ZFS_TYPE_VOLUME, "ISCSIOPTIONS");
400         zprop_register_hidden(ZFS_PROP_STMF_SHAREINFO, "stmf_sbd_lu",
401             PROP_TYPE_STRING, PROP_INHERIT, ZFS_TYPE_VOLUME,
402             "STMF_SBD_LU");
403         zprop_register_hidden(ZFS_PROP_GUID, "guid", PROP_TYPE_NUMBER,
404             PROP_READONLY, ZFS_TYPE_DATASET, "GUID");
405         zprop_register_hidden(ZFS_PROP_USERACCOUNTING, "useraccounting",
406             PROP_TYPE_NUMBER, PROP_READONLY, ZFS_TYPE_DATASET,
407             "USERACCOUNTING");
408         zprop_register_hidden(ZFS_PROP_UNIQUE, "unique", PROP_TYPE_NUMBER,
409             PROP_READONLY, ZFS_TYPE_DATASET, "UNIQUE");
410         zprop_register_hidden(ZFS_PROP_OBJSETID, "objsetid", PROP_TYPE_NUMBER,
411             PROP_READONLY, ZFS_TYPE_DATASET, "OBJSETID");
412         zprop_register_hidden(ZFS_PROP_INCONSISTENT, "inconsistent",
413             PROP_TYPE_NUMBER, PROP_READONLY, ZFS_TYPE_DATASET, "INCONSISTENT");

415         /* oddball properties */
416         zprop_register_impl(ZFS_PROP_CREATION, "creation", PROP_TYPE_NUMBER, 0,
417             NULL, PROP_READONLY, ZFS_TYPE_DATASET,
418             "<date>", "CREATION", B_FALSE, B_TRUE, NULL);
419 }
```

*unchanged_portion_omitted_*

```
**********************************************************
    1392 Mon Oct  7 19:17:13 2013
new/usr/src/grub/capability
4185 New hash algorithm support
**********************************************************
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
  22 # Copyright (c) 2012 by Delphix. All rights reserved.
  23 # Copyright 2013 Saso Kiselkov.  All rights reserved.
  24 #
  25 # This file defines the current capabilities of GRUB over and above that
  26 # supported by the standard distribution
  27 #
  28 # The version field contains the version of the associated GRUB software.  The
  29 # version is incremented by 1 each time there is a bugfix or enhancement to
  30 # GRUB necessitating that the boot blocks be reinstalled for that fix or
  31 # enhancement to take effect.
  32 #
  33 VERSION=23
  32 VERSION=22
  34 dboot
  35 xVM
  36 zfs
  37 findroot
```

```
**********************************************************
   43727 Mon Oct  7 19:17:14 2013
new/usr/src/grub/grub-0.97/stage2/fsys_zfs.c
4185 New hash algorithm support
**********************************************************
______unchanged_portion_omitted_

 136 /* Checksum Table and Values */
 137 zio_checksum_info_t zio_checksum_table[ZIO_CHECKSUM_FUNCTIONS] = {
 138         {{NULL,                 NULL},                  0, 0,   "inherit"},
 139         {{NULL,                 NULL},                  0, 0,   "on"},
 140         {{zio_checksum_off,     zio_checksum_off},      0, 0,   "off"},
 141         {{zio_checksum_SHA256,  zio_checksum_SHA256},   1, 1,   "label"},
 142         {{zio_checksum_SHA256,  zio_checksum_SHA256},   1, 1,   "gang_header"},
 143         {{NULL,                 NULL},                  0, 0,   "zilog"},
 144         {{fletcher_2_native,    fletcher_2_byteswap},   0, 0,   "fletcher2"},
 145         {{fletcher_4_native,    fletcher_4_byteswap},   1, 0,   "fletcher4"},
 146         {{zio_checksum_SHA256,  zio_checksum_SHA256},   1, 0,   "SHA256"},
 147         {{NULL,                 NULL},                  0, 0,   "zilog2"},
 148         {{zio_checksum_SHA512,  NULL},                  0, 0,   "SHA512"}
 149 };

 151 /*
 152  * zio_checksum_verify: Provides support for checksum verification.
 153  *
 154  * Fletcher2, Fletcher4, SHA-256 and SHA-512/256 are supported.
 153  * Fletcher2, Fletcher4, and SHA256 are supported.
 155  *
 156  * Return:
 157  *      -1 = Failure
 158  *       0 = Success
 159  */
 160 static int
 161 zio_checksum_verify(blkptr_t *bp, char *data, int size)
 162 {
 163         zio_cksum_t zc = bp->blk_cksum;
 164         uint32_t checksum = BP_GET_CHECKSUM(bp);
 165         int byteswap = BP_SHOULD_BYTESWAP(bp);
 166         zio_eck_t *zec = (zio_eck_t *)(data + size) - 1;
 167         zio_checksum_info_t *ci = &zio_checksum_table[checksum];
 168         zio_cksum_t actual_cksum, expected_cksum;

 170         /* byteswap is not supported */
 171         if (byteswap)
 172                 return (-1);

 174         if (checksum >= ZIO_CHECKSUM_FUNCTIONS || ci->ci_func[0] == NULL)
 175                 return (-1);

 177         if (ci->ci_eck) {
 178                 expected_cksum = zec->zec_cksum;
 179                 zec->zec_cksum = zc;
 180                 ci->ci_func[0](data, size, &actual_cksum);
 181                 zec->zec_cksum = expected_cksum;
 182                 zc = expected_cksum;

 184         } else {
 185                 ci->ci_func[byteswap](data, size, &actual_cksum);
 186         }

 188         if ((actual_cksum.zc_word[0] - zc.zc_word[0]) |
 189             (actual_cksum.zc_word[1] - zc.zc_word[1]) |
 190             (actual_cksum.zc_word[2] - zc.zc_word[2]) |
 191             (actual_cksum.zc_word[3] - zc.zc_word[3]))
 192                 return (-1);
```



```
 194         return (0);
 195 }
______unchanged_portion_omitted_

 960 /*
 961  * List of pool features that the grub implementation of ZFS supports for
 962  * read. Note that features that are only required for write do not need
 963  * to be listed here since grub opens pools in read-only mode.
 964  *
 965  * When this list is updated the version number in usr/src/grub/capability
 966  * must be incremented to ensure the new grub gets installed.
 967  */
 968 static const char *spa_feature_names[] = {
 969         "org.illumos:lz4_compress",
 970         "org.illumos:sha512",
 971         NULL
 972 };
______unchanged_portion_omitted_
```

****
6491 Mon Oct 7 19:17:14 2013
new/usr/src/grub/grub-0.97/stage2/fsys_zfs.h
4185 New hash algorithm support
****
_____***unchanged_portion_omitted_***

```
 174 /*
 175  * FAT ZAP data structures
 176  */
 177 #define ZFS_CRC64_POLY 0xC96C5795D7870F42ULL /* ECMA-182, reflected form */
 178 #define ZAP_HASH_IDX(hash, n)   (((n) == 0) ? 0 : ((hash) >> (64 - (n))))
 179 #define CHAIN_END       0xffff  /* end of the chunk chain */

 181 /*
 182  * The amount of space within the chunk available for the array is:
 183  * chunk size - space for type (1) - space for next pointer (2)
 184  */
 185 #define ZAP_LEAF_ARRAY_BYTES (ZAP_LEAF_CHUNKSIZE - 3)

 187 #define ZAP_LEAF_HASH_SHIFT(bs) (bs - 5)
 188 #define ZAP_LEAF_HASH_NUMENTRIES(bs) (1 << ZAP_LEAF_HASH_SHIFT(bs))
 189 #define LEAF_HASH(bs, h) \
 190         ((ZAP_LEAF_HASH_NUMENTRIES(bs)-1) & \
 191         ((h) >> (64 - ZAP_LEAF_HASH_SHIFT(bs)-l->l_hdr.lh_prefix_len)))

 193 /*
 194  * The amount of space available for chunks is:
 195  * block size shift - hash entry size (2) * number of hash
 196  * entries - header space (2*chunksize)
 197  */
 198 #define ZAP_LEAF_NUMCHUNKS(bs) \
 199         (((1<<bs) - 2*ZAP_LEAF_HASH_NUMENTRIES(bs)) / \
 200         ZAP_LEAF_CHUNKSIZE - 2)

 202 /*
 203  * The chunks start immediately after the hash table.  The end of the
 204  * hash table is at l_hash + HASH_NUMENTRIES, which we simply cast to a
 205  * chunk_t.
 206  */
 207 #define ZAP_LEAF_CHUNK(l, bs, idx) \
 208         ((zap_leaf_chunk_t *)(l->l_hash + ZAP_LEAF_HASH_NUMENTRIES(bs)))[idx]
 209 #define ZAP_LEAF_ENTRY(l, bs, idx) (&ZAP_LEAF_CHUNK(l, bs, idx).l_entry)

 211 extern void fletcher_2_native(const void *, uint64_t, zio_cksum_t *);
 212 extern void fletcher_2_byteswap(const void *, uint64_t, zio_cksum_t *);
 213 extern void fletcher_4_native(const void *, uint64_t, zio_cksum_t *);
 214 extern void fletcher_4_byteswap(const void *, uint64_t, zio_cksum_t *);
 215 extern void zio_checksum_SHA256(const void *, uint64_t, zio_cksum_t *);
 216 extern void zio_checksum_SHA512(const void *, uint64_t, zio_cksum_t *);
 217 extern int lzjb_decompress(void *, void *, size_t, size_t);
 218 extern int lz4_decompress(void *, void *, size_t, size_t);

 220 #endif  /* FSYS_ZFS */

 222 #endif /* !_FSYS_ZFS_H */
```

****
2572 Mon Oct 7 19:17:14 2013
new/usr/src/grub/grub-0.97/stage2/zfs-include/zio.h
4185 New hash algorithm support
****
_____***unchanged_portion_omitted_***

```
 59 enum zio_checksum {
 60         ZIO_CHECKSUM_INHERIT = 0,
 61         ZIO_CHECKSUM_ON,
 62         ZIO_CHECKSUM_OFF,
 63         ZIO_CHECKSUM_LABEL,
 64         ZIO_CHECKSUM_GANG_HEADER,
 65         ZIO_CHECKSUM_ZILOG,
 66         ZIO_CHECKSUM_FLETCHER_2,
 67         ZIO_CHECKSUM_FLETCHER_4,
 68         ZIO_CHECKSUM_SHA256,
 69         ZIO_CHECKSUM_ZILOG2,
 70         ZIO_CHECKSUM_SHA512,
 71         ZIO_CHECKSUM_FUNCTIONS
 72 };
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    9827 Mon Oct  7 19:17:14 2013
new/usr/src/grub/grub-0.97/stage2/zfs_sha256.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 *  GRUB  --  GRand Unified Bootloader
 *  Copyright (C) 1999,2000,2001,2002,2003,2004  Free Software Foundation, Inc.
 *
 *  This program is free software; you can redistribute it and/or modify
 *  it under the terms of the GNU General Public License as published by
 *  the Free Software Foundation; either version 2 of the License, or
 *  (at your option) any later version.
 *
 *  This program is distributed in the hope that it will be useful,
 *  but WITHOUT ANY WARRANTY; without even the implied warranty of
 *  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 *  GNU General Public License for more details.
 *
 *  You should have received a copy of the GNU General Public License
 *  along with this program; if not, write to the Free Software
 *  Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */
/*
 * Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2013 Saso Kiselkov.  All rights reserved.
 */

#pragma ident   "%Z%%M% %I%     %E% SMI"

#include "fsys_zfs.h"

/*
 * SHA-256 and SHA-512/256 hashes, as specified in FIPS 180-4, available at:
 * SHA-256 checksum, as specified in FIPS 180-2, available at:
 * http://csrc.nist.gov/cryptval
 *
 * This is a very compact implementation of SHA-256 and SHA-512/256.
 * This is a very compact implementation of SHA-256.
 * It is designed to be simple and portable, not to be fast.
 */

/*
 * The literal definitions according to FIPS180-4 would be:
 * The literal definitions according to FIPS180-2 would be:
 *
 *      Ch(x, y, z)     (((x) & (y)) ^ ((~(x)) & (z)))
 *      Maj(x, y, z)    (((x) & (y)) | ((x) & (z)) | ((y) & (z)))
 *
 * We use logical equivalents which require one less op.
 */
#define Ch(x, y, z)     ((z) ^ ((x) & ((y) ^ (z))))
#define Maj(x, y, z)    (((x) & (y)) ^ ((z) & ((x) ^ (y))))
#define ROTR(x, n)      (((x) >> (n)) | ((x) << ((sizeof (x) * NBBY)-(n))))
#define Rot32(x, s)     (((x) >> s) | ((x) << (32 - s)))
#define SIGMA0(x)       (Rot32(x, 2) ^ Rot32(x, 13) ^ Rot32(x, 22))
#define SIGMA1(x)       (Rot32(x, 6) ^ Rot32(x, 11) ^ Rot32(x, 25))
#define sigma0(x)       (Rot32(x, 7) ^ Rot32(x, 18) ^ ((x) >> 3))
#define sigma1(x)       (Rot32(x, 17) ^ Rot32(x, 19) ^ ((x) >> 10))

/* SHA-224/256 operations */
#define BIGSIGMA0_256(x)        (ROTR(x, 2) ^ ROTR(x, 13) ^ ROTR(x, 22))
#define BIGSIGMA1_256(x)        (ROTR(x, 6) ^ ROTR(x, 11) ^ ROTR(x, 25))
```



```
#define SIGMA0_256(x)           (ROTR(x, 7) ^ ROTR(x, 18) ^ ((x) >> 3))
#define SIGMA1_256(x)           (ROTR(x, 17) ^ ROTR(x, 19) ^ ((x) >> 10))

/* SHA-384/512 operations */
#define BIGSIGMA0_512(x)        (ROTR((x), 28) ^ ROTR((x), 34) ^ ROTR((x), 39))
#define BIGSIGMA1_512(x)        (ROTR((x), 14) ^ ROTR((x), 18) ^ ROTR((x), 41))
#define SIGMA0_512(x)           (ROTR((x), 1) ^ ROTR((x), 8) ^ ((x) >> 7))
#define SIGMA1_512(x)           (ROTR((x), 19) ^ ROTR((x), 61) ^ ((x) >> 6))

/* SHA-256 round constants */
static const uint32_t SHA256_K[64] = {
        0x428a2f98, 0x71374491, 0xb5c0fbcf, 0xe9b5dba5,
        0x3956c25b, 0x59f111f1, 0x923f82a4, 0xab1c5ed5,
        0xd807aa98, 0x12835b01, 0x243185be, 0x550c7dc3,
        0x72be5d74, 0x80deb1fe, 0x9bdc06a7, 0xc19bf174,
        0xe49b69c1, 0xefbe4786, 0x0fc19dc6, 0x240ca1cc,
        0x2de92c6f, 0x4a7484aa, 0x5cb0a9dc, 0x76f988da,
        0x983e5152, 0xa831c66d, 0xb00327c8, 0xbf597fc7,
        0xc6e00bf3, 0xd5a79147, 0x06ca6351, 0x14292967,
        0x27b70a85, 0x2e1b2138, 0x4d2c6dfc, 0x53380d13,
        0x650a7354, 0x766a0abb, 0x81c2c92e, 0x92722c85,
        0xa2bfe8a1, 0xa81a664b, 0xc24b8b70, 0xc76c51a3,
        0xd192e819, 0xd6990624, 0xf40e3585, 0x106aa070,
        0x19a4c116, 0x1e376c08, 0x2748774c, 0x34b0bcb5,
        0x391c0cb3, 0x4ed8aa4a, 0x5b9cca4f, 0x682e6ff3,
        0x748f82ee, 0x78a5636f, 0x84c87814, 0x8cc70208,
        0x90befffa, 0xa4506ceb, 0xbef9a3f7, 0xc67178f2
};

/* SHA-512 round constants */
static const uint64_t SHA512_K[80] = {
        0x428A2F98D728AE22ULL, 0x7137449123EF65CDULL,
        0xB5C0FBCFEC4D3B2FULL, 0xE9B5DBA58189DBBCULL,
        0x3956C25BF348B538ULL, 0x59F111F1B605D019ULL,
        0x923F82A4AF194F9BULL, 0xAB1C5ED5DA6D8118ULL,
        0xD807AA98A3030242ULL, 0x12835B0145706FBEULL,
        0x243185BE4EE4B28CULL, 0x550C7DC3D5FFB4E2ULL,
        0x72BE5D74F27B896FULL, 0x80DEB1FE3B1696B1ULL,
        0x9BDC06A725C71235ULL, 0xC19BF174CF692694ULL,
        0xE49B69C19EF14AD2ULL, 0xEFBE4786384F25E3ULL,
        0x0FC19DC68B8CD5B5ULL, 0x240CA1CC77AC9C65ULL,
        0x2DE92C6F592B0275ULL, 0x4A7484AA6EA6E483ULL,
        0x5CB0A9DCBD41FBD4ULL, 0x76F988DA831153B5ULL,
        0x983E5152EE66DFABULL, 0xA831C66D2DB43210ULL,
        0xB00327C898FB213FULL, 0xBF597FC7BEEF0EE4ULL,
        0xC6E00BF33DA88FC2ULL, 0xD5A79147930AA725ULL,
        0x06CA6351E003826FULL, 0x142929670A0E6E70ULL,
        0x27B70A8546D22FFCULL, 0x2E1B21385C26C926ULL,
        0x4D2C6DFC5AC42AEDULL, 0x53380D139D95B3DFULL,
        0x650A73548BAF63DEULL, 0x766A0ABB3C77B2A8ULL,
        0x81C2C92E47EDAEE6ULL, 0x92722C851482353BULL,
        0xA2BFE8A14CF10364ULL, 0xA81A664BBC423001ULL,
        0xC24B8B70D0F89791ULL, 0xC76C51A30654BE30ULL,
        0xD192E819D6EF5218ULL, 0xD69906245565A910ULL,
        0xF40E35855771202AULL, 0x106AA07032BBD1B8ULL,
        0x19A4C116B8D2D0C8ULL, 0x1E376C085141AB53ULL,
        0x2748774CDF8EEB99ULL, 0x34B0BCB5E19B48A8ULL,
        0x391C0CB3C5C95A63ULL, 0x4ED8AA4AE3418ACBULL,
        0x5B9CCA4F7763E373ULL, 0x682E6FF3D6B2B8A3ULL,
        0x748F82EE5DEFB2FCULL, 0x78A5636F43172F60ULL,
        0x84C87814A1F0AB72ULL, 0x8CC702081A6439ECULL,
        0x90BEFFFA23631E28ULL, 0xA4506CEBDE82BDE9ULL,
        0xBEF9A3F7B2C67915ULL, 0xC67178F2E372532BULL,
        0xCA273ECEEA26619CULL, 0xD186B8C721C0C207ULL,
        0xEADA7DD6CDE0EB1EULL, 0xF57D4F7FEE6ED178ULL,
        0x06F067AA72176FBAULL, 0x0A637DC5A2C898A6ULL,
```

```
        0x113F9804BEF90DAEULL, 0x1B710B35131C471BULL,
        0x28DB77F523047D84ULL, 0x32CAAB7B40C72493ULL,
        0x3C9EBE0A15C9BEBCULL, 0x431D67C49C100D4CULL,
        0x4CC5D4BECB3E42B6ULL, 0x597F299CFC657E2AULL,
        0x5FCB6FAB3AD6FAECULL, 0x6C44198C4A475817ULL
};

static void
SHA256Transform(uint32_t *H, const uint8_t *cp)
{
        uint32_t a, b, c, d, e, f, g, h, t, T1, T2, W[64];

        /* copy chunk into the first 16 words of the message schedule */
        for (t = 0; t < 16; t++, cp +=  sizeof (uint32_t))
 76     for (t = 0; t < 16; t++, cp += 4)
132             W[t] = (cp[0] << 24) | (cp[1] << 16) | (cp[2] << 8) | cp[3];

        /* extend the first 16 words into the remaining 48 words */
        for (t = 16; t < 64; t++)
                W[t] = SIGMA1_256(W[t - 2]) + W[t - 7] +
                    SIGMA0_256(W[t - 15]) + W[t - 16];
 80             W[t] = sigma1(W[t - 2]) + W[t - 7] +
 81                 sigma0(W[t - 15]) + W[t - 16];

        /* init working variables to the current hash value */
        a = H[0]; b = H[1]; c = H[2]; d = H[3];
        e = H[4]; f = H[5]; g = H[6]; h = H[7];

        /* iterate the compression function for all rounds of the hash */
        for (t = 0; t < 64; t++) {
                T1 = h + BIGSIGMA1_256(e) + Ch(e, f, g) + SHA256_K[t] + W[t];
                T2 = BIGSIGMA0_256(a) + Maj(a, b, c);
 87             T1 = h + SIGMA1(e) + Ch(e, f, g) + SHA256_K[t] + W[t];
 88             T2 = SIGMA0(a) + Maj(a, b, c);
                h = g; g = f; f = e; e = d + T1;
                d = c; c = b; b = a; a = T1 + T2;
        }

        /* add the compressed chunk to the current hash value */
        H[0] += a; H[1] += b; H[2] += c; H[3] += d;
        H[4] += e; H[5] += f; H[6] += g; H[7] += h;
}

static void
SHA512Transform(uint64_t *H, const uint8_t *cp)
 97 void
 98 zio_checksum_SHA256(const void *buf, uint64_t size, zio_cksum_t *zcp)
{
        uint64_t a, b, c, d, e, f, g, h, t, T1, T2, W[80];

        /* copy chunk into the first 16 words of the message schedule */
        for (t = 0; t < 16; t++, cp += sizeof (uint64_t))
                W[t] = ((uint64_t)cp[0] << 56) | ((uint64_t)cp[1] << 48) |
                    ((uint64_t)cp[2] << 40) | ((uint64_t)cp[3] << 32) |
                    (cp[4] << 24) | (cp[5] << 16) | (cp[6] << 8) | cp[7];

        /* extend the first 16 words into the remaining 64 words */
        for (t = 16; t < 80; t++)
                W[t] = SIGMA1_512(W[t - 2]) + W[t - 7] +
                    SIGMA0_512(W[t - 15]) + W[t - 16];

        /* init working variables to the current hash value */
        a = H[0]; b = H[1]; c = H[2]; d = H[3];
        e = H[4]; f = H[5]; g = H[6]; h = H[7];

        /* iterate the compression function for all rounds of the hash */
```



```
        for (t = 0; t < 80; t++) {
                T1 = h + BIGSIGMA1_512(e) + Ch(e, f, g) + SHA512_K[t] + W[t];
                T2 = BIGSIGMA0_512(a) + Maj(a, b, c);
                h = g; g = f; f = e; e = d + T1;
                d = c; c = b; b = a; a = T1 + T2;
        }

        /* add the compressed chunk to the current hash value */
        H[0] += a; H[1] += b; H[2] += c; H[3] += d;
        H[4] += e; H[5] += f; H[6] += g; H[7] += h;
}

/*
 * Implements the SHA-224 and SHA-256 hash algos - to select between them
 * pass the appropriate initial values of 'H' and truncate the last 32 bits
 * in case of SHA-224.
 */
static void
SHA256(uint32_t *H, const void *buf, uint64_t size, zio_cksum_t *zcp)
{
100     uint32_t H[8] = { 0x6a09e667, 0xbb67ae85, 0x3c6ef372, 0xa54ff53a,
101         0x510e527f, 0x9b05688c, 0x1f83d9ab, 0x5be0cd19 };
197     uint8_t         pad[128];
198     unsigned        padsize = size & 63;
199     unsigned        i;
103     int padsize = size & 63;
104     int i;

        /* process all blocks up to the last one */
        for (i = 0; i < size - padsize; i += 64)
                SHA256Transform(H, (uint8_t *)buf + i);

        /* process the last block and padding */
        for (i = 0; i < padsize; i++)
                pad[i] = ((uint8_t *)buf)[i];

        for (pad[padsize++] = 0x80; (padsize & 63) != 56; padsize++)
                pad[padsize] = 0;

        for (i = 0; i < 8; i++)
                pad[padsize++] = (size << 3) >> (56 - 8 * i);

        for (i = 0; i < padsize; i += 64)
                SHA256Transform(H, pad + i);

        ZIO_SET_CHECKSUM(zcp,
            (uint64_t)H[0] << 32 | H[1],
            (uint64_t)H[2] << 32 | H[3],
            (uint64_t)H[4] << 32 | H[5],
            (uint64_t)H[6] << 32 | H[7]);
}

/*
 * Implements the SHA-384, SHA-512 and SHA-512/t hash algos - to select
 * between them pass the appropriate initial values for 'H'. The output
 * of this function is truncated to the first 256 bits that fit into 'zcp'.
 */
static void
SHA512(uint64_t *H, const void *buf, uint64_t size, zio_cksum_t *zcp)
{
        uint8_t         pad[256];
        unsigned        padsize = size & 127;
        unsigned        i;

        /* process all blocks up to the last one */
        for (i = 0; i < size - padsize; i += 128)
```

```
239                 SHA512Transform(H, (uint8_t *)buf + i);

241         /* process the last block and padding */
242         for (i = 0; i < padsize; i++)
243                 pad[i] = ((uint8_t *)buf)[i];

245         for (pad[padsize++] = 0x80; (padsize & 127) != 120; padsize++)
246                 pad[padsize] = 0;

248         for (i = 0; i < 8; i++)
249                 pad[padsize++] = (size << 3) >> (120 - 8 * i);

251         for (i = 0; i < padsize; i += 128)
252                 SHA512Transform(H, pad + i);

254         /* truncate the output to the first 256 bits which fit into 'zcp' */
255         ZIO_SET_CHECKSUM(zcp, H[0], H[1], H[2], H[3]);
256 }

258 void
259 zio_checksum_SHA256(const void *buf, uint64_t size, zio_cksum_t *zcp)
260 {
261         /* SHA-256 as per FIPS 180-4. */
262         uint32_t        H[] = {
263                 0x6a09e667, 0xbb67ae85, 0x3c6ef372, 0xa54ff53a,
264                 0x510e527f, 0x9b05688c, 0x1f83d9ab, 0x5be0cd19
265         };
266         SHA256(H, buf, size, zcp);
267 }

269 void
270 zio_checksum_SHA512(const void *buf, uint64_t size, zio_cksum_t *zcp)
271 {
272         /* SHA-512/256 as per FIPS 180-4. */
273         uint64_t        H[] = {
274                 0x22312194FC2BF72CULL, 0x9F555FA3C84C64C2ULL,
275                 0x2393B86B6F53B151ULL, 0x963877195940EABDULL,
276                 0x96283EE2A88EFFE3ULL, 0xBE5E1E2553863992ULL,
277                 0x2B0199FC2C85B8AAULL, 0x0EB72DDC81C52CA2ULL
278         };
279         SHA512(H, buf, size, zcp);
280 }
```

*unchanged_portion_omitted_*

```
**********************************************************
    1473 Mon Oct  7 19:17:14 2013
new/usr/src/lib/libmd/Makefile
4185 New hash algorithm support
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2013 Saso Kiselkov. All rights reserved.
#

include         $(SRC)/lib/Makefile.lib

$(SPARC_BLD)CAPDIR = capabilities

SUBDIRS=        $(MACH) $(BUILD64) $(MACH64)

HDRS =          md4.h md5.h sha1.h sha2.h skein.h
HDRS =          md4.h md5.h sha1.h sha2.h
HDRDIR =        common

all :=          TARGET= all
clean :=        TARGET= clean
clobber :=      TARGET= clobber
install :=      TARGET= install
lint :=         TARGET= lint

.KEEP_STATE:

.PARALLEL:      $(SUBDIRS)

all clean clobber install lint: \
                $(CAPDIR) .WAIT $(SUBDIRS)

install_h:      $(ROOTHDRS)

check:          $(CHECKHDRS)

$(CAPDIR) \
$(SUBDIRS):     FRC
                @cd $@; pwd; $(MAKE) $(TARGET)

FRC:

include         $(SRC)/lib/Makefile.targ
```

```
**********************************************************
    1338 Mon Oct  7 19:17:15 2013
new/usr/src/lib/libmd/Makefile.com
4185 New hash algorithm support
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2013 Saso Kiselkov. All rights reserved.
#

LIBS =          $(DYNLIB) $(LINTLIB)
SRCS =          $(COMDIR)/edonr/edonr.c \
                $(COMDIR)/md4/md4.c \
SRCS =          $(COMDIR)/md4/md4.c \
                $(COMDIR)/md5/md5.c \
                $(COMDIR)/sha1/sha1.c \
                $(COMDIR)/sha2/sha2.c \
                $(COMDIR)/skein/skein.c \
                $(COMDIR)/skein/skein_block.c
                $(COMDIR)/sha2/sha2.c

COMDIR =        $(SRC)/common/crypto
SRCDIR =        ../common
MAPFILEDIR =    $(SRCDIR)

CFLAGS +=       $(CCVERBOSE)
CPPFLAGS +=     -I$(SRCDIR)
LDLIBS +=       -lc

$(LINTLIB) :=   SRCS = $(SRCDIR)/$(LINTSRC)
```

```
**********************************************************
    1717 Mon Oct  7 19:17:15 2013
new/usr/src/lib/libmd/Makefile.targ
4185 New hash algorithm support
**********************************************************
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #

  22 #
  23 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
  24 # Copyright 2013 Saso Kiselkov. All rights reserved.
  25 #

  27 COMDIR =        $(SRC)/common/crypto

  29 pics/%.o: $(COMDIR)/edonr/%.c
  30                 $(COMPILE.c) -I$(COMDIR)/edonr -o $@ $<
  31                 $(POST_PROCESS_O)

  33 pics/%.o: $(COMDIR)/md4/%.c
  34                 $(COMPILE.c) -I$(COMDIR)/md4 -o $@ $<
  35                 $(POST_PROCESS_O)

  37 pics/%.o: $(COMDIR)/md5/%.c
  38                 $(COMPILE.c) -I$(COMDIR)/md5 $(INLINES) -o $@ $<
  39                 $(POST_PROCESS_O)

  41 pics/%.o: $(COMDIR)/sha1/%.c
  42                 $(COMPILE.c) -I$(COMDIR)/sha1 -o $@ $<
  43                 $(POST_PROCESS_O)

  45 pics/sha1_asm.o: $(COMDIR)/sha1/sparc/$(PLATFORM)/sha1_asm.s
  46                 $(COMPILE.s) -o pics/sha1_asm.o \
  47                 $(COMDIR)/sha1/sparc/$(PLATFORM)/sha1_asm.s
  48                 $(POST_PROCESS_O)

  50 pics/%.o: $(COMDIR)/sha2/%.c
  51                 $(COMPILE.c) -I$(COMDIR)/sha2 -o $@ $<
  52                 $(POST_PROCESS_O)

  54 pics/%.o: $(COMDIR)/skein/%.c
  55                 $(COMPILE.c) -I$(COMDIR)/skein -o $@ $<
  56                 $(POST_PROCESS_O)

  58 include        $(SRC)/lib/Makefile.targ
```

```
**********************************************************
    1876 Mon Oct  7 19:17:15 2013
new/usr/src/lib/libmd/amd64/Makefile
4185 New hash algorithm support
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2013 Saso Kiselkov. All rights reserved.
#

LIBRARY =       libmd.a
VERS =          .1

OBJECTS =       edonr.o md4.o md5.o sha1.o sha2.o skein.o skein_block.o
OBJECTS =       md4.o md5.o sha1.o sha2.o

include         $(SRC)/lib/Makefile.lib
include         $(SRC)/lib/Makefile.rootfs
include         $(SRC)/lib/Makefile.lib.64
include         ../Makefile.com

EXTPICS =       pics/md5_amd64.o \
                pics/sha1-x86_64.o \
                pics/sha512-x86_64.o \
                pics/sha256-x86_64.o

CLEANFILES +=   $(EXTPICS) \
                $(EXTPICS:pics/%.o=%.s)

# This prevents <sys/asm_linkage.h> from including C source:
AS_CPPFLAGS += -D_ASM

.KEPP_STATE:

all:            $(LIBS)

install:        all $(ROOTLIBS64) $(ROOTLINKS64) $(ROOTLINT64)

lint:           lintcheck

pics/%.o: %.s
                $(COMPILE.s) -o $@ ${@F:.o=.s}
                $(POST_PROCESS_O)

md5_amd64.s: $(COMDIR)/md5/amd64/md5_amd64.pl
                $(PERL) $? $@
```



```
sha1-x86_64.s: $(COMDIR)/sha1/amd64/sha1-x86_64.pl
                $(PERL) $? $@

sha512-x86_64.s: $(COMDIR)/sha2/amd64/sha512-x86_64.pl
                $(PERL) $? $@

sha256-x86_64.s: $(COMDIR)/sha2/amd64/sha512-x86_64.pl
                $(PERL) $? $@

include         ../Makefile.targ
```

```
**********************************************************
    2646 Mon Oct  7 19:17:15 2013
new/usr/src/lib/libmd/common/mapfile-vers
4185 New hash algorithm support
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
  23 # Copyright 2013 Saso Kiselkov. All rights reserved.
  24 #

  26 #
  27 # MAPFILE HEADER START
  28 #
  29 # WARNING:  STOP NOW.  DO NOT MODIFY THIS FILE.
  30 # Object versioning must comply with the rules detailed in
  31 #
  32 #       usr/src/lib/README.mapfiles
  33 #
  34 # You should not be making modifications here until you've read the most current
  35 # copy of that file. If you need help, contact a gatekeeper for guidance.
  36 #
  37 # MAPFILE HEADER END
  38 #

  40 $mapfile_version 2

  42 # Note:
  43 #
  44 #       SHA256Update, SHA384Update, and SHA512Update are all
  45 #       weak aliases for SHA2Update.
  46 #
  47 #       SHA256Final, SHA384Final, and SHA512Final are all
  48 #       weak aliases for SHA2Final
  49 #
  50 # We use NODYNSORT to keep the weak aliases out of the .SUNW_dymsymsort
  51 # ELF section. As a result, ld will put the two main symbols in.
  52 #


  55 SYMBOL_VERSION SUNW_1.1 {
  56     global:
  57         MD4Final;
  58         MD4Init;
  59         MD4Update;
  60         md5_calc;
  61         MD5Final        { FLAGS = NODIRECT };   # Addr interpose from testsuite
```



```
  62         MD5Init         { FLAGS = NODIRECT };   # Addr interpose from testsuite
  63         MD5Update       { FLAGS = NODIRECT };   # Addr interpose from testsuite
  64         SHA1Final;
  65         SHA1Init;
  66         SHA1Update;
  67         SHA256Final     { FLAGS = NODYNSORT };
  68         SHA256Init;
  69         SHA256Update    { FLAGS = NODYNSORT };
  70         SHA2Final;
  71         SHA2Init;
  72         SHA2Update;
  73         SHA384Final     { FLAGS = NODYNSORT };
  74         SHA384Init;
  75         SHA384Update    { FLAGS = NODYNSORT };
  76         SHA512Final     { FLAGS = NODYNSORT };
  77         SHA512Init;
  78         SHA512Update    { FLAGS = NODYNSORT };
  79         Skein1024_Final;
  80         Skein1024_Final_Pad;
  81         Skein1024_Init;
  82         Skein1024_InitExt;
  83         Skein1024_Output;
  84         Skein1024_Update;
  85         Skein_256_Final;
  86         Skein_256_Final_Pad;
  87         Skein_256_Init;
  88         Skein_256_InitExt;
  89         Skein_256_Output;
  90         Skein_256_Update;
  91         Skein_512_Final;
  92         Skein_512_Final_Pad;
  93         Skein_512_Init;
  94         Skein_512_InitExt;
  95         Skein_512_Output;
  96         Skein_512_Update;
  97         EdonRFinal;
  98         EdonRHash;
  99         EdonRInit;
 100         EdonRUpdate;
 101     local:
 102         *;
 103 };
```

*_____unchanged_portion_omitted_*

```
**********************************************************
     965 Mon Oct  7 19:17:15 2013
new/usr/src/lib/libmd/common/skein.h
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2013 Saso Kiselkov.  All rights reserved.
 */

#ifndef _SKEIN_H
#define _SKEIN_H

#include <sys/skein.h>

#endif  /* _SKEIN_H */
```

```
**********************************************************
    1253 Mon Oct  7 19:17:15 2013
new/usr/src/lib/libmd/i386/Makefile
4185 New hash algorithm support
**********************************************************
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #

  22 #
  23 # Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
  24 # Copyright 2013 Saso Kiselkov. All rights reserved.
  25 #

  27 LIBRARY =       libmd.a
  28 VERS =          .1

  30 OBJECTS =       edonr.o md4.o md5.o sha1.o sha2.o skein.o skein_block.o
  29 OBJECTS =       md4.o md5.o sha1.o sha2.o

  32 include         $(SRC)/lib/Makefile.lib
  33 include         $(SRC)/lib/Makefile.rootfs
  34 include         ../Makefile.com

  36 .KEEP_STATE:

  38 all:            $(LIBS)

  40 install:        all $(ROOTLIBS) $(ROOTLINKS) $(ROOTLINT)

  42 lint:           lintcheck

  44 include         ../Makefile.targ
```

```
**********************************************************
    1229 Mon Oct  7 19:17:16 2013
new/usr/src/lib/libmd/inc.flg
4185 New hash algorithm support
**********************************************************
```

```
#!/bin/sh
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright 2013 Saso Kiselkov.  All rights reserved.
#ident  "%Z%%M% %I%     %E% SMI"

find_files "s.*" usr/src/common/crypto/edonr
find_files "s.*" usr/src/common/crypto/md4
find_files "s.*" usr/src/common/crypto/md5
find_files "s.*" usr/src/common/crypto/sha1
find_files "s.*" usr/src/common/crypto/sha2
find_files "s.*" usr/src/common/crypto/skein
```

```
**********************************************************
    1412 Mon Oct  7 19:17:16 2013
new/usr/src/lib/libmd/sparc/Makefile
4185 New hash algorithm support
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2013 Saso Kiselkov. All rights reserved.
#

LIBRARY =       libmd.a
VERS =          .1

OBJECTS =       edonr.o md4.o md5.o sha1.o sha2.o skein.o skein_block.o
OBJECTS =       md4.o md5.o sha1.o sha2.o

include         $(SRC)/lib/Makefile.lib
include         $(SRC)/lib/Makefile.rootfs
include         ../Makefile.com

# Symbol capabilities objects are added for sun4u and sun4v.
EXTPICS =       ../capabilities/sun4u/sparc/pics/symcap.o \
                ../capabilities/sun4v/sparc/pics/symcap.o

.KEEP_STATE:

all:            $(LIBS)

install:        all $(ROOTLIBS) $(ROOTLINKS) $(ROOTLINT)

lint:           lintcheck

include         ../Makefile.targ
```

****
1459 Mon Oct 7 19:17:16 2013
new/usr/src/lib/libmd/sparcv9/Makefile
4185 New hash algorithm support
****

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2013 Saso Kiselkov. All rights reserved.
#

LIBRARY =       libmd.a
VERS =          .1

OBJECTS =       edonr.o md4.o md5.o sha1.o sha2.o skein.o skein_block.o
OBJECTS =       md4.o md5.o sha1.o sha2.o

include         $(SRC)/lib/Makefile.lib
include         $(SRC)/lib/Makefile.rootfs
include         $(SRC)/lib/Makefile.lib.64
include         ../Makefile.com

# Symbol capabilities objects are added for sun4u and sun4v.
EXTPICS =       ../capabilities/sun4u/sparcv9/pics/symcap.o \
                ../capabilities/sun4v/sparcv9/pics/symcap.o

.KEEP_STATE:

all:            $(LIBS)

install:        all $(ROOTLIBS64) $(ROOTLINKS64) $(ROOTLINT64)

lint:           lintcheck

include         ../Makefile.targ
```

```
**********************************************************
  112025 Mon Oct  7 19:17:16 2013
new/usr/src/lib/libzfs/common/libzfs_dataset.c
4185 New hash algorithm support
**********************************************************
______unchanged_portion_omitted_

1325 void
1326 zfs_setprop_error(libzfs_handle_t *hdl, zfs_prop_t prop, int err,
1327     char *errbuf)
1328 {
1329         switch (err) {

1331         case ENOSPC:
1332                 /*
1333                  * For quotas and reservations, ENOSPC indicates
1334                  * something different; setting a quota or reservation
1335                  * doesn't use any disk space.
1336                  */
1337                 switch (prop) {
1338                 case ZFS_PROP_QUOTA:
1339                 case ZFS_PROP_REFQUOTA:
1340                         zfs_error_aux(hdl, dgettext(TEXT_DOMAIN,
1341                             "size is less than current used or "
1342                             "reserved space"));
1343                         (void) zfs_error(hdl, EZFS_PROPSPACE, errbuf);
1344                         break;

1346                 case ZFS_PROP_RESERVATION:
1347                 case ZFS_PROP_REFRESERVATION:
1348                         zfs_error_aux(hdl, dgettext(TEXT_DOMAIN,
1349                             "size is greater than available space"));
1350                         (void) zfs_error(hdl, EZFS_PROPSPACE, errbuf);
1351                         break;

1353                 default:
1354                         (void) zfs_standard_error(hdl, err, errbuf);
1355                         break;
1356                 }
1357                 break;

1359         case EBUSY:
1360                 (void) zfs_standard_error(hdl, EBUSY, errbuf);
1361                 break;

1363         case EROFS:
1364                 (void) zfs_error(hdl, EZFS_DSREADONLY, errbuf);
1365                 break;

1367         case ENOTSUP:
1368                 zfs_error_aux(hdl, dgettext(TEXT_DOMAIN,
1369                     "pool and or dataset must be upgraded to set this "
1370                     "property or value"));
1371                 (void) zfs_error(hdl, EZFS_BADVERSION, errbuf);
1372                 break;

1374         case ERANGE:
1375                 if (prop == ZFS_PROP_COMPRESSION) {
1376                         (void) zfs_error_aux(hdl, dgettext(TEXT_DOMAIN,
1377                             "property setting is not allowed on "
1378                             "bootable datasets"));
1379                         (void) zfs_error(hdl, EZFS_NOTSUP, errbuf);
1380                 } else if (prop == ZFS_PROP_CHECKSUM ||
1381                     prop == ZFS_PROP_DEDUP) {
1382                         (void) zfs_error_aux(hdl, dgettext(TEXT_DOMAIN,
1383                             "property setting is not allowed on "
```



```
1384                             "root pools"));
1385                         (void) zfs_error(hdl, EZFS_NOTSUP, errbuf);
1386                 } else {
1387                         (void) zfs_standard_error(hdl, err, errbuf);
1388                 }
1389                 break;

1391         case EINVAL:
1392                 if (prop == ZPROP_INVAL) {
1393                         (void) zfs_error(hdl, EZFS_BADPROP, errbuf);
1394                 } else {
1395                         (void) zfs_standard_error(hdl, err, errbuf);
1396                 }
1397                 break;

1399         case EOVERFLOW:
1400                 /*
1401                  * This platform can't address a volume this big.
1402                  */
1403 #ifdef _ILP32
1404                 if (prop == ZFS_PROP_VOLSIZE) {
1405                         (void) zfs_error(hdl, EZFS_VOLTOOBIG, errbuf);
1406                         break;
1407                 }
1408 #endif
1409                 /* FALLTHROUGH */
1410         default:
1411                 (void) zfs_standard_error(hdl, err, errbuf);
1412         }
1413 }
______unchanged_portion_omitted_
```

```
**********************************************************
  111355 Mon Oct  7 19:17:16 2013
new/usr/src/man/man1m/zfs.1m
4185 New hash algorithm support
**********************************************************
   1 '\" t
   2 .\"
   3 .\" CDDL HEADER START
   4 .\"
   5 .\" The contents of this file are subject to the terms of the
   6 .\" Common Development and Distribution License (the "License").
   7 .\" You may not use this file except in compliance with the License.
   8 .\"
   9 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
  10 .\" or http://www.opensolaris.org/os/licensing.
  11 .\" See the License for the specific language governing permissions
  12 .\" and limitations under the License.
  13 .\"
  14 .\" When distributing Covered Code, include this CDDL HEADER in each
  15 .\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  16 .\" If applicable, add the following below this CDDL HEADER, with the
  17 .\" fields enclosed by brackets "[]" replaced with your own identifying
  18 .\" information: Portions Copyright [yyyy] [name of copyright owner]
  19 .\"
  20 .\" CDDL HEADER END
  21 .\"
  22 .\"
  23 .\" Copyright (c) 2009 Sun Microsystems, Inc. All Rights Reserved.
  24 .\" Copyright 2011 Joshua M. Clulow <josh@sysmgr.org>
  25 .\" Copyright (c) 2012 by Delphix. All rights reserved.
  26 .\" Copyright (c) 2013 by Saso Kiselkov. All rights reserved.
  27 .\" Copyright 2013 Nexenta Systems, Inc.  All Rights Reserved.
  28 .\" Copyright (c) 2013, Joyent, Inc. All rights reserved.
  29 .\"
  30 .TH ZFS 1M "Jan 26, 2013"
  31 .SH NAME
  32 zfs \- configures ZFS file systems
  33 .SH SYNOPSIS
  34 .LP
  35 .nf
  36 \fBzfs\fR [\fB-?\fR]
  37 .fi

  39 .LP
  40 .nf
  41 \fBzfs\fR \fBcreate\fR [\fB-p\fR] [\fB-o\fR \fIproperty\fR=\fIvalue\fR]... \fIfi
  42 .fi

  44 .LP
  45 .nf
  46 \fBzfs\fR \fBcreate\fR [\fB-ps\fR] [\fB-b\fR \fIblocksize\fR] [\fB-o\fR \fIprope
  47 .fi

  49 .LP
  50 .nf
  51 \fBzfs\fR \fBdestroy\fR [\fB-fnpRrv\fR] \fIfilesystem\fR|\fIvolume\fR
  52 .fi

  54 .LP
  55 .nf
  56 \fBzfs\fR \fBdestroy\fR [\fB-dnpRrv\fR] \fIfilesystem\fR|\fIvolume\fR@\fIsnap\fR
  57 .fi

  59 .LP
  60 .nf
  61 \fBzfs\fR \fBsnapshot\fR [\fB-r\fR] [\fB-o\fR \fIproperty\fR=\fIvalue\fR]...
```



```
  62       \fIfilesystem@snapname\fR|\fIvolume@snapname\fR...
  63 .fi

  65 .LP
  66 .nf
  67 \fBzfs\fR \fBrollback\fR [\fB-rRf\fR] \fIsnapshot\fR
  68 .fi

  70 .LP
  71 .nf
  72 \fBzfs\fR \fBclone\fR [\fB-p\fR] [\fB-o\fR \fIproperty\fR=\fIvalue\fR]... \fIsna
  73 .fi

  75 .LP
  76 .nf
  77 \fBzfs\fR \fBpromote\fR \fIclone-filesystem\fR
  78 .fi

  80 .LP
  81 .nf
  82 \fBzfs\fR \fBrename\fR [\fB-f\fR] \fIfilesystem\fR|\fIvolume\fR|\fIsnapshot\fR
  83      \fIfilesystem\fR|\fIvolume\fR|\fIsnapshot\fR
  84 .fi

  86 .LP
  87 .nf
  88 \fBzfs\fR \fBrename\fR [\fB-fp\fR] \fIfilesystem\fR|\fIvolume\fR \fIfilesystem\f
  89 .fi

  91 .LP
  92 .nf
  93 \fBzfs\fR \fBrename\fR \fB-r\fR \fIsnapshot\fR \fIsnapshot\fR
  94 .fi

  96 .LP
  97 .nf
  98 \fBzfs\fR \fBlist\fR [\fB-r\fR|\fB-d\fR \fIdepth\fR][\fB-H\fR][\fB-o\fR \fIprope
  99      [\fB-s\fR \fIproperty\fR]... [\fB-S\fR \fIproperty\fR]... [\fIfilesystem\fR
 100 .fi

 102 .LP
 103 .nf
 104 \fBzfs\fR \fBset\fR \fIproperty\fR=\fIvalue\fR \fIfilesystem\fR|\fIvolume\fR|\fI
 105 .fi

 107 .LP
 108 .nf
 109 \fBzfs\fR \fBget\fR [\fB-r\fR|\fB-d\fR \fIdepth\fR][\fB-Hp\fR][\fB-o\fR \fIfield
 110     [\fB-s\fR \fIsource\fR[,\fIsource\fR]...] \fBall\fR | \fIproperty\fR[,\fIpro
 111     \fIfilesystem\fR|\fIvolume\fR|\fIsnapshot\fR...
 112 .fi

 114 .LP
 115 .nf
 116 \fBzfs\fR \fBinherit\fR [\fB-r\fR] \fIproperty\fR \fIfilesystem\fR|\fIvolume|sna
 117 .fi

 119 .LP
 120 .nf
 121 \fBzfs\fR \fBupgrade\fR [\fB-v\fR]
 122 .fi

 124 .LP
 125 .nf
 126 \fBzfs\fR \fBupgrade\fR [\fB-r\fR] [\fB-V\fR \fIversion\fR] \fB-a\fR | \fIfilesy
 127 .fi
```

```
.LP
.nf
\fBzfs\fR \fBuserspace\fR [\fB-Hinp\fR] [\fB-o\fR \fIfield\fR[,\fIfield\fR]...]
    [\fB-S\fR \fIfield\fR]... [\fB-t\fR \fItype\fR[,\fItype\fR]...] \fIfilesyste
.fi

.LP
.nf
\fBzfs\fR \fBgroupspace\fR [\fB-Hinp\fR] [\fB-o\fR \fIfield\fR[,\fIfield\fR]...]
    [\fB-S\fR \fIfield\fR]... [\fB-t\fR \fItype\fR[,\fItype\fR]...] \fIfilesyste
.fi

.LP
.nf
\fBzfs\fR \fBmount\fR
.fi

.LP
.nf
\fBzfs\fR \fBmount\fR [\fB-vO\fR] [\fB-o \fIoptions\fR\fR] \fB-a\fR | \fIfilesys
.fi

.LP
.nf
\fBzfs\fR \fBunmount\fR [\fB-f\fR] \fB-a\fR | \fIfilesystem\fR|\fImountpoint\fR
.fi

.LP
.nf
\fBzfs\fR \fBshare\fR \fB-a\fR | \fIfilesystem\fR
.fi

.LP
.nf
\fBzfs\fR \fBunshare\fR \fB-a\fR \fIfilesystem\fR|\fImountpoint\fR
.fi

.LP
.nf
\fBzfs\fR \fBsend\fR [\fB-DnPpRrv\fR] [\fB-\fR[\fBiI\fR] \fIsnapshot\fR] \fIsnap
.fi

.LP
.nf
\fBzfs\fR \fBreceive\fR [\fB-vnFu\fR] \fIfilesystem\fR|\fIvolume\fR|\fIsnapshot\
.fi

.LP
.nf
\fBzfs\fR \fBreceive\fR [\fB-vnFu\fR] [\fB-d\fR|\fB-e\fR] \fIfilesystem\fR
.fi

.LP
.nf
\fBzfs\fR \fBallow\fR \fIfilesystem\fR|\fIvolume\fR
.fi

.LP
.nf
\fBzfs\fR \fBallow\fR [\fB-ldug\fR] \fIuser\fR|\fIgroup\fR[,\fIuser\fR|\fIgroup\
     \fIperm\fR|\fI@setname\fR[,\fIperm\fR|\fI@setname\fR]... \fIfilesystem\fR|\
.fi

.LP
.nf
```



```
\fBzfs\fR \fBallow\fR [\fB-ld\fR] \fB-e\fR|\fBeveryone\fR \fIperm\fR|@\fIsetname
     \fIfilesystem\fR|\fIvolume\fR
.fi

.LP
.nf
\fBzfs\fR \fBallow\fR \fB-c\fR \fIperm\fR|@\fIsetname\fR[,\fIperm\fR|\fI@setname
.fi

.LP
.nf
\fBzfs\fR \fBallow\fR \fB-s\fR @\fIsetname\fR \fIperm\fR|@\fIsetname\fR[,\fIperm
.fi

.LP
.nf
\fBzfs\fR \fBunallow\fR [\fB-rldug\fR] \fIuser\fR|\fIgroup\fR[,\fIuser\fR|\fIgro
     [\fIperm\fR|@\fIsetname\fR[,\fIperm\fR|\fI@setname\fR]...] \fIfilesystem\fR
.fi

.LP
.nf
\fBzfs\fR \fBunallow\fR [\fB-rld\fR] \fB-e\fR|\fBeveryone\fR [\fIperm\fR|@\fIset
     \fIfilesystem\fR|\fIvolume\fR
.fi

.LP
.nf
\fBzfs\fR \fBunallow\fR [\fB-r\fR] \fB-c\fR [\fIperm\fR|@\fIsetname\fR[,\fIperm\
.fi

.LP
.nf
\fBzfs\fR \fBunallow\fR [\fB-r\fR] \fB-s\fR @\fIsetname\fR [\fIperm\fR|@\fIsetna
     \fIfilesystem\fR|\fIvolume\fR
.fi

.LP
.nf
\fBzfs\fR \fBhold\fR [\fB-r\fR] \fItag\fR \fIsnapshot\fR...
.fi

.LP
.nf
\fBzfs\fR \fBholds\fR [\fB-r\fR] \fIsnapshot\fR...
.fi

.LP
.nf
\fBzfs\fR \fBrelease\fR [\fB-r\fR] \fItag\fR \fIsnapshot\fR...
.fi

.LP
.nf
\fBzfs\fR \fBdiff\fR [\fB-FHt\fR] \fIsnapshot\fR \fIsnapshot|filesystem\fR

.SH DESCRIPTION
.sp
.LP
The \fBzfs\fR command configures \fBZFS\fR datasets within a \fBZFS\fR storage
pool, as described in \fBzpool\fR(1M). A dataset is identified by a unique path
within the \fBZFS\fR namespace. For example:
.sp
.in +2
.nf
pool/{filesystem,volume,snapshot}
```

```
.fi
.in -2
.sp

.sp
.LP
where the maximum length of a dataset name is \fBMAXNAMELEN\fR (256 bytes).
.sp
.LP
A dataset can be one of the following:
.sp
.ne 2
.na
\fB\fIfile system\fR\fR
.ad
.sp .6
.RS 4n
A \fBZFS\fR dataset of type \fBfilesystem\fR can be mounted within the standard
system namespace and behaves like other file systems. While \fBZFS\fR file
systems are designed to be \fBPOSIX\fR compliant, known issues exist that
prevent compliance in some cases. Applications that depend on standards
conformance might fail due to nonstandard behavior when checking file system
free space.
.RE

.sp
.ne 2
.na
\fB\fIvolume\fR\fR
.ad
.sp .6
.RS 4n
A logical volume exported as a raw or block device. This type of dataset should
only be used under special circumstances. File systems are typically used in
most environments.
.RE

.sp
.ne 2
.na
\fB\fIsnapshot\fR\fR
.ad
.sp .6
.RS 4n
A read-only version of a file system or volume at a given point in time. It is
specified as \fIfilesystem@name\fR or \fIvolume@name\fR.
.RE

.SS "ZFS File System Hierarchy"
.sp
.LP
A \fBZFS\fR storage pool is a logical collection of devices that provide space
for datasets. A storage pool is also the root of the \fBZFS\fR file system
hierarchy.
.sp
.LP
The root of the pool can be accessed as a file system, such as mounting and
unmounting, taking snapshots, and setting properties. The physical storage
characteristics, however, are managed by the \fBzpool\fR(1M) command.
.sp
.LP
See \fBzpool\fR(1M) for more information on creating and administering pools.
.SS "Snapshots"
.sp
.LP
A snapshot is a read-only copy of a file system or volume. Snapshots can be
```



```
created extremely quickly, and initially consume no additional space within the
pool. As data within the active dataset changes, the snapshot consumes more
data than would otherwise be shared with the active dataset.
.sp
.LP
Snapshots can have arbitrary names. Snapshots of volumes can be cloned or
rolled back, but cannot be accessed independently.
.sp
.LP
File system snapshots can be accessed under the \fB\&.zfs/snapshot\fR directory
in the root of the file system. Snapshots are automatically mounted on demand
and may be unmounted at regular intervals. The visibility of the \fB\&.zfs\fR
directory can be controlled by the \fBsnapdir\fR property.
.SS "Clones"
.sp
.LP
A clone is a writable volume or file system whose initial contents are the same
as another dataset. As with snapshots, creating a clone is nearly
instantaneous, and initially consumes no additional space.
.sp
.LP
Clones can only be created from a snapshot. When a snapshot is cloned, it
creates an implicit dependency between the parent and child. Even though the
clone is created somewhere else in the dataset hierarchy, the original snapshot
cannot be destroyed as long as a clone exists. The \fBorigin\fR property
exposes this dependency, and the \fBdestroy\fR command lists any such
dependencies, if they exist.
.sp
.LP
The clone parent-child dependency relationship can be reversed by using the
\fBpromote\fR subcommand. This causes the "origin" file system to become a
clone of the specified file system, which makes it possible to destroy the file
system that the clone was created from.
.SS "Mount Points"
.sp
.LP
Creating a \fBZFS\fR file system is a simple operation, so the number of file
systems per system is likely to be numerous. To cope with this, \fBZFS\fR
automatically manages mounting and unmounting file systems without the need to
edit the \fB/etc/vfstab\fR file. All automatically managed file systems are
mounted by \fBZFS\fR at boot time.
.sp
.LP
By default, file systems are mounted under \fB/\fIpath\fR\fR, where \fIpath\fR
is the name of the file system in the \fBZFS\fR namespace. Directories are
created and destroyed as needed.
.sp
.LP
A file system can also have a mount point set in the \fBmountpoint\fR property.
This directory is created as needed, and \fBZFS\fR automatically mounts the
file system when the \fBzfs mount -a\fR command is invoked (without editing
\fB/etc/vfstab\fR). The \fBmountpoint\fR property can be inherited, so if
\fBpool/home\fR has a mount point of \fB/export/stuff\fR, then
\fBpool/home/user\fR automatically inherits a mount point of
\fB/export/stuff/user\fR.
.sp
.LP
A file system \fBmountpoint\fR property of \fBnone\fR prevents the file system
from being mounted.
.sp
.LP
If needed, \fBZFS\fR file systems can also be managed with traditional tools
(\fBmount\fR, \fBumount\fR, \fB/etc/vfstab\fR). If a file system's mount point
is set to \fBlegacy\fR, \fBZFS\fR makes no attempt to manage the file system,
and the administrator is responsible for mounting and unmounting the file
system.
```

```
392 .SS "Zones"
393 .sp
394 .LP
395 A \fBZFS\fR file system can be added to a non-global zone by using the
396 \fBzonecfg\fR \fBadd fs\fR subcommand. A \fBZFS\fR file system that is added to
397 a non-global zone must have its \fBmountpoint\fR property set to \fBlegacy\fR.
398 .sp
399 .LP
400 The physical properties of an added file system are controlled by the global
401 administrator. However, the zone administrator can create, modify, or destroy
402 files within the added file system, depending on how the file system is
403 mounted.
404 .sp
405 .LP
406 A dataset can also be delegated to a non-global zone by using the \fBzonecfg\fR
407 \fBadd dataset\fR subcommand. You cannot delegate a dataset to one zone and the
408 children of the same dataset to another zone. The zone administrator can change
409 properties of the dataset or any of its children. However, the \fBquota\fR
410 property is controlled by the global administrator.
411 .sp
412 .LP
413 A \fBZFS\fR volume can be added as a device to a non-global zone by using the
414 \fBzonecfg\fR \fBadd device\fR subcommand. However, its physical properties can
415 be modified only by the global administrator.
416 .sp
417 .LP
418 For more information about \fBzonecfg\fR syntax, see \fBzonecfg\fR(1M).
419 .sp
420 .LP
421 After a dataset is delegated to a non-global zone, the \fBzoned\fR property is
422 automatically set. A zoned file system cannot be mounted in the global zone,
423 since the zone administrator might have to set the mount point to an
424 unacceptable value.
425 .sp
426 .LP
427 The global administrator can forcibly clear the \fBzoned\fR property, though
428 this should be done with extreme care. The global administrator should verify
429 that all the mount points are acceptable before clearing the property.
430 .SS "Native Properties"
431 .sp
432 .LP
433 Properties are divided into two types, native properties and user-defined (or
434 "user") properties. Native properties either export internal statistics or
435 control \fBZFS\fR behavior. In addition, native properties are either editable
436 or read-only. User properties have no effect on \fBZFS\fR behavior, but you can
437 use them to annotate datasets in a way that is meaningful in your environment.
438 For more information about user properties, see the "User Properties" section,
439 below.
440 .sp
441 .LP
442 Every dataset has a set of properties that export statistics about the dataset
443 as well as control various behaviors. Properties are inherited from the parent
444 unless overridden by the child. Some properties apply only to certain types of
445 datasets (file systems, volumes, or snapshots).
446 .sp
447 .LP
448 The values of numeric properties can be specified using human-readable suffixes
449 (for example, \fBk\fR, \fBKB\fR, \fBM\fR, \fBGb\fR, and so forth, up to \fBZ\fR
450 for zettabyte). The following are all valid (and equal) specifications:
451 .sp
452 .in +2
453 .nf
454 1536M, 1.5g, 1.50GB
455 .fi
456 .in -2
457 .sp
```



```
459 .sp
460 .LP
461 The values of non-numeric properties are case sensitive and must be lowercase,
462 except for \fBmountpoint\fR, \fBsharenfs\fR, and \fBsharesmb\fR.
463 .sp
464 .LP
465 The following native properties consist of read-only statistics about the
466 dataset. These properties can be neither set, nor inherited. Native properties
467 apply to all dataset types unless otherwise noted.
468 .sp
469 .ne 2
470 .na
471 \fB\fBavailable\fR\fR
472 .ad
473 .sp .6
474 .RS 4n
475 The amount of space available to the dataset and all its children, assuming
476 that there is no other activity in the pool. Because space is shared within a
477 pool, availability can be limited by any number of factors, including physical
478 pool size, quotas, reservations, or other datasets within the pool.
479 .sp
480 This property can also be referred to by its shortened column name,
481 \fBavail\fR.
482 .RE

484 .sp
485 .ne 2
486 .na
487 \fB\fBcompressratio\fR\fR
488 .ad
489 .sp .6
490 .RS 4n
491 For non-snapshots, the compression ratio achieved for the \fBused\fR
492 space of this dataset, expressed as a multiplier.  The \fBused\fR
493 property includes descendant datasets, and, for clones, does not include
494 the space shared with the origin snapshot.  For snapshots, the
495 \fBcompressratio\fR is the same as the \fBrefcompressratio\fR property.
496 Compression can be turned on by running: \fBzfs set compression=on
497 \fIdataset\fR\fR. The default value is \fBoff\fR.
498 .RE

500 .sp
501 .ne 2
502 .na
503 \fB\fBcreation\fR\fR
504 .ad
505 .sp .6
506 .RS 4n
507 The time this dataset was created.
508 .RE

510 .sp
511 .ne 2
512 .na
513 \fB\fBclones\fR\fR
514 .ad
515 .sp .6
516 .RS 4n
517 For snapshots, this property is a comma-separated list of filesystems or
518 volumes which are clones of this snapshot.  The clones' \fBorigin\fR property
519 is this snapshot.  If the \fBclones\fR property is not empty, then this
520 snapshot can not be destroyed (even with the \fB-r\fR or \fB-f\fR options).
521 .RE

523 .sp
```

```
.ne 2
.na
\fB\fBdefer_destroy\fR\fR
.ad
.sp .6
.RS 4n
This property is \fBon\fR if the snapshot has been marked for deferred destroy
by using the \fBzfs destroy\fR \fB-d\fR command. Otherwise, the property is
\fBoff\fR.
.RE

.sp
.ne 2
.na
\fB\fBlogicalreferenced\fR\fR
.ad
.sp .6
.RS 4n
The amount of space that is "logically" accessible by this dataset.  See
the \fBreferenced\fR property.  The logical space ignores the effect of
the \fBcompression\fR and \fBcopies\fR properties, giving a quantity
closer to the amount of data that applications see.  However, it does
include space consumed by metadata.
.sp
This property can also be referred to by its shortened column name,
\fBlrefer\fR.
.RE

.sp
.ne 2
.na
\fB\fBlogicalused\fR\fR
.ad
.sp .6
.RS 4n
The amount of space that is "logically" consumed by this dataset and all
its descendents.  See the \fBused\fR property.  The logical space
ignores the effect of the \fBcompression\fR and \fBcopies\fR properties,
giving a quantity closer to the amount of data that applications see.
However, it does include space consumed by metadata.
.sp
This property can also be referred to by its shortened column name,
\fBlused\fR.
.RE

.sp
.ne 2
.na
\fB\fBmounted\fR\fR
.ad
.sp .6
.RS 4n
For file systems, indicates whether the file system is currently mounted. This
property can be either \fByes\fR or \fBno\fR.
.RE

.sp
.ne 2
.na
\fB\fBorigin\fR\fR
.ad
.sp .6
.RS 4n
For cloned file systems or volumes, the snapshot from which the clone was
created.  See also the \fBclones\fR property.
.RE
```



```
.sp
.ne 2
.na
\fB\fBreferenced\fR\fR
.ad
.sp .6
.RS 4n
The amount of data that is accessible by this dataset, which may or may not be
shared with other datasets in the pool. When a snapshot or clone is created, it
initially references the same amount of space as the file system or snapshot it
was created from, since its contents are identical.
.sp
This property can also be referred to by its shortened column name,
\fBrefer\fR.
.RE

.sp
.ne 2
.na
\fB\fBrefcompressratio\fR\fR
.ad
.sp .6
.RS 4n
The compression ratio achieved for the \fBreferenced\fR space of this
dataset, expressed as a multiplier.  See also the \fBcompressratio\fR
property.
.RE

.sp
.ne 2
.na
\fB\fBtype\fR\fR
.ad
.sp .6
.RS 4n
The type of dataset: \fBfilesystem\fR, \fBvolume\fR, or \fBsnapshot\fR.
.RE

.sp
.ne 2
.na
\fB\fBused\fR\fR
.ad
.sp .6
.RS 4n
The amount of space consumed by this dataset and all its descendents. This is
the value that is checked against this dataset's quota and reservation. The
space used does not include this dataset's reservation, but does take into
account the reservations of any descendent datasets. The amount of space that a
dataset consumes from its parent, as well as the amount of space that are freed
if this dataset is recursively destroyed, is the greater of its space used and
its reservation.
.sp
When snapshots (see the "Snapshots" section) are created, their space is
initially shared between the snapshot and the file system, and possibly with
previous snapshots. As the file system changes, space that was previously
shared becomes unique to the snapshot, and counted in the snapshot's space
used. Additionally, deleting snapshots can increase the amount of space unique
to (and used by) other snapshots.
.sp
The amount of space used, available, or referenced does not take into account
pending changes. Pending changes are generally accounted for within a few
seconds. Committing a change to a disk using \fBfsync\fR(3c) or \fBO_SYNC\fR
does not necessarily guarantee that the space usage information is updated
immediately.
```

```
656 .RE

658 .sp
659 .ne 2
660 .na
661 \fB\fBusedby*\fR\fR
662 .ad
663 .sp .6
664 .RS 4n
665 The \fBusedby*\fR properties decompose the \fBused\fR properties into the
666 various reasons that space is used. Specifically, \fBused\fR =
667 \fBusedbychildren\fR + \fBusedbydataset\fR + \fBusedbyrefreservation\fR +,
668 \fBusedbysnapshots\fR. These properties are only available for datasets created
669 on \fBzpool\fR "version 13" pools.
670 .RE

672 .sp
673 .ne 2
674 .na
675 \fB\fBusedbychildren\fR\fR
676 .ad
677 .sp .6
678 .RS 4n
679 The amount of space used by children of this dataset, which would be freed if
680 all the dataset's children were destroyed.
681 .RE

683 .sp
684 .ne 2
685 .na
686 \fB\fBusedbydataset\fR\fR
687 .ad
688 .sp .6
689 .RS 4n
690 The amount of space used by this dataset itself, which would be freed if the
691 dataset were destroyed (after first removing any \fBrefreservation\fR and
692 destroying any necessary snapshots or descendents).
693 .RE

695 .sp
696 .ne 2
697 .na
698 \fB\fBusedbyrefreservation\fR\fR
699 .ad
700 .sp .6
701 .RS 4n
702 The amount of space used by a \fBrefreservation\fR set on this dataset, which
703 would be freed if the \fBrefreservation\fR was removed.
704 .RE

706 .sp
707 .ne 2
708 .na
709 \fB\fBusedbysnapshots\fR\fR
710 .ad
711 .sp .6
712 .RS 4n
713 The amount of space consumed by snapshots of this dataset. In particular, it is
714 the amount of space that would be freed if all of this dataset's snapshots were
715 destroyed. Note that this is not simply the sum of the snapshots' \fBused\fR
716 properties because space can be shared by multiple snapshots.
717 .RE

719 .sp
720 .ne 2
721 .na
```



```
722 \fB\fBuserused@\fR\fIuser\fR\fR
723 .ad
724 .sp .6
725 .RS 4n
726 The amount of space consumed by the specified user in this dataset. Space is
727 charged to the owner of each file, as displayed by \fBls\fR \fB-l\fR. The
728 amount of space charged is displayed by \fBdu\fR and \fBls\fR \fB-s\fR. See the
729 \fBzfs userspace\fR subcommand for more information.
730 .sp
731 Unprivileged users can access only their own space usage. The root user, or a
732 user who has been granted the \fBuserused\fR privilege with \fBzfs allow\fR,
733 can access everyone's usage.
734 .sp
735 The \fBuserused@\fR... properties are not displayed by \fBzfs get all\fR. The
736 user's name must be appended after the \fB@\fR symbol, using one of the
737 following forms:
738 .RS +4
739 .TP
740 .ie t \(bu
741 .el o
742 \fIPOSIX name\fR (for example, \fBjoe\fR)
743 .RE
744 .RS +4
745 .TP
746 .ie t \(bu
747 .el o
748 \fIPOSIX numeric ID\fR (for example, \fB789\fR)
749 .RE
750 .RS +4
751 .TP
752 .ie t \(bu
753 .el o
754 \fISID name\fR (for example, \fBjoe.smith@mydomain\fR)
755 .RE
756 .RS +4
757 .TP
758 .ie t \(bu
759 .el o
760 \fISID numeric ID\fR (for example, \fBS-1-123-456-789\fR)
761 .RE
762 .RE

764 .sp
765 .ne 2
766 .na
767 \fB\fBuserrefs\fR\fR
768 .ad
769 .sp .6
770 .RS 4n
771 This property is set to the number of user holds on this snapshot. User holds
772 are set by using the \fBzfs hold\fR command.
773 .RE

775 .sp
776 .ne 2
777 .na
778 \fB\fBgroupused@\fR\fIgroup\fR\fR
779 .ad
780 .sp .6
781 .RS 4n
782 The amount of space consumed by the specified group in this dataset. Space is
783 charged to the group of each file, as displayed by \fBls\fR \fB-l\fR. See the
784 \fBuserused@\fR\fIuser\fR property for more information.
785 .sp
786 Unprivileged users can only access their own groups' space usage. The root
787 user, or a user who has been granted the \fBgroupused\fR privilege with \fBzfs
```

allow\fR, can access all groups' usage.
.RE

.sp
.ne 2
.na
\fB\fBvolblocksize\fR=\fIblocksize\fR\fR
.ad
.sp .6
.RS 4n
For volumes, specifies the block size of the volume. The \fBblocksize\fR cannot be changed once the volume has been written, so it should be set at volume creation time. The default \fBblocksize\fR for volumes is 8 Kbytes. Any power of 2 from 512 bytes to 128 Kbytes is valid.
.sp
This property can also be referred to by its shortened column name, \fBvolblock\fR.
.RE

.sp
.ne 2
.na
\fB\fBwritten\fR\fR
.ad
.sp .6
.RS 4n
The amount of \fBreferenced\fR space written to this dataset since the previous snapshot.
.RE

.sp
.ne 2
.na
\fB\fBwritten@\fR\fIsnapshot\fR\fR
.ad
.sp .6
.RS 4n
The amount of \fBreferenced\fR space written to this dataset since the specified snapshot. This is the space that is referenced by this dataset but was not referenced by the specified snapshot.
.sp
The \fIsnapshot\fR may be specified as a short snapshot name (just the part after the \fB@\fR), in which case it will be interpreted as a snapshot in the same filesystem as this dataset.
The \fIsnapshot\fR be a full snapshot name (\fIfilesystem\fR@\fIsnapshot\fR), which for clones may be a snapshot in the origin's filesystem (or the origin of the origin's filesystem, etc).
.RE

.sp
.LP
The following native properties can be used to change the behavior of a \fBZFS\fR dataset.
.sp
.ne 2
.na
\fB\fBaclinherit\fR=\fBdiscard\fR | \fBnoallow\fR | \fBrestricted\fR | \fBpassthrough\fR | \fBpassthrough-x\fR\fR
.ad
.sp .6
.RS 4n
Controls how \fBACL\fR entries are inherited when files and directories are created. A file system with an \fBaclinherit\fR property of \fBdiscard\fR does not inherit any \fBACL\fR entries. A file system with an \fBaclinherit\fR property value of \fBnoallow\fR only inherits inheritable \fBACL\fR entries that specify "deny" permissions. The property value \fBrestricted\fR (the

default) removes the \fBwrite_acl\fR and \fBwrite_owner\fR permissions when the \fBACL\fR entry is inherited. A file system with an \fBaclinherit\fR property value of \fBpassthrough\fR inherits all inheritable \fBACL\fR entries without any modifications made to the \fBACL\fR entries when they are inherited. A file system with an \fBaclinherit\fR property value of \fBpassthrough-x\fR has the same meaning as \fBpassthrough\fR, except that the \fBowner@\fR, \fBgroup@\fR, and \fBeveryone@\fR \fBACE\fRs inherit the execute permission only if the file creation mode also requests the execute bit.
.sp
When the property value is set to \fBpassthrough\fR, files are created with a mode determined by the inheritable \fBACE\fRs. If no inheritable \fBACE\fRs exist that affect the mode, then the mode is set in accordance to the requested mode from the application.
.RE

.sp
.ne 2
.na
\fB\fBaclmode\fR=\fBdiscard\fR | \fBgroupmask\fR | \fBpassthrough\fR\fR | \fBres
.ad
.sp .6
.RS 4n
Controls how an \fBACL\fR is modified during \fBchmod\fR(2). A file system with an \fBaclmode\fR property of \fBdiscard\fR (the default) deletes all \fBACL\fR entries that do not represent the mode of the file. An \fBaclmode\fR property of \fBgroupmask\fR reduces permissions granted in all \fBALLOW\fR entries found in the \fBACL\fR such that they are no greater than the group permissions specified by \fBchmod\fR(2). A file system with an \fBaclmode\fR property of \fBpassthrough\fR indicates that no changes are made to the \fBACL\fR other than creating or updating the necessary \fBACL\fR entries to represent the new mode of the file or directory. An \fBaclmode\fR property of \fBrestricted\fR will cause the \fBchmod\fR(2) operation to return an error when used on any file or directory which has a non-trivial \fBACL\fR whose entries can not be represented by a mode. \fBchmod\fR(2) is required to change the set user ID, set group ID, or sticky bits on a file or directory, as they do not have equivalent \fBACL\fR entries. In order to use \fBchmod\fR(2) on a file or directory with a non-trivial \fBACL\fR when \fBaclmode\fR is set to \fBrestricted\fR, you must first remove all \fBACL\fR entries which do not represent the current mode.
.RE

.sp
.ne 2
.na
\fB\fBatime\fR=\fBon\fR | \fBoff\fR\fR
.ad
.sp .6
.RS 4n
Controls whether the access time for files is updated when they are read. Turning this property off avoids producing write traffic when reading files and can result in significant performance gains, though it might confuse mailers and other similar utilities. The default value is \fBon\fR.
.RE

.sp
.ne 2
.na
\fB\fBcanmount\fR=\fBon\fR | \fBoff\fR | \fBnoauto\fR\fR
.ad
.sp .6
.RS 4n
If this property is set to \fBoff\fR, the file system cannot be mounted, and is ignored by \fBzfs mount -a\fR. Setting this property to \fBoff\fR is similar to setting the \fBmountpoint\fR property to \fBnone\fR, except that the dataset still has a normal \fBmountpoint\fR property, which can be inherited. Setting this property to \fBoff\fR allows datasets to be used solely as a mechanism to

inherit properties. One example of setting \fBcanmount=\fR\fBoff\fR is to have two datasets with the same \fBmountpoint\fR, so that the children of both datasets appear in the same directory, but might have different inherited characteristics.
.sp
When the \fBnoauto\fR option is set, a dataset can only be mounted and unmounted explicitly. The dataset is not mounted automatically when the dataset is created or imported, nor is it mounted by the \fBzfs mount -a\fR command or unmounted by the \fBzfs unmount -a\fR command.
.sp
This property is not inherited.
.RE

.sp
.ne 2
.na
\fB\fBchecksum\fR=\fBon\fR | \fBoff\fR | \fBfletcher2\fR | \fBfletcher4\fR |
\fBsha256\fR | \fBnoparity\fR | \fBsha512\fR | \fBskein\fR | \fBedonr\fR\fR
(fBsha256(fR | *(fBnoparity*(fR *(fR
.ad
.sp .6
.RS 4n
Controls the checksum used to verify data integrity. The default value is \fBon\fR, which automatically selects an appropriate algorithm (currently, \fBfletcher4\fR, but this may change in future releases). The value \fBoff\fR disables integrity checking on user data. The value \fBnoparity\fR not only disables integrity but also disables maintaining parity for user data. This setting is used internally by a dump device residing on a RAID-Z pool and should not be used by any other dataset. Disabling checksums is \fBNOT\fR a recommended practice.
.sp
The \fBsha512\fR, \fBskein\fR and \fBedonr\fR checksum algorithms require enabling the appropriate features on the pool. Please see \fBzpool-features(5)\fR for more information on these algorithms.
.sp
Changing this property affects only newly-written data.
.RE

.sp
.ne 2
.na
\fB\fBcompression\fR=\fBon\fR | \fBoff\fR | \fBlzjb\fR | \fBgzip\fR |
\fBgzip-\fR\fIN\fR | \fBzle\fR\fR | \fBlz4\fR
.ad
.sp .6
.RS 4n
Controls the compression algorithm used for this dataset. The \fBlzjb\fR compression algorithm is optimized for performance while providing decent data compression. Setting compression to \fBon\fR uses the \fBlzjb\fR compression algorithm. The \fBgzip\fR compression algorithm uses the same compression as the \fBgzip\fR(1) command. You can specify the \fBgzip\fR level by using the value \fBgzip-\fR\fIN\fR where \fIN\fR is an integer from 1 (fastest) to 9 (best compression ratio). Currently, \fBgzip\fR is equivalent to \fBgzip-6\fR (which is also the default for \fBgzip\fR(1)). The \fBzle\fR compression algorithm compresses runs of zeros.
.sp
The \fBlz4\fR compression algorithm is a high-performance replacement for the \fBlzjb\fR algorithm. It features significantly faster compression and decompression, as well as a moderately higher compression ratio than \fBlzjb\fR, but can only be used on pools with the \fBlz4_compress\fR feature set to \fIenabled\fR. See \fBzpool-features\fR(5) for details on ZFS feature flags and the \fBlz4_compress\fR feature.
.sp
This property can also be referred to by its shortened column name \fBcompress\fR. Changing this property affects only newly-written data.

.RE

.sp
.ne 2
.na
\fB\fBcopies\fR=\fB1\fR | \fB2\fR | \fB3\fR\fR
.ad
.sp .6
.RS 4n
Controls the number of copies of data stored for this dataset. These copies are in addition to any redundancy provided by the pool, for example, mirroring or RAID-Z. The copies are stored on different disks, if possible. The space used by multiple copies is charged to the associated file and dataset, changing the \fBused\fR property and counting against quotas and reservations.
.sp
Changing this property only affects newly-written data. Therefore, set this property at file system creation time by using the \fB-o\fR \fBcopies=\fR\fIN\fR option.
.RE

.sp
.ne 2
.na
\fB\fBdevices\fR=\fBon\fR | \fBoff\fR\fR
.ad
.sp .6
.RS 4n
Controls whether device nodes can be opened on this file system. The default value is \fBon\fR.
.RE

.sp
.ne 2
.na
\fB\fBexec\fR=\fBon\fR | \fBoff\fR\fR
.ad
.sp .6
.RS 4n
Controls whether processes can be executed from within this file system. The default value is \fBon\fR.
.RE

.sp
.ne 2
.na
\fB\fBmountpoint\fR=\fIpath\fR | \fBnone\fR | \fBlegacy\fR\fR
.ad
.sp .6
.RS 4n
Controls the mount point used for this file system. See the "Mount Points" section for more information on how this property is used.
.sp
When the \fBmountpoint\fR property is changed for a file system, the file system and any children that inherit the mount point are unmounted. If the new value is \fBlegacy\fR, then they remain unmounted. Otherwise, they are automatically remounted in the new location if the property was previously \fBlegacy\fR or \fBnone\fR, or if they were mounted before the property was changed. In addition, any shared file systems are unshared and shared in the new location.
.RE

.sp
.ne 2
.na
\fB\fBnbmand\fR=\fBon\fR | \fBoff\fR\fR
.ad

```
.sp .6
.RS 4n
Controls whether the file system should be mounted with \fBnbmand\fR (Non
Blocking mandatory locks). This is used for \fBCIFS\fR clients. Changes to this
property only take effect when the file system is umounted and remounted. See
\fBmount\fR(1M) for more information on \fBnbmand\fR mounts.
.RE

.sp
.ne 2
.na
\fB\fBprimarycache\fR=\fBall\fR | \fBnone\fR | \fBmetadata\fR\fR
.ad
.sp .6
.RS 4n
Controls what is cached in the primary cache (ARC). If this property is set to
\fBall\fR, then both user data and metadata is cached. If this property is set
to \fBnone\fR, then neither user data nor metadata is cached. If this property
is set to \fBmetadata\fR, then only metadata is cached. The default value is
\fBall\fR.
.RE

.sp
.ne 2
.na
\fB\fBquota\fR=\fIsize\fR | \fBnone\fR\fR
.ad
.sp .6
.RS 4n
Limits the amount of space a dataset and its descendents can consume. This
property enforces a hard limit on the amount of space used. This includes all
space consumed by descendents, including file systems and snapshots. Setting a
quota on a descendent of a dataset that already has a quota does not override
the ancestor's quota, but rather imposes an additional limit.
.sp
Quotas cannot be set on volumes, as the \fBvolsize\fR property acts as an
implicit quota.
.RE

.sp
.ne 2
.na
\fB\fBuserquota@\fR\fIuser\fR=\fIsize\fR | \fBnone\fR\fR
.ad
.sp .6
.RS 4n
Limits the amount of space consumed by the specified user. User space
consumption is identified by the \fBuserspace@\fR\fIuser\fR property.
.sp
Enforcement of user quotas may be delayed by several seconds. This delay means
that a user might exceed their quota before the system notices that they are
over quota and begins to refuse additional writes with the \fBEDQUOT\fR error
message . See the \fBzfs userspace\fR subcommand for more information.
.sp
Unprivileged users can only access their own groups' space usage. The root
user, or a user who has been granted the \fBuserquota\fR privilege with \fBzfs
allow\fR, can get and set everyone's quota.
.sp
This property is not available on volumes, on file systems before version 4, or
on pools before version 15. The \fBuserquota@\fR... properties are not
displayed by \fBzfs get all\fR. The user's name must be appended after the
\fB@\fR symbol, using one of the following forms:
.RS +4
.TP
.ie t \(bu
.el o
```



```
\fIPOSIX name\fR (for example, \fBjoe\fR)
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fIPOSIX numeric ID\fR (for example, \fB789\fR)
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fISID name\fR (for example, \fBjoe.smith@mydomain\fR)
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fISID numeric ID\fR (for example, \fBS-1-123-456-789\fR)
.RE
.RE

.sp
.ne 2
.na
\fB\fBgroupquota@\fR\fIgroup\fR=\fIsize\fR | \fBnone\fR\fR
.ad
.sp .6
.RS 4n
Limits the amount of space consumed by the specified group. Group space
consumption is identified by the \fBuserquota@\fR\fIuser\fR property.
.sp
Unprivileged users can access only their own groups' space usage. The root
user, or a user who has been granted the \fBgroupquota\fR privilege with \fBzfs
allow\fR, can get and set all groups' quotas.
.RE

.sp
.ne 2
.na
\fB\fBreadonly\fR=\fBon\fR | \fBoff\fR\fR
.ad
.sp .6
.RS 4n
Controls whether this dataset can be modified. The default value is \fBoff\fR.
.sp
This property can also be referred to by its shortened column name,
\fBrdonly\fR.
.RE

.sp
.ne 2
.na
\fB\fBrecordsize\fR=\fIsize\fR\fR
.ad
.sp .6
.RS 4n
Specifies a suggested block size for files in the file system. This property is
designed solely for use with database workloads that access files in fixed-size
records. \fBZFS\fR automatically tunes block sizes according to internal
algorithms optimized for typical access patterns.
.sp
For databases that create very large files but access them in small random
chunks, these algorithms may be suboptimal. Specifying a \fBrecordsize\fR
greater than or equal to the record size of the database can result in
significant performance gains. Use of this property for general purpose file
```

systems is strongly discouraged, and may adversely affect performance.
.sp
The size specified must be a power of two greater than or equal to 512 and less than or equal to 128 Kbytes.
.sp
Changing the file system's \fBrecordsize\fR affects only files created afterward; existing files are unaffected.
.sp
This property can also be referred to by its shortened column name, \fBrecsize\fR.
.RE

.sp
.ne 2
.na
\fB\fBrefquota\fR=\fIsize\fR | \fBnone\fR\fR
.ad
.sp .6
.RS 4n
Limits the amount of space a dataset can consume. This property enforces a hard limit on the amount of space used. This hard limit does not include space used by descendents, including file systems and snapshots.
.RE

.sp
.ne 2
.na
\fB\fBrefreservation\fR=\fIsize\fR | \fBnone\fR\fR
.ad
.sp .6
.RS 4n
The minimum amount of space guaranteed to a dataset, not including its descendents. When the amount of space used is below this value, the dataset is treated as if it were taking up the amount of space specified by \fBrefreservation\fR. The \fBrefreservation\fR reservation is accounted for in the parent datasets' space used, and counts against the parent datasets' quotas and reservations.
.sp
If \fBrefreservation\fR is set, a snapshot is only allowed if there is enough free pool space outside of this reservation to accommodate the current number of "referenced" bytes in the dataset.
.sp
This property can also be referred to by its shortened column name, \fBrefreserv\fR.
.RE

.sp
.ne 2
.na
\fB\fBreservation\fR=\fIsize\fR | \fBnone\fR\fR
.ad
.sp .6
.RS 4n
The minimum amount of space guaranteed to a dataset and its descendents. When the amount of space used is below this value, the dataset is treated as if it were taking up the amount of space specified by its reservation. Reservations are accounted for in the parent datasets' space used, and count against the parent datasets' quotas and reservations.
.sp
This property can also be referred to by its shortened column name, \fBreserv\fR.
.RE

.sp
.ne 2
.na

\fB\fBsecondarycache\fR=\fBall\fR | \fBnone\fR | \fBmetadata\fR\fR
.ad
.sp .6
.RS 4n
Controls what is cached in the secondary cache (L2ARC). If this property is set to \fBall\fR, then both user data and metadata is cached. If this property is set to \fBnone\fR, then neither user data nor metadata is cached. If this property is set to \fBmetadata\fR, then only metadata is cached. The default value is \fBall\fR.
.RE

.sp
.ne 2
.na
\fB\fBsetuid\fR=\fBon\fR | \fBoff\fR\fR
.ad
.sp .6
.RS 4n
Controls whether the set-\fBUID\fR bit is respected for the file system. The default value is \fBon\fR.
.RE

.sp
.ne 2
.na
\fB\fBshareiscsi\fR=\fBon\fR | \fBoff\fR\fR
.ad
.sp .6
.RS 4n
Like the \fBsharenfs\fR property, \fBshareiscsi\fR indicates whether a \fBZFS\fR volume is exported as an \fBiSCSI\fR target. The acceptable values for this property are \fBon\fR, \fBoff\fR, and \fBtype=disk\fR. The default value is \fBoff\fR. In the future, other target types might be supported. For example, \fBtape\fR.
.sp
You might want to set \fBshareiscsi=on\fR for a file system so that all \fBZFS\fR volumes within the file system are shared by default. However, setting this property on a file system has no direct effect.
.RE

.sp
.ne 2
.na
\fB\fBsharesmb\fR=\fBon\fR | \fBoff\fR | \fIopts\fR\fR
.ad
.sp .6
.RS 4n
Controls whether the file system is shared by using the Solaris \fBCIFS\fR service, and what options are to be used. A file system with the \fBsharesmb\fR property set to \fBoff\fR is managed through traditional tools such as \fBsharemgr\fR(1M). Otherwise, the file system is automatically shared and unshared with the \fBzfs share\fR and \fBzfs unshare\fR commands. If the property is set to \fBon\fR, the \fBsharemgr\fR(1M) command is invoked with no options. Otherwise, the \fBsharemgr\fR(1M) command is invoked with options equivalent to the contents of this property.
.sp
Because \fBSMB\fR shares requires a resource name, a unique resource name is constructed from the dataset name. The constructed name is a copy of the dataset name except that the characters in the dataset name, which would be illegal in the resource name, are replaced with underscore (\fB_\fR) characters. A pseudo property "name" is also supported that allows you to replace the data set name with a specified name. The specified name is then used to replace the prefix dataset in the case of inheritance. For example, if the dataset \fBdata/home/john\fR is set to \fBname=john\fR, then \fBdata/home/john\fR has a resource name of \fBjohn\fR. If a child dataset of \fBdata/home/john/backups\fR, it has a resource name of \fBjohn_backups\fR.

```
.sp
When SMB shares are created, the SMB share name appears as an entry in the
\fB\&.zfs/shares\fR directory. You can use the \fBls\fR or \fBchmod\fR command
to display the share-level ACLs on the entries in this directory.
.sp
When the \fBsharesmb\fR property is changed for a dataset, the dataset and any
children inheriting the property are re-shared with the new options, only if
the property was previously set to \fBoff\fR, or if they were shared before the
property was changed. If the new property is set to \fBoff\fR, the file systems
are unshared.
.RE

.sp
.ne 2
.na
\fB\fBsharenfs\fR=\fBon\fR | \fBoff\fR | \fIopts\fR\fR
.ad
.sp .6
.RS 4n
Controls whether the file system is shared via \fBNFS\fR, and what options are
used. A file system with a \fBsharenfs\fR property of \fBoff\fR is managed
through traditional tools such as \fBshare\fR(1M), \fBunshare\fR(1M), and
\fBdfstab\fR(4). Otherwise, the file system is automatically shared and
unshared with the \fBzfs share\fR and \fBzfs unshare\fR commands. If the
property is set to \fBon\fR, the \fBshare\fR(1M) command is invoked with no
options. Otherwise, the \fBshare\fR(1M) command is invoked with options
equivalent to the contents of this property.
.sp
When the \fBsharenfs\fR property is changed for a dataset, the dataset and any
children inheriting the property are re-shared with the new options, only if
the property was previously \fBoff\fR, or if they were shared before the
property was changed. If the new property is \fBoff\fR, the file systems are
unshared.
.RE

.sp
.ne 2
.na
\fB\fBlogbias\fR = \fBlatency\fR | \fBthroughput\fR\fR
.ad
.sp .6
.RS 4n
Provide a hint to ZFS about handling of synchronous requests in this dataset.
If \fBlogbias\fR is set to \fBlatency\fR (the default), ZFS will use pool log
devices (if configured) to handle the requests at low latency. If \fBlogbias\fR
is set to \fBthroughput\fR, ZFS will not use configured pool log devices. ZFS
will instead optimize synchronous operations for global pool throughput and
efficient use of resources.
.RE

.sp
.ne 2
.na
\fB\fBsnapdir\fR=\fBhidden\fR | \fBvisible\fR\fR
.ad
.sp .6
.RS 4n
Controls whether the \fB\&.zfs\fR directory is hidden or visible in the root of
the file system as discussed in the "Snapshots" section. The default value is
\fBhidden\fR.
.RE

.sp
.ne 2
.na
\fB\fBsync\fR=\fBdefault\fR | \fBalways\fR | \fBdisabled\fR\fR
```



```
.ad
.sp .6
.RS 4n
Controls the behavior of synchronous requests (e.g. fsync, O_DSYNC).
\fBdefault\fR is the POSIX specified behavior of ensuring all synchronous
requests are written to stable storage and all devices are flushed to ensure
data is not cached by device controllers (this is the default). \fBalways\fR
causes every file system transaction to be written and flushed before its
system call returns. This has a large performance penalty. \fBdisabled\fR
disables synchronous requests. File system transactions are only committed to
stable storage periodically. This option will give the highest performance.
However, it is very dangerous as ZFS would be ignoring the synchronous
transaction demands of applications such as databases or NFS.  Administrators
should only use this option when the risks are understood.
.RE

.sp
.ne 2
.na
\fB\fBversion\fR=\fB1\fR | \fB2\fR | \fBcurrent\fR\fR
.ad
.sp .6
.RS 4n
The on-disk version of this file system, which is independent of the pool
version. This property can only be set to later supported versions. See the
\fBzfs upgrade\fR command.
.RE

.sp
.ne 2
.na
\fB\fBvolsize\fR=\fIsize\fR\fR
.ad
.sp .6
.RS 4n
For volumes, specifies the logical size of the volume. By default, creating a
volume establishes a reservation of equal size. For storage pools with a
version number of 9 or higher, a \fBrefreservation\fR is set instead. Any
changes to \fBvolsize\fR are reflected in an equivalent change to the
reservation (or \fBrefreservation\fR). The \fBvolsize\fR can only be set to a
multiple of \fBvolblocksize\fR, and cannot be zero.
.sp
The reservation is kept equal to the volume’s logical size to prevent
unexpected behavior for consumers. Without the reservation, the volume could
run out of space, resulting in undefined behavior or data corruption, depending
on how the volume is used. These effects can also occur when the volume size is
changed while it is in use (particularly when shrinking the size). Extreme care
should be used when adjusting the volume size.
.sp
Though not recommended, a "sparse volume" (also known as "thin provisioning")
can be created by specifying the \fB-s\fR option to the \fBzfs create -V\fR
command, or by changing the reservation after the volume has been created. A
"sparse volume" is a volume where the reservation is less then the volume size.
Consequently, writes to a sparse volume can fail with \fBENOSPC\fR when the
pool is low on space. For a sparse volume, changes to \fBvolsize\fR are not
reflected in the reservation.
.RE

.sp
.ne 2
.na
\fB\fBvscan\fR=\fBon\fR | \fBoff\fR\fR
.ad
.sp .6
.RS 4n
Controls whether regular files should be scanned for viruses when a file is
```

opened and closed. In addition to enabling this property, the virus scan service must also be enabled for virus scanning to occur. The default value is \fBoff\fR.
.RE

.sp
.ne 2
.na
\fB\fBxattr\fR=\fBon\fR | \fBoff\fR\fR
.ad
.sp .6
.RS 4n
Controls whether extended attributes are enabled for this file system. The default value is \fBon\fR.
.RE

.sp
.ne 2
.na
\fB\fBzoned\fR=\fBon\fR | \fBoff\fR\fR
.ad
.sp .6
.RS 4n
Controls whether the dataset is managed from a non-global zone. See the "Zones" section for more information. The default value is \fBoff\fR.
.RE

.sp
.LP
The following three properties cannot be changed after the file system is created, and therefore, should be set when the file system is created. If the properties are not set with the \fBzfs create\fR or \fBzpool create\fR commands, these properties are inherited from the parent dataset. If the parent dataset lacks these properties due to having been created prior to these features being supported, the new file system will have the default values for these properties.
.sp
.ne 2
.na
\fB\fBcasesensitivity\fR=\fBsensitive\fR | \fBinsensitive\fR | \fBmixed\fR\fR
.ad
.sp .6
.RS 4n
Indicates whether the file name matching algorithm used by the file system should be case-sensitive, case-insensitive, or allow a combination of both styles of matching. The default value for the \fBcasesensitivity\fR property is \fBsensitive\fR. Traditionally, UNIX and POSIX file systems have case-sensitive file names.
.sp
The \fBmixed\fR value for the \fBcasesensitivity\fR property indicates that the file system can support requests for both case-sensitive and case-insensitive matching behavior. Currently, case-insensitive matching behavior on a file system that supports mixed behavior is limited to the Solaris CIFS server product. For more information about the \fBmixed\fR value behavior, see the \fISolaris ZFS Administration Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fBnormalization\fR = \fBnone\fR | \fBformC\fR | \fBformD\fR | \fBformKC\fR | \fBformKD\fR\fR
.ad
.sp .6
.RS 4n
Indicates whether the file system should perform a \fBunicode\fR normalization

of file names whenever two file names are compared, and which normalization algorithm should be used. File names are always stored unmodified, names are normalized as part of any comparison process. If this property is set to a legal value other than \fBnone\fR, and the \fButf8only\fR property was left unspecified, the \fButf8only\fR property is automatically set to \fBon\fR. The default value of the \fBnormalization\fR property is \fBnone\fR. This property cannot be changed after the file system is created.
.RE

.sp
.ne 2
.na
\fB\fButf8only\fR=\fBon\fR | \fBoff\fR\fR
.ad
.sp .6
.RS 4n
Indicates whether the file system should reject file names that include characters that are not present in the \fBUTF-8\fR character code set. If this property is explicitly set to \fBoff\fR, the normalization property must either not be explicitly set or be set to \fBnone\fR. The default value for the \fButf8only\fR property is \fBoff\fR. This property cannot be changed after the file system is created.
.RE

.sp
.LP
The \fBcasesensitivity\fR, \fBnormalization\fR, and \fButf8only\fR properties are also new permissions that can be assigned to non-privileged users by using the \fBZFS\fR delegated administration feature.
.SS "Temporary Mount Point Properties"
.sp
.LP
When a file system is mounted, either through \fBmount\fR(1M) for legacy mounts or the \fBzfs mount\fR command for normal file systems, its mount options are set according to its properties. The correlation between properties and mount options is as follows:
.sp
.in +2
.nf
    PROPERTY   MOUNT OPTION
     devices   devices/nodevices
     exec   exec/noexec
     readonly   ro/rw
     setuid   setuid/nosetuid
     xattr   xattr/noxattr
.fi
.in -2
.sp

.sp
.LP
In addition, these options can be set on a per-mount basis using the \fB-o\fR option, without affecting the property that is stored on disk. The values specified on the command line override the values stored in the dataset. The \fB-nosuid\fR option is an alias for \fBnodevices,nosetuid\fR. These properties are reported as "temporary" by the \fBzfs get\fR command. If the properties are changed while the dataset is mounted, the new setting overrides any temporary settings.
.SS "User Properties"
.sp
.LP
In addition to the standard native properties, \fBZFS\fR supports arbitrary user properties. User properties have no effect on \fBZFS\fR behavior, but applications or administrators can use them to annotate datasets (file systems, volumes, and snapshots).
.sp

```
.LP
User property names must contain a colon (\fB:\fR) character to distinguish
them from native properties. They may contain lowercase letters, numbers, and
the following punctuation characters: colon (\fB:\fR), dash (\fB-\fR), period
(\fB\&.\fR), and underscore (\fB_\fR). The expected convention is that the
property name is divided into two portions such as
\fImodule\fR\fB:\fR\fIproperty\fR, but this namespace is not enforced by
\fBZFS\fR. User property names can be at most 256 characters, and cannot begin
with a dash (\fB-\fR).
.sp
.LP
When making programmatic use of user properties, it is strongly suggested to
use a reversed \fBDNS\fR domain name for the \fImodule\fR component of property
names to reduce the chance that two independently-developed packages use the
same property name for different purposes. Property names beginning with
\fBcom.sun\fR. are reserved for use by Sun Microsystems.
.sp
.LP
The values of user properties are arbitrary strings, are always inherited, and
are never validated. All of the commands that operate on properties (\fBzfs
list\fR, \fBzfs get\fR, \fBzfs set\fR, and so forth) can be used to manipulate
both native properties and user properties. Use the \fBzfs inherit\fR command
to clear a user property . If the property is not defined in any parent
dataset, it is removed entirely. Property values are limited to 1024
characters.
.SS "ZFS Volumes as Swap or Dump Devices"
.sp
.LP
During an initial installation a swap device and dump device are created on
\fBZFS\fR volumes in the \fBZFS\fR root pool. By default, the swap area size is
based on 1/2 the size of physical memory up to 2 Gbytes. The size of the dump
device depends on the kernel’s requirements at installation time. Separate
\fBZFS\fR volumes must be used for the swap area and dump devices. Do not swap
to a file on a \fBZFS\fR file system. A \fBZFS\fR swap file configuration is
not supported.
.sp
.LP
If you need to change your swap area or dump device after the system is
installed or upgraded, use the \fBswap\fR(1M) and \fBdumpadm\fR(1M) commands.
If you need to change the size of your swap area or dump device, see the
\fISolaris ZFS Administration Guide\fR.
.SH SUBCOMMANDS
.sp
.LP
All subcommands that modify state are logged persistently to the pool in their
original form.
.sp
.ne 2
.na
\fB\fBzfs ?\fR\fR
.ad
.sp .6
.RS 4n
Displays a help message.
.RE

.sp
.ne 2
.na
\fB\fBzfs create\fR [\fB-p\fR] [\fB-o\fR \fIproperty\fR=\fIvalue\fR]...
\fIfilesystem\fR\fR
.ad
.sp .6
.RS 4n
Creates a new \fBZFS\fR file system. The file system is automatically mounted
according to the \fBmountpoint\fR property inherited from the parent.
```



```
.sp
.ne 2
.na
\fB\fB-p\fR\fR
.ad
.sp .6
.RS 4n
Creates all the non-existing parent datasets. Datasets created in this manner
are automatically mounted according to the \fBmountpoint\fR property inherited
from their parent. Any property specified on the command line using the
\fB-o\fR option is ignored. If the target filesystem already exists, the
operation completes successfully.
.RE

.sp
.ne 2
.na
\fB\fB-o\fR \fIproperty\fR=\fIvalue\fR\fR
.ad
.sp .6
.RS 4n
Sets the specified property as if the command \fBzfs set\fR
\fIproperty\fR=\fIvalue\fR was invoked at the same time the dataset was
created. Any editable \fBZFS\fR property can also be set at creation time.
Multiple \fB-o\fR options can be specified. An error results if the same
property is specified in multiple \fB-o\fR options.
.RE

.RE

.sp
.ne 2
.na
\fB\fBzfs create\fR [\fB-ps\fR] [\fB-b\fR \fIblocksize\fR] [\fB-o\fR
\fIproperty\fR=\fIvalue\fR]... \fB-V\fR \fIsize\fR \fIvolume\fR\fR
.ad
.sp .6
.RS 4n
Creates a volume of the given size. The volume is exported as a block device in
\fB/dev/zvol/{dsk,rdsk}/\fR\fIpath\fR, where \fIpath\fR is the name of the
volume in the \fBZFS\fR namespace. The size represents the logical size as
exported by the device. By default, a reservation of equal size is created.
.sp
\fIsize\fR is automatically rounded up to the nearest 128 Kbytes to ensure that
the volume has an integral number of blocks regardless of \fIblocksize\fR.
.sp
.ne 2
.na
\fB\fB-p\fR\fR
.ad
.sp .6
.RS 4n
Creates all the non-existing parent datasets. Datasets created in this manner
are automatically mounted according to the \fBmountpoint\fR property inherited
from their parent. Any property specified on the command line using the
\fB-o\fR option is ignored. If the target filesystem already exists, the
operation completes successfully.
.RE

.sp
.ne 2
.na
\fB\fB-s\fR\fR
.ad
.sp .6
.RS 4n
```

Creates a sparse volume with no reservation. See \fBvolsize\fR in the Native Properties section for more information about sparse volumes.
.RE

.sp
.ne 2
.na
\fB\fB-o\fR \fIproperty\fR=\fIvalue\fR\fR
.ad
.sp .6
.RS 4n
Sets the specified property as if the \fBzfs set\fR \fIproperty\fR=\fIvalue\fR command was invoked at the same time the dataset was created. Any editable \fBZFS\fR property can also be set at creation time. Multiple \fB-o\fR options can be specified. An error results if the same property is specified in multiple \fB-o\fR options.
.RE

.sp
.ne 2
.na
\fB\fB-b\fR \fIblocksize\fR\fR
.ad
.sp .6
.RS 4n
Equivalent to \fB-o\fR \fBvolblocksize\fR=\fIblocksize\fR. If this option is specified in conjunction with \fB-o\fR \fBvolblocksize\fR, the resulting behavior is undefined.
.RE

.RE

.sp
.ne 2
.na
\fBzfs destroy\fR [\fB-fnpRrv\fR] \fIfilesystem\fR|\fIvolume\fR
.ad
.sp .6
.RS 4n
Destroys the given dataset. By default, the command unshares any file systems that are currently shared, unmounts any file systems that are currently mounted, and refuses to destroy a dataset that has active dependents (children or clones).
.sp
.ne 2
.na
\fB\fB-r\fR\fR
.ad
.sp .6
.RS 4n
Recursively destroy all children.
.RE

.sp
.ne 2
.na
\fB\fB-R\fR\fR
.ad
.sp .6
.RS 4n
Recursively destroy all dependents, including cloned file systems outside the target hierarchy.
.RE

.sp
.ne 2

.na
\fB\fB-f\fR\fR
.ad
.sp .6
.RS 4n
Force an unmount of any file systems using the \fBunmount -f\fR command. This option has no effect on non-file systems or unmounted file systems.
.RE

.sp
.ne 2
.na
\fB\fB-n\fR\fR
.ad
.sp .6
.RS 4n
Do a dry-run ("No-op") deletion. No data will be deleted. This is useful in conjunction with the \fB-v\fR or \fB-p\fR flags to determine what data would be deleted.
.RE

.sp
.ne 2
.na
\fB\fB-p\fR\fR
.ad
.sp .6
.RS 4n
Print machine-parsable verbose information about the deleted data.
.RE

.sp
.ne 2
.na
\fB\fB-v\fR\fR
.ad
.sp .6
.RS 4n
Print verbose information about the deleted data.
.RE
.sp
Extreme care should be taken when applying either the \fB-r\fR or the \fB-R\fR options, as they can destroy large portions of a pool and cause unexpected behavior for mounted file systems in use.
.RE

.sp
.ne 2
.na
\fBzfs destroy\fR [\fB-dnpRrv\fR] \fIfilesystem\fR|\fIvolume\fR@\fIsnap\fR[%\fIs
.ad
.sp .6
.RS 4n
The given snapshots are destroyed immediately if and only if the \fBzfs destroy\fR command without the \fB-d\fR option would have destroyed it. Such immediate destruction would occur, for example, if the snapshot had no clones and the user-initiated reference count were zero.
.sp
If a snapshot does not qualify for immediate destruction, it is marked for deferred deletion. In this state, it exists as a usable, visible snapshot until both of the preconditions listed above are met, at which point it is destroyed.
.sp
An inclusive range of snapshots may be specified by separating the first and last snapshots with a percent sign.
The first and/or last snapshots may be left blank, in which case the filesystem’s oldest or newest snapshot will be implied.

```
1843 .sp
1844 Multiple snapshots
1845 (or ranges of snapshots) of the same filesystem or volume may be specified
1846 in a comma-separated list of snapshots.
1847 Only the snapshot's short name (the
1848 part after the \fB@\fR) should be specified when using a range or
1849 comma-separated list to identify multiple snapshots.
1850 .sp
1851 .ne 2
1852 .na
1853 \fB\fB-d\fR\fR
1854 .ad
1855 .sp .6
1856 .RS 4n
1857 Defer snapshot deletion.
1858 .RE

1860 .sp
1861 .ne 2
1862 .na
1863 \fB\fB-r\fR\fR
1864 .ad
1865 .sp .6
1866 .RS 4n
1867 Destroy (or mark for deferred deletion) all snapshots with this name in
1868 descendent file systems.
1869 .RE

1871 .sp
1872 .ne 2
1873 .na
1874 \fB\fB-R\fR\fR
1875 .ad
1876 .sp .6
1877 .RS 4n
1878 Recursively destroy all clones of these snapshots, including the clones,
1879 snapshots, and children.  If this flag is specified, the \fB-d\fR flag will
1880 have no effect.
1881 .RE

1883 .sp
1884 .ne 2
1885 .na
1886 \fB\fB-n\fR\fR
1887 .ad
1888 .sp .6
1889 .RS 4n
1890 Do a dry-run ("No-op") deletion.  No data will be deleted.  This is
1891 useful in conjunction with the \fB-v\fR or \fB-p\fR flags to determine what
1892 data would be deleted.
1893 .RE

1895 .sp
1896 .ne 2
1897 .na
1898 \fB\fB-p\fR\fR
1899 .ad
1900 .sp .6
1901 .RS 4n
1902 Print machine-parsable verbose information about the deleted data.
1903 .RE

1905 .sp
1906 .ne 2
1907 .na
1908 \fB\fB-v\fR\fR
```



```
1909 .ad
1910 .sp .6
1911 .RS 4n
1912 Print verbose information about the deleted data.
1913 .RE

1915 .sp
1916 Extreme care should be taken when applying either the \fB-r\fR or the \fB-R\fR
1917 options, as they can destroy large portions of a pool and cause unexpected
1918 behavior for mounted file systems in use.
1919 .RE

1921 .sp
1922 .ne 2
1923 .na
1924 \fB\fBzfs snapshot\fR [\fB-r\fR] [\fB-o\fR \fIproperty\fR=\fIvalue\fR]...
1925 \fIfilesystem@snapname\fR|\fIvolume@snapname\fR\fR...
1926 .ad
1927 .sp .6
1928 .RS 4n
1929 Creates snapshots with the given names. All previous modifications by
1930 successful system calls to the file system are part of the snapshots.
1931 Snapshots are taken atomically, so that all snapshots correspond to the same
1932 moment in time. See the "Snapshots" section for details.
1933 .sp
1934 .ne 2
1935 .na
1936 \fB\fB-r\fR\fR
1937 .ad
1938 .sp .6
1939 .RS 4n
1940 Recursively create snapshots of all descendent datasets
1941 .RE

1943 .sp
1944 .ne 2
1945 .na
1946 \fB\fB-o\fR \fIproperty\fR=\fIvalue\fR\fR
1947 .ad
1948 .sp .6
1949 .RS 4n
1950 Sets the specified property; see \fBzfs create\fR for details.
1951 .RE

1953 .RE

1955 .sp
1956 .ne 2
1957 .na
1958 \fB\fBzfs rollback\fR [\fB-rRf\fR] \fIsnapshot\fR\fR
1959 .ad
1960 .sp .6
1961 .RS 4n
1962 Roll back the given dataset to a previous snapshot. When a dataset is rolled
1963 back, all data that has changed since the snapshot is discarded, and the
1964 dataset reverts to the state at the time of the snapshot. By default, the
1965 command refuses to roll back to a snapshot other than the most recent one. In
1966 order to do so, all intermediate snapshots must be destroyed by specifying the
1967 \fB-r\fR option.
1968 .sp
1969 The \fB-rR\fR options do not recursively destroy the child snapshots of a
1970 recursive snapshot. Only the top-level recursive snapshot is destroyed by
1971 either of these options. To completely roll back a recursive snapshot, you must
1972 rollback the individual child snapshots.
1973 .sp
1974 .ne 2
```

```
.na
\fB\fB-r\fR\fR
.ad
.sp .6
.RS 4n
Recursively destroy any snapshots more recent than the one specified.
.RE

.sp
.ne 2
.na
\fB\fB-R\fR\fR
.ad
.sp .6
.RS 4n
Recursively destroy any more recent snapshots, as well as any clones of those
snapshots.
.RE

.sp
.ne 2
.na
\fB\fB-f\fR\fR
.ad
.sp .6
.RS 4n
Used with the \fB-R\fR option to force an unmount of any clone file systems
that are to be destroyed.
.RE

.RE

.sp
.ne 2
.na
\fB\fBzfs clone\fR [\fB-p\fR] [\fB-o\fR \fIproperty\fR=\fIvalue\fR]...
\fIsnapshot\fR \fIfilesystem\fR|\fIvolume\fR\fR
.ad
.sp .6
.RS 4n
Creates a clone of the given snapshot. See the "Clones" section for details.
The target dataset can be located anywhere in the \fBZFS\fR hierarchy, and is
created as the same type as the original.
.sp
.ne 2
.na
\fB\fB-p\fR\fR
.ad
.sp .6
.RS 4n
Creates all the non-existing parent datasets. Datasets created in this manner
are automatically mounted according to the \fBmountpoint\fR property inherited
from their parent. If the target filesystem or volume already exists, the
operation completes successfully.
.RE

.sp
.ne 2
.na
\fB\fB-o\fR \fIproperty\fR=\fIvalue\fR\fR
.ad
.sp .6
.RS 4n
Sets the specified property; see \fBzfs create\fR for details.
.RE
```



```
.RE

.sp
.ne 2
.na
\fB\fBzfs promote\fR \fIclone-filesystem\fR\fR
.ad
.sp .6
.RS 4n
Promotes a clone file system to no longer be dependent on its "origin"
snapshot. This makes it possible to destroy the file system that the clone was
created from. The clone parent-child dependency relationship is reversed, so
that the origin file system becomes a clone of the specified file system.
.sp
The snapshot that was cloned, and any snapshots previous to this snapshot, are
now owned by the promoted clone. The space they use moves from the origin file
system to the promoted clone, so enough space must be available to accommodate
these snapshots. No new space is consumed by this operation, but the space
accounting is adjusted. The promoted clone must not have any conflicting
snapshot names of its own. The \fBrename\fR subcommand can be used to rename
any conflicting snapshots.
.RE

.sp
.ne 2
.na
\fB\fBzfs rename\fR [\fB-f\fR] \fIfilesystem\fR|\fIvolume\fR|\fIsnapshot\fR\fR
.ad
.br
.na
\fB\fIfilesystem\fR|\fIvolume\fR|\fIsnapshot\fR\fR
.ad
.br
.na
\fB\fBzfs rename\fR [\fB-fp\fR] \fIfilesystem\fR|\fIvolume\fR
\fIfilesystem\fR|\fIvolume\fR\fR
.ad
.sp .6
.RS 4n
Renames the given dataset. The new target can be located anywhere in the
\fBZFS\fR hierarchy, with the exception of snapshots. Snapshots can only be
renamed within the parent file system or volume. When renaming a snapshot, the
parent file system of the snapshot does not need to be specified as part of the
second argument. Renamed file systems can inherit new mount points, in which
case they are unmounted and remounted at the new mount point.
.sp
.ne 2
.na
\fB\fB-p\fR\fR
.ad
.sp .6
.RS 4n
Creates all the nonexistent parent datasets. Datasets created in this manner
are automatically mounted according to the \fBmountpoint\fR property inherited
from their parent.
.RE

.sp
.ne 2
.na
\fB\fB-f\fR\fR
.ad
.sp .6
.RS 4n
Force unmount any filesystems that need to be unmounted in the process.
.RE
```

```
.RE

.sp
.ne 2
.na
\fB\fBzfs rename\fR \fB-r\fR \fIsnapshot\fR \fIsnapshot\fR\fR
.ad
.sp .6
.RS 4n
Recursively rename the snapshots of all descendent datasets. Snapshots are the
only dataset that can be renamed recursively.
.RE

.sp
.ne 2
.na
\fB\fBzfs\fR \fBlist\fR [\fB-r\fR|\fB-d\fR \fIdepth\fR] [\fB-H\fR] [\fB-o\fR
\fIproperty\fR[,\fIproperty\fR]...] [ \fB-t\fR \fItype\fR[,\fItype\fR]...] [ \fB
\fIproperty\fR ]... [ \fB-S\fR \fIproperty\fR ]...
[\fIfilesystem\fR|\fIvolume\fR|\fIsnapshot\fR]...\fR
.ad
.sp .6
.RS 4n
Lists the property information for the given datasets in tabular form. If
specified, you can list property information by the absolute pathname or the
relative pathname. By default, all file systems and volumes are displayed.
Snapshots are displayed if the \fBlistsnaps\fR property is \fBon\fR (the
default is \fBoff\fR) . The following fields are displayed,
\fBname,used,available,referenced,mountpoint\fR.
.sp
.ne 2
.na
\fB\fB-H\fR\fR
.ad
.sp .6
.RS 4n
Used for scripting mode. Do not print headers and separate fields by a single
tab instead of arbitrary white space.
.RE

.sp
.ne 2
.na
\fB\fB-r\fR\fR
.ad
.sp .6
.RS 4n
Recursively display any children of the dataset on the command line.
.RE

.sp
.ne 2
.na
\fB\fB-d\fR \fIdepth\fR\fR
.ad
.sp .6
.RS 4n
Recursively display any children of the dataset, limiting the recursion to
\fIdepth\fR. A depth of \fB1\fR will display only the dataset and its direct
children.
.RE

.sp
.ne 2
.na
```



```
\fB\fB-o\fR \fIproperty\fR\fR
.ad
.sp .6
.RS 4n
A comma-separated list of properties to display. The property must be:
.RS +4
.TP
.ie t \(bu
.el o
One of the properties described in the "Native Properties" section
.RE
.RS +4
.TP
.ie t \(bu
.el o
A user property
.RE
.RS +4
.TP
.ie t \(bu
.el o
The value \fBname\fR to display the dataset name
.RE
.RS +4
.TP
.ie t \(bu
.el o
The value \fBspace\fR to display space usage properties on file systems and
volumes. This is a shortcut for specifying \fB-o
name,avail,used,usedsnap,usedds,usedrefreserv,usedchild\fR \fB-t
filesystem,volume\fR syntax.
.RE
.RE

.sp
.ne 2
.na
\fB\fB-s\fR \fIproperty\fR\fR
.ad
.sp .6
.RS 4n
A property for sorting the output by column in ascending order based on the
value of the property. The property must be one of the properties described in
the "Properties" section, or the special value \fBname\fR to sort by the
dataset name. Multiple properties can be specified at one time using multiple
\fB-s\fR property options. Multiple \fB-s\fR options are evaluated from left to
right in decreasing order of importance.
.sp
The following is a list of sorting criteria:
.RS +4
.TP
.ie t \(bu
.el o
Numeric types sort in numeric order.
.RE
.RS +4
.TP
.ie t \(bu
.el o
String types sort in alphabetical order.
.RE
.RS +4
.TP
.ie t \(bu
.el o
Types inappropriate for a row sort that row to the literal bottom, regardless
```

```
2239 of the specified ordering.
2240 .RE
2241 .RS +4
2242 .TP
2243 .ie t \(bu
2244 .el o
2245 If no sorting options are specified the existing behavior of \fBzfs list\fR is
2246 preserved.
2247 .RE
2248 .RE

2250 .sp
2251 .ne 2
2252 .na
2253 \fB\fB-S\fR \fIproperty\fR\fR
2254 .ad
2255 .sp .6
2256 .RS 4n
2257 Same as the \fB-s\fR option, but sorts by property in descending order.
2258 .RE

2260 .sp
2261 .ne 2
2262 .na
2263 \fB\fB-t\fR \fItype\fR\fR
2264 .ad
2265 .sp .6
2266 .RS 4n
2267 A comma-separated list of types to display, where \fItype\fR is one of
2268 \fBfilesystem\fR, \fBsnapshot\fR , \fBvolume\fR, or \fBall\fR. For example,
2269 specifying \fB-t snapshot\fR displays only snapshots.
2270 .RE

2272 .RE

2274 .sp
2275 .ne 2
2276 .na
2277 \fB\fBzfs set\fR \fIproperty\fR=\fIvalue\fR
2278 \fIfilesystem\fR|\fIvolume\fR|\fIsnapshot\fR...\fR
2279 .ad
2280 .sp .6
2281 .RS 4n
2282 Sets the property to the given value for each dataset. Only some properties can
2283 be edited. See the "Properties" section for more information on what properties
2284 can be set and acceptable values. Numeric values can be specified as exact
2285 values, or in a human-readable form with a suffix of \fBB\fR, \fBK\fR, \fBM\fR,
2286 \fBG\fR, \fBT\fR, \fBP\fR, \fBE\fR, \fBZ\fR (for bytes, kilobytes, megabytes,
2287 gigabytes, terabytes, petabytes, exabytes, or zettabytes, respectively). User
2288 properties can be set on snapshots. For more information, see the "User
2289 Properties" section.
2290 .RE

2292 .sp
2293 .ne 2
2294 .na
2295 \fB\fBzfs get\fR [\fB-r\fR|\fB-d\fR \fIdepth\fR] [\fB-Hp\fR] [\fB-o\fR
2296 \fIfield\fR[,\fIfield\fR]... [\fB-t\fR \fItype\fR[,\fItype\fR]...] [\fB-s\fR \fI
2297 \fIproperty\fR[,\fIproperty\fR]... \fIfilesystem\fR|\fIvolume\fR|\fIsnapshot\fR.
2298 .ad
2299 .sp .6
2300 .RS 4n
2301 Displays properties for the given datasets. If no datasets are specified, then
2302 the command displays properties for all datasets on the system. For each
2303 property, the following columns are displayed:
2304 .sp
```



```
2305 .in +2
2306 .nf
2307     name      Dataset name
2308      property  Property name
2309      value     Property value
2310      source    Property source. Can either be local, default,
2311                temporary, inherited, or none (-).
2312 .fi
2313 .in -2
2314 .sp

2316 All columns are displayed by default, though this can be controlled by using
2317 the \fB-o\fR option. This command takes a comma-separated list of properties as
2318 described in the "Native Properties" and "User Properties" sections.
2319 .sp
2320 The special value \fBall\fR can be used to display all properties that apply to
2321 the given dataset's type (filesystem, volume, or snapshot).
2322 .sp
2323 .ne 2
2324 .na
2325 \fB\fB-r\fR\fR
2326 .ad
2327 .sp .6
2328 .RS 4n
2329 Recursively display properties for any children.
2330 .RE

2332 .sp
2333 .ne 2
2334 .na
2335 \fB\fB-d\fR \fIdepth\fR\fR
2336 .ad
2337 .sp .6
2338 .RS 4n
2339 Recursively display any children of the dataset, limiting the recursion to
2340 \fIdepth\fR. A depth of \fB1\fR will display only the dataset and its direct
2341 children.
2342 .RE

2344 .sp
2345 .ne 2
2346 .na
2347 \fB\fB-H\fR\fR
2348 .ad
2349 .sp .6
2350 .RS 4n
2351 Display output in a form more easily parsed by scripts. Any headers are
2352 omitted, and fields are explicitly separated by a single tab instead of an
2353 arbitrary amount of space.
2354 .RE

2356 .sp
2357 .ne 2
2358 .na
2359 \fB\fB-o\fR \fIfield\fR\fR
2360 .ad
2361 .sp .6
2362 .RS 4n
2363 A comma-separated list of columns to display. \fBname,property,value,source\fR
2364 is the default value.
2365 .RE

2367 .sp
2368 .ne 2
2369 .na
2370 \fB\fB-s\fR \fIsource\fR\fR
```

```
.ad
.sp .6
.RS 4n
A comma-separated list of sources to display. Those properties coming from a
source other than those in this list are ignored. Each source must be one of
the following: \fBlocal,default,inherited,temporary,none\fR. The default value
is all sources.
.RE

.sp
.ne 2
.na
\fB\fB-p\fR\fR
.ad
.sp .6
.RS 4n
Display numbers in parseable (exact) values.
.RE

.RE

.sp
.ne 2
.na
\fB\fBzfs inherit\fR [\fB-r\fR] \fIproperty\fR
\fIfilesystem\fR|\fIvolume\fR|\fIsnapshot\fR...\fR
.ad
.sp .6
.RS 4n
Clears the specified property, causing it to be inherited from an ancestor. If
no ancestor has the property set, then the default value is used. See the
"Properties" section for a listing of default values, and details on which
properties can be inherited.
.sp
.ne 2
.na
\fB\fB-r\fR\fR
.ad
.sp .6
.RS 4n
Recursively inherit the given property for all children.
.RE

.RE

.sp
.ne 2
.na
\fB\fBzfs upgrade\fR [\fB-v\fR]\fR
.ad
.sp .6
.RS 4n
Displays a list of file systems that are not the most recent version.
.RE

.sp
.ne 2
.na
\fB\fBzfs upgrade\fR [\fB-r\fR] [\fB-V\fR \fIversion\fR] [\fB-a\fR |
\fIfilesystem\fR]\fR
.ad
.sp .6
.RS 4n
Upgrades file systems to a new on-disk version. Once this is done, the file
systems will no longer be accessible on systems running older versions of the
software. \fBzfs send\fR streams generated from new snapshots of these file
```



```
systems cannot be accessed on systems running older versions of the software.
.sp
In general, the file system version is independent of the pool version. See
\fBzpool\fR(1M) for information on the \fBzpool upgrade\fR command.
.sp
In some cases, the file system version and the pool version are interrelated
and the pool version must be upgraded before the file system version can be
upgraded.
.sp
.ne 2
.na
\fB\fB-a\fR\fR
.ad
.sp .6
.RS 4n
Upgrade all file systems on all imported pools.
.RE

.sp
.ne 2
.na
\fB\fIfilesystem\fR\fR
.ad
.sp .6
.RS 4n
Upgrade the specified file system.
.RE

.sp
.ne 2
.na
\fB\fB-r\fR\fR
.ad
.sp .6
.RS 4n
Upgrade the specified file system and all descendent file systems
.RE

.sp
.ne 2
.na
\fB\fB-V\fR \fIversion\fR\fR
.ad
.sp .6
.RS 4n
Upgrade to the specified \fIversion\fR. If the \fB-V\fR flag is not specified,
this command upgrades to the most recent version. This option can only be used
to increase the version number, and only up to the most recent version
supported by this software.
.RE

.RE

.sp
.ne 2
.na
\fBzfs\fR \fBuserspace\fR [\fB-Hinp\fR] [\fB-o\fR \fIfield\fR[,\fIfield\fR]...]
[\fB-s\fR \fIfield\fR]...
[\fB-S\fR \fIfield\fR]...
[\fB-t\fR \fItype\fR[,\fItype\fR]...] \fIfilesystem\fR|\fIsnapshot\fR
.ad
.sp .6
.RS 4n
Displays space consumed by, and quotas on, each user in the specified
filesystem or snapshot. This corresponds to the \fBuserused@\fR\fIuser\fR and
\fBuserquota@\fR\fIuser\fR properties.
```

```
.sp
.ne 2
.na
\fB\fB-n\fR\fR
.ad
.sp .6
.RS 4n
Print numeric ID instead of user/group name.
.RE

.sp
.ne 2
.na
\fB\fB-H\fR\fR
.ad
.sp .6
.RS 4n
Do not print headers, use tab-delimited output.
.RE

.sp
.ne 2
.na
\fB\fB-p\fR\fR
.ad
.sp .6
.RS 4n
Use exact (parsable) numeric output.
.RE

.sp
.ne 2
.na
\fB\fB-o\fR \fIfield\fR[,\fIfield\fR]...\fR
.ad
.sp .6
.RS 4n
Display only the specified fields from the following
set: \fBtype, name, used, quota\fR. The default is to display all fields.
.RE

.sp
.ne 2
.na
\fB\fB-s\fR \fIfield\fR\fR
.ad
.sp .6
.RS 4n
Sort output by this field. The \fIs\fR and \fIS\fR flags may be specified
multiple times to sort first by one field, then by another. The default is
\fB-s type\fR \fB-s name\fR.
.RE

.sp
.ne 2
.na
\fB\fB-S\fR \fIfield\fR\fR
.ad
.sp .6
.RS 4n
Sort by this field in reverse order. See \fB-s\fR.
.RE

.sp
.ne 2
.na
```



```
\fB\fB-t\fR \fItype\fR[,\fItype\fR]...\fR
.ad
.sp .6
.RS 4n
Print only the specified types from the following
set: \fBall, posixuser, smbuser, posixgroup, smbgroup\fR. The default
is \fB-t posixuser,smbuser\fR. The default can be changed to include group
types.
.RE

.sp
.ne 2
.na
\fB\fB-i\fR\fR
.ad
.sp .6
.RS 4n
Translate SID to POSIX ID. The POSIX ID may be ephemeral if no mapping exists.
Normal POSIX interfaces (for example, \fBstat\fR(2), \fBls\fR \fB-l\fR) perform
this translation, so the \fB-i\fR option allows the output from \fBzfs
userspace\fR to be compared directly with those utilities. However, \fB-i\fR
may lead to confusion if some files were created by an SMB user before a
SMB-to-POSIX name mapping was established. In such a case, some files will be ow
by the SMB entity and some by the POSIX entity. However, the \fB-i\fR option
will report that the POSIX entity has the total usage and quota for both.
.RE

.RE

.sp
.ne 2
.na
\fBzfs\fR \fBgroupspace\fR [\fB-Hinp\fR] [\fB-o\fR \fIfield\fR[,\fIfield\fR]...]
[\fB-s\fR \fIfield\fR]...
[\fB-S\fR \fIfield\fR]...
[\fB-t\fR \fItype\fR[,\fItype\fR]...] \fIfilesystem\fR|\fIsnapshot\fR
.ad
.sp .6
.RS 4n
Displays space consumed by, and quotas on, each group in the specified
filesystem or snapshot. This subcommand is identical to \fBzfs userspace\fR,
except that the default types to display are \fB-t posixgroup,smbgroup\fR.
.RE

.sp
.ne 2
.na
\fB\fBzfs mount\fR\fR
.ad
.sp .6
.RS 4n
Displays all \fBZFS\fR file systems currently mounted.
.RE

.sp
.ne 2
.na
\fB\fBzfs mount\fR [\fB-vO\fR] [\fB-o\fR \fIoptions\fR] \fB-a\fR |
\fIfilesystem\fR\fR
.ad
.sp .6
.RS 4n
Mounts \fBZFS\fR file systems. Invoked automatically as part of the boot
process.
.sp
.ne 2
```

```
.na
\fB\fB-o\fR \fIoptions\fR\fR
.ad
.sp .6
.RS 4n
An optional, comma-separated list of mount options to use temporarily for the
duration of the mount. See the "Temporary Mount Point Properties" section for
details.
.RE

.sp
.ne 2
.na
\fB\fB-O\fR\fR
.ad
.sp .6
.RS 4n
Perform an overlay mount. See \fBmount\fR(1M) for more information.
.RE

.sp
.ne 2
.na
\fB\fB-v\fR\fR
.ad
.sp .6
.RS 4n
Report mount progress.
.RE

.sp
.ne 2
.na
\fB\fB-a\fR\fR
.ad
.sp .6
.RS 4n
Mount all available \fBZFS\fR file systems. Invoked automatically as part of
the boot process.
.RE

.sp
.ne 2
.na
\fB\fIfilesystem\fR\fR
.ad
.sp .6
.RS 4n
Mount the specified filesystem.
.RE

.RE

.sp
.ne 2
.na
\fB\fBzfs unmount\fR [\fB-f\fR] \fB-a\fR | \fIfilesystem\fR|\fImountpoint\fR\fR
.ad
.sp .6
.RS 4n
Unmounts currently mounted \fBZFS\fR file systems. Invoked automatically as
part of the shutdown process.
.sp
.ne 2
.na
\fB\fB-f\fR\fR
```



```
.ad
.sp .6
.RS 4n
Forcefully unmount the file system, even if it is currently in use.
.RE

.sp
.ne 2
.na
\fB\fB-a\fR\fR
.ad
.sp .6
.RS 4n
Unmount all available \fBZFS\fR file systems. Invoked automatically as part of
the boot process.
.RE

.sp
.ne 2
.na
\fB\fIfilesystem\fR|\fImountpoint\fR\fR
.ad
.sp .6
.RS 4n
Unmount the specified filesystem. The command can also be given a path to a
\fBZFS\fR file system mount point on the system.
.RE

.RE

.sp
.ne 2
.na
\fB\fBzfs share\fR \fB-a\fR | \fIfilesystem\fR\fR
.ad
.sp .6
.RS 4n
Shares available \fBZFS\fR file systems.
.sp
.ne 2
.na
\fB\fB-a\fR\fR
.ad
.sp .6
.RS 4n
Share all available \fBZFS\fR file systems. Invoked automatically as part of
the boot process.
.RE

.sp
.ne 2
.na
\fB\fIfilesystem\fR\fR
.ad
.sp .6
.RS 4n
Share the specified filesystem according to the \fBsharenfs\fR and
\fBsharesmb\fR properties. File systems are shared when the \fBsharenfs\fR or
\fBsharesmb\fR property is set.
.RE

.RE

.sp
.ne 2
.na
```

\fB\fBzfs unshare\fR \fB-a\fR | \fIfilesystem\fR|\fImountpoint\fR\fR
.ad
.sp .6
.RS 4n
Unshares currently shared \fBZFS\fR file systems. This is invoked automatically as part of the shutdown process.
.sp
.ne 2
.na
\fB\fB-a\fR\fR
.ad
.sp .6
.RS 4n
Unshare all available \fBZFS\fR file systems. Invoked automatically as part of the boot process.
.RE

.sp
.ne 2
.na
\fB\fIfilesystem\fR|\fImountpoint\fR\fR
.ad
.sp .6
.RS 4n
Unshare the specified filesystem. The command can also be given a path to a \fBZFS\fR file system shared on the system.
.RE

.RE

.sp
.ne 2
.na
\fBzfs send\fR [\fB-DnPpRv\fR] [\fB-\fR[\fBiI\fR] \fIsnapshot\fR] \fIsnapshot\fR
.ad
.sp .6
.RS 4n
Creates a stream representation of the second \fIsnapshot\fR, which is written to standard output. The output can be redirected to a file or to a different system (for example, using \fBssh\fR(1). By default, a full stream is generated.
.sp
.ne 2
.na
\fB\fB-i\fR \fIsnapshot\fR\fR
.ad
.sp .6
.RS 4n
Generate an incremental stream from the first \fIsnapshot\fR to the second \fIsnapshot\fR. The incremental source (the first \fIsnapshot\fR) can be specified as the last component of the snapshot name (for example, the part after the \fB@\fR), and it is assumed to be from the same file system as the second \fIsnapshot\fR.
.sp
If the destination is a clone, the source may be the origin snapshot, which must be fully specified (for example, \fBpool/fs@origin\fR, not just \fB@origin\fR).
.RE

.sp
.ne 2
.na
\fB\fB-I\fR \fIsnapshot\fR\fR
.ad
.sp .6
.RS 4n

Generate a stream package that sends all intermediary snapshots from the first snapshot to the second snapshot. For example, \fB-I @a fs@d\fR is similar to \fB-i @a fs@b; -i @b fs@c; -i @c fs@d\fR. The incremental source snapshot may be specified as with the \fB-i\fR option.
.RE

.sp
.ne 2
.na
\fB\fB-R\fR\fR
.ad
.sp .6
.RS 4n
Generate a replication stream package, which will replicate the specified filesystem, and all descendent file systems, up to the named snapshot. When received, all properties, snapshots, descendent file systems, and clones are preserved.
.sp
If the \fB-i\fR or \fB-I\fR flags are used in conjunction with the \fB-R\fR flag, an incremental replication stream is generated. The current values of properties, and current snapshot and file system names are set when the stream is received. If the \fB-F\fR flag is specified when this stream is received, snapshots and file systems that do not exist on the sending side are destroyed.
.RE

.sp
.ne 2
.na
\fB\fB-D\fR\fR
.ad
.sp .6
.RS 4n
Generate a deduplicated stream. Blocks which would have been sent multiple times in the send stream will only be sent once. The receiving system must also support this feature to recieve a deduplicated stream. This flag can be used regardless of the dataset’s \fBdedup\fR property, but performance **will be much better if the filesystem uses a dedup-capable checksum (e.g.**
will be much better if the filesystem uses a dedup-capable checksum (eg.
\fBsha256\fR).
.RE

.sp
.ne 2
.na
\fB\fB-p\fR\fR
.ad
.sp .6
.RS 4n
Include the dataset’s properties in the stream. This flag is implicit when \fB-R\fR is specified. The receiving system must also support this feature.
.RE

.sp
.ne 2
.na
\fB\fB-n\fR\fR
.ad
.sp .6
.RS 4n
Do a dry-run ("No-op") send. Do not generate any actual send data. This is useful in conjunction with the \fB-v\fR or \fB-P\fR flags to determine what data will be sent.
.RE

.sp
.ne 2

```
.na
\fB\fB-P\fR\fR
.ad
.sp .6
.RS 4n
Print machine-parsable verbose information about the stream package generated.
.RE

.sp
.ne 2
.na
\fB\fB-v\fR\fR
.ad
.sp .6
.RS 4n
Print verbose information about the stream package generated.  This information
includes a per-second report of how much data has been sent.
.RE

The format of the stream is committed. You will be able to receive your streams
on future versions of \fBZFS\fR.
.RE

.sp
.ne 2
.na
\fB\fBzfs receive\fR [\fB-vnFu\fR]
\fIfilesystem\fR|\fIvolume\fR|\fIsnapshot\fR\fR
.ad
.br
.na
\fB\fBzfs receive\fR [\fB-vnFu\fR] [\fB-d\fR|\fB-e\fR] \fIfilesystem\fR\fR
.ad
.sp .6
.RS 4n
Creates a snapshot whose contents are as specified in the stream provided on
standard input. If a full stream is received, then a new file system is created
as well. Streams are created using the \fBzfs send\fR subcommand, which by
default creates a full stream. \fBzfs recv\fR can be used as an alias for
\fBzfs receive\fR.
.sp
If an incremental stream is received, then the destination file system must
already exist, and its most recent snapshot must match the incremental stream's
source. For \fBzvols\fR, the destination device link is destroyed and
recreated, which means the \fBzvol\fR cannot be accessed during the
\fBreceive\fR operation.
.sp
When a snapshot replication package stream that is generated by using the
\fBzfs send\fR \fB-R\fR command is  received, any snapshots that do not exist
on the sending location are destroyed by using the \fBzfs destroy\fR \fB-d\fR
command.
.sp
The name of the snapshot (and file system, if a full stream is received) that
this subcommand creates depends on the argument type and the use of the
\fB-d\fR or \fB-e\fR options.
.sp
If the argument is a snapshot name, the specified \fIsnapshot\fR is created. If
the argument is a file system or volume name, a snapshot with the same name as
the sent snapshot is created within the specified \fIfilesystem\fR or
\fIvolume\fR.  If neither of the \fB-d\fR or \fB-e\fR options are specified,
the provided target snapshot name is used exactly as provided.
.sp
The \fB-d\fR and \fB-e\fR options cause the file system name of the target
snapshot to be determined by appending a portion of the sent snapshot's name to
the specified target \fIfilesystem\fR. If the \fB-d\fR option is specified, all
but the first element of the sent snapshot's file system path (usually the
```



```
pool name) is used and any required intermediate file systems within the
specified one are created.  If the \fB-e\fR option is specified, then only the
last element of the sent snapshot's file system name (i.e. the name of the
source file system itself) is used as the target file system name.
.sp
.ne 2
.na
\fB\fB-d\fR\fR
.ad
.sp .6
.RS 4n
Discard the first element of the sent snapshot's file system name, using
the remaining elements to determine the name of the target file system for
the new snapshot as described in the paragraph above.
.RE

.sp
.ne 2
.na
\fB\fB-e\fR\fR
.ad
.sp .6
.RS 4n
Discard all but the last element of the sent snapshot's file system name,
using that element to determine the name of the target file system for
the new snapshot as described in the paragraph above.
.RE

.sp
.ne 2
.na
\fB\fB-u\fR\fR
.ad
.sp .6
.RS 4n
File system that is associated with the received stream is not mounted.
.RE

.sp
.ne 2
.na
\fB\fB-v\fR\fR
.ad
.sp .6
.RS 4n
Print verbose information about the stream and the time required to perform the
receive operation.
.RE

.sp
.ne 2
.na
\fB\fB-n\fR\fR
.ad
.sp .6
.RS 4n
Do not actually receive the stream. This can be useful in conjunction with the
\fB-v\fR option to verify the name the receive operation would use.
.RE

.sp
.ne 2
.na
\fB\fB-F\fR\fR
.ad
.sp .6
```

```
.RS 4n
Force a rollback of the file system to the most recent snapshot before
performing the receive operation. If receiving an incremental replication
stream (for example, one generated by \fBzfs send -R -[iI]\fR), destroy
snapshots and file systems that do not exist on the sending side.
.RE

.RE

.sp
.ne 2
.na
\fB\fBzfs allow\fR \fIfilesystem\fR | \fIvolume\fR\fR
.ad
.sp .6
.RS 4n
Displays permissions that have been delegated on the specified filesystem or
volume. See the other forms of \fBzfs allow\fR for more information.
.RE

.sp
.ne 2
.na
\fB\fBzfs allow\fR [\fB-ldug\fR] \fIuser\fR|\fIgroup\fR[,\fIuser\fR|\fIgroup\fR]
\fIperm\fR|@\fIsetname\fR[,\fIperm\fR|@\fIsetname\fR]... \fIfilesystem\fR|\fIvol
.ad
.br
.na
\fB\fBzfs allow\fR [\fB-ld\fR] \fB-e\fR|\fBeveryone\fR \fIperm\fR|@\fIsetname\fR
\fIfilesystem\fR|\fIvolume\fR\fR
.ad
.sp .6
.RS 4n
Delegates \fBZFS\fR administration permission for the file systems to
non-privileged users.
.sp
.ne 2
.na
[\fB-ug\fR] \fIuser\fR|\fIgroup\fR[,\fIuser\fR|\fIgroup\fR]...
.ad
.sp .6
.RS 4n
Specifies to whom the permissions are delegated. Multiple entities can be
specified as a comma-separated list. If neither of the \fB-ug\fR options are
specified, then the argument is interpreted preferentially as the keyword
\fBeveryone,\fR then as a user name, and lastly as a group name. To specify a us
or group named "everyone", use the \fB-u\fR or \fB-g\fR options. To specify a
group with the same name as a user, use the \fB-g\fR options.
.RE

.sp
.ne 2
.na
\fB-e\fR|\fBeveryone\fR
.ad
.sp .6
.RS 4n
Specifies that the permissions be delegated to everyone.
.RE

.sp
.ne 2
.na
\fIperm\fR|@\fIsetname\fR[,\fIperm\fR|@\fIsetname\fR]...
.ad
.sp .6
```



```
.RS 4n
The permissions to delegate. Multiple permissions
may be specified as a comma-separated list. Permission names are the same as
\fBZFS\fR subcommand and property names. See the property list below. Property
set names, which begin with an at sign (\fB@\fR) , may be specified. See the
\fB-s\fR form below for details.
.RE

.sp
.ne 2
.na
[\fB-ld\fR] \fIfilesystem\fR|\fIvolume\fR
.ad
.sp .6
.RS 4n
Specifies where the permissions are delegated. If neither of the \fB-ld\fR
options are specified, or both are, then the permissions are allowed for the
file system or volume, and all of its descendents. If only the \fB-l\fR option
is used, then is allowed "locally" only for the specified file system. If only
the \fB-d\fR option is used, then is allowed only for the descendent file
systems.
.RE

.RE

.sp
.LP
Permissions are generally the ability to use a \fBZFS\fR subcommand or change a
\fBZFS\fR property. The following permissions are available:
.sp
.in +2
.nf
NAME            TYPE           NOTES
allow           subcommand     Must also have the permission that is being
                               allowed
clone           subcommand     Must also have the 'create' ability and 'mount'
                               ability in the origin file system
create          subcommand     Must also have the 'mount' ability
destroy         subcommand     Must also have the 'mount' ability
diff            subcommand     Allows lookup of paths within a dataset
                               given an object number, and the ability to
                               create snapshots necessary to 'zfs diff'.
mount           subcommand     Allows mount/umount of ZFS datasets
promote         subcommand     Must also have the 'mount'
                               and 'promote' ability in the origin file system
receive         subcommand     Must also have the 'mount' and 'create' ability
rename          subcommand     Must also have the 'mount' and 'create'
                               ability in the new parent
rollback        subcommand     Must also have the 'mount' ability
send            subcommand
share           subcommand     Allows sharing file systems over NFS or SMB
                               protocols
snapshot        subcommand     Must also have the 'mount' ability
groupquota      other          Allows accessing any groupquota@... property
groupused       other          Allows reading any groupused@... property
userprop        other          Allows changing any user property
userquota       other          Allows accessing any userquota@... property
userused        other          Allows reading any userused@... property

aclinherit      property
aclmode         property
atime           property
canmount        property
casesensitivity property
checksum        property
compression     property
```

```
3162 copies           property
3163 devices          property
3164 exec             property
3165 mountpoint       property
3166 nbmand           property
3167 normalization    property
3168 primarycache     property
3169 quota            property
3170 readonly         property
3171 recordsize       property
3172 refquota         property
3173 refreservation   property
3174 reservation      property
3175 secondarycache   property
3176 setuid           property
3177 shareiscsi       property
3178 sharenfs         property
3179 sharesmb         property
3180 snapdir          property
3181 utf8only         property
3182 version          property
3183 volblocksize     property
3184 volsize          property
3185 vscan            property
3186 xattr            property
3187 zoned            property
3188 .fi
3189 .in -2
3190 .sp

3192 .sp
3193 .ne 2
3194 .na
3195 \fB\fBzfs allow\fR \fB-c\fR \fIperm\fR|@\fIsetname\fR[,\fIperm\fR|@\fIsetname\fR
3196 \fIfilesystem\fR|\fIvolume\fR\fR
3197 .ad
3198 .sp .6
3199 .RS 4n
3200 Sets "create time" permissions. These permissions are granted (locally) to the
3201 creator of any newly-created descendent file system.
3202 .RE

3204 .sp
3205 .ne 2
3206 .na
3207 \fB\fBzfs allow\fR \fB-s\fR @\fIsetname\fR \fIperm\fR|@\fIsetname\fR[,\fIperm\fR
3208 \fIfilesystem\fR|\fIvolume\fR\fR
3209 .ad
3210 .sp .6
3211 .RS 4n
3212 Defines or adds permissions to a permission set. The set can be used by other
3213 \fBzfs allow\fR commands for the specified file system and its descendents.
3214 Sets are evaluated dynamically, so changes to a set are immediately reflected.
3215 Permission sets follow the same naming restrictions as ZFS file systems, but
3216 the name must begin with an "at sign" (\fB@\fR), and can be no more than 64
3217 characters long.
3218 .RE

3220 .sp
3221 .ne 2
3222 .na
3223 \fB\fBzfs unallow\fR [\fB-rldug\fR] \fIuser\fR|\fIgroup\fR[,\fIuser\fR|\fIgroup\
3224 [\fIperm\fR|@\fIsetname\fR[,\fIperm\fR|@\fIsetname\fR]...] \fIfilesystem\fR|\fIv
3225 .ad
3226 .br
3227 .na
```



```
3228 \fB\fBzfs unallow\fR [\fB-rld\fR] \fB-e\fR|\fBeveryone\fR [\fIperm\fR|@\fIsetnam
3229 \fIfilesystem\fR|\fIvolume\fR\fR
3230 .ad
3231 .br
3232 .na
3233 \fB\fBzfs unallow\fR [\fB-r\fR] \fB-c\fR [\fIperm\fR|@\fIsetname\fR[,\fIperm\fR|
3234 .ad
3235 .br
3236 .na
3237 \fB\fIfilesystem\fR|\fIvolume\fR\fR
3238 .ad
3239 .sp .6
3240 .RS 4n
3241 Removes permissions that were granted with the \fBzfs allow\fR command. No
3242 permissions are explicitly denied, so other permissions granted are still in
3243 effect. For example, if the permission is granted by an ancestor. If no
3244 permissions are specified, then all permissions for the specified \fIuser\fR,
3245 \fIgroup\fR, or everyone are removed. Specifying \fBeveryone\fR (or using the
3246 \fB-e\fR option) only removes the permissions that were granted to everyone,
3247 not all permissions for every user and group. See the \fBzfs allow\fR command
3248 for a description of the \fB-ldugec\fR options.
3249 .sp
3250 .ne 2
3251 .na
3252 \fB\fB-r\fR\fR
3253 .ad
3254 .sp .6
3255 .RS 4n
3256 Recursively remove the permissions from this file system and all descendents.
3257 .RE

3259 .RE

3261 .sp
3262 .ne 2
3263 .na
3264 \fB\fBzfs unallow\fR [\fB-r\fR] \fB-s\fR @\fIsetname\fR
3265 [\fIperm\fR|@\fIsetname\fR[,\fIperm\fR|@\fIsetname\fR]...]\fR
3266 .ad
3267 .br
3268 .na
3269 \fB\fIfilesystem\fR|\fIvolume\fR\fR
3270 .ad
3271 .sp .6
3272 .RS 4n
3273 Removes permissions from a permission set. If no permissions are specified,
3274 then all permissions are removed, thus removing the set entirely.
3275 .RE

3277 .sp
3278 .ne 2
3279 .na
3280 \fB\fBzfs hold\fR [\fB-r\fR] \fItag\fR \fIsnapshot\fR...\fR
3281 .ad
3282 .sp .6
3283 .RS 4n
3284 Adds a single reference, named with the \fItag\fR argument, to the specified
3285 snapshot or snapshots. Each snapshot has its own tag namespace, and tags must
3286 be unique within that space.
3287 .sp
3288 If a hold exists on a snapshot, attempts to destroy that snapshot by using the
3289 \fBzfs destroy\fR command return \fBEBUSY\fR.
3290 .sp
3291 .ne 2
3292 .na
3293 \fB\fB-r\fR\fR
```

```
.ad
.sp .6
.RS 4n
Specifies that a hold with the given tag is applied recursively to the
snapshots of all descendent file systems.
.RE

.RE

.sp
.ne 2
.na
\fB\fBzfs holds\fR [\fB-r\fR] \fIsnapshot\fR...\fR
.ad
.sp .6
.RS 4n
Lists all existing user references for the given snapshot or snapshots.
.sp
.ne 2
.na
\fB\fB-r\fR\fR
.ad
.sp .6
.RS 4n
Lists the holds that are set on the named descendent snapshots, in addition to
listing the holds on the named snapshot.
.RE

.RE

.sp
.ne 2
.na
\fB\fBzfs release\fR [\fB-r\fR] \fItag\fR \fIsnapshot\fR...\fR
.ad
.sp .6
.RS 4n
Removes a single reference, named with the \fItag\fR argument, from the
specified snapshot or snapshots. The tag must already exist for each snapshot.
.sp
If a hold exists on a snapshot, attempts to destroy that snapshot by using the
\fBzfs destroy\fR command return \fBEBUSY\fR.
.sp
.ne 2
.na
\fB\fB-r\fR\fR
.ad
.sp .6
.RS 4n
Recursively releases a hold with the given tag on the snapshots of all
descendent file systems.
.RE

.sp
.ne 2
.na
\fBzfs diff\fR [\fB-FHt\fR] \fIsnapshot\fR \fIsnapshot|filesystem\fR
.ad
.sp .6
.RS 4n
Display the difference between a snapshot of a given filesystem and another
snapshot of that filesystem from a later time or the current contents of the
filesystem.  The first column is a character indicating the type of change,
the other columns indicate pathname, new pathname (in case of rename), change
in link count, and optionally file type and/or change time.
```



```
The types of change are:
.in +2
.nf
-       The path has been removed
+       The path has been created
M       The path has been modified
R       The path has been renamed
.fi
.in -2
.sp
.ne 2
.na
\fB-F\fR
.ad
.sp .6
.RS 4n
Display an indication of the type of file, in a manner similar to the \fB-F\fR
option of \fBls\fR(1).
.in +2
.nf
B       Block device
C       Character device
/       Directory
>       Door
|       Named pipe
@       Symbolic link
P       Event port
=       Socket
F       Regular file
.fi
.in -2
.RE
.sp
.ne 2
.na
\fB-H\fR
.ad
.sp .6
.RS 4n
Give more parseable tab-separated output, without header lines and without arrow
.RE
.sp
.ne 2
.na
\fB-t\fR
.ad
.sp .6
.RS 4n
Display the path’s inode change time as the first column of output.
.RE

.SH EXAMPLES
.LP
\fBExample 1 \fRCreating a ZFS File System Hierarchy
.sp
.LP
The following commands create a file system named \fBpool/home\fR and a file
system named \fBpool/home/bob\fR. The mount point \fB/export/home\fR is set for
the parent file system, and is automatically inherited by the child file
system.

.sp
.in +2
.nf
# \fBzfs create pool/home\fR
# \fBzfs set mountpoint=/export/home pool/home\fR
```

```
# \fBzfs create pool/home/bob\fR
.fi
.in -2
.sp

.LP
\fBExample 2 \fRCreating a ZFS Snapshot
.sp
.LP
The following command creates a snapshot named \fByesterday\fR. This snapshot
is mounted on demand in the \fB\&.zfs/snapshot\fR directory at the root of the
\fBpool/home/bob\fR file system.

.sp
.in +2
.nf
# \fBzfs snapshot pool/home/bob@yesterday\fR
.fi
.in -2
.sp

.LP
\fBExample 3 \fRCreating and Destroying Multiple Snapshots
.sp
.LP
The following command creates snapshots named \fByesterday\fR of
\fBpool/home\fR and all of its descendent file systems. Each snapshot is
mounted on demand in the \fB\&.zfs/snapshot\fR directory at the root of its
file system. The second command destroys the newly created snapshots.

.sp
.in +2
.nf
# \fBzfs snapshot -r pool/home@yesterday\fR
# \fBzfs destroy -r pool/home@yesterday\fR
.fi
.in -2
.sp

.LP
\fBExample 4 \fRDisabling and Enabling File System Compression
.sp
.LP
The following command disables the \fBcompression\fR property for all file
systems under \fBpool/home\fR. The next command explicitly enables
\fBcompression\fR for \fBpool/home/anne\fR.

.sp
.in +2
.nf
# \fBzfs set compression=off pool/home\fR
# \fBzfs set compression=on pool/home/anne\fR
.fi
.in -2
.sp

.LP
\fBExample 5 \fRListing ZFS Datasets
.sp
.LP
The following command lists all active file systems and volumes in the system.
Snapshots are displayed if the \fBlistsnaps\fR property is \fBon\fR. The
default is \fBoff\fR. See \fBzpool\fR(1M) for more information on pool
properties.

.sp
```



```
.in +2
.nf
# \fBzfs list\fR
    NAME                    USED  AVAIL  REFER  MOUNTPOINT
    pool                    450K   457G    18K  /pool
    pool/home               315K   457G    21K  /export/home
    pool/home/anne           18K   457G    18K  /export/home/anne
    pool/home/bob           276K   457G   276K  /export/home/bob
.fi
.in -2
.sp

.LP
\fBExample 6 \fRSetting a Quota on a ZFS File System
.sp
.LP
The following command sets a quota of 50 Gbytes for \fBpool/home/bob\fR.

.sp
.in +2
.nf
# \fBzfs set quota=50G pool/home/bob\fR
.fi
.in -2
.sp

.LP
\fBExample 7 \fRListing ZFS Properties
.sp
.LP
The following command lists all properties for \fBpool/home/bob\fR.

.sp
.in +2
.nf
# \fBzfs get all pool/home/bob\fR
NAME           PROPERTY              VALUE                  SOURCE
pool/home/bob  type                  filesystem             -
pool/home/bob  creation              Tue Jul 21 15:53 2009  -
pool/home/bob  used                  21K                    -
pool/home/bob  available             20.0G                  -
pool/home/bob  referenced            21K                    -
pool/home/bob  compressratio         1.00x                  -
pool/home/bob  mounted               yes                    -
pool/home/bob  quota                 20G                    local
pool/home/bob  reservation           none                   default
pool/home/bob  recordsize            128K                   default
pool/home/bob  mountpoint            /pool/home/bob         default
pool/home/bob  sharenfs              off                    default
pool/home/bob  checksum              on                     default
pool/home/bob  compression           on                     local
pool/home/bob  atime                 on                     default
pool/home/bob  devices               on                     default
pool/home/bob  exec                  on                     default
pool/home/bob  setuid                on                     default
pool/home/bob  readonly              off                    default
pool/home/bob  zoned                 off                    default
pool/home/bob  snapdir               hidden                 default
pool/home/bob  aclmode               discard                default
pool/home/bob  aclinherit            restricted             default
pool/home/bob  canmount              on                     default
pool/home/bob  shareiscsi            off                    default
pool/home/bob  xattr                 on                     default
pool/home/bob  copies                1                      default
pool/home/bob  version               4                      -
pool/home/bob  utf8only              off                    -
```

```
pool/home/bob  normalization         none                  -
pool/home/bob  casesensitivity       sensitive             -
pool/home/bob  vscan                 off                   default
pool/home/bob  nbmand                off                   default
pool/home/bob  sharesmb              off                   default
pool/home/bob  refquota              none                  default
pool/home/bob  refreservation        none                  default
pool/home/bob  primarycache          all                   default
pool/home/bob  secondarycache        all                   default
pool/home/bob  usedbysnapshots       0                     -
pool/home/bob  usedbydataset         21K                   -
pool/home/bob  usedbychildren        0                     -
pool/home/bob  usedbyrefreservation  0                     -
.fi
.in -2
.sp

.sp
.LP
The following command gets a single property value.

.sp
.in +2
.nf
# \fBzfs get -H -o value compression pool/home/bob\fR
on
.fi
.in -2
.sp

.sp
.LP
The following command lists all properties with local settings for
\fBpool/home/bob\fR.

.sp
.in +2
.nf
# \fBzfs get -r -s local -o name,property,value all pool/home/bob\fR
NAME           PROPERTY              VALUE
pool/home/bob  quota                 20G
pool/home/bob  compression           on
.fi
.in -2
.sp

.LP
\fBExample 8 \fRRolling Back a ZFS File System
.sp
.LP
The following command reverts the contents of \fBpool/home/anne\fR to the
snapshot named \fByesterday\fR, deleting all intermediate snapshots.

.sp
.in +2
.nf
# \fBzfs rollback -r pool/home/anne@yesterday\fR
.fi
.in -2
.sp

.LP
\fBExample 9 \fRCreating a ZFS Clone
.sp
.LP
The following command creates a writable file system whose initial contents are
```



```
the same as \fBpool/home/bob@yesterday\fR.

.sp
.in +2
.nf
# \fBzfs clone pool/home/bob@yesterday pool/clone\fR
.fi
.in -2
.sp

.LP
\fBExample 10 \fRPromoting a ZFS Clone
.sp
.LP
The following commands illustrate how to test out changes to a file system, and
then replace the original file system with the changed one, using clones, clone
promotion, and renaming:

.sp
.in +2
.nf
# \fBzfs create pool/project/production\fR
  populate /pool/project/production with data
# \fBzfs snapshot pool/project/production@today\fR
# \fBzfs clone pool/project/production@today pool/project/beta\fR
make changes to /pool/project/beta and test them
# \fBzfs promote pool/project/beta\fR
# \fBzfs rename pool/project/production pool/project/legacy\fR
# \fBzfs rename pool/project/beta pool/project/production\fR
once the legacy version is no longer needed, it can be destroyed
# \fBzfs destroy pool/project/legacy\fR
.fi
.in -2
.sp

.LP
\fBExample 11 \fRInheriting ZFS Properties
.sp
.LP
The following command causes \fBpool/home/bob\fR and \fBpool/home/anne\fR to
inherit the \fBchecksum\fR property from their parent.

.sp
.in +2
.nf
# \fBzfs inherit checksum pool/home/bob pool/home/anne\fR
.fi
.in -2
.sp

.LP
\fBExample 12 \fRRemotely Replicating ZFS Data
.sp
.LP
The following commands send a full stream and then an incremental stream to a
remote machine, restoring them into \fBpoolB/received/fs@a\fRand
\fBpoolB/received/fs@b\fR, respectively. \fBpoolB\fR must contain the file
system \fBpoolB/received\fR, and must not initially contain
\fBpoolB/received/fs\fR.

.sp
.in +2
.nf
# \fBzfs send pool/fs@a | \e\fR
   \fBssh host zfs receive poolB/received/fs@a\fR
# \fBzfs send -i a pool/fs@b | ssh host \e\fR
```

```
3690    \fBzfs receive poolB/received/fs\fR
3691 .fi
3692 .in -2
3693 .sp

3695 .LP
3696 \fBExample 13 \fRUsing the \fBzfs receive\fR \fB-d\fR Option
3697 .sp
3698 .LP
3699 The following command sends a full stream of \fBpoolA/fsA/fsB@snap\fR to a
3700 remote machine, receiving it into \fBpoolB/received/fsA/fsB@snap\fR. The
3701 \fBfsA/fsB@snap\fR portion of the received snapshot's name is determined from
3702 the name of the sent snapshot. \fBpoolB\fR must contain the file system
3703 \fBpoolB/received\fR. If \fBpoolB/received/fsA\fR does not exist, it is created
3704 as an empty file system.

3706 .sp
3707 .in +2
3708 .nf
3709 # \fBzfs send poolA/fsA/fsB@snap | \e
3710    ssh host zfs receive -d poolB/received\fR
3711 .fi
3712 .in -2
3713 .sp

3715 .LP
3716 \fBExample 14 \fRSetting User Properties
3717 .sp
3718 .LP
3719 The following example sets the user-defined \fBcom.example:department\fR
3720 property for a dataset.

3722 .sp
3723 .in +2
3724 .nf
3725 # \fBzfs set com.example:department=12345 tank/accounting\fR
3726 .fi
3727 .in -2
3728 .sp

3730 .LP
3731 \fBExample 15 \fRCreating a ZFS Volume as an iSCSI Target Device
3732 .sp
3733 .LP
3734 The following example shows how to create a \fBZFS\fR volume as an \fBiSCSI\fR
3735 target.

3737 .sp
3738 .in +2
3739 .nf
3740 # \fBzfs create -V 2g pool/volumes/vol1\fR
3741 # \fBzfs set shareiscsi=on pool/volumes/vol1\fR
3742 # \fBiscsitadm list target\fR
3743 Target: pool/volumes/vol1
3744  iSCSI Name:
3745  iqn.1986-03.com.sun:02:7b4b02a6-3277-eb1b-e686-a24762c52a8c
3746  Connections: 0
3747 .fi
3748 .in -2
3749 .sp

3751 .sp
3752 .LP
3753 After the \fBiSCSI\fR target is created, set up the \fBiSCSI\fR initiator. For
3754 more information about the Solaris \fBiSCSI\fR initiator, see
3755 \fBiscsitadm\fR(1M).
```



```
3756 .LP
3757 \fBExample 16 \fRPerforming a Rolling Snapshot
3758 .sp
3759 .LP
3760 The following example shows how to maintain a history of snapshots with a
3761 consistent naming scheme. To keep a week's worth of snapshots, the user
3762 destroys the oldest snapshot, renames the remaining snapshots, and then creates
3763 a new snapshot, as follows:

3765 .sp
3766 .in +2
3767 .nf
3768 # \fBzfs destroy -r pool/users@7daysago\fR
3769 # \fBzfs rename -r pool/users@6daysago @7daysago\fR
3770 # \fBzfs rename -r pool/users@5daysago @6daysago\fR
3771 # \fBzfs rename -r pool/users@yesterday @5daysago\fR
3772 # \fBzfs rename -r pool/users@yesterday @4daysago\fR
3773 # \fBzfs rename -r pool/users@yesterday @3daysago\fR
3774 # \fBzfs rename -r pool/users@yesterday @2daysago\fR
3775 # \fBzfs rename -r pool/users@today @yesterday\fR
3776 # \fBzfs snapshot -r pool/users@today\fR
3777 .fi
3778 .in -2
3779 .sp

3781 .LP
3782 \fBExample 17 \fRSetting \fBsharenfs\fR Property Options on a ZFS File System
3783 .sp
3784 .LP
3785 The following commands show how to set \fBsharenfs\fR property options to
3786 enable \fBrw\fR access for a set of \fBIP\fR addresses and to enable root
3787 access for system \fBneo\fR on the \fBtank/home\fR file system.

3789 .sp
3790 .in +2
3791 .nf
3792 # \fBzfs set sharenfs='rw=@123.123.0.0/16,root=neo' tank/home\fR
3793 .fi
3794 .in -2
3795 .sp

3797 .sp
3798 .LP
3799 If you are using \fBDNS\fR for host name resolution, specify the fully
3800 qualified hostname.

3802 .LP
3803 \fBExample 18 \fRDelegating ZFS Administration Permissions on a ZFS Dataset
3804 .sp
3805 .LP
3806 The following example shows how to set permissions so that user \fBcindys\fR
3807 can create, destroy, mount, and take snapshots on \fBtank/cindys\fR. The
3808 permissions on \fBtank/cindys\fR are also displayed.

3810 .sp
3811 .in +2
3812 .nf
3813 # \fBzfs allow cindys create,destroy,mount,snapshot tank/cindys\fR
3814 # \fBzfs allow tank/cindys\fR
3815 -------------------------------------------------------------
3816 Local+Descendent permissions on (tank/cindys)
3817          user cindys create,destroy,mount,snapshot
3818 -------------------------------------------------------------
3819 .fi
3820 .in -2
3821 .sp
```

```
.sp
.LP
Because the \fBtank/cindys\fR mount point permission is set to 755 by default,
user \fBcindys\fR will be unable to mount file systems under \fBtank/cindys\fR.
Set an \fBACL\fR similar to the following syntax to provide mount point access:
.sp
.in +2
.nf
# \fBchmod A+user:cindys:add_subdirectory:allow /tank/cindys\fR
.fi
.in -2
.sp

.LP
\fBExample 19 \fRDelegating Create Time Permissions on a ZFS Dataset
.sp
.LP
The following example shows how to grant anyone in the group \fBstaff\fR to
create file systems in \fBtank/users\fR. This syntax also allows staff members
to destroy their own file systems, but not destroy anyone else's file system.
The permissions on \fBtank/users\fR are also displayed.

.sp
.in +2
.nf
# \fBzfs allow staff create,mount tank/users\fR
# \fBzfs allow -c destroy tank/users\fR
# \fBzfs allow tank/users\fR
-------------------------------------------------------------
Create time permissions on (tank/users)
          create,destroy
Local+Descendent permissions on (tank/users)
          group staff create,mount
-------------------------------------------------------------
.fi
.in -2
.sp

.LP
\fBExample 20 \fRDefining and Granting a Permission Set on a ZFS Dataset
.sp
.LP
The following example shows how to define and grant a permission set on the
\fBtank/users\fR file system. The permissions on \fBtank/users\fR are also
displayed.

.sp
.in +2
.nf
# \fBzfs allow -s @pset create,destroy,snapshot,mount tank/users\fR
# \fBzfs allow staff @pset tank/users\fR
# \fBzfs allow tank/users\fR
-------------------------------------------------------------
Permission sets on (tank/users)
        @pset create,destroy,mount,snapshot
Create time permissions on (tank/users)
        create,destroy
Local+Descendent permissions on (tank/users)
        group staff @pset,create,mount
-------------------------------------------------------------
.fi
.in -2
.sp

.LP
```



```
\fBExample 21 \fRDelegating Property Permissions on a ZFS Dataset
.sp
.LP
The following example shows to grant the ability to set quotas and reservations
on the \fBusers/home\fR file system. The permissions on \fBusers/home\fR are
also displayed.

.sp
.in +2
.nf
# \fBzfs allow cindys quota,reservation users/home\fR
# \fBzfs allow users/home\fR
-------------------------------------------------------------
Local+Descendent permissions on (users/home)
        user cindys quota,reservation
-------------------------------------------------------------
cindys% \fBzfs set quota=10G users/home/marks\fR
cindys% \fBzfs get quota users/home/marks\fR
NAME              PROPERTY  VALUE             SOURCE
users/home/marks  quota     10G               local
.fi
.in -2
.sp

.LP
\fBExample 22 \fRRemoving ZFS Delegated Permissions on a ZFS Dataset
.sp
.LP
The following example shows how to remove the snapshot permission from the
\fBstaff\fR group on the \fBtank/users\fR file system. The permissions on
\fBtank/users\fR are also displayed.

.sp
.in +2
.nf
# \fBzfs unallow staff snapshot tank/users\fR
# \fBzfs allow tank/users\fR
-------------------------------------------------------------
Permission sets on (tank/users)
        @pset create,destroy,mount,snapshot
Create time permissions on (tank/users)
        create,destroy
Local+Descendent permissions on (tank/users)
        group staff @pset,create,mount
-------------------------------------------------------------
.fi
.in -2
.sp

.LP
\fBExample 23\fR Showing the differences between a snapshot and a ZFS Dataset
.sp
.LP
The following example shows how to see what has changed between a prior
snapshot of a ZFS Dataset and its current state.  The \fB-F\fR option is used
to indicate type information for the files affected.

.sp
.in +2
.nf
# zfs diff -F tank/test@before tank/test
M       /       /tank/test/
M       F       /tank/test/linked      (+1)
R       F       /tank/test/oldname -> /tank/test/newname
-       F       /tank/test/deleted
+       F       /tank/test/created
```

```
3954 M       F       /tank/test/modified
3955 .fi
3956 .in -2
3957 .sp

3959 .SH EXIT STATUS
3960 .sp
3961 .LP
3962 The following exit values are returned:
3963 .sp
3964 .ne 2
3965 .na
3966 \fB\fB0\fR\fR
3967 .ad
3968 .sp .6
3969 .RS 4n
3970 Successful completion.
3971 .RE

3973 .sp
3974 .ne 2
3975 .na
3976 \fB\fB1\fR\fR
3977 .ad
3978 .sp .6
3979 .RS 4n
3980 An error occurred.
3981 .RE

3983 .sp
3984 .ne 2
3985 .na
3986 \fB\fB2\fR\fR
3987 .ad
3988 .sp .6
3989 .RS 4n
3990 Invalid command line options were specified.
3991 .RE

3993 .SH ATTRIBUTES
3994 .sp
3995 .LP
3996 See \fBattributes\fR(5) for descriptions of the following attributes:
3997 .sp

3999 .sp
4000 .TS
4001 box;
4002 c | c
4003 l | l .
4004 ATTRIBUTE TYPE  ATTRIBUTE VALUE
4005 _
4006 Interface Stability     Committed
4007 .TE

4009 .SH SEE ALSO
4010 .sp
4011 .LP
4012 \fBssh\fR(1), \fBiscsitadm\fR(1M), \fBmount\fR(1M), \fBshare\fR(1M),
4013 \fBsharemgr\fR(1M), \fBunshare\fR(1M), \fBzonecfg\fR(1M), \fBzpool\fR(1M),
4014 \fBchmod\fR(2), \fBstat\fR(2), \fBwrite\fR(2), \fBfsync\fR(3C),
4015 \fBdfstab\fR(4), \fBacl\fR(5), \fBattributes\fR(5)
4016 .sp
4017 .LP
4018 See the \fBgzip\fR(1) man page, which is not part of the SunOS man page
4019 collection.
```



```
4020 .sp
4021 .LP
4022 For information about using the \fBZFS\fR web-based management tool and other
4023 \fBZFS\fR features, see the \fISolaris ZFS Administration Guide\fR.
```

```
**********************************************************
   13237 Mon Oct  7 19:17:17 2013
new/usr/src/man/man5/zpool-features.5
4185 New hash algorithm support
**********************************************************
'\" te
.\" Copyright (c) 2013 by Delphix. All rights reserved.
.\" Copyright (c) 2013 by Saso Kiselkov. All rights reserved.
.\" Copyright (c) 2013, Joyent, Inc. All rights reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" and Distribution License (the "License").  You may not use this file except
.\" in compliance with the License. You can obtain a copy of the license at
.\" usr/src/OPENSOLARIS.LICENSE or http://www.opensolaris.org/os/licensing.
.\"
.\" See the License for the specific language governing permissions and
.\" limitations under the License. When distributing Covered Code, include this
.\" CDDL HEADER in each file and include the License file at
.\" usr/src/OPENSOLARIS.LICENSE.  If applicable, add the following below this
.\" CDDL HEADER, with the fields enclosed by brackets "[]" replaced with your
.\" own identifying information:
.\" Portions Copyright [yyyy] [name of copyright owner]
.TH ZPOOL-FEATURES 5 "Aug 27, 2013"
.SH NAME
zpool\-features \- ZFS pool feature descriptions
.SH DESCRIPTION
.sp
.LP
ZFS pool on\-disk format versions are specified via "features" which replace
the old on\-disk format numbers (the last supported on\-disk format number is
28). To enable a feature on a pool use the \fBupgrade\fR subcommand of the
\fBzpool\fR(1M) command, or set the \fBfeature@\fR\fIfeature_name\fR property
to \fBenabled\fR.
.sp
.LP
The pool format does not affect file system version compatibility or the ability
to send file systems between pools.
.sp
.LP
Since most features can be enabled independently of each other the on\-disk
format of the pool is specified by the set of all features marked as
\fBactive\fR on the pool. If the pool was created by another software version
this set may include unsupported features.
.SS "Identifying features"
.sp
.LP
Every feature has a guid of the form \fIcom.example:feature_name\fR. The reverse
DNS name ensures that the feature's guid is unique across all ZFS
implementations. When unsupported features are encountered on a pool they will
be identified by their guids. Refer to the documentation for the ZFS
implementation that created the pool for information about those features.
.sp
.LP
Each supported feature also has a short name. By convention a feature's short
name is the portion of its guid which follows the ':' (e.g.
\fIcom.example:feature_name\fR would have the short name \fIfeature_name\fR),
however a feature's short name may differ across ZFS implementations if
following the convention would result in name conflicts.
.SS "Feature states"
.sp
.LP
Features can be in one of three states:
.sp
.ne 2
.na
\fB\fBactive\fR\fR
.ad
```



```
.RS 12n
This feature's on\-disk format changes are in effect on the pool. Support for
this feature is required to import the pool in read\-write mode. If this
feature is not read-only compatible, support is also required to import the pool
in read\-only mode (see "Read\-only compatibility").
.RE

.sp
.ne 2
.na
\fB\fBenabled\fR\fR
.ad
.RS 12n
An administrator has marked this feature as enabled on the pool, but the
feature's on\-disk format changes have not been made yet. The pool can still be
imported by software that does not support this feature, but changes may be made
to the on\-disk format at any time which will move the feature to the
\fBactive\fR state. Some features may support returning to the \fBenabled\fR
state after becoming \fBactive\fR. See feature\-specific documentation for
details.
.RE

.sp
.ne 2
.na
\fBdisabled\fR
.ad
.RS 12n
This feature's on\-disk format changes have not been made and will not be made
unless an administrator moves the feature to the \fBenabled\fR state. Features
cannot be disabled once they have been enabled.
.RE

.sp
.LP
The state of supported features is exposed through pool properties of the form
\fIfeature@short_name\fR.
.SS "Read\-only compatibility"
.sp
.LP
Some features may make on\-disk format changes that do not interfere with other
software's ability to read from the pool. These features are referred to as
"read\-only compatible". If all unsupported features on a pool are read\-only
compatible, the pool can be imported in read\-only mode by setting the
\fBreadonly\fR property during import (see \fBzpool\fR(1M) for details on
importing pools).
.SS "Unsupported features"
.sp
.LP
For each unsupported feature enabled on an imported pool a pool property
named \fIunsupported@feature_guid\fR will indicate why the import was allowed
despite the unsupported feature. Possible values for this property are:

.sp
.ne 2
.na
\fB\fBinactive\fR\fR
.ad
.RS 12n
The feature is in the \fBenabled\fR state and therefore the pool's on\-disk
format is still compatible with software that does not support this feature.
.RE

.sp
.ne 2
.na
```

```
\fB\fBreadonly\fR\fR
.ad
.RS 12n
The feature is read\-only compatible and the pool has been imported in
read\-only mode.
.RE

.SS "Feature dependencies"
.sp
.LP
Some features depend on other features being enabled in order to function
properly. Enabling a feature will automatically enable any features it
depends on.
.SH FEATURES
.sp
.LP
The following features are supported on this system:
.sp
.ne 2
.na
\fB\fBasync_destroy\fR\fR
.ad
.RS 4n
.TS
l l .
GUID    com.delphix:async_destroy
READ\-ONLY COMPATIBLE   yes
DEPENDENCIES    none
.TE

Destroying a file system requires traversing all of its data in order to
return its used space to the pool. Without \fBasync_destroy\fR the file system
is not fully removed until all space has been reclaimed. If the destroy
operation is interrupted by a reboot or power outage the next attempt to open
the pool will need to complete the destroy operation synchronously.

When \fBasync_destroy\fR is enabled the file system's data will be reclaimed
by a background process, allowing the destroy operation to complete without
traversing the entire file system. The background process is able to resume
interrupted destroys after the pool has been opened, eliminating the need
to finish interrupted destroys as part of the open operation. The amount
of space remaining to be reclaimed by the background process is available
through the \fBfreeing\fR property.

This feature is only \fBactive\fR while \fBfreeing\fR is non\-zero.
.RE

.sp
.ne 2
.na
\fB\fBempty_bpobj\fR\fR
.ad
.RS 4n
.TS
l l .
GUID    com.delphix:empty_bpobj
READ\-ONLY COMPATIBLE   yes
DEPENDENCIES    none
.TE

This feature increases the performance of creating and using a large
number of snapshots of a single filesystem or volume, and also reduces
the disk space required.

When there are many snapshots, each snapshot uses many Block Pointer
Objects (bpobj's) to track blocks associated with that snapshot.
```



```
However, in common use cases, most of these bpobj's are empty.  This
feature allows us to create each bpobj on-demand, thus eliminating the
empty bpobjs.

This feature is \fBactive\fR while there are any filesystems, volumes,
or snapshots which were created after enabling this feature.
.RE

.sp
.ne 2
.na
\fB\fBlz4_compress\fR\fR
.ad
.RS 4n
.TS
l l .
GUID    org.illumos:lz4_compress
READ\-ONLY COMPATIBLE   no
DEPENDENCIES    none
.TE

\fBlz4\fR is a high-performance real-time compression algorithm that
features significantly faster compression and decompression as well as a
higher compression ratio than the older \fBlzjb\fR compression.
Typically, \fBlz4\fR compression is approximately 50% faster on
compressible data and 200% faster on incompressible data than
\fBlzjb\fR. It is also approximately 80% faster on decompression, while
giving approximately 10% better compression ratio.

When the \fBlz4_compress\fR feature is set to \fBenabled\fR, the
administrator can turn on \fBlz4\fR compression on any dataset on the
pool using the \fBzfs\fR(1M) command. Please note that doing so will
immediately activate the \fBlz4_compress\fR feature on the underlying
pool (even before any data is written). Since this feature is not
read-only compatible, this operation will render the pool unimportable
on systems without support for the \fBlz4_compress\fR feature. At the
moment, this operation cannot be reversed. Booting off of
\fBlz4\fR-compressed root pools is supported.
.RE

.sp
.ne 2
.na
\fB\fBspacemap_histogram\fR\fR
.ad
.RS 4n
.TS
l l .
GUID    com.delphix:spacemap_histogram
READ\-ONLY COMPATIBLE   yes
DEPENDENCIES    none
.TE

This features allows ZFS to maintain more information about how free space
is organized within the pool. If this feature is \fBenabled\fR, ZFS will
set this feature to \fBactive\fR when a new space map object is created or
an existing space map is upgraded to the new format. Once the feature is
\fBactive\fR, it will remain in that state until the pool is destroyed.
.RE

.sp
.ne 2
.na
\fB\fBmulti_vdev_crash_dump\fR\fR
.ad
.RS 4n
```

```
.TS
l l .
GUID    com.joyent:multi_vdev_crash_dump
READ\-ONLY COMPATIBLE   no
DEPENDENCIES    none
.TE

This feature allows a dump device to be configured with a pool comprised
of multiple vdevs.  Those vdevs may be arranged in any mirrored or raidz
configuration.

When the \fBmulti_vdev_crash_dump\fR feature is set to \fBenabled\fR,
the administrator can use the \fBdumpadm\fR(1M) command to configure a
dump device on a pool comprised of multiple vdevs.

.RE

.sp
.ne 2
.na
\fB\fBsha512\fR\fR
.ad
.RS 4n
.TS
l l .
GUID    org.illumos:sha512
READ\-ONLY COMPATIBLE   no
DEPENDENCIES    none
.TE

This feature enables the use of the SHA-512/256 truncated hash algorithm
(FIPS 180-4) for checksum and dedup. The native 64-bit arithemtic of
SHA-512 provides an approximate 50% performance boost over SHA-256 on
64-bit hardware and is thus a good minimum-change replacement candidate
for systems where hash performance is important, but these systems
cannot for whatever reason utilize the faster \fBskein\fR and
\fBedonr\fR algorithms.

When the \fBsha512\fR feature is set to \fBenabled\fR, the administrator
can turn on the \fBsha512\fR checksum on any dataset using the
\fBzfs\fR(1M) command. Please note that doing so will immediately
activate the \fBsha512\fR feature on the underlying pool (even before
any data is written). Since this feature is not read-only compatible,
this operation will render the pool unimportable on systems without
support for the \fBsha512\fR feature. At the moment, this operation
cannot be reversed. Booting off of pools utilizing SHA-512/256 is
supported, provided that the appropriate GRUB stage2 module is
installed.

.RE

.sp
.ne 2
.na
\fB\fBskein\fR\fR
.ad
.RS 4n
.TS
l l .
GUID    org.illumos:skein
READ\-ONLY COMPATIBLE   no
DEPENDENCIES    none
.TE

This feature enables the use of the Skein hash algorithm for checksum
and dedup. Skein is a high-performance secure hash algorithm that was a
```



```
finalist in the NIST SHA-3 competition. It provides a very high security
margin and high performance on 64-bit hardware (80% faster than
SHA-256). This implementation also utilizes the new salted checksumming
functionality in ZFS, which means that the checksum is pre-seeded with a
secret 256-bit random key (stored on the pool) before being fed the data
block to be checksummed. Thus the produced checksums are unique to a
given pool, preventing hash collision attacks on systems with dedup.

When the \fBskein\fR feature is set to \fBenabled\fR, the administrator
can turn on the \fBskein\fR checksum on any dataset using the
\fBzfs\fR(1M) command. Please note that doing so will immediately
activate the \fBskein\fR feature on the underlying pool (even before any
data is written). Since this feature is not read-only compatible, this
operation will render the pool unimportable on systems without support
for the \fBskein\fR feature. At the moment, this operation cannot be
reversed. Booting off of pools using \fBskein\fR is \fBNOT\fR supported
-- any attempt to enable \fBskein\fR on a root pool will fail with an
error.

.RE

.sp
.ne 2
.na
\fB\fBedonr\fR\fR
.ad
.RS 4n
.TS
l l .
GUID    org.illumos:edonr
READ\-ONLY COMPATIBLE   no
DEPENDENCIES    none
.TE

This feature enables the use of the Edon-R hash algorithm for checksum
and dedup. Edon-R is a very high-performance hash algorithm that was part
of the NIST SHA-3 competition. It provides extremely high hash
performance (over 350% faster than SHA-256), but was not selected
because of its unsuitability as a general purpose secure hash algorithm.
This implementation utilizes the new salted checksumming functionality
in ZFS, which means that the checksum is pre-seeded with a secret
256-bit random key (stored on the pool) before being fed the data block
to be checksummed. Thus the produced checksums are unique to a given
pool, blocking hash collision attacks on systems with dedup.

When the \fBedonr\fR feature is set to \fBenabled\fR, the administrator
can turn on the \fBedonr\fR checksum on any dataset using the
\fBzfs\fR(1M) command. Please note that doing so will immediately
activate the \fBedonr\fR feature on the underlying pool (even before any
data is written). Since this feature is not read-only compatible, this
operation will render the pool unimportable on systems without support
for the \fBedonr\fR feature. At the moment, this operation cannot be
reversed. Booting off of pools using \fBedonr\fR is \fBNOT\fR supported
-- any attempt to enable \fBedonr\fR on a root pool will fail with an
error.

.SH "SEE ALSO"
\fBzpool\fR(1M)
```

```
**********************************************************
   90388 Mon Oct  7 19:17:17 2013
new/usr/src/pkg/manifests/system-header.mf
4185 New hash algorithm support
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #

  22 #
  23 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
  24 # Copyright (c) 2012 by Delphix. All rights reserved.
  25 # Copyright 2012 Nexenta Systems, Inc. All rights reserved.
  26 # Copyright 2013 Saso Kiselkov. All rights reserved.
  27 #

  29 set name=pkg.fmri value=pkg:/system/header@$(PKGVERS)
  30 set name=pkg.description \
  31     value="SunOS C/C++ header files for general development of software"
  32 set name=pkg.summary value="SunOS Header Files"
  33 set name=info.classification value=org.opensolaris.category.2008:System/Core
  34 set name=variant.arch value=$(ARCH)
  35 dir path=usr group=sys
  36 dir path=usr/include
  37 $(i386_ONLY)dir path=usr/include/$(ARCH64)
  38 $(i386_ONLY)dir path=usr/include/$(ARCH64)/sys
  39 dir path=usr/include/arpa
  40 dir path=usr/include/asm
  41 dir path=usr/include/ast
  42 dir path=usr/include/bsm
  43 dir path=usr/include/dat
  44 dir path=usr/include/des
  45 dir path=usr/include/gssapi
  46 dir path=usr/include/hal
  47 $(i386_ONLY)dir path=usr/include/ia32
  48 $(i386_ONLY)dir path=usr/include/ia32/sys
  49 dir path=usr/include/inet
  50 dir path=usr/include/inet/kssl
  51 dir path=usr/include/ipp
  52 dir path=usr/include/ipp/ipgpc
  53 dir path=usr/include/iso
  54 dir path=usr/include/kerberosv5
  55 dir path=usr/include/libpolkit
  56 dir path=usr/include/net
  57 dir path=usr/include/netinet
  58 dir path=usr/include/nfs
  59 dir path=usr/include/protocols
  60 dir path=usr/include/rpc
  61 dir path=usr/include/rpcsvc
```



```
  62 dir path=usr/include/sasl
  63 dir path=usr/include/scsi
  64 dir path=usr/include/scsi/plugins
  65 dir path=usr/include/scsi/plugins/ses
  66 dir path=usr/include/scsi/plugins/ses/framework
  67 dir path=usr/include/scsi/plugins/ses/vendor
  68 dir path=usr/include/scsi/plugins/smp
  69 dir path=usr/include/scsi/plugins/smp/engine
  70 dir path=usr/include/scsi/plugins/smp/framework
  71 dir path=usr/include/security
  72 dir path=usr/include/sharefs
  73 dir path=usr/include/sys
  74 dir path=usr/include/sys/av
  75 dir path=usr/include/sys/contract
  76 dir path=usr/include/sys/crypto
  77 dir path=usr/include/sys/dktp
  78 dir path=usr/include/sys/fc4
  79 dir path=usr/include/sys/fm
  80 dir path=usr/include/sys/fm/cpu
  81 dir path=usr/include/sys/fm/fs
  82 dir path=usr/include/sys/fm/io
  83 $(sparc_ONLY)dir path=usr/include/sys/fpu
  84 dir path=usr/include/sys/fs
  85 dir path=usr/include/sys/hotplug
  86 dir path=usr/include/sys/hotplug/pci
  87 dir path=usr/include/sys/ib
  88 dir path=usr/include/sys/ib/adapters
  89 dir path=usr/include/sys/ib/adapters/hermon
  90 dir path=usr/include/sys/ib/adapters/tavor
  91 dir path=usr/include/sys/ib/clients
  92 dir path=usr/include/sys/ib/clients/ibd
  93 dir path=usr/include/sys/ib/clients/of
  94 dir path=usr/include/sys/ib/clients/of/rdma
  95 dir path=usr/include/sys/ib/clients/of/sol_ofs
  96 dir path=usr/include/sys/ib/clients/of/sol_ucma
  97 dir path=usr/include/sys/ib/clients/of/sol_umad
  98 dir path=usr/include/sys/ib/clients/of/sol_uverbs
  99 dir path=usr/include/sys/ib/ibnex
 100 dir path=usr/include/sys/ib/ibtl
 101 dir path=usr/include/sys/ib/ibtl/impl
 102 dir path=usr/include/sys/ib/mgt
 103 dir path=usr/include/sys/ib/mgt/ibmf
 104 dir path=usr/include/sys/iso
 105 dir path=usr/include/sys/lvm
 106 dir path=usr/include/sys/pcmcia
 107 dir path=usr/include/sys/proc
 108 dir path=usr/include/sys/rsm
 109 $(i386_ONLY)dir path=usr/include/sys/sata group=sys
 110 dir path=usr/include/sys/scsi
 111 dir path=usr/include/sys/scsi/adapters
 112 dir path=usr/include/sys/scsi/conf
 113 dir path=usr/include/sys/scsi/generic
 114 dir path=usr/include/sys/scsi/impl
 115 dir path=usr/include/sys/scsi/targets
 116 dir path=usr/include/sys/sysevent
 117 dir path=usr/include/sys/tsol
 118 dir path=usr/include/tsol
 119 dir path=usr/include/uuid
 120 $(sparc_ONLY)dir path=usr/include/v7
 121 $(sparc_ONLY)dir path=usr/include/v7/sys
 122 $(sparc_ONLY)dir path=usr/include/v9
 123 $(sparc_ONLY)dir path=usr/include/v9/sys
 124 dir path=usr/include/vm
 125 dir path=usr/platform group=sys
 126 $(sparc_ONLY)dir path=usr/platform/SUNW,A70 group=sys
 127 $(sparc_ONLY)dir path=usr/platform/SUNW,Netra-CP2300 group=sys
```

```
$(sparc_ONLY)dir path=usr/platform/SUNW,Netra-CP2300/include
$(sparc_ONLY)dir path=usr/platform/SUNW,Netra-CP3010 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Netra-CP3010/include
$(sparc_ONLY)dir path=usr/platform/SUNW,Netra-T12 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Netra-T4 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,SPARC-Enterprise group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Serverblade1 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Blade-100 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Blade-1000 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Blade-1500 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Blade-2500 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-15000 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-280R group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-480R group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-880 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V215 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V240 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V250 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V440 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V445 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V490 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V890 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-2 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-250 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-4 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-Enterprise group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-Enterprise-10000 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,UltraSPARC-IIe-NetraCT-40 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,UltraSPARC-IIe-NetraCT-60 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,UltraSPARC-IIi-Netract group=sys
$(i386_ONLY)dir path=usr/platform/i86pc group=sys
$(i386_ONLY)dir path=usr/platform/i86pc/include
$(i386_ONLY)dir path=usr/platform/i86pc/include/sys
$(i386_ONLY)dir path=usr/platform/i86pc/include/vm
$(i386_ONLY)dir path=usr/platform/i86xpv group=sys
$(i386_ONLY)dir path=usr/platform/i86xpv/include
$(i386_ONLY)dir path=usr/platform/i86xpv/include/sys
$(i386_ONLY)dir path=usr/platform/i86xpv/include/vm
$(sparc_ONLY)dir path=usr/platform/sun4u group=sys
$(sparc_ONLY)dir path=usr/platform/sun4u/include
$(sparc_ONLY)dir path=usr/platform/sun4u/include/sys
$(sparc_ONLY)dir path=usr/platform/sun4u/include/sys/i2c
$(sparc_ONLY)dir path=usr/platform/sun4u/include/sys/i2c/clients
$(sparc_ONLY)dir path=usr/platform/sun4u/include/sys/i2c/misc
$(sparc_ONLY)dir path=usr/platform/sun4u/include/vm
$(sparc_ONLY)dir path=usr/platform/sun4v group=sys
$(sparc_ONLY)dir path=usr/platform/sun4v/include
$(sparc_ONLY)dir path=usr/platform/sun4v/include/sys
$(sparc_ONLY)dir path=usr/platform/sun4v/include/vm
dir path=usr/share
dir path=usr/share/man
dir path=usr/share/man/man3head
dir path=usr/share/man/man4
dir path=usr/share/man/man5
dir path=usr/share/man/man7i
dir path=usr/share/src group=sys
dir path=usr/share/src/uts
$(i386_ONLY)dir path=usr/share/src/uts/i86pc
$(i386_ONLY)dir path=usr/share/src/uts/i86xpv
$(sparc_ONLY)dir path=usr/share/src/uts/sun4u
$(sparc_ONLY)dir path=usr/share/src/uts/sun4v
dir path=usr/xpg4
dir path=usr/xpg4/include
$(i386_ONLY)file path=usr/include/$(ARCH64)/sys/kdi_regs.h
$(i386_ONLY)file path=usr/include/$(ARCH64)/sys/privmregs.h
```



```
$(i386_ONLY)file path=usr/include/$(ARCH64)/sys/privregs.h
file path=usr/include/aio.h
file path=usr/include/alloca.h
file path=usr/include/apptrace.h
file path=usr/include/apptrace_impl.h
file path=usr/include/ar.h
file path=usr/include/archives.h
file path=usr/include/arpa/ftp.h
file path=usr/include/arpa/inet.h
file path=usr/include/arpa/nameser.h
file path=usr/include/arpa/nameser_compat.h
file path=usr/include/arpa/telnet.h
file path=usr/include/arpa/tftp.h
$(i386_ONLY)file path=usr/include/asm/atomic.h
$(i386_ONLY)file path=usr/include/asm/bitmap.h
$(i386_ONLY)file path=usr/include/asm/byteorder.h
$(i386_ONLY)file path=usr/include/asm/clock.h
$(i386_ONLY)file path=usr/include/asm/cpu.h
$(i386_ONLY)file path=usr/include/asm/cpuvar.h
$(sparc_ONLY)file path=usr/include/asm/flush.h
$(i386_ONLY)file path=usr/include/asm/htable.h
$(i386_ONLY)file path=usr/include/asm/mmu.h
file path=usr/include/asm/sunddi.h
file path=usr/include/asm/thread.h
file path=usr/include/assert.h
file path=usr/include/ast/align.h
file path=usr/include/ast/ast.h
file path=usr/include/ast/ast_botch.h
file path=usr/include/ast/ast_ccode.h
file path=usr/include/ast/ast_common.h
file path=usr/include/ast/ast_dir.h
file path=usr/include/ast/ast_dirent.h
file path=usr/include/ast/ast_fcntl.h
file path=usr/include/ast/ast_float.h
file path=usr/include/ast/ast_fs.h
file path=usr/include/ast/ast_getopt.h
file path=usr/include/ast/ast_iconv.h
file path=usr/include/ast/ast_lib.h
file path=usr/include/ast/ast_limits.h
file path=usr/include/ast/ast_map.h
file path=usr/include/ast/ast_mmap.h
file path=usr/include/ast/ast_mode.h
file path=usr/include/ast/ast_namval.h
file path=usr/include/ast/ast_ndbm.h
file path=usr/include/ast/ast_nl_types.h
file path=usr/include/ast/ast_param.h
file path=usr/include/ast/ast_standards.h
file path=usr/include/ast/ast_std.h
file path=usr/include/ast/ast_stdio.h
file path=usr/include/ast/ast_sys.h
file path=usr/include/ast/ast_time.h
file path=usr/include/ast/ast_tty.h
file path=usr/include/ast/ast_version.h
file path=usr/include/ast/ast_vfork.h
file path=usr/include/ast/ast_wait.h
file path=usr/include/ast/ast_wchar.h
file path=usr/include/ast/ast_windows.h
file path=usr/include/ast/bytesex.h
file path=usr/include/ast/ccode.h
file path=usr/include/ast/cdt.h
file path=usr/include/ast/cmd.h
file path=usr/include/ast/cmdext.h
file path=usr/include/ast/debug.h
file path=usr/include/ast/dirent.h
file path=usr/include/ast/dlldefs.h
file path=usr/include/ast/dt.h
```

```
260 file path=usr/include/ast/endian.h
261 file path=usr/include/ast/error.h
262 file path=usr/include/ast/find.h
263 file path=usr/include/ast/fnmatch.h
264 file path=usr/include/ast/fnv.h
265 file path=usr/include/ast/fs3d.h
266 file path=usr/include/ast/fts.h
267 file path=usr/include/ast/ftw.h
268 file path=usr/include/ast/ftwalk.h
269 file path=usr/include/ast/getopt.h
270 file path=usr/include/ast/glob.h
271 file path=usr/include/ast/hash.h
272 file path=usr/include/ast/hashkey.h
273 file path=usr/include/ast/hashpart.h
274 file path=usr/include/ast/history.h
275 file path=usr/include/ast/iconv.h
276 file path=usr/include/ast/ip6.h
277 file path=usr/include/ast/lc.h
278 file path=usr/include/ast/ls.h
279 file path=usr/include/ast/magic.h
280 file path=usr/include/ast/magicid.h
281 file path=usr/include/ast/mc.h
282 file path=usr/include/ast/mime.h
283 file path=usr/include/ast/mnt.h
284 file path=usr/include/ast/modecanon.h
285 file path=usr/include/ast/modex.h
286 file path=usr/include/ast/namval.h
287 file path=usr/include/ast/nl_types.h
288 file path=usr/include/ast/nval.h
289 file path=usr/include/ast/option.h
290 file path=usr/include/ast/preroot.h
291 file path=usr/include/ast/proc.h
292 file path=usr/include/ast/prototyped.h
293 file path=usr/include/ast/re_comp.h
294 file path=usr/include/ast/recfmt.h
295 file path=usr/include/ast/regex.h
296 file path=usr/include/ast/regexp.h
297 file path=usr/include/ast/sfdisc.h
298 file path=usr/include/ast/sfio.h
299 file path=usr/include/ast/sfio_s.h
300 file path=usr/include/ast/sfio_t.h
301 file path=usr/include/ast/shcmd.h
302 file path=usr/include/ast/shell.h
303 file path=usr/include/ast/sig.h
304 file path=usr/include/ast/stack.h
305 file path=usr/include/ast/stak.h
306 file path=usr/include/ast/stdio.h
307 file path=usr/include/ast/stk.h
308 file path=usr/include/ast/sum.h
309 file path=usr/include/ast/swap.h
310 file path=usr/include/ast/tar.h
311 file path=usr/include/ast/times.h
312 file path=usr/include/ast/tm.h
313 file path=usr/include/ast/tmx.h
314 file path=usr/include/ast/tok.h
315 file path=usr/include/ast/tv.h
316 file path=usr/include/ast/usage.h
317 file path=usr/include/ast/vdb.h
318 file path=usr/include/ast/vecargs.h
319 file path=usr/include/ast/vmalloc.h
320 file path=usr/include/ast/wait.h
321 file path=usr/include/ast/wchar.h
322 file path=usr/include/ast/wordexp.h
323 file path=usr/include/atomic.h
324 file path=usr/include/attr.h
325 file path=usr/include/auth_attr.h
```



```
326 file path=usr/include/bsm/adt.h
327 file path=usr/include/bsm/adt_event.h
328 file path=usr/include/bsm/audit.h
329 file path=usr/include/bsm/audit_kernel.h
330 file path=usr/include/bsm/audit_kevents.h
331 file path=usr/include/bsm/audit_record.h
332 file path=usr/include/bsm/audit_uevents.h
333 file path=usr/include/bsm/devices.h
334 file path=usr/include/bsm/libbsm.h
335 file path=usr/include/config_admin.h
336 file path=usr/include/cpio.h
337 file path=usr/include/crypt.h
338 file path=usr/include/cryptoutil.h
339 file path=usr/include/ctype.h
340 file path=usr/include/curses.h
341 file path=usr/include/dat/dat.h
342 file path=usr/include/dat/dat_error.h
343 file path=usr/include/dat/dat_platform_specific.h
344 file path=usr/include/dat/dat_redirection.h
345 file path=usr/include/dat/dat_registry.h
346 file path=usr/include/dat/dat_vendor_specific.h
347 file path=usr/include/dat/udat.h
348 file path=usr/include/dat/udat_config.h
349 file path=usr/include/dat/udat_redirection.h
350 file path=usr/include/dat/udat_vendor_specific.h
351 file path=usr/include/deflt.h
352 file path=usr/include/des/des.h
353 file path=usr/include/des/desdata.h
354 file path=usr/include/des/softdes.h
355 file path=usr/include/device_info.h
356 file path=usr/include/devid.h
357 file path=usr/include/devmgmt.h
358 file path=usr/include/devpoll.h
359 file path=usr/include/dial.h
360 file path=usr/include/dirent.h
361 file path=usr/include/dlfcn.h
362 file path=usr/include/door.h
363 file path=usr/include/elf.h
364 file path=usr/include/err.h
365 file path=usr/include/errno.h
366 file path=usr/include/eti.h
367 file path=usr/include/euc.h
368 file path=usr/include/exacct.h
369 file path=usr/include/exacct_impl.h
370 file path=usr/include/exec_attr.h
371 file path=usr/include/execinfo.h
372 file path=usr/include/fatal.h
373 file path=usr/include/fcntl.h
374 file path=usr/include/float.h
375 file path=usr/include/fmtmsg.h
376 file path=usr/include/fnmatch.h
377 file path=usr/include/form.h
378 file path=usr/include/ftw.h
379 file path=usr/include/gelf.h
380 file path=usr/include/getopt.h
381 file path=usr/include/getwidth.h
382 file path=usr/include/glob.h
383 file path=usr/include/grp.h
384 file path=usr/include/gssapi/gssapi.h
385 file path=usr/include/gssapi/gssapi_ext.h
386 file path=usr/include/hal/libhal-storage.h
387 file path=usr/include/hal/libhal.h
388 $(i386_ONLY)file path=usr/include/ia32/sys/asm_linkage.h
389 $(i386_ONLY)file path=usr/include/ia32/sys/kdi_regs.h
390 $(i386_ONLY)file path=usr/include/ia32/sys/machtypes.h
391 $(i386_ONLY)file path=usr/include/ia32/sys/privmregs.h
```

```
$(i386_ONLY)file path=usr/include/ia32/sys/privregs.h
$(i386_ONLY)file path=usr/include/ia32/sys/psw.h
$(i386_ONLY)file path=usr/include/ia32/sys/pte.h
$(i386_ONLY)file path=usr/include/ia32/sys/reg.h
$(i386_ONLY)file path=usr/include/ia32/sys/stack.h
$(i386_ONLY)file path=usr/include/ia32/sys/trap.h
$(i386_ONLY)file path=usr/include/ia32/sys/traptrace.h
file path=usr/include/iconv.h
file path=usr/include/idmap.h
file path=usr/include/ieeefp.h
file path=usr/include/ifaddrs.h
file path=usr/include/inet/arp.h
file path=usr/include/inet/common.h
file path=usr/include/inet/ip.h
file path=usr/include/inet/ip6.h
file path=usr/include/inet/ip6_asp.h
file path=usr/include/inet/ip_arp.h
file path=usr/include/inet/ip_ftable.h
file path=usr/include/inet/ip_if.h
file path=usr/include/inet/ip_ire.h
file path=usr/include/inet/ip_multi.h
file path=usr/include/inet/ip_netinfo.h
file path=usr/include/inet/ip_rts.h
file path=usr/include/inet/ip_stack.h
file path=usr/include/inet/ipclassifier.h
file path=usr/include/inet/ipdrop.h
file path=usr/include/inet/ipnet.h
file path=usr/include/inet/ipp_common.h
file path=usr/include/inet/kssl/ksslapi.h
file path=usr/include/inet/led.h
file path=usr/include/inet/mi.h
file path=usr/include/inet/mib2.h
file path=usr/include/inet/nd.h
file path=usr/include/inet/optcom.h
file path=usr/include/inet/sctp_itf.h
file path=usr/include/inet/snmpcom.h
file path=usr/include/inet/tcp.h
file path=usr/include/inet/tcp_sack.h
file path=usr/include/inet/tcp_stack.h
file path=usr/include/inet/tcp_stats.h
file path=usr/include/inet/tunables.h
file path=usr/include/inet/wifi_ioctl.h
file path=usr/include/inttypes.h
file path=usr/include/ipmp.h
file path=usr/include/ipmp_admin.h
file path=usr/include/ipmp_mpathd.h
file path=usr/include/ipmp_query.h
file path=usr/include/ipp/ipgpc/ipgpc.h
file path=usr/include/ipp/ipp.h
file path=usr/include/ipp/ipp_config.h
file path=usr/include/ipp/ipp_impl.h
file path=usr/include/ipp/ippctl.h
file path=usr/include/iso/ctype_c99.h
file path=usr/include/iso/ctype_iso.h
file path=usr/include/iso/limits_iso.h
file path=usr/include/iso/locale_iso.h
file path=usr/include/iso/setjmp_iso.h
file path=usr/include/iso/signal_iso.h
file path=usr/include/iso/stdarg_c99.h
file path=usr/include/iso/stdarg_iso.h
file path=usr/include/iso/stddef_iso.h
file path=usr/include/iso/stdio_c99.h
file path=usr/include/iso/stdio_iso.h
file path=usr/include/iso/stdlib_c99.h
file path=usr/include/iso/stdlib_iso.h
file path=usr/include/iso/string_iso.h
```



```
file path=usr/include/iso/time_iso.h
file path=usr/include/iso/wchar_c99.h
file path=usr/include/iso/wchar_iso.h
file path=usr/include/iso/wctype_c99.h
file path=usr/include/iso/wctype_iso.h
file path=usr/include/iso646.h
file path=usr/include/kerberosv5/com_err.h
file path=usr/include/kerberosv5/krb5.h
file path=usr/include/kerberosv5/mit-sipb-copyright.h
file path=usr/include/kerberosv5/mit_copyright.h
file path=usr/include/klpd.h
file path=usr/include/kmfapi.h
file path=usr/include/kmftypes.h
file path=usr/include/kstat.h
file path=usr/include/kvm.h
file path=usr/include/langinfo.h
file path=usr/include/lastlog.h
file path=usr/include/lber.h
file path=usr/include/ldap.h
file path=usr/include/libcontract.h
file path=usr/include/libctf.h
file path=usr/include/libdevice.h
file path=usr/include/libdevinfo.h
file path=usr/include/libdladm.h
file path=usr/include/libdlbridge.h
file path=usr/include/libdlib.h
file path=usr/include/libdllink.h
file path=usr/include/libdlpi.h
file path=usr/include/libdlvlan.h
file path=usr/include/libelf.h
$(i386_ONLY)file path=usr/include/libfdisk.h
file path=usr/include/libfstyp.h
file path=usr/include/libfstyp_module.h
file path=usr/include/libgen.h
file path=usr/include/libgrubmgmt.h
file path=usr/include/libintl.h
file path=usr/include/libipmi.h
file path=usr/include/libipp.h
file path=usr/include/libnvpair.h
file path=usr/include/libnwam.h
file path=usr/include/libpolkit/libpolkit.h
file path=usr/include/librcm.h
file path=usr/include/libscf.h
file path=usr/include/libscf_priv.h
file path=usr/include/libshare.h
file path=usr/include/libsvm.h
file path=usr/include/libsysevent.h
file path=usr/include/libsysevent_impl.h
file path=usr/include/libtsnet.h
$(sparc_ONLY)file path=usr/include/libv12n.h
file path=usr/include/libw.h
file path=usr/include/libzfs.h
file path=usr/include/libzfs_core.h
file path=usr/include/libzoneinfo.h
file path=usr/include/limits.h
file path=usr/include/linenum.h
file path=usr/include/link.h
file path=usr/include/listen.h
file path=usr/include/locale.h
file path=usr/include/macros.h
file path=usr/include/maillock.h
file path=usr/include/malloc.h
file path=usr/include/md4.h
file path=usr/include/md5.h
file path=usr/include/mdiox.h
file path=usr/include/mdmn_changelog.h
```

```
524 file path=usr/include/memory.h
525 file path=usr/include/menu.h
526 file path=usr/include/meta.h
527 file path=usr/include/meta_basic.h
528 file path=usr/include/meta_runtime.h
529 file path=usr/include/metacl.h
530 file path=usr/include/metad.h
531 file path=usr/include/metadyn.h
532 file path=usr/include/metamed.h
533 file path=usr/include/metamhd.h
534 file path=usr/include/mhdx.h
535 file path=usr/include/mon.h
536 file path=usr/include/monetary.h
537 file path=usr/include/mp.h
538 file path=usr/include/mqueue.h
539 file path=usr/include/mtmalloc.h
540 file path=usr/include/nan.h
541 file path=usr/include/ndbm.h
542 file path=usr/include/ndpd.h
543 file path=usr/include/net/af.h
544 file path=usr/include/net/bridge.h
545 file path=usr/include/net/if.h
546 file path=usr/include/net/if_arp.h
547 file path=usr/include/net/if_dl.h
548 file path=usr/include/net/if_types.h
549 file path=usr/include/net/pfkeyv2.h
550 file path=usr/include/net/pfpolicy.h
551 file path=usr/include/net/ppp-comp.h
552 file path=usr/include/net/ppp_defs.h
553 file path=usr/include/net/pppio.h
554 file path=usr/include/net/radix.h
555 file path=usr/include/net/route.h
556 file path=usr/include/net/trill.h
557 file path=usr/include/net/vjcompress.h
558 file path=usr/include/netconfig.h
559 file path=usr/include/netdb.h
560 file path=usr/include/netdir.h
561 file path=usr/include/netinet/arp.h
562 file path=usr/include/netinet/dhcp.h
563 file path=usr/include/netinet/dhcp6.h
564 file path=usr/include/netinet/icmp6.h
565 file path=usr/include/netinet/icmp_var.h
566 file path=usr/include/netinet/if_ether.h
567 file path=usr/include/netinet/igmp.h
568 file path=usr/include/netinet/igmp_var.h
569 file path=usr/include/netinet/in.h
570 file path=usr/include/netinet/in_pcb.h
571 file path=usr/include/netinet/in_systm.h
572 file path=usr/include/netinet/in_var.h
573 file path=usr/include/netinet/ip.h
574 file path=usr/include/netinet/ip6.h
575 file path=usr/include/netinet/ip_icmp.h
576 file path=usr/include/netinet/ip_mroute.h
577 file path=usr/include/netinet/ip_var.h
578 file path=usr/include/netinet/pim.h
579 file path=usr/include/netinet/sctp.h
580 file path=usr/include/netinet/tcp.h
581 file path=usr/include/netinet/tcp_debug.h
582 file path=usr/include/netinet/tcp_fsm.h
583 file path=usr/include/netinet/tcp_seq.h
584 file path=usr/include/netinet/tcp_timer.h
585 file path=usr/include/netinet/tcp_var.h
586 file path=usr/include/netinet/tcpip.h
587 file path=usr/include/netinet/udp.h
588 file path=usr/include/netinet/udp_var.h
589 file path=usr/include/netinet/vrrp.h
```



```
590 file path=usr/include/nfs/auth.h
591 file path=usr/include/nfs/export.h
592 file path=usr/include/nfs/lm.h
593 file path=usr/include/nfs/mapid.h
594 file path=usr/include/nfs/mount.h
595 file path=usr/include/nfs/nfs.h
596 file path=usr/include/nfs/nfs4.h
597 file path=usr/include/nfs/nfs4_attr.h
598 file path=usr/include/nfs/nfs4_clnt.h
599 file path=usr/include/nfs/nfs4_db_impl.h
600 file path=usr/include/nfs/nfs4_idmap_impl.h
601 file path=usr/include/nfs/nfs4_kprot.h
602 file path=usr/include/nfs/nfs_acl.h
603 file path=usr/include/nfs/nfs_clnt.h
604 file path=usr/include/nfs/nfs_cmd.h
605 file path=usr/include/nfs/nfs_log.h
606 file path=usr/include/nfs/nfs_sec.h
607 file path=usr/include/nfs/nfsid_map.h
608 file path=usr/include/nfs/nfssys.h
609 file path=usr/include/nfs/rnode.h
610 file path=usr/include/nfs/rnode4.h
611 file path=usr/include/nl_types.h
612 file path=usr/include/nlist.h
613 file path=usr/include/note.h
614 file path=usr/include/nss_common.h
615 file path=usr/include/nss_dbdefs.h
616 file path=usr/include/nss_netdir.h
617 file path=usr/include/nsswitch.h
618 file path=usr/include/panel.h
619 file path=usr/include/paths.h
620 file path=usr/include/pcsample.h
621 file path=usr/include/pfmt.h
622 file path=usr/include/pkgdev.h
623 file path=usr/include/pkginfo.h
624 file path=usr/include/pkglocs.h
625 file path=usr/include/pkgstrct.h
626 file path=usr/include/pkgtrans.h
627 file path=usr/include/poll.h
628 file path=usr/include/port.h
629 file path=usr/include/priv.h
630 file path=usr/include/proc_service.h
631 file path=usr/include/procfs.h
632 file path=usr/include/prof.h
633 file path=usr/include/prof_attr.h
634 file path=usr/include/project.h
635 file path=usr/include/protocols/dumprestore.h
636 file path=usr/include/protocols/routed.h
637 file path=usr/include/protocols/rwhod.h
638 file path=usr/include/protocols/timed.h
639 file path=usr/include/pthread.h
640 file path=usr/include/pw.h
641 file path=usr/include/pwd.h
642 file path=usr/include/rcm_module.h
643 file path=usr/include/rctl.h
644 file path=usr/include/re_comp.h
645 file path=usr/include/regex.h
646 file path=usr/include/regexp.h
647 file path=usr/include/regexpr.h
648 file path=usr/include/resolv.h
649 file path=usr/include/rje.h
650 file path=usr/include/rp_plugin.h
651 file path=usr/include/rpc/auth.h
652 file path=usr/include/rpc/auth_des.h
653 file path=usr/include/rpc/auth_sys.h
654 file path=usr/include/rpc/auth_unix.h
655 file path=usr/include/rpc/bootparam.h
```

```
 656 file path=usr/include/rpc/clnt.h
 657 file path=usr/include/rpc/clnt_soc.h
 658 file path=usr/include/rpc/clnt_stat.h
 659 file path=usr/include/rpc/des_crypt.h
 660 $(sparc_ONLY)file path=usr/include/rpc/ib.h
 661 file path=usr/include/rpc/key_prot.h
 662 file path=usr/include/rpc/nettype.h
 663 file path=usr/include/rpc/pmap_clnt.h
 664 file path=usr/include/rpc/pmap_prot.h
 665 file path=usr/include/rpc/pmap_prot.x
 666 file path=usr/include/rpc/pmap_rmt.h
 667 file path=usr/include/rpc/raw.h
 668 file path=usr/include/rpc/rpc.h
 669 file path=usr/include/rpc/rpc_com.h
 670 file path=usr/include/rpc/rpc_msg.h
 671 file path=usr/include/rpc/rpc_rdma.h
 672 file path=usr/include/rpc/rpc_sztypes.h
 673 file path=usr/include/rpc/rpcb_clnt.h
 674 file path=usr/include/rpc/rpcb_prot.h
 675 file path=usr/include/rpc/rpcb_prot.x
 676 file path=usr/include/rpc/rpcent.h
 677 file path=usr/include/rpc/rpcsec_gss.h
 678 file path=usr/include/rpc/rpcsys.h
 679 file path=usr/include/rpc/svc.h
 680 file path=usr/include/rpc/svc_auth.h
 681 file path=usr/include/rpc/svc_mt.h
 682 file path=usr/include/rpc/svc_soc.h
 683 file path=usr/include/rpc/types.h
 684 file path=usr/include/rpc/xdr.h
 685 file path=usr/include/rpcsvc/autofs_prot.h
 686 file path=usr/include/rpcsvc/autofs_prot.x
 687 file path=usr/include/rpcsvc/bootparam.h
 688 file path=usr/include/rpcsvc/bootparam_prot.h
 689 file path=usr/include/rpcsvc/bootparam_prot.x
 690 file path=usr/include/rpcsvc/dbm.h
 691 file path=usr/include/rpcsvc/key_prot.x
 692 file path=usr/include/rpcsvc/mount.h
 693 file path=usr/include/rpcsvc/mount.x
 694 file path=usr/include/rpcsvc/nfs4_prot.h
 695 file path=usr/include/rpcsvc/nfs4_prot.x
 696 file path=usr/include/rpcsvc/nfs_acl.h
 697 file path=usr/include/rpcsvc/nfs_acl.x
 698 file path=usr/include/rpcsvc/nfs_prot.h
 699 file path=usr/include/rpcsvc/nfs_prot.x
 700 file path=usr/include/rpcsvc/nis.h
 701 file path=usr/include/rpcsvc/nis.x
 702 file path=usr/include/rpcsvc/nis_db.h
 703 file path=usr/include/rpcsvc/nis_object.x
 704 file path=usr/include/rpcsvc/nislib.h
 705 file path=usr/include/rpcsvc/nlm_prot.h
 706 file path=usr/include/rpcsvc/nlm_prot.x
 707 file path=usr/include/rpcsvc/nsm_addr.h
 708 file path=usr/include/rpcsvc/nsm_addr.x
 709 file path=usr/include/rpcsvc/rex.h
 710 file path=usr/include/rpcsvc/rex.x
 711 file path=usr/include/rpcsvc/rpc_sztypes.h
 712 file path=usr/include/rpcsvc/rpc_sztypes.x
 713 file path=usr/include/rpcsvc/rquota.h
 714 file path=usr/include/rpcsvc/rquota.x
 715 file path=usr/include/rpcsvc/rstat.h
 716 file path=usr/include/rpcsvc/rstat.x
 717 file path=usr/include/rpcsvc/rusers.h
 718 file path=usr/include/rpcsvc/rusers.x
 719 file path=usr/include/rpcsvc/rwall.h
 720 file path=usr/include/rpcsvc/rwall.x
 721 file path=usr/include/rpcsvc/sm_inter.h
```



```
 722 file path=usr/include/rpcsvc/sm_inter.x
 723 file path=usr/include/rpcsvc/spray.h
 724 file path=usr/include/rpcsvc/spray.x
 725 file path=usr/include/rpcsvc/ufs_prot.h
 726 file path=usr/include/rpcsvc/ufs_prot.x
 727 file path=usr/include/rpcsvc/yp.x
 728 file path=usr/include/rpcsvc/yp_prot.h
 729 file path=usr/include/rpcsvc/ypclnt.h
 730 file path=usr/include/rpcsvc/yppasswd.h
 731 file path=usr/include/rpcsvc/ypupd.h
 732 file path=usr/include/rsmapi.h
 733 file path=usr/include/rtld_db.h
 734 file path=usr/include/sac.h
 735 file path=usr/include/sasl/prop.h
 736 file path=usr/include/sasl/sasl.h
 737 file path=usr/include/sasl/saslplug.h
 738 file path=usr/include/sasl/saslutil.h
 739 file path=usr/include/sched.h
 740 file path=usr/include/schedctl.h
 741 file path=usr/include/scsi/libscsi.h
 742 file path=usr/include/scsi/libses.h
 743 file path=usr/include/scsi/libses_plugin.h
 744 file path=usr/include/scsi/libsmp.h
 745 file path=usr/include/scsi/libsmp_plugin.h
 746 file path=usr/include/scsi/plugins/ses/framework/libses.h
 747 file path=usr/include/scsi/plugins/ses/framework/ses2.h
 748 file path=usr/include/scsi/plugins/ses/framework/ses2_impl.h
 749 file path=usr/include/scsi/plugins/ses/vendor/sun.h
 750 file path=usr/include/sdp.h
 751 file path=usr/include/search.h
 752 file path=usr/include/secdb.h
 753 file path=usr/include/security/auditd.h
 754 file path=usr/include/security/cryptoki.h
 755 file path=usr/include/security/pam_appl.h
 756 file path=usr/include/security/pam_modules.h
 757 file path=usr/include/security/pkcs11.h
 758 file path=usr/include/security/pkcs11f.h
 759 file path=usr/include/security/pkcs11t.h
 760 file path=usr/include/semaphore.h
 761 file path=usr/include/setjmp.h
 762 file path=usr/include/sgtty.h
 763 file path=usr/include/sha1.h
 764 file path=usr/include/sha2.h
 765 file path=usr/include/shadow.h
 766 file path=usr/include/sharefs/share.h
 767 file path=usr/include/sharefs/sharefs.h
 768 file path=usr/include/sharefs/sharetab.h
 769 file path=usr/include/siginfo.h
 770 file path=usr/include/signal.h
 771 file path=usr/include/sip.h
 772 file path=usr/include/skein.h
 773 file path=usr/include/smbios.h
 774 file path=usr/include/spawn.h
 775 $(i386_ONLY)file path=usr/include/stack_unwind.h
 776 file path=usr/include/stdarg.h
 777 file path=usr/include/stdbool.h
 778 file path=usr/include/stddef.h
 779 file path=usr/include/stdint.h
 780 file path=usr/include/stdio.h
 781 file path=usr/include/stdio_ext.h
 782 file path=usr/include/stdio_impl.h
 783 file path=usr/include/stdio_tag.h
 784 file path=usr/include/stdlib.h
 785 file path=usr/include/storclass.h
 786 file path=usr/include/string.h
 787 file path=usr/include/strings.h
```

```
file path=usr/include/stropts.h
file path=usr/include/syms.h
file path=usr/include/synch.h
file path=usr/include/sys/acct.h
file path=usr/include/sys/acctctl.h
file path=usr/include/sys/acl.h
file path=usr/include/sys/acl_impl.h
file path=usr/include/sys/acpi_drv.h
file path=usr/include/sys/aio.h
file path=usr/include/sys/aio_impl.h
file path=usr/include/sys/aio_req.h
file path=usr/include/sys/aiocb.h
file path=usr/include/sys/archsystm.h
file path=usr/include/sys/ascii.h
file path=usr/include/sys/asm_linkage.h
file path=usr/include/sys/asynch.h
file path=usr/include/sys/atomic.h
file path=usr/include/sys/attr.h
file path=usr/include/sys/autoconf.h
file path=usr/include/sys/auxv.h
file path=usr/include/sys/auxv_386.h
file path=usr/include/sys/auxv_SPARC.h
file path=usr/include/sys/av/iec61883.h
file path=usr/include/sys/avintr.h
file path=usr/include/sys/avl.h
file path=usr/include/sys/avl_impl.h
file path=usr/include/sys/bitmap.h
file path=usr/include/sys/bitset.h
file path=usr/include/sys/bl.h
file path=usr/include/sys/blkdev.h
file path=usr/include/sys/bofi.h
file path=usr/include/sys/bofi_impl.h
file path=usr/include/sys/bootconf.h
$(i386_ONLY)file path=usr/include/sys/bootregs.h
file path=usr/include/sys/bootstat.h
$(i386_ONLY)file path=usr/include/sys/bootsvcs.h
file path=usr/include/sys/bpp_io.h
file path=usr/include/sys/brand.h
file path=usr/include/sys/buf.h
file path=usr/include/sys/bufmod.h
file path=usr/include/sys/bustypes.h
file path=usr/include/sys/byteorder.h
file path=usr/include/sys/callb.h
file path=usr/include/sys/callo.h
file path=usr/include/sys/cap_util.h
file path=usr/include/sys/ccompile.h
file path=usr/include/sys/cdio.h
file path=usr/include/sys/cis.h
file path=usr/include/sys/cis_handlers.h
file path=usr/include/sys/cis_protos.h
file path=usr/include/sys/cladm.h
file path=usr/include/sys/class.h
file path=usr/include/sys/clconf.h
file path=usr/include/sys/cmlb.h
file path=usr/include/sys/cmn_err.h
$(sparc_ONLY)file path=usr/include/sys/cmpregs.h
file path=usr/include/sys/compress.h
file path=usr/include/sys/condvar.h
file path=usr/include/sys/condvar_impl.h
file path=usr/include/sys/conf.h
file path=usr/include/sys/consdev.h
file path=usr/include/sys/console.h
file path=usr/include/sys/consplat.h
file path=usr/include/sys/contract.h
file path=usr/include/sys/contract/device.h
file path=usr/include/sys/contract/device_impl.h
```



```
file path=usr/include/sys/contract/process.h
file path=usr/include/sys/contract/process_impl.h
file path=usr/include/sys/contract_impl.h
$(i386_ONLY)file path=usr/include/sys/controlregs.h
file path=usr/include/sys/copyops.h
file path=usr/include/sys/core.h
file path=usr/include/sys/corectl.h
file path=usr/include/sys/cpc_impl.h
file path=usr/include/sys/cpc_pcbe.h
file path=usr/include/sys/cpr.h
file path=usr/include/sys/cpu.h
file path=usr/include/sys/cpucaps.h
file path=usr/include/sys/cpucaps_impl.h
file path=usr/include/sys/cpupart.h
file path=usr/include/sys/cpuvar.h
file path=usr/include/sys/crc32.h
file path=usr/include/sys/cred.h
file path=usr/include/sys/cred_impl.h
file path=usr/include/sys/crtctl.h
file path=usr/include/sys/crypto/api.h
file path=usr/include/sys/crypto/common.h
file path=usr/include/sys/crypto/ioctl.h
file path=usr/include/sys/crypto/ioctladmin.h
file path=usr/include/sys/crypto/spi.h
file path=usr/include/sys/cs.h
file path=usr/include/sys/cs_priv.h
file path=usr/include/sys/cs_strings.h
file path=usr/include/sys/cs_stubs.h
file path=usr/include/sys/cs_types.h
file path=usr/include/sys/csiioctl.h
file path=usr/include/sys/ctf.h
file path=usr/include/sys/ctf_api.h
file path=usr/include/sys/ctfs.h
file path=usr/include/sys/ctfs_impl.h
file path=usr/include/sys/ctype.h
file path=usr/include/sys/cyclic.h
file path=usr/include/sys/cyclic_impl.h
file path=usr/include/sys/dacf.h
file path=usr/include/sys/dacf_impl.h
file path=usr/include/sys/damap.h
file path=usr/include/sys/damap_impl.h
file path=usr/include/sys/dc_ki.h
file path=usr/include/sys/ddi.h
file path=usr/include/sys/ddi_hp.h
file path=usr/include/sys/ddi_hp_impl.h
file path=usr/include/sys/ddi_impldefs.h
file path=usr/include/sys/ddi_implfuncs.h
file path=usr/include/sys/ddi_intr.h
file path=usr/include/sys/ddi_intr_impl.h
file path=usr/include/sys/ddi_isa.h
file path=usr/include/sys/ddi_obsolete.h
file path=usr/include/sys/ddi_periodic.h
file path=usr/include/sys/ddidevmap.h
file path=usr/include/sys/ddidmareq.h
file path=usr/include/sys/ddifm.h
file path=usr/include/sys/ddifm_impl.h
file path=usr/include/sys/ddimapreq.h
file path=usr/include/sys/ddipropdefs.h
file path=usr/include/sys/dditypes.h
file path=usr/include/sys/debug.h
$(i386_ONLY)file path=usr/include/sys/debugreg.h
file path=usr/include/sys/des.h
file path=usr/include/sys/devcache.h
file path=usr/include/sys/devcache_impl.h
file path=usr/include/sys/devctl.h
file path=usr/include/sys/devfm.h
```

```
 920 file path=usr/include/sys/devid_cache.h
 921 file path=usr/include/sys/devinfo_impl.h
 922 file path=usr/include/sys/devops.h
 923 file path=usr/include/sys/devpolicy.h
 924 file path=usr/include/sys/devpoll.h
 925 file path=usr/include/sys/dirent.h
 926 file path=usr/include/sys/disp.h
 927 file path=usr/include/sys/dkbad.h
 928 file path=usr/include/sys/dkio.h
 929 file path=usr/include/sys/dklabel.h
 930 $(sparc_ONLY)file path=usr/include/sys/dkmpio.h
 931 $(i386_ONLY)file path=usr/include/sys/dktp/altsctr.h
 932 $(i386_ONLY)file path=usr/include/sys/dktp/cmpkt.h
 933 file path=usr/include/sys/dktp/dadkio.h
 934 file path=usr/include/sys/dktp/fdisk.h
 935 file path=usr/include/sys/dl.h
 936 file path=usr/include/sys/dld.h
 937 file path=usr/include/sys/dlpi.h
 938 file path=usr/include/sys/dls_mgmt.h
 939 $(i386_ONLY)file path=usr/include/sys/dma_engine.h
 940 file path=usr/include/sys/dma_i8237A.h
 941 file path=usr/include/sys/dnlc.h
 942 file path=usr/include/sys/door.h
 943 file path=usr/include/sys/door_data.h
 944 file path=usr/include/sys/door_impl.h
 945 file path=usr/include/sys/dumphdr.h
 946 file path=usr/include/sys/ecppio.h
 947 file path=usr/include/sys/ecppreg.h
 948 file path=usr/include/sys/ecppsys.h
 949 file path=usr/include/sys/ecppvar.h
 950 file path=usr/include/sys/edonr.h
 951 file path=usr/include/sys/efi_partition.h
 952 file path=usr/include/sys/elf.h
 953 file path=usr/include/sys/elf_386.h
 954 file path=usr/include/sys/elf_SPARC.h
 955 file path=usr/include/sys/elf_amd64.h
 956 file path=usr/include/sys/elf_notes.h
 957 file path=usr/include/sys/elftypes.h
 958 file path=usr/include/sys/epm.h
 959 file path=usr/include/sys/errno.h
 960 file path=usr/include/sys/errorq.h
 961 file path=usr/include/sys/errorq_impl.h
 962 file path=usr/include/sys/esunddi.h
 963 file path=usr/include/sys/ethernet.h
 964 file path=usr/include/sys/euc.h
 965 file path=usr/include/sys/eucioctl.h
 966 file path=usr/include/sys/exacct.h
 967 file path=usr/include/sys/exacct_catalog.h
 968 file path=usr/include/sys/exacct_impl.h
 969 file path=usr/include/sys/exec.h
 970 file path=usr/include/sys/exechdr.h
 971 file path=usr/include/sys/fault.h
 972 file path=usr/include/sys/fbio.h
 973 file path=usr/include/sys/fbuf.h
 974 file path=usr/include/sys/fc4/fc.h
 975 file path=usr/include/sys/fc4/fc_transport.h
 976 file path=usr/include/sys/fc4/fcal.h
 977 file path=usr/include/sys/fc4/fcal_linkapp.h
 978 file path=usr/include/sys/fc4/fcal_transport.h
 979 file path=usr/include/sys/fc4/fcio.h
 980 file path=usr/include/sys/fc4/fcp.h
 981 file path=usr/include/sys/fc4/linkapp.h
 982 file path=usr/include/sys/fcntl.h
 983 file path=usr/include/sys/fdbuffer.h
 984 file path=usr/include/sys/fdio.h
 985 $(sparc_ONLY)file path=usr/include/sys/fdreg.h
```



```
 986 $(sparc_ONLY)file path=usr/include/sys/fdvar.h
 987 file path=usr/include/sys/feature_tests.h
 988 file path=usr/include/sys/fem.h
 989 file path=usr/include/sys/file.h
 990 file path=usr/include/sys/filio.h
 991 file path=usr/include/sys/flock.h
 992 file path=usr/include/sys/flock_impl.h
 993 $(sparc_ONLY)file path=usr/include/sys/fm/cpu/SPARC64-VI.h
 994 $(sparc_ONLY)file path=usr/include/sys/fm/cpu/UltraSPARC-II.h
 995 $(sparc_ONLY)file path=usr/include/sys/fm/cpu/UltraSPARC-III.h
 996 $(sparc_ONLY)file path=usr/include/sys/fm/cpu/UltraSPARC-T1.h
 997 file path=usr/include/sys/fm/fs/zfs.h
 998 file path=usr/include/sys/fm/io/ddi.h
 999 file path=usr/include/sys/fm/io/disk.h
1000 file path=usr/include/sys/fm/io/opl_mc_fm.h
1001 file path=usr/include/sys/fm/io/pci.h
1002 file path=usr/include/sys/fm/io/scsi.h
1003 file path=usr/include/sys/fm/io/sun4upci.h
1004 file path=usr/include/sys/fm/protocol.h
1005 file path=usr/include/sys/fm/util.h
1006 file path=usr/include/sys/fork.h
1007 $(i386_ONLY)file path=usr/include/sys/fp.h
1008 $(sparc_ONLY)file path=usr/include/sys/fpu/fpu_simulator.h
1009 $(sparc_ONLY)file path=usr/include/sys/fpu/fpusystm.h
1010 $(sparc_ONLY)file path=usr/include/sys/fpu/globals.h
1011 $(sparc_ONLY)file path=usr/include/sys/fpu/ieee.h
1012 file path=usr/include/sys/frame.h
1013 file path=usr/include/sys/fs/autofs.h
1014 file path=usr/include/sys/fs/cachefs_dir.h
1015 file path=usr/include/sys/fs/cachefs_dlog.h
1016 file path=usr/include/sys/fs/cachefs_filegrp.h
1017 file path=usr/include/sys/fs/cachefs_fs.h
1018 file path=usr/include/sys/fs/cachefs_fscache.h
1019 file path=usr/include/sys/fs/cachefs_ioctl.h
1020 file path=usr/include/sys/fs/cachefs_log.h
1021 file path=usr/include/sys/fs/decomp.h
1022 file path=usr/include/sys/fs/dv_node.h
1023 file path=usr/include/sys/fs/fifonode.h
1024 file path=usr/include/sys/fs/hsfs_isospec.h
1025 file path=usr/include/sys/fs/hsfs_node.h
1026 file path=usr/include/sys/fs/hsfs_rrip.h
1027 file path=usr/include/sys/fs/hsfs_spec.h
1028 file path=usr/include/sys/fs/hsfs_susp.h
1029 file path=usr/include/sys/fs/lofs_info.h
1030 file path=usr/include/sys/fs/lofs_node.h
1031 file path=usr/include/sys/fs/mntdata.h
1032 file path=usr/include/sys/fs/namenode.h
1033 file path=usr/include/sys/fs/pc_dir.h
1034 file path=usr/include/sys/fs/pc_fs.h
1035 file path=usr/include/sys/fs/pc_label.h
1036 file path=usr/include/sys/fs/pc_node.h
1037 file path=usr/include/sys/fs/pxfs_ki.h
1038 file path=usr/include/sys/fs/sdev_impl.h
1039 file path=usr/include/sys/fs/snode.h
1040 file path=usr/include/sys/fs/swapnode.h
1041 file path=usr/include/sys/fs/tmp.h
1042 file path=usr/include/sys/fs/tmpnode.h
1043 file path=usr/include/sys/fs/udf_inode.h
1044 file path=usr/include/sys/fs/udf_volume.h
1045 file path=usr/include/sys/fs/ufs_acl.h
1046 file path=usr/include/sys/fs/ufs_bio.h
1047 file path=usr/include/sys/fs/ufs_filio.h
1048 file path=usr/include/sys/fs/ufs_fs.h
1049 file path=usr/include/sys/fs/ufs_fsdir.h
1050 file path=usr/include/sys/fs/ufs_inode.h
1051 file path=usr/include/sys/fs/ufs_lockfs.h
```

```
1052 file path=usr/include/sys/fs/ufs_log.h
1053 file path=usr/include/sys/fs/ufs_mount.h
1054 file path=usr/include/sys/fs/ufs_panic.h
1055 file path=usr/include/sys/fs/ufs_prot.h
1056 file path=usr/include/sys/fs/ufs_quota.h
1057 file path=usr/include/sys/fs/ufs_snap.h
1058 file path=usr/include/sys/fs/ufs_trans.h
1059 file path=usr/include/sys/fs/zfs.h
1060 file path=usr/include/sys/fs_reparse.h
1061 file path=usr/include/sys/fs_subr.h
1062 file path=usr/include/sys/fsid.h
1063 $(sparc_ONLY)file path=usr/include/sys/fsr.h
1064 file path=usr/include/sys/fss.h
1065 file path=usr/include/sys/fssnap.h
1066 file path=usr/include/sys/fssnap_if.h
1067 file path=usr/include/sys/fsspriocntl.h
1068 file path=usr/include/sys/fstyp.h
1069 file path=usr/include/sys/ftrace.h
1070 file path=usr/include/sys/fx.h
1071 file path=usr/include/sys/fxpriocntl.h
1072 file path=usr/include/sys/gfs.h
1073 file path=usr/include/sys/gld.h
1074 file path=usr/include/sys/gldpriv.h
1075 file path=usr/include/sys/group.h
1076 file path=usr/include/sys/hdio.h
1077 file path=usr/include/sys/hook.h
1078 file path=usr/include/sys/hook_event.h
1079 file path=usr/include/sys/hook_impl.h
1080 file path=usr/include/sys/hotplug/hpcsvc.h
1081 file path=usr/include/sys/hotplug/hpctrl.h
1082 file path=usr/include/sys/hotplug/pci/pcicfg.h
1083 file path=usr/include/sys/hotplug/pci/pcihp.h
1084 file path=usr/include/sys/hwconf.h
1085 $(i386_ONLY)file path=usr/include/sys/hypervisor.h
1086 $(i386_ONLY)file path=usr/include/sys/i8272A.h
1087 file path=usr/include/sys/ia.h
1088 file path=usr/include/sys/iapriocntl.h
1089 file path=usr/include/sys/ib/adapters/hermon/hermon_ioctl.h
1090 file path=usr/include/sys/ib/adapters/mlnx_umap.h
1091 file path=usr/include/sys/ib/adapters/tavor/tavor_ioctl.h
1092 file path=usr/include/sys/ib/clients/ibd/ibd.h
1093 file path=usr/include/sys/ib/clients/of/ofa_solaris.h
1094 file path=usr/include/sys/ib/clients/of/ofed_kernel.h
1095 file path=usr/include/sys/ib/clients/of/rdma/ib_addr.h
1096 file path=usr/include/sys/ib/clients/of/rdma/ib_user_mad.h
1097 file path=usr/include/sys/ib/clients/of/rdma/ib_user_sa.h
1098 file path=usr/include/sys/ib/clients/of/rdma/ib_user_verbs.h
1099 file path=usr/include/sys/ib/clients/of/rdma/ib_verbs.h
1100 file path=usr/include/sys/ib/clients/of/rdma/rdma_cm.h
1101 file path=usr/include/sys/ib/clients/of/rdma/rdma_user_cm.h
1102 file path=usr/include/sys/ib/clients/of/sol_ofs/sol_cma.h
1103 file path=usr/include/sys/ib/clients/of/sol_ofs/sol_ib_cma.h
1104 file path=usr/include/sys/ib/clients/of/sol_ofs/sol_kverb_impl.h
1105 file path=usr/include/sys/ib/clients/of/sol_ofs/sol_ofs_common.h
1106 file path=usr/include/sys/ib/clients/of/sol_ucma/sol_rdma_user_cm.h
1107 file path=usr/include/sys/ib/clients/of/sol_ucma/sol_ucma.h
1108 file path=usr/include/sys/ib/clients/of/sol_umad/sol_umad.h
1109 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs.h
1110 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs2ucma.h
1111 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs_comp.h
1112 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs_event.h
1113 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs_hca.h
1114 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs_qp.h
1115 file path=usr/include/sys/ib/ib_pkt_hdrs.h
1116 file path=usr/include/sys/ib/ib_types.h
1117 file path=usr/include/sys/ib/ibnex/ibnex_devctl.h
```



```
1118 file path=usr/include/sys/ib/ibtl/ibci.h
1119 file path=usr/include/sys/ib/ibtl/ibti.h
1120 file path=usr/include/sys/ib/ibtl/ibti_cm.h
1121 file path=usr/include/sys/ib/ibtl/ibti_common.h
1122 file path=usr/include/sys/ib/ibtl/ibtl_ci_types.h
1123 file path=usr/include/sys/ib/ibtl/ibtl_status.h
1124 file path=usr/include/sys/ib/ibtl/ibtl_types.h
1125 file path=usr/include/sys/ib/ibtl/ibvti.h
1126 file path=usr/include/sys/ib/ibtl/impl/ibtl_util.h
1127 file path=usr/include/sys/ib/mgt/ib_dm_attr.h
1128 file path=usr/include/sys/ib/mgt/ib_mad.h
1129 file path=usr/include/sys/ib/mgt/ibmf/ibmf.h
1130 file path=usr/include/sys/ib/mgt/ibmf/ibmf_msg.h
1131 file path=usr/include/sys/ib/mgt/ibmf/ibmf_saa.h
1132 file path=usr/include/sys/ib/mgt/ibmf/ibmf_utils.h
1133 file path=usr/include/sys/ib/mgt/sa_recs.h
1134 file path=usr/include/sys/ib/mgt/sm_attr.h
1135 file path=usr/include/sys/ibpart.h
1136 file path=usr/include/sys/id32.h
1137 file path=usr/include/sys/id_space.h
1138 file path=usr/include/sys/idmap.h
1139 file path=usr/include/sys/inline.h
1140 file path=usr/include/sys/instance.h
1141 file path=usr/include/sys/int_const.h
1142 file path=usr/include/sys/int_fmtio.h
1143 file path=usr/include/sys/int_limits.h
1144 file path=usr/include/sys/int_types.h
1145 file path=usr/include/sys/inttypes.h
1146 file path=usr/include/sys/ioccom.h
1147 file path=usr/include/sys/ioctl.h
1148 $(i386_ONLY)file path=usr/include/sys/iommulib.h
1149 file path=usr/include/sys/ipc.h
1150 file path=usr/include/sys/ipc_impl.h
1151 file path=usr/include/sys/ipc_rctl.h
1152 file path=usr/include/sys/isa_defs.h
1153 file path=usr/include/sys/iso/signal_iso.h
1154 file path=usr/include/sys/jioctl.h
1155 file path=usr/include/sys/kbd.h
1156 file path=usr/include/sys/kbdreg.h
1157 file path=usr/include/sys/kbio.h
1158 file path=usr/include/sys/kcpc.h
1159 file path=usr/include/sys/kd.h
1160 file path=usr/include/sys/kdi.h
1161 file path=usr/include/sys/kdi_impl.h
1162 file path=usr/include/sys/kdi_machimpl.h
1163 $(i386_ONLY)file path=usr/include/sys/kdi_regs.h
1164 file path=usr/include/sys/kiconv.h
1165 file path=usr/include/sys/kidmap.h
1166 file path=usr/include/sys/klpd.h
1167 file path=usr/include/sys/klwp.h
1168 file path=usr/include/sys/kmem.h
1169 file path=usr/include/sys/kmem_impl.h
1170 file path=usr/include/sys/kobj.h
1171 file path=usr/include/sys/kobj_impl.h
1172 file path=usr/include/sys/ksocket.h
1173 file path=usr/include/sys/kstat.h
1174 file path=usr/include/sys/kstr.h
1175 file path=usr/include/sys/ksyms.h
1176 file path=usr/include/sys/ksynch.h
1177 file path=usr/include/sys/lc_core.h
1178 file path=usr/include/sys/ldterm.h
1179 file path=usr/include/sys/lgrp.h
1180 file path=usr/include/sys/lgrp_user.h
1181 file path=usr/include/sys/link.h
1182 file path=usr/include/sys/list.h
1183 file path=usr/include/sys/list_impl.h
```

```
1184 file path=usr/include/sys/llc1.h
1185 file path=usr/include/sys/loadavg.h
1186 file path=usr/include/sys/localedef.h
1187 file path=usr/include/sys/lock.h
1188 file path=usr/include/sys/lockfs.h
1189 file path=usr/include/sys/lofi.h
1190 file path=usr/include/sys/log.h
1191 file path=usr/include/sys/logindmux.h
1192 file path=usr/include/sys/lvm/md_basic.h
1193 file path=usr/include/sys/lvm/md_convert.h
1194 file path=usr/include/sys/lvm/md_crc.h
1195 file path=usr/include/sys/lvm/md_hotspares.h
1196 file path=usr/include/sys/lvm/md_mddb.h
1197 file path=usr/include/sys/lvm/md_mdiox.h
1198 file path=usr/include/sys/lvm/md_mhdx.h
1199 file path=usr/include/sys/lvm/md_mirror.h
1200 file path=usr/include/sys/lvm/md_mirror_shared.h
1201 file path=usr/include/sys/lvm/md_names.h
1202 file path=usr/include/sys/lvm/md_notify.h
1203 file path=usr/include/sys/lvm/md_raid.h
1204 file path=usr/include/sys/lvm/md_rename.h
1205 file path=usr/include/sys/lvm/md_sp.h
1206 file path=usr/include/sys/lvm/md_stripe.h
1207 file path=usr/include/sys/lvm/md_trans.h
1208 file path=usr/include/sys/lvm/mdio.h
1209 file path=usr/include/sys/lvm/mdmed.h
1210 file path=usr/include/sys/lvm/mdmn_commd.h
1211 file path=usr/include/sys/lvm/mdvar.h
1212 file path=usr/include/sys/lwp.h
1213 file path=usr/include/sys/lwp_timer_impl.h
1214 file path=usr/include/sys/lwp_upimutex_impl.h
1215 file path=usr/include/sys/mac.h
1216 file path=usr/include/sys/mac_ether.h
1217 file path=usr/include/sys/mac_flow.h
1218 file path=usr/include/sys/mac_provider.h
1219 file path=usr/include/sys/machelf.h
1220 file path=usr/include/sys/machlock.h
1221 file path=usr/include/sys/machsig.h
1222 file path=usr/include/sys/machtypes.h
1223 file path=usr/include/sys/map.h
1224 $(i386_ONLY)file path=usr/include/sys/mc.h
1225 $(i386_ONLY)file path=usr/include/sys/mc_amd.h
1226 $(i386_ONLY)file path=usr/include/sys/mc_intel.h
1227 $(i386_ONLY)file path=usr/include/sys/mca_amd.h
1228 $(i386_ONLY)file path=usr/include/sys/mca_x86.h
1229 file path=usr/include/sys/md4.h
1230 file path=usr/include/sys/md5.h
1231 file path=usr/include/sys/md5_consts.h
1232 file path=usr/include/sys/mdi_impldefs.h
1233 file path=usr/include/sys/mem.h
1234 file path=usr/include/sys/mem_config.h
1235 file path=usr/include/sys/memlist.h
1236 file path=usr/include/sys/mhd.h
1237 file path=usr/include/sys/mii.h
1238 file path=usr/include/sys/miiregs.h
1239 file path=usr/include/sys/mkdev.h
1240 file path=usr/include/sys/mman.h
1241 file path=usr/include/sys/mmapobj.h
1242 file path=usr/include/sys/mntent.h
1243 file path=usr/include/sys/mntio.h
1244 file path=usr/include/sys/mnttab.h
1245 file path=usr/include/sys/modctl.h
1246 file path=usr/include/sys/mode.h
1247 file path=usr/include/sys/model.h
1248 file path=usr/include/sys/modhash.h
1249 file path=usr/include/sys/modhash_impl.h
```



```
1250 file path=usr/include/sys/mount.h
1251 file path=usr/include/sys/mouse.h
1252 file path=usr/include/sys/msacct.h
1253 file path=usr/include/sys/msg.h
1254 file path=usr/include/sys/msg_impl.h
1255 file path=usr/include/sys/msio.h
1256 file path=usr/include/sys/msreg.h
1257 file path=usr/include/sys/mtio.h
1258 file path=usr/include/sys/multidata.h
1259 file path=usr/include/sys/mutex.h
1260 $(i386_ONLY)file path=usr/include/sys/mutex_impl.h
1261 file path=usr/include/sys/nbmlock.h
1262 file path=usr/include/sys/ndi_impldefs.h
1263 file path=usr/include/sys/ndifm.h
1264 file path=usr/include/sys/netconfig.h
1265 file path=usr/include/sys/neti.h
1266 file path=usr/include/sys/netstack.h
1267 file path=usr/include/sys/nexusdefs.h
1268 file path=usr/include/sys/note.h
1269 file path=usr/include/sys/nvpair.h
1270 file path=usr/include/sys/nvpair_impl.h
1271 file path=usr/include/sys/objfs.h
1272 file path=usr/include/sys/objfs_impl.h
1273 file path=usr/include/sys/obpdefs.h
1274 file path=usr/include/sys/old_procfs.h
1275 file path=usr/include/sys/open.h
1276 file path=usr/include/sys/openpromio.h
1277 file path=usr/include/sys/panic.h
1278 file path=usr/include/sys/param.h
1279 file path=usr/include/sys/pathconf.h
1280 file path=usr/include/sys/pathname.h
1281 file path=usr/include/sys/pattr.h
1282 file path=usr/include/sys/pbio.h
1283 file path=usr/include/sys/pcb.h
1284 file path=usr/include/sys/pccard.h
1285 file path=usr/include/sys/pci.h
1286 $(i386_ONLY)file path=usr/include/sys/pcic_reg.h
1287 $(i386_ONLY)file path=usr/include/sys/pcic_var.h
1288 file path=usr/include/sys/pcie.h
1289 file path=usr/include/sys/pcmcia.h
1290 file path=usr/include/sys/pcmcia/pcata.h
1291 file path=usr/include/sys/pcmcia/pcser_conf.h
1292 file path=usr/include/sys/pcmcia/pcser_io.h
1293 file path=usr/include/sys/pcmcia/pcser_manuspec.h
1294 file path=usr/include/sys/pcmcia/pcser_reg.h
1295 file path=usr/include/sys/pcmcia/pcser_var.h
1296 file path=usr/include/sys/pctypes.h
1297 file path=usr/include/sys/pfmod.h
1298 file path=usr/include/sys/pg.h
1299 file path=usr/include/sys/pghw.h
1300 file path=usr/include/sys/physmem.h
1301 $(i386_ONLY)file path=usr/include/sys/pic.h
1302 $(i386_ONLY)file path=usr/include/sys/pit.h
1303 file path=usr/include/sys/pkp_hash.h
1304 file path=usr/include/sys/pm.h
1305 $(i386_ONLY)file path=usr/include/sys/pmem.h
1306 file path=usr/include/sys/policy.h
1307 file path=usr/include/sys/poll.h
1308 file path=usr/include/sys/poll_impl.h
1309 file path=usr/include/sys/pool.h
1310 file path=usr/include/sys/pool_impl.h
1311 file path=usr/include/sys/pool_pset.h
1312 file path=usr/include/sys/port.h
1313 file path=usr/include/sys/port_impl.h
1314 file path=usr/include/sys/port_kernel.h
1315 file path=usr/include/sys/ppmio.h
```

```
1316 file path=usr/include/sys/priocntl.h
1317 file path=usr/include/sys/priv.h
1318 file path=usr/include/sys/priv_const.h
1319 file path=usr/include/sys/priv_impl.h
1320 file path=usr/include/sys/priv_names.h
1321 $(i386_ONLY)file path=usr/include/sys/privmregs.h
1322 $(i386_ONLY)file path=usr/include/sys/privregs.h
1323 file path=usr/include/sys/prnio.h
1324 file path=usr/include/sys/proc.h
1325 file path=usr/include/sys/proc/prdata.h
1326 file path=usr/include/sys/processor.h
1327 file path=usr/include/sys/procfs.h
1328 file path=usr/include/sys/procfs_isa.h
1329 file path=usr/include/sys/procset.h
1330 file path=usr/include/sys/project.h
1331 $(i386_ONLY)file path=usr/include/sys/prom_emul.h
1332 $(i386_ONLY)file path=usr/include/sys/prom_isa.h
1333 $(i386_ONLY)file path=usr/include/sys/prom_plat.h
1334 file path=usr/include/sys/promif.h
1335 file path=usr/include/sys/promimpl.h
1336 file path=usr/include/sys/protosw.h
1337 file path=usr/include/sys/prsystm.h
1338 file path=usr/include/sys/pset.h
1339 file path=usr/include/sys/psw.h
1340 $(i386_ONLY)file path=usr/include/sys/pte.h
1341 file path=usr/include/sys/ptem.h
1342 file path=usr/include/sys/ptms.h
1343 file path=usr/include/sys/ptyvar.h
1344 file path=usr/include/sys/queue.h
1345 file path=usr/include/sys/raidioctl.h
1346 file path=usr/include/sys/ramdisk.h
1347 file path=usr/include/sys/random.h
1348 file path=usr/include/sys/rctl.h
1349 file path=usr/include/sys/rctl_impl.h
1350 file path=usr/include/sys/rds.h
1351 file path=usr/include/sys/reboot.h
1352 file path=usr/include/sys/refstr.h
1353 file path=usr/include/sys/refstr_impl.h
1354 file path=usr/include/sys/reg.h
1355 file path=usr/include/sys/regset.h
1356 file path=usr/include/sys/resource.h
1357 file path=usr/include/sys/rlioctl.h
1358 file path=usr/include/sys/rsm/rsm.h
1359 file path=usr/include/sys/rsm/rsm_common.h
1360 file path=usr/include/sys/rsm/rsmapi_common.h
1361 file path=usr/include/sys/rsm/rsmka_path_int.h
1362 file path=usr/include/sys/rsm/rsmndi.h
1363 file path=usr/include/sys/rsm/rsmpi.h
1364 file path=usr/include/sys/rsm/rsmpi_driver.h
1365 file path=usr/include/sys/rt.h
1366 $(i386_ONLY)file path=usr/include/sys/rtc.h
1367 file path=usr/include/sys/rtpriocntl.h
1368 file path=usr/include/sys/rwlock.h
1369 file path=usr/include/sys/rwlock_impl.h
1370 file path=usr/include/sys/rwstlock.h
1371 file path=usr/include/sys/sad.h
1372 $(i386_ONLY)file path=usr/include/sys/sata/sata_defs.h
1373 $(i386_ONLY)file path=usr/include/sys/sata/sata_hba.h
1374 file path=usr/include/sys/schedctl.h
1375 $(sparc_ONLY)file path=usr/include/sys/scsi/adapters/ifpio.h
1376 file path=usr/include/sys/scsi/adapters/scsi_vhci.h
1377 $(sparc_ONLY)file path=usr/include/sys/scsi/adapters/sfvar.h
1378 file path=usr/include/sys/scsi/conf/autoconf.h
1379 file path=usr/include/sys/scsi/conf/device.h
1380 file path=usr/include/sys/scsi/generic/commands.h
1381 file path=usr/include/sys/scsi/generic/dad_mode.h
```



```
1382 file path=usr/include/sys/scsi/generic/inquiry.h
1383 file path=usr/include/sys/scsi/generic/message.h
1384 file path=usr/include/sys/scsi/generic/mode.h
1385 file path=usr/include/sys/scsi/generic/persist.h
1386 file path=usr/include/sys/scsi/generic/sense.h
1387 file path=usr/include/sys/scsi/generic/sff_frames.h
1388 file path=usr/include/sys/scsi/generic/smp_frames.h
1389 file path=usr/include/sys/scsi/generic/status.h
1390 file path=usr/include/sys/scsi/impl/commands.h
1391 file path=usr/include/sys/scsi/impl/inquiry.h
1392 file path=usr/include/sys/scsi/impl/mode.h
1393 file path=usr/include/sys/scsi/impl/scsi_reset_notify.h
1394 file path=usr/include/sys/scsi/impl/scsi_sas.h
1395 file path=usr/include/sys/scsi/impl/sense.h
1396 file path=usr/include/sys/scsi/impl/services.h
1397 file path=usr/include/sys/scsi/impl/smp_transport.h
1398 file path=usr/include/sys/scsi/impl/spc3_types.h
1399 file path=usr/include/sys/scsi/impl/status.h
1400 file path=usr/include/sys/scsi/impl/transport.h
1401 file path=usr/include/sys/scsi/impl/types.h
1402 file path=usr/include/sys/scsi/impl/uscsi.h
1403 file path=usr/include/sys/scsi/impl/usmp.h
1404 file path=usr/include/sys/scsi/scsi.h
1405 file path=usr/include/sys/scsi/scsi_address.h
1406 file path=usr/include/sys/scsi/scsi_ctl.h
1407 file path=usr/include/sys/scsi/scsi_fm.h
1408 file path=usr/include/sys/scsi/scsi_params.h
1409 file path=usr/include/sys/scsi/scsi_pkt.h
1410 file path=usr/include/sys/scsi/scsi_resource.h
1411 file path=usr/include/sys/scsi/scsi_types.h
1412 file path=usr/include/sys/scsi/scsi_watch.h
1413 file path=usr/include/sys/scsi/targets/sddef.h
1414 file path=usr/include/sys/scsi/targets/ses.h
1415 file path=usr/include/sys/scsi/targets/sesio.h
1416 file path=usr/include/sys/scsi/targets/sgendef.h
1417 file path=usr/include/sys/scsi/targets/smp.h
1418 $(sparc_ONLY)file path=usr/include/sys/scsi/targets/ssddef.h
1419 file path=usr/include/sys/scsi/targets/stdef.h
1420 $(i386_ONLY)file path=usr/include/sys/segment.h
1421 $(i386_ONLY)file path=usr/include/sys/segments.h
1422 file path=usr/include/sys/select.h
1423 file path=usr/include/sys/sem.h
1424 file path=usr/include/sys/sem_impl.h
1425 file path=usr/include/sys/sema_impl.h
1426 file path=usr/include/sys/semaphore.h
1427 file path=usr/include/sys/sendfile.h
1428 $(sparc_ONLY)file path=usr/include/sys/ser_async.h
1429 file path=usr/include/sys/ser_sync.h
1430 $(sparc_ONLY)file path=usr/include/sys/ser_zscc.h
1431 file path=usr/include/sys/serializer.h
1432 file path=usr/include/sys/session.h
1433 file path=usr/include/sys/sha1.h
1434 file path=usr/include/sys/sha2.h
1435 file path=usr/include/sys/share.h
1436 file path=usr/include/sys/shm.h
1437 file path=usr/include/sys/shm_impl.h
1438 file path=usr/include/sys/sid.h
1439 file path=usr/include/sys/siginfo.h
1440 file path=usr/include/sys/signal.h
1441 file path=usr/include/sys/skein.h
1442 file path=usr/include/sys/sleepq.h
1443 file path=usr/include/sys/smbios.h
1444 file path=usr/include/sys/smbios_impl.h
1445 file path=usr/include/sys/smedia.h
1446 file path=usr/include/sys/sobject.h
1447 $(sparc_ONLY)file path=usr/include/sys/socal_cq_defs.h
```

```
$(sparc_ONLY)file path=usr/include/sys/socalio.h
$(sparc_ONLY)file path=usr/include/sys/socalmap.h
$(sparc_ONLY)file path=usr/include/sys/socalreg.h
$(sparc_ONLY)file path=usr/include/sys/socalvar.h
file path=usr/include/sys/socket.h
file path=usr/include/sys/socket_impl.h
file path=usr/include/sys/socket_proto.h
file path=usr/include/sys/socketvar.h
file path=usr/include/sys/sockio.h
file path=usr/include/sys/spl.h
file path=usr/include/sys/squeue.h
file path=usr/include/sys/squeue_impl.h
file path=usr/include/sys/sservice.h
file path=usr/include/sys/stack.h
file path=usr/include/sys/stat.h
file path=usr/include/sys/stat_impl.h
file path=usr/include/sys/statfs.h
file path=usr/include/sys/statvfs.h
file path=usr/include/sys/stdbool.h
file path=usr/include/sys/stdint.h
file path=usr/include/sys/stermio.h
file path=usr/include/sys/stream.h
file path=usr/include/sys/strft.h
file path=usr/include/sys/strlog.h
file path=usr/include/sys/strmdep.h
file path=usr/include/sys/stropts.h
file path=usr/include/sys/strredir.h
file path=usr/include/sys/strstat.h
file path=usr/include/sys/strsubr.h
file path=usr/include/sys/strsun.h
file path=usr/include/sys/strtty.h
file path=usr/include/sys/sunddi.h
file path=usr/include/sys/sunldi.h
file path=usr/include/sys/sunldi_impl.h
file path=usr/include/sys/sunmdi.h
file path=usr/include/sys/sunndi.h
file path=usr/include/sys/sunpm.h
file path=usr/include/sys/suntpi.h
file path=usr/include/sys/suntty.h
file path=usr/include/sys/swap.h
file path=usr/include/sys/synch.h
file path=usr/include/sys/syscall.h
file path=usr/include/sys/sysconf.h
file path=usr/include/sys/sysconfig.h
file path=usr/include/sys/sysconfig_impl.h
file path=usr/include/sys/sysdc.h
file path=usr/include/sys/sysdc_impl.h
file path=usr/include/sys/sysevent.h
file path=usr/include/sys/sysevent/ap_driver.h
file path=usr/include/sys/sysevent/dev.h
file path=usr/include/sys/sysevent/domain.h
file path=usr/include/sys/sysevent/dr.h
file path=usr/include/sys/sysevent/env.h
file path=usr/include/sys/sysevent/eventdefs.h
file path=usr/include/sys/sysevent/ipmp.h
file path=usr/include/sys/sysevent/pwrctl.h
file path=usr/include/sys/sysevent/svm.h
file path=usr/include/sys/sysevent/vrrp.h
file path=usr/include/sys/sysevent_impl.h
$(i386_ONLY)file path=usr/include/sys/sysi86.h
file path=usr/include/sys/sysinfo.h
file path=usr/include/sys/syslog.h
file path=usr/include/sys/sysmacros.h
file path=usr/include/sys/systeminfo.h
file path=usr/include/sys/systm.h
file path=usr/include/sys/t_kuser.h
```



```
file path=usr/include/sys/t_lock.h
file path=usr/include/sys/task.h
file path=usr/include/sys/taskq.h
file path=usr/include/sys/taskq_impl.h
file path=usr/include/sys/telioctl.h
file path=usr/include/sys/termio.h
file path=usr/include/sys/termios.h
file path=usr/include/sys/termiox.h
file path=usr/include/sys/thread.h
file path=usr/include/sys/ticlts.h
file path=usr/include/sys/ticots.h
file path=usr/include/sys/ticotsord.h
file path=usr/include/sys/tihdr.h
file path=usr/include/sys/time.h
file path=usr/include/sys/time_impl.h
file path=usr/include/sys/time_std_impl.h
file path=usr/include/sys/timeb.h
file path=usr/include/sys/timer.h
file path=usr/include/sys/times.h
file path=usr/include/sys/timex.h
file path=usr/include/sys/timod.h
file path=usr/include/sys/tirdwr.h
file path=usr/include/sys/tiuser.h
file path=usr/include/sys/tl.h
file path=usr/include/sys/tnf.h
file path=usr/include/sys/tnf_com.h
file path=usr/include/sys/tnf_probe.h
file path=usr/include/sys/tnf_writer.h
file path=usr/include/sys/todio.h
file path=usr/include/sys/tpicommon.h
file path=usr/include/sys/trap.h
$(i386_ONLY)file path=usr/include/sys/traptrace.h
file path=usr/include/sys/ts.h
file path=usr/include/sys/tsol/label.h
file path=usr/include/sys/tsol/label_macro.h
file path=usr/include/sys/tsol/priv.h
file path=usr/include/sys/tsol/tndb.h
file path=usr/include/sys/tsol/tsyscall.h
file path=usr/include/sys/tspriocntl.h
$(i386_ONLY)file path=usr/include/sys/tss.h
file path=usr/include/sys/ttcompat.h
file path=usr/include/sys/ttold.h
file path=usr/include/sys/tty.h
file path=usr/include/sys/ttychars.h
file path=usr/include/sys/ttydev.h
$(sparc_ONLY)file path=usr/include/sys/ttymux.h
$(sparc_ONLY)file path=usr/include/sys/ttymuxuser.h
file path=usr/include/sys/tuneable.h
file path=usr/include/sys/turnstile.h
file path=usr/include/sys/types.h
file path=usr/include/sys/types32.h
file path=usr/include/sys/tzfile.h
file path=usr/include/sys/u8_textprep.h
file path=usr/include/sys/uadmin.h
$(i386_ONLY)file path=usr/include/sys/ucode.h
file path=usr/include/sys/ucontext.h
file path=usr/include/sys/uio.h
file path=usr/include/sys/ulimit.h
file path=usr/include/sys/un.h
file path=usr/include/sys/unistd.h
file path=usr/include/sys/user.h
file path=usr/include/sys/ustat.h
file path=usr/include/sys/utime.h
file path=usr/include/sys/utrap.h
file path=usr/include/sys/utsname.h
file path=usr/include/sys/utssys.h
```

```
1580 file path=usr/include/sys/uuid.h
1581 file path=usr/include/sys/va_impl.h
1582 file path=usr/include/sys/va_list.h
1583 file path=usr/include/sys/var.h
1584 file path=usr/include/sys/varargs.h
1585 file path=usr/include/sys/vfs.h
1586 file path=usr/include/sys/vfs_opreg.h
1587 file path=usr/include/sys/vfstab.h
1588 file path=usr/include/sys/videodev2.h
1589 file path=usr/include/sys/visual_io.h
1590 file path=usr/include/sys/vm.h
1591 file path=usr/include/sys/vm_usage.h
1592 file path=usr/include/sys/vmem.h
1593 file path=usr/include/sys/vmem_impl.h
1594 file path=usr/include/sys/vmem_impl_user.h
1595 file path=usr/include/sys/vmparam.h
1596 file path=usr/include/sys/vmsystm.h
1597 file path=usr/include/sys/vnode.h
1598 file path=usr/include/sys/vt.h
1599 file path=usr/include/sys/vtdaemon.h
1600 file path=usr/include/sys/vtoc.h
1601 file path=usr/include/sys/vtrace.h
1602 file path=usr/include/sys/vuid_event.h
1603 file path=usr/include/sys/vuid_queue.h
1604 file path=usr/include/sys/vuid_state.h
1605 file path=usr/include/sys/vuid_store.h
1606 file path=usr/include/sys/vuid_wheel.h
1607 file path=usr/include/sys/wait.h
1608 file path=usr/include/sys/waitq.h
1609 file path=usr/include/sys/watchpoint.h
1610 $(i386_ONLY)file path=usr/include/sys/x86_archext.h
1611 $(i386_ONLY)file path=usr/include/sys/xen_errno.h
1612 file path=usr/include/sys/xti_inet.h
1613 file path=usr/include/sys/xti_osi.h
1614 file path=usr/include/sys/xti_xtiopt.h
1615 file path=usr/include/sys/zcons.h
1616 file path=usr/include/sys/zmod.h
1617 file path=usr/include/sys/zone.h
1618 $(sparc_ONLY)file path=usr/include/sys/zsdev.h
1619 file path=usr/include/sysexits.h
1620 file path=usr/include/syslog.h
1621 file path=usr/include/tar.h
1622 file path=usr/include/tcpd.h
1623 file path=usr/include/term.h
1624 file path=usr/include/termcap.h
1625 file path=usr/include/termio.h
1626 file path=usr/include/termios.h
1627 file path=usr/include/thread.h
1628 file path=usr/include/thread_db.h
1629 file path=usr/include/time.h
1630 file path=usr/include/tiuser.h
1631 file path=usr/include/tsol/label.h
1632 file path=usr/include/tzfile.h
1633 file path=usr/include/ucontext.h
1634 file path=usr/include/ucred.h
1635 file path=usr/include/uid_stp.h
1636 file path=usr/include/ulimit.h
1637 file path=usr/include/umem.h
1638 file path=usr/include/umem_impl.h
1639 file path=usr/include/unctrl.h
1640 file path=usr/include/unistd.h
1641 file path=usr/include/user_attr.h
1642 file path=usr/include/userdefs.h
1643 file path=usr/include/ustat.h
1644 file path=usr/include/utility.h
1645 file path=usr/include/utime.h
```



```
1646 file path=usr/include/utmp.h
1647 file path=usr/include/utmpx.h
1648 file path=usr/include/uuid/uuid.h
1649 $(sparc_ONLY)file path=usr/include/v7/sys/machpcb.h
1650 $(sparc_ONLY)file path=usr/include/v7/sys/machtrap.h
1651 $(sparc_ONLY)file path=usr/include/v7/sys/mutex_impl.h
1652 $(sparc_ONLY)file path=usr/include/v7/sys/privregs.h
1653 $(sparc_ONLY)file path=usr/include/v7/sys/prom_isa.h
1654 $(sparc_ONLY)file path=usr/include/v7/sys/psr.h
1655 $(sparc_ONLY)file path=usr/include/v7/sys/traptrace.h
1656 $(sparc_ONLY)file path=usr/include/v9/sys/asi.h
1657 $(sparc_ONLY)file path=usr/include/v9/sys/machpcb.h
1658 $(sparc_ONLY)file path=usr/include/v9/sys/machtrap.h
1659 $(sparc_ONLY)file path=usr/include/v9/sys/membar.h
1660 $(sparc_ONLY)file path=usr/include/v9/sys/mutex_impl.h
1661 $(sparc_ONLY)file path=usr/include/v9/sys/privregs.h
1662 $(sparc_ONLY)file path=usr/include/v9/sys/prom_isa.h
1663 $(sparc_ONLY)file path=usr/include/v9/sys/psr_compat.h
1664 $(sparc_ONLY)file path=usr/include/v9/sys/vis_simulator.h
1665 file path=usr/include/valtools.h
1666 file path=usr/include/values.h
1667 file path=usr/include/varargs.h
1668 file path=usr/include/vm/anon.h
1669 file path=usr/include/vm/as.h
1670 file path=usr/include/vm/faultcode.h
1671 file path=usr/include/vm/hat.h
1672 file path=usr/include/vm/kpm.h
1673 file path=usr/include/vm/page.h
1674 file path=usr/include/vm/pvn.h
1675 file path=usr/include/vm/rm.h
1676 file path=usr/include/vm/seg.h
1677 file path=usr/include/vm/seg_dev.h
1678 file path=usr/include/vm/seg_enum.h
1679 file path=usr/include/vm/seg_kmem.h
1680 file path=usr/include/vm/seg_kp.h
1681 file path=usr/include/vm/seg_kpm.h
1682 file path=usr/include/vm/seg_map.h
1683 file path=usr/include/vm/seg_spt.h
1684 file path=usr/include/vm/seg_vn.h
1685 file path=usr/include/vm/vpage.h
1686 file path=usr/include/vm/vpm.h
1687 file path=usr/include/volmgt.h
1688 file path=usr/include/wait.h
1689 file path=usr/include/wchar.h
1690 file path=usr/include/wchar_impl.h
1691 file path=usr/include/wctype.h
1692 file path=usr/include/widec.h
1693 file path=usr/include/wordexp.h
1694 file path=usr/include/xti.h
1695 file path=usr/include/xti_inet.h
1696 file path=usr/include/zone.h
1697 file path=usr/include/zonestat.h
1698 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/acpidev.h
1699 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/amd_iommu.h
1700 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/asm_misc.h
1701 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/clock.h
1702 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/cram.h
1703 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/ddi_subrdefs.h
1704 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/debug_info.h
1705 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/fastboot.h
1706 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/mach_mmu.h
1707 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machclock.h
1708 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machcpuvar.h
1709 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machparam.h
1710 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machprivregs.h
1711 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machsystm.h
```

```
1712 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machthread.h
1713 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/memnode.h
1714 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/pc_mmu.h
1715 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/psm.h
1716 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/psm_defs.h
1717 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/psm_modctl.h
1718 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/psm_types.h
1719 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/rm_platter.h
1720 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/sbd_ioctl.h
1721 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/smp_impldefs.h
1722 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/vm_machparam.h
1723 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/x_call.h
1724 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/xc_levels.h
1725 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/xsvc.h
1726 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/hat_i86.h
1727 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/hat_pte.h
1728 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/hment.h
1729 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/htable.h
1730 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/kboot_mmu.h
1731 $(i386_ONLY)file path=usr/platform/i86xpv/include/sys/balloon.h
1732 $(i386_ONLY)file path=usr/platform/i86xpv/include/sys/machprivregs.h
1733 $(i386_ONLY)file path=usr/platform/i86xpv/include/sys/xen_mmu.h
1734 $(i386_ONLY)file path=usr/platform/i86xpv/include/sys/xpv_impl.h
1735 $(i386_ONLY)file path=usr/platform/i86xpv/include/vm/seg_mf.h
1736 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/ac.h
1737 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/async.h
1738 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cheetahregs.h
1739 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cherrystone.h
1740 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/clock.h
1741 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cmp.h
1742 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cpc_ultra.h
1743 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cpr_impl.h
1744 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cpu_impl.h
1745 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cpu_sgnblk_defs.h
1746 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cvc.h
1747 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/daktari.h
1748 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/ddi_subrdefs.h
1749 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/dvma.h
1750 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/ecc_kstat.h
1751 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/eeprom.h
1752 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/envctrl.h
1753 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/envctrl_gen.h
1754 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/envctrl_ue250.h
1755 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/envctrl_ue450.h
1756 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/environ.h
1757 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/errclassify.h
1758 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/fhc.h
1759 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/gpio_87317.h
1760 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/hpc3130_events.h
1761 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/clients/hpc3130.h
1762 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/clients/i2c_client.h
1763 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/clients/lm75.h
1764 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/clients/max1617.h
1765 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/clients/pcf8591.h
1766 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/clients/ssc050.h
1767 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/misc/i2c_svc.h
1768 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/idprom.h
1769 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/intr.h
1770 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/intreg.h
1771 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/iocache.h
1772 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/iommu.h
1773 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/ivintr.h
1774 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/lom_io.h
1775 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machasi.h
1776 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machclock.h
1777 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machcpuvar.h
```



```
1778 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machparam.h
1779 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machsystm.h
1780 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machthread.h
1781 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/mem_cache.h
1782 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/memlist_plat.h
1783 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/memnode.h
1784 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/mmu.h
1785 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/nexusdebug.h
1786 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/opl_hwdesc.h
1787 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/opl_module.h
1788 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/prom_debug.h
1789 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/prom_plat.h
1790 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/pte.h
1791 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sbd_ioctl.h
1792 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/scb.h
1793 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/scsb_led.h
1794 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/simmstat.h
1795 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/spitregs.h
1796 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sram.h
1797 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/starfire.h
1798 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sun4asi.h
1799 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sysctrl.h
1800 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sysioerr.h
1801 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sysiosbus.h
1802 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/tod.h
1803 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/todmostek.h
1804 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/trapstat.h
1805 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/traptrace.h
1806 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/vis.h
1807 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/vm_machparam.h
1808 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/x_call.h
1809 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/xc_impl.h
1810 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/zsmach.h
1811 $(sparc_ONLY)file path=usr/platform/sun4u/include/vm/hat_sfmmu.h
1812 $(sparc_ONLY)file path=usr/platform/sun4u/include/vm/mach_sfmmu.h
1813 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/clock.h
1814 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/cmp.h
1815 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/cpc_ultra.h
1816 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/cpu_sgnblk_defs.h
1817 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ddi_subrdefs.h
1818 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ds_pri.h
1819 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ds_snmp.h
1820 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/dvma.h
1821 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/eeprom.h
1822 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/fcode.h
1823 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/hsvc.h
1824 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/hypervisor_api.h
1825 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/idprom.h
1826 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/intr.h
1827 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/intreg.h
1828 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ivintr.h
1829 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machasi.h
1830 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machclock.h
1831 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machcpuvar.h
1832 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machintreg.h
1833 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machparam.h
1834 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machsystm.h
1835 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machthread.h
1836 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/memlist_plat.h
1837 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/memnode.h
1838 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/mmu.h
1839 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/nexusdebug.h
1840 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/niagaraasi.h
1841 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/niagararegs.h
1842 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ntwdt.h
1843 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/pri.h
```

```
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/prom_debug.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/prom_plat.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/pte.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/qcn.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/scb.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/soft_state.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/sun4asi.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/tod.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/trapstat.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/traptrace.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/vis.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/vm_machparam.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/x_call.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/xc_impl.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/sys/zsmach.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/vm/hat_sfmmu.h
$(sparc_ONLY)file path=usr/platform/sun4v/include/vm/mach_sfmmu.h
file path=usr/share/man/man3head/acct.h.3head
file path=usr/share/man/man3head/aio.h.3head
file path=usr/share/man/man3head/ar.h.3head
file path=usr/share/man/man3head/archives.h.3head
file path=usr/share/man/man3head/assert.h.3head
file path=usr/share/man/man3head/complex.h.3head
file path=usr/share/man/man3head/cpio.h.3head
file path=usr/share/man/man3head/dirent.h.3head
file path=usr/share/man/man3head/errno.h.3head
file path=usr/share/man/man3head/fcntl.h.3head
file path=usr/share/man/man3head/fenv.h.3head
file path=usr/share/man/man3head/float.h.3head
file path=usr/share/man/man3head/floatingpoint.h.3head
file path=usr/share/man/man3head/fmtmsg.h.3head
file path=usr/share/man/man3head/fnmatch.h.3head
file path=usr/share/man/man3head/ftw.h.3head
file path=usr/share/man/man3head/glob.h.3head
file path=usr/share/man/man3head/grp.h.3head
file path=usr/share/man/man3head/iconv.h.3head
file path=usr/share/man/man3head/if.h.3head
file path=usr/share/man/man3head/in.h.3head
file path=usr/share/man/man3head/inet.h.3head
file path=usr/share/man/man3head/inttypes.h.3head
file path=usr/share/man/man3head/ipc.h.3head
file path=usr/share/man/man3head/iso646.h.3head
file path=usr/share/man/man3head/langinfo.h.3head
file path=usr/share/man/man3head/libgen.h.3head
file path=usr/share/man/man3head/libintl.h.3head
file path=usr/share/man/man3head/limits.h.3head
file path=usr/share/man/man3head/locale.h.3head
file path=usr/share/man/man3head/math.h.3head
file path=usr/share/man/man3head/mman.h.3head
file path=usr/share/man/man3head/monetary.h.3head
file path=usr/share/man/man3head/mqueue.h.3head
file path=usr/share/man/man3head/msg.h.3head
file path=usr/share/man/man3head/ndbm.h.3head
file path=usr/share/man/man3head/netdb.h.3head
file path=usr/share/man/man3head/nl_types.h.3head
file path=usr/share/man/man3head/poll.h.3head
file path=usr/share/man/man3head/pthread.h.3head
file path=usr/share/man/man3head/pwd.h.3head
file path=usr/share/man/man3head/regex.h.3head
file path=usr/share/man/man3head/resource.h.3head
file path=usr/share/man/man3head/sched.h.3head
file path=usr/share/man/man3head/search.h.3head
file path=usr/share/man/man3head/select.h.3head
file path=usr/share/man/man3head/sem.h.3head
file path=usr/share/man/man3head/semaphore.h.3head
file path=usr/share/man/man3head/setjmp.h.3head
```



```
file path=usr/share/man/man3head/shm.h.3head
file path=usr/share/man/man3head/siginfo.h.3head
file path=usr/share/man/man3head/signal.h.3head
file path=usr/share/man/man3head/socket.h.3head
file path=usr/share/man/man3head/spawn.h.3head
file path=usr/share/man/man3head/stat.h.3head
file path=usr/share/man/man3head/statvfs.h.3head
file path=usr/share/man/man3head/stdbool.h.3head
file path=usr/share/man/man3head/stddef.h.3head
file path=usr/share/man/man3head/stdint.h.3head
file path=usr/share/man/man3head/stdio.h.3head
file path=usr/share/man/man3head/stdlib.h.3head
file path=usr/share/man/man3head/string.h.3head
file path=usr/share/man/man3head/strings.h.3head
file path=usr/share/man/man3head/stropts.h.3head
file path=usr/share/man/man3head/syslog.h.3head
file path=usr/share/man/man3head/tar.h.3head
file path=usr/share/man/man3head/tcp.h.3head
file path=usr/share/man/man3head/termios.h.3head
file path=usr/share/man/man3head/tgmath.h.3head
file path=usr/share/man/man3head/time.h.3head
file path=usr/share/man/man3head/timeb.h.3head
file path=usr/share/man/man3head/times.h.3head
file path=usr/share/man/man3head/types.h.3head
file path=usr/share/man/man3head/types32.h.3head
file path=usr/share/man/man3head/ucontext.h.3head
file path=usr/share/man/man3head/uio.h.3head
file path=usr/share/man/man3head/ulimit.h.3head
file path=usr/share/man/man3head/un.h.3head
file path=usr/share/man/man3head/unistd.h.3head
file path=usr/share/man/man3head/utime.h.3head
file path=usr/share/man/man3head/utmpx.h.3head
file path=usr/share/man/man3head/utsname.h.3head
file path=usr/share/man/man3head/values.h.3head
file path=usr/share/man/man3head/wait.h.3head
file path=usr/share/man/man3head/wchar.h.3head
file path=usr/share/man/man3head/wctype.h.3head
file path=usr/share/man/man3head/wordexp.h.3head
file path=usr/share/man/man4/note.4
file path=usr/share/man/man5/prof.5
file path=usr/share/man/man7i/cdio.7i
file path=usr/share/man/man7i/dkio.7i
file path=usr/share/man/man7i/fbio.7i
file path=usr/share/man/man7i/fdio.7i
file path=usr/share/man/man7i/hdio.7i
file path=usr/share/man/man7i/iec61883.7i
file path=usr/share/man/man7i/mhd.7i
file path=usr/share/man/man7i/mtio.7i
file path=usr/share/man/man7i/prnio.7i
file path=usr/share/man/man7i/quotactl.7i
file path=usr/share/man/man7i/sesio.7i
file path=usr/share/man/man7i/sockio.7i
file path=usr/share/man/man7i/streamio.7i
file path=usr/share/man/man7i/termio.7i
file path=usr/share/man/man7i/termiox.7i
file path=usr/share/man/man7i/uscsi.7i
file path=usr/share/man/man7i/visual_io.7i
file path=usr/share/man/man7i/vt.7i
file path=usr/xpg4/include/curses.h
file path=usr/xpg4/include/term.h
file path=usr/xpg4/include/unctrl.h
legacy pkg=SUNWhea \
    desc="SunOS C/C++ header files for general development of software" \
    name="SunOS Header Files"
license cr_Sun license=cr_Sun
license lic_CDDL license=lic_CDDL
```

```
1976 license license_in_headers license=license_in_headers
1977 license usr/src/lib/pkcs11/include/THIRDPARTYLICENSE \
1978     license=usr/src/lib/pkcs11/include/THIRDPARTYLICENSE
1979 link path=usr/include/iso/assert_iso.h target=../assert.h
1980 link path=usr/include/iso/errno_iso.h target=../errno.h
1981 link path=usr/include/iso/float_iso.h target=../float.h
1982 link path=usr/include/iso/iso646_iso.h target=../iso646.h
1983 $(sparc_ONLY)link path=usr/platform/SUNW,A70/include target=../sun4u/include
1984 $(sparc_ONLY)link path=usr/platform/SUNW,Netra-T12/include \
1985     target=../sun4u/include
1986 $(sparc_ONLY)link path=usr/platform/SUNW,Netra-T4/include \
1987     target=../sun4u/include
1988 $(sparc_ONLY)link path=usr/platform/SUNW,SPARC-Enterprise/include \
1989     target=../sun4u/include
1990 $(sparc_ONLY)link path=usr/platform/SUNW,Serverblade1/include \
1991     target=../sun4u/include
1992 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Blade-100/include \
1993     target=../sun4u/include
1994 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Blade-1000/include \
1995     target=../sun4u/include
1996 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Blade-1500/include \
1997     target=../sun4u/include
1998 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Blade-2500/include \
1999     target=../sun4u/include
2000 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-15000/include \
2001     target=../sun4u/include
2002 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-280R/include \
2003     target=../sun4u/include
2004 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-480R/include \
2005     target=../sun4u/include
2006 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-880/include \
2007     target=../sun4u/include
2008 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V215/include \
2009     target=../sun4u/include
2010 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V240/include \
2011     target=../sun4u/include
2012 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V250/include \
2013     target=../sun4u/include
2014 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V440/include \
2015     target=../sun4u/include
2016 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V445/include \
2017     target=../sun4u/include
2018 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V490/include \
2019     target=../sun4u/include
2020 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V890/include \
2021     target=../sun4u/include
2022 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire/include \
2023     target=../sun4u/include
2024 $(sparc_ONLY)link path=usr/platform/SUNW,Ultra-2/include \
2025     target=../sun4u/include
2026 $(sparc_ONLY)link path=usr/platform/SUNW,Ultra-250/include \
2027     target=../sun4u/include
2028 $(sparc_ONLY)link path=usr/platform/SUNW,Ultra-4/include \
2029     target=../sun4u/include
2030 $(sparc_ONLY)link path=usr/platform/SUNW,Ultra-Enterprise-10000/include \
2031     target=../sun4u/include
2032 $(sparc_ONLY)link path=usr/platform/SUNW,Ultra-Enterprise/include \
2033     target=../sun4u/include
2034 $(sparc_ONLY)link path=usr/platform/SUNW,UltraSPARC-IIe-NetraCT-40/include \
2035     target=../sun4u/include
2036 $(sparc_ONLY)link path=usr/platform/SUNW,UltraSPARC-IIe-NetraCT-60/include \
2037     target=../sun4u/include
2038 $(sparc_ONLY)link path=usr/platform/SUNW,UltraSPARC-IIi-Netract/include \
2039     target=../sun4u/include
2040 link path=usr/share/man/man3head/acct.3head target=acct.h.3head
2041 link path=usr/share/man/man3head/aio.3head target=aio.h.3head
```



```
2042 link path=usr/share/man/man3head/ar.3head target=ar.h.3head
2043 link path=usr/share/man/man3head/archives.3head target=archives.h.3head
2044 link path=usr/share/man/man3head/assert.3head target=assert.h.3head
2045 link path=usr/share/man/man3head/complex.3head target=complex.h.3head
2046 link path=usr/share/man/man3head/cpio.3head target=cpio.h.3head
2047 link path=usr/share/man/man3head/dirent.3head target=dirent.h.3head
2048 link path=usr/share/man/man3head/errno.3head target=errno.h.3head
2049 link path=usr/share/man/man3head/fcntl.3head target=fcntl.h.3head
2050 link path=usr/share/man/man3head/fenv.3head target=fenv.h.3head
2051 link path=usr/share/man/man3head/float.3head target=float.h.3head
2052 link path=usr/share/man/man3head/floatingpoint.3head \
2053     target=floatingpoint.h.3head
2054 link path=usr/share/man/man3head/fmtmsg.3head target=fmtmsg.h.3head
2055 link path=usr/share/man/man3head/fnmatch.3head target=fnmatch.h.3head
2056 link path=usr/share/man/man3head/ftw.3head target=ftw.h.3head
2057 link path=usr/share/man/man3head/glob.3head target=glob.h.3head
2058 link path=usr/share/man/man3head/grp.3head target=grp.h.3head
2059 link path=usr/share/man/man3head/iconv.3head target=iconv.h.3head
2060 link path=usr/share/man/man3head/if.3head target=if.h.3head
2061 link path=usr/share/man/man3head/in.3head target=in.h.3head
2062 link path=usr/share/man/man3head/inet.3head target=inet.h.3head
2063 link path=usr/share/man/man3head/inttypes.3head target=inttypes.h.3head
2064 link path=usr/share/man/man3head/ipc.3head target=ipc.h.3head
2065 link path=usr/share/man/man3head/iso646.3head target=iso646.h.3head
2066 link path=usr/share/man/man3head/langinfo.3head target=langinfo.h.3head
2067 link path=usr/share/man/man3head/libgen.3head target=libgen.h.3head
2068 link path=usr/share/man/man3head/libintl.3head target=libintl.h.3head
2069 link path=usr/share/man/man3head/limits.3head target=limits.h.3head
2070 link path=usr/share/man/man3head/locale.3head target=locale.h.3head
2071 link path=usr/share/man/man3head/math.3head target=math.h.3head
2072 link path=usr/share/man/man3head/mman.3head target=mman.h.3head
2073 link path=usr/share/man/man3head/monetary.3head target=monetary.h.3head
2074 link path=usr/share/man/man3head/mqueue.3head target=mqueue.h.3head
2075 link path=usr/share/man/man3head/msg.3head target=msg.h.3head
2076 link path=usr/share/man/man3head/ndbm.3head target=ndbm.h.3head
2077 link path=usr/share/man/man3head/netdb.3head target=netdb.h.3head
2078 link path=usr/share/man/man3head/nl_types.3head target=nl_types.h.3head
2079 link path=usr/share/man/man3head/poll.3head target=poll.h.3head
2080 link path=usr/share/man/man3head/pthread.3head target=pthread.h.3head
2081 link path=usr/share/man/man3head/pwd.3head target=pwd.h.3head
2082 link path=usr/share/man/man3head/regex.3head target=regex.h.3head
2083 link path=usr/share/man/man3head/resource.3head target=resource.h.3head
2084 link path=usr/share/man/man3head/sched.3head target=sched.h.3head
2085 link path=usr/share/man/man3head/search.3head target=search.h.3head
2086 link path=usr/share/man/man3head/select.3head target=select.h.3head
2087 link path=usr/share/man/man3head/sem.3head target=sem.h.3head
2088 link path=usr/share/man/man3head/semaphore.3head target=semaphore.h.3head
2089 link path=usr/share/man/man3head/setjmp.3head target=setjmp.h.3head
2090 link path=usr/share/man/man3head/shm.3head target=shm.h.3head
2091 link path=usr/share/man/man3head/siginfo.3head target=siginfo.h.3head
2092 link path=usr/share/man/man3head/signal.3head target=signal.h.3head
2093 link path=usr/share/man/man3head/socket.3head target=socket.h.3head
2094 link path=usr/share/man/man3head/spawn.3head target=spawn.h.3head
2095 link path=usr/share/man/man3head/stat.3head target=stat.h.3head
2096 link path=usr/share/man/man3head/statvfs.3head target=statvfs.h.3head
2097 link path=usr/share/man/man3head/stdbool.3head target=stdbool.h.3head
2098 link path=usr/share/man/man3head/stddef.3head target=stddef.h.3head
2099 link path=usr/share/man/man3head/stdint.3head target=stdint.h.3head
2100 link path=usr/share/man/man3head/stdio.3head target=stdio.h.3head
2101 link path=usr/share/man/man3head/stdlib.3head target=stdlib.h.3head
2102 link path=usr/share/man/man3head/string.3head target=string.h.3head
2103 link path=usr/share/man/man3head/strings.3head target=strings.h.3head
2104 link path=usr/share/man/man3head/stropts.3head target=stropts.h.3head
2105 link path=usr/share/man/man3head/syslog.3head target=syslog.h.3head
2106 link path=usr/share/man/man3head/tar.3head target=tar.h.3head
2107 link path=usr/share/man/man3head/tcp.3head target=tcp.h.3head
```

```
2108 link path=usr/share/man/man3head/termios.3head target=termios.h.3head
2109 link path=usr/share/man/man3head/tgmath.3head target=tgmath.h.3head
2110 link path=usr/share/man/man3head/time.3head target=time.h.3head
2111 link path=usr/share/man/man3head/timeb.3head target=timeb.h.3head
2112 link path=usr/share/man/man3head/times.3head target=times.h.3head
2113 link path=usr/share/man/man3head/types.3head target=types.h.3head
2114 link path=usr/share/man/man3head/types32.3head target=types32.h.3head
2115 link path=usr/share/man/man3head/ucontext.3head target=ucontext.h.3head
2116 link path=usr/share/man/man3head/uio.3head target=uio.h.3head
2117 link path=usr/share/man/man3head/ulimit.3head target=ulimit.h.3head
2118 link path=usr/share/man/man3head/un.3head target=un.h.3head
2119 link path=usr/share/man/man3head/unistd.3head target=unistd.h.3head
2120 link path=usr/share/man/man3head/utime.3head target=utime.h.3head
2121 link path=usr/share/man/man3head/utmpx.3head target=utmpx.h.3head
2122 link path=usr/share/man/man3head/utsname.3head target=utsname.h.3head
2123 link path=usr/share/man/man3head/values.3head target=values.h.3head
2124 link path=usr/share/man/man3head/wait.3head target=wait.h.3head
2125 link path=usr/share/man/man3head/wchar.3head target=wchar.h.3head
2126 link path=usr/share/man/man3head/wctype.3head target=wctype.h.3head
2127 link path=usr/share/man/man3head/wordexp.3head target=wordexp.h.3head
2128 $(i386_ONLY)link path=usr/share/src/uts/i86pc/sys \
2129     target=../../../../platform/i86pc/include/sys
2130 $(i386_ONLY)link path=usr/share/src/uts/i86pc/vm \
2131     target=../../../../platform/i86pc/include/vm
2132 $(i386_ONLY)link path=usr/share/src/uts/i86xpv/sys \
2133     target=../../../../platform/i86xpv/include/sys
2134 $(i386_ONLY)link path=usr/share/src/uts/i86xpv/vm \
2135     target=../../../../platform/i86xpv/include/vm
2136 $(sparc_ONLY)link path=usr/share/src/uts/sun4u/sys \
2137     target=../../../../platform/sun4u/include/sys
2138 $(sparc_ONLY)link path=usr/share/src/uts/sun4u/vm \
2139     target=../../../../platform/sun4u/include/vm
2140 $(sparc_ONLY)link path=usr/share/src/uts/sun4v/sys \
2141     target=../../../../platform/sun4v/include/sys
2142 $(sparc_ONLY)link path=usr/share/src/uts/sun4v/vm \
2143     target=../../../../platform/sun4v/include/vm
```

```
**********************************************************
   45572 Mon Oct  7 19:17:17 2013
new/usr/src/pkg/manifests/system-kernel.mf
4185 New hash algorithm support
**********************************************************
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #

  22 #
  23 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
  24 # Copyright 2013 Saso Kiselkov. All rights reserved.
  25 #

  27 #
  28 # The default for payload-bearing actions in this package is to appear in the
  29 # global zone only.  See the include file for greater detail, as well as
  30 # information about overriding the defaults.
  31 #
  32 <include global_zone_only_component>
  33 <include system-kernel.man1m.inc>
  34 <include system-kernel.man2.inc>
  35 <include system-kernel.man4.inc>
  36 <include system-kernel.man5.inc>
  37 <include system-kernel.man7.inc>
  38 <include system-kernel.man7d.inc>
  39 <include system-kernel.man7fs.inc>
  40 <include system-kernel.man7m.inc>
  41 <include system-kernel.man7p.inc>
  42 <include system-kernel.man9.inc>
  43 <include system-kernel.man9e.inc>
  44 <include system-kernel.man9f.inc>
  45 <include system-kernel.man9p.inc>
  46 <include system-kernel.man9s.inc>
  47 set name=pkg.fmri value=pkg:/system/kernel@$(PKGVERS)
  48 set name=pkg.description \
  49     value="core kernel software for a specific instruction-set architecture"
  50 set name=pkg.summary value="Core Solaris Kernel"
  51 set name=info.classification value=org.opensolaris.category.2008:System/Core
  52 set name=variant.arch value=$(ARCH)
  53 dir path=boot group=sys
  54 $(i386_ONLY)dir path=boot/acpi group=sys
  55 $(i386_ONLY)dir path=boot/acpi/tables group=sys
  56 dir path=boot/solaris group=sys
  57 dir path=boot/solaris/bin group=sys
  58 dir path=etc group=sys
  59 dir path=etc/crypto group=sys
  60 dir path=etc/sock2path.d group=sys
  61 dir path=kernel group=sys
```



```
  62 $(i386_ONLY)dir path=kernel/$(ARCH64) group=sys
  63 dir path=kernel/crypto group=sys
  64 dir path=kernel/crypto/$(ARCH64) group=sys
  65 dir path=kernel/dacf group=sys
  66 dir path=kernel/dacf/$(ARCH64) group=sys
  67 dir path=kernel/drv group=sys
  68 dir path=kernel/drv/$(ARCH64) group=sys
  69 dir path=kernel/exec group=sys
  70 dir path=kernel/exec/$(ARCH64) group=sys
  71 dir path=kernel/fs group=sys
  72 dir path=kernel/fs/$(ARCH64) group=sys
  73 dir path=kernel/ipp group=sys
  74 dir path=kernel/ipp/$(ARCH64) group=sys
  75 dir path=kernel/kiconv group=sys
  76 dir path=kernel/kiconv/$(ARCH64) group=sys
  77 dir path=kernel/mac group=sys
  78 dir path=kernel/mac/$(ARCH64) group=sys
  79 dir path=kernel/misc group=sys
  80 dir path=kernel/misc/$(ARCH64) group=sys
  81 dir path=kernel/misc/scsi_vhci group=sys
  82 dir path=kernel/misc/scsi_vhci/$(ARCH64) group=sys
  83 dir path=kernel/sched group=sys
  84 dir path=kernel/sched/$(ARCH64) group=sys
  85 dir path=kernel/socketmod group=sys
  86 dir path=kernel/socketmod/$(ARCH64) group=sys
  87 dir path=kernel/strmod group=sys
  88 dir path=kernel/strmod/$(ARCH64) group=sys
  89 dir path=kernel/sys group=sys
  90 dir path=kernel/sys/$(ARCH64) group=sys
  91 dir path=lib
  92 dir path=lib/svc
  93 dir path=lib/svc/manifest group=sys
  94 dir path=lib/svc/manifest/system group=sys
  95 dir path=lib/svc/method
  96 dir path=usr/share/man
  97 dir path=usr/share/man/man1m
  98 dir path=usr/share/man/man2
  99 dir path=usr/share/man/man3
 100 dir path=usr/share/man/man4
 101 dir path=usr/share/man/man5
 102 dir path=usr/share/man/man7d
 103 dir path=usr/share/man/man7fs
 104 dir path=usr/share/man/man7m
 105 dir path=usr/share/man/man7p
 106 dir path=usr/share/man/man9
 107 dir path=usr/share/man/man9e
 108 dir path=usr/share/man/man9f
 109 dir path=usr/share/man/man9p
 110 dir path=usr/share/man/man9s
 111 $(i386_ONLY)driver name=acpi_drv perms="* 0666 root sys"
 112 driver name=aggr perms="* 0666 root sys"
 113 driver name=arp perms="arp 0666 root sys"
 114 driver name=bl perms="* 0666 root sys"
 115 driver name=bridge clone_perms="bridge 0666 root sys" \
 116     policy="read_priv_set=net_rawaccess write_priv_set=net_rawaccess"
 117 $(sparc_ONLY)driver name=bscbus alias=SUNW,bscbus
 118 $(i386_ONLY)driver name=bscbus alias=SVI0101
 119 $(sparc_ONLY)driver name=bscv alias=SUNW,bscv perms="* 0644 root sys"
 120 $(i386_ONLY)driver name=bscv
 121 driver name=clone
 122 driver name=cn perms="* 0620 root tty"
 123 driver name=conskbd perms="kbd 0666 root sys"
 124 driver name=consms perms="mouse 0666 root sys"
 125 driver name=cpuid perms="self 0644 root sys"
 126 $(i386_ONLY)driver name=cpunex alias=cpus
 127 driver name=crypto perms="crypto 0666 root sys"
```

```
128 driver name=cryptoadm perms="cryptoadm 0644 root sys"
129 $(sparc_ONLY)driver name=dad alias=ide-disk perms="* 0640 root sys"
130 driver name=devinfo perms="devinfo 0640 root sys" \
131     perms="devinfo,ro 0444 root sys"
132 driver name=dld perms="* 0666 root sys"
133 driver name=dlpistub perms="* 0666 root sys"
134 $(sparc_ONLY)driver name=i8042 alias=8042
135 $(i386_ONLY)driver name=i8042
136 driver name=icmp perms="icmp 0666 root sys" \
137     policy="read_priv_set=net_icmpaccess write_priv_set=net_icmpaccess"
138 driver name=icmp6 perms="icmp6 0666 root sys" \
139     policy="read_priv_set=net_icmpaccess write_priv_set=net_icmpaccess"
140 $(i386_ONLY)driver name=intel_nb5000 \
141     alias=pci8086,25c0 \
142     alias=pci8086,25d0 \
143     alias=pci8086,25d4 \
144     alias=pci8086,25d8 \
145     alias=pci8086,3600 \
146     alias=pci8086,4000 \
147     alias=pci8086,4001 \
148     alias=pci8086,4003 \
149     alias=pci8086,65c0
150 $(i386_ONLY)driver name=intel_nhm \
151     alias=pci8086,3423 \
152     alias=pci8086,372a
153 $(i386_ONLY)driver name=intel_nhmex alias=pci8086,3438
154 driver name=ip perms="ip 0666 root sys" \
155     policy="read_priv_set=net_rawaccess write_priv_set=net_rawaccess"
156 driver name=ip6 perms="ip6 0666 root sys" \
157     policy="read_priv_set=net_rawaccess write_priv_set=net_rawaccess"
158 driver name=ipnet perms="lo0 0666 root sys" \
159     policy="read_priv_set=net_observability write_priv_set=net_observability"
160 driver name=ippctl
161 driver name=ipsecah perms="ipsecah 0666 root sys" \
162     policy="read_priv_set=sys_ip_config write_priv_set=sys_ip_config"
163 driver name=ipsecesp perms="ipsecesp 0666 root sys" \
164     policy="read_priv_set=sys_ip_config write_priv_set=sys_ip_config"
165 driver name=iptun
166 driver name=iwscn
167 driver name=kb8042 alias=pnpPNP,303
168 driver name=keysock perms="keysock 0666 root sys" \
169     policy="read_priv_set=sys_ip_config write_priv_set=sys_ip_config"
170 driver name=kmdb
171 driver name=kssl perms="* 0666 root sys"
172 driver name=llc1 clone_perms="llc1 0666 root sys"
173 driver name=lofi perms="* 0600 root sys" perms="ctl 0644 root sys"
174 driver name=log perms="conslog 0666 root sys" perms="log 0640 root sys"
175 $(i386_ONLY)driver name=mc-amd \
176     alias=pci1022,1100 \
177     alias=pci1022,1101 \
178     alias=pci1022,1102
179 driver name=mm perms="allkmem 0600 root sys" perms="kmem 0640 root sys" \
180     perms="mem 0640 root sys" perms="null 0666 root sys" \
181     perms="zero 0666 root sys" \
182     policy="allkmem read_priv_set=all write_priv_set=all" \
183     policy="kmem read_priv_set=none write_priv_set=all" \
184     policy="mem read_priv_set=none write_priv_set=all"
185 driver name=mouse8042 alias=pnpPNP,f03
186 $(i386_ONLY)driver name=mpt class=scsi \
187     alias=pci1000,30 \
188     alias=pci1000,50 \
189     alias=pci1000,54 \
190     alias=pci1000,56 \
191     alias=pci1000,58 \
192     alias=pci1000,62 \
193     alias=pciex1000,56 \
```



```
194     alias=pciex1000,58 \
195     alias=pciex1000,62
196 driver name=nulldriver \
197     alias=scsa,nodev \
198     alias=scsa,probe
199 driver name=openeepr perms="openprom 0640 root sys" policy=write_priv_set=all
200 driver name=options
201 $(sparc_ONLY)driver name=pci_pci class=pci \
202     alias=pci1011,1 \
203     alias=pci1011,21 \
204     alias=pci1011,24 \
205     alias=pci1011,25 \
206     alias=pci1011,26 \
207     alias=pci1014,22 \
208     alias=pciclass,060400
209 $(i386_ONLY)driver name=pci_pci class=pci \
210     alias=pci1011,1 \
211     alias=pci1011,21 \
212     alias=pci1014,22 \
213     alias=pciclass,060400 \
214     alias=pciclass,060401
215 $(sparc_ONLY)driver name=pcieb \
216     alias=pciex108e,9010 \
217     alias=pciex108e,9020 \
218     alias=pciex10b5,8114 \
219     alias=pciex10b5,8516 \
220     alias=pciex10b5,8517 \
221     alias=pciex10b5,8518 \
222     alias=pciex10b5,8532 \
223     alias=pciex10b5,8533 \
224     alias=pciex10b5,8548 \
225     alias=pciexclass,060400
226 $(i386_ONLY)driver name=pcieb \
227     alias=pciexclass,060400 \
228     alias=pciexclass,060401
229 $(sparc_ONLY)driver name=pcieb_bcm alias=pciex1166,103
230 driver name=physmem perms="* 0600 root sys"
231 driver name=poll perms="* 0666 root sys"
232 $(sparc_ONLY)driver name=power alias=ali1535d+-power
233 $(i386_ONLY)driver name=power
234 driver name=pseudo alias=zconsnex
235 driver name=ptc perms="* 0666 root sys"
236 driver name=ptsl perms="* 0666 root sys"
237 $(sparc_ONLY)driver name=ramdisk alias=SUNW,ramdisk perms="* 0600 root sys" \
238     perms="ctl 0644 root sys"
239 $(i386_ONLY)driver name=ramdisk perms="* 0600 root sys" \
240     perms="ctl 0644 root sys"
241 driver name=random perms="* 0644 root sys" policy=write_priv_set=sys_devices
242 driver name=rts perms="rts 0666 root sys"
243 driver name=sad perms="admin 0666 root sys" perms="user 0666 root sys"
244 driver name=scsi_vhci class=scsi-self-identifying perms="* 0666 root sys" \
245     policy="devctl write_priv_set=sys_devices"
246 $(sparc_ONLY)driver name=sd perms="* 0640 root sys" \
247     alias=ide-cdrom \
248     alias=scsiclass,00 \
249     alias=scsiclass,05
250 $(i386_ONLY)driver name=sd perms="* 0640 root sys" \
251     alias=scsiclass,00 \
252     alias=scsiclass,05
253 driver name=sgen perms="* 0600 root sys" \
254     alias=scsa,08.bfcp \
255     alias=scsa,08.bvhci
256 driver name=simnet clone_perms="simnet 0666 root sys" perms="* 0666 root sys"
257 $(i386_ONLY)driver name=smbios perms="smbios 0444 root sys"
258 driver name=softmac
259 driver name=spdsock perms="spdsock 0666 root sys" \
```

```
260     policy="read_priv_set=sys_ip_config write_priv_set=sys_ip_config"
261 driver name=st alias=scsiclass,01 perms="* 0666 root sys"
262 driver name=sy perms="tty 0666 root tty"
263 driver name=sysevent perms="* 0600 root sys"
264 driver name=sysmsg perms="msglog 0600 root sys" perms="sysmsg 0600 root sys"
265 driver name=tcp perms="tcp 0666 root sys"
266 driver name=tcp6 perms="tcp6 0666 root sys"
267 driver name=tl perms="* 0666 root sys" clone_perms="ticlts 0666 root sys" \
268     clone_perms="ticots 0666 root sys" clone_perms="ticotsord 0666 root sys"
269 $(sparc_ONLY)driver name=ttymux alias=multiplexer
270 $(i386_ONLY)driver name=tzmon
271 $(sparc_ONLY)driver name=uata \
272     alias=pci1095,646 \
273     alias=pci1095,649 \
274     alias=pci1095,680 \
275     alias=pci10b9,5229 \
276     alias=pci10b9,5288 class=dada class=scsi
277 $(i386_ONLY)driver name=ucode perms="* 0644 root sys"
278 driver name=udp perms="udp 0666 root sys"
279 driver name=udp6 perms="udp6 0666 root sys"
280 $(i386_ONLY)driver name=vgatext \
281     alias=pciclass,000100 \
282     alias=pciclass,030000 \
283     alias=pciclass,030001 \
284     alias=pnpPNP,900
285 driver name=vnic clone_perms="vnic 0666 root sys" perms="* 0666 root sys"
286 driver name=wc perms="* 0600 root sys"
287 $(i386_ONLY)file path=boot/solaris/bin/create_diskmap group=sys mode=0555
288 file path=boot/solaris/bin/create_ramdisk group=sys mode=0555
289 file path=boot/solaris/bin/extract_boot_filelist group=sys mode=0555
290 $(i386_ONLY)file path=boot/solaris/bin/mbr group=sys mode=0555
291 $(i386_ONLY)file path=boot/solaris/bin/symdef group=sys mode=0555
292 $(i386_ONLY)file path=boot/solaris/bin/update_grub group=sys mode=0555
293 file path=boot/solaris/filelist.ramdisk group=sys
294 file path=boot/solaris/filelist.safe group=sys
295 file path=etc/crypto/kcf.conf group=sys \
296     original_name=SUNWckr:etc/crypto/kcf.conf preserve=true
297 file path=etc/name_to_sysnum group=sys \
298     original_name=SUNWckr:etc/name_to_sysnum preserve=renameold
299 file path=etc/sock2path.d/system%2Fkernel group=sys
300 file path=etc/system group=sys original_name=SUNWckr:etc/system preserve=true
301 $(i386_ONLY)file path=kernel/$(ARCH64)/genunix group=sys mode=0755
302 file path=kernel/crypto/$(ARCH64)/aes group=sys mode=0755
303 file path=kernel/crypto/$(ARCH64)/arcfour group=sys mode=0755
304 file path=kernel/crypto/$(ARCH64)/blowfish group=sys mode=0755
305 file path=kernel/crypto/$(ARCH64)/des group=sys mode=0755
306 file path=kernel/crypto/$(ARCH64)/ecc group=sys mode=0755
307 file path=kernel/crypto/$(ARCH64)/edonr group=sys mode=0755
308 file path=kernel/crypto/$(ARCH64)/md4 group=sys mode=0755
309 file path=kernel/crypto/$(ARCH64)/md5 group=sys mode=0755
310 file path=kernel/crypto/$(ARCH64)/rsa group=sys mode=0755
311 file path=kernel/crypto/$(ARCH64)/sha1 group=sys mode=0755
312 file path=kernel/crypto/$(ARCH64)/sha2 group=sys mode=0755
313 file path=kernel/crypto/$(ARCH64)/skein group=sys mode=0755
314 file path=kernel/crypto/$(ARCH64)/swrand group=sys mode=0755
315 $(i386_ONLY)file path=kernel/crypto/aes group=sys mode=0755
316 $(i386_ONLY)file path=kernel/crypto/arcfour group=sys mode=0755
317 $(i386_ONLY)file path=kernel/crypto/blowfish group=sys mode=0755
318 $(i386_ONLY)file path=kernel/crypto/des group=sys mode=0755
319 $(i386_ONLY)file path=kernel/crypto/ecc group=sys mode=0755
320 $(i386_ONLY)file path=kernel/crypto/edonr group=sys mode=0755
321 $(i386_ONLY)file path=kernel/crypto/md4 group=sys mode=0755
322 $(i386_ONLY)file path=kernel/crypto/md5 group=sys mode=0755
323 $(i386_ONLY)file path=kernel/crypto/rsa group=sys mode=0755
324 $(i386_ONLY)file path=kernel/crypto/sha1 group=sys mode=0755
325 $(i386_ONLY)file path=kernel/crypto/sha2 group=sys mode=0755
```



```
326 $(i386_ONLY)file path=kernel/crypto/skein group=sys mode=0755
327 $(i386_ONLY)file path=kernel/crypto/swrand group=sys mode=0755
328 $(sparc_ONLY)file path=kernel/dacf/$(ARCH64)/consconfig_dacf group=sys \
329     mode=0755
330 file path=kernel/dacf/$(ARCH64)/net_dacf group=sys mode=0755
331 $(i386_ONLY)file path=kernel/dacf/net_dacf group=sys mode=0755
332 $(i386_ONLY)file path=kernel/drv/$(ARCH64)/acpi_drv group=sys
333 $(i386_ONLY)file path=kernel/drv/$(ARCH64)/acpi_toshiba group=sys
334 file path=kernel/drv/$(ARCH64)/aggr group=sys
335 file path=kernel/drv/$(ARCH64)/arp group=sys
336 file path=kernel/drv/$(ARCH64)/bl group=sys
337 file path=kernel/drv/$(ARCH64)/bridge group=sys
338 $(i386_ONLY)file path=kernel/drv/$(ARCH64)/bscbus group=sys
339 $(i386_ONLY)file path=kernel/drv/$(ARCH64)/bscv group=sys
340 file path=kernel/drv/$(ARCH64)/clone group=sys
341 file path=kernel/drv/$(ARCH64)/cn group=sys
342 file path=kernel/drv/$(ARCH64)/conskbd group=sys
343 file path=kernel/drv/$(ARCH64)/consms group=sys
344 file path=kernel/drv/$(ARCH64)/cpuid group=sys
345 $(i386_ONLY)file path=kernel/drv/$(ARCH64)/cpunex group=sys
346 file path=kernel/drv/$(ARCH64)/crypto group=sys
347 file path=kernel/drv/$(ARCH64)/cryptoadm group=sys
348 $(sparc_ONLY)file path=kernel/drv/$(ARCH64)/dad group=sys
349 file path=kernel/drv/$(ARCH64)/devinfo group=sys
350 file path=kernel/drv/$(ARCH64)/dld group=sys
351 file path=kernel/drv/$(ARCH64)/dlpistub group=sys
352 file path=kernel/drv/$(ARCH64)/i8042 group=sys
353 file path=kernel/drv/$(ARCH64)/icmp group=sys
354 file path=kernel/drv/$(ARCH64)/icmp6 group=sys
355 $(i386_ONLY)file path=kernel/drv/$(ARCH64)/intel_nb5000 group=sys
356 $(i386_ONLY)file path=kernel/drv/$(ARCH64)/intel_nhm group=sys
357 $(i386_ONLY)file path=kernel/drv/$(ARCH64)/intel_nhmex group=sys
358 file path=kernel/drv/$(ARCH64)/ip group=sys
359 file path=kernel/drv/$(ARCH64)/ip6 group=sys
360 file path=kernel/drv/$(ARCH64)/ipnet group=sys
361 file path=kernel/drv/$(ARCH64)/ippctl group=sys
362 file path=kernel/drv/$(ARCH64)/ipsecah group=sys
363 file path=kernel/drv/$(ARCH64)/ipsecesp group=sys
364 file path=kernel/drv/$(ARCH64)/iptun group=sys
365 file path=kernel/drv/$(ARCH64)/iwscn group=sys
366 file path=kernel/drv/$(ARCH64)/kb8042 group=sys
367 file path=kernel/drv/$(ARCH64)/keysock group=sys
368 file path=kernel/drv/$(ARCH64)/kmdb group=sys
369 file path=kernel/drv/$(ARCH64)/kssl group=sys
370 file path=kernel/drv/$(ARCH64)/llc1 group=sys
371 file path=kernel/drv/$(ARCH64)/lofi group=sys
372 file path=kernel/drv/$(ARCH64)/log group=sys
373 $(i386_ONLY)file path=kernel/drv/$(ARCH64)/mc-amd group=sys
374 file path=kernel/drv/$(ARCH64)/mm group=sys
375 file path=kernel/drv/$(ARCH64)/mouse8042 group=sys
376 $(i386_ONLY)file path=kernel/drv/$(ARCH64)/mpt group=sys
377 file path=kernel/drv/$(ARCH64)/nulldriver group=sys
378 file path=kernel/drv/$(ARCH64)/openeepr group=sys
379 file path=kernel/drv/$(ARCH64)/options group=sys
380 file path=kernel/drv/$(ARCH64)/pci_pci group=sys
381 file path=kernel/drv/$(ARCH64)/pcieb group=sys
382 $(sparc_ONLY)file path=kernel/drv/$(ARCH64)/pcieb_bcm group=sys
383 file path=kernel/drv/$(ARCH64)/physmem group=sys
384 file path=kernel/drv/$(ARCH64)/poll group=sys
385 $(i386_ONLY)file path=kernel/drv/$(ARCH64)/power group=sys
386 file path=kernel/drv/$(ARCH64)/pseudo group=sys
387 file path=kernel/drv/$(ARCH64)/ptc group=sys
388 file path=kernel/drv/$(ARCH64)/ptsl group=sys
389 file path=kernel/drv/$(ARCH64)/ramdisk group=sys
390 file path=kernel/drv/$(ARCH64)/random group=sys
391 file path=kernel/drv/$(ARCH64)/rts group=sys
```

```
file path=kernel/drv/$(ARCH64)/sad group=sys
file path=kernel/drv/$(ARCH64)/scsi_vhci group=sys
file path=kernel/drv/$(ARCH64)/sd group=sys
file path=kernel/drv/$(ARCH64)/sgen group=sys
file path=kernel/drv/$(ARCH64)/simnet group=sys
$(i386_ONLY)file path=kernel/drv/$(ARCH64)/smbios group=sys
file path=kernel/drv/$(ARCH64)/softmac group=sys
file path=kernel/drv/$(ARCH64)/spdsock group=sys
file path=kernel/drv/$(ARCH64)/st group=sys
file path=kernel/drv/$(ARCH64)/sy group=sys
file path=kernel/drv/$(ARCH64)/sysevent group=sys
file path=kernel/drv/$(ARCH64)/sysmsg group=sys
file path=kernel/drv/$(ARCH64)/tcp group=sys
file path=kernel/drv/$(ARCH64)/tcp6 group=sys
file path=kernel/drv/$(ARCH64)/tl group=sys
$(sparc_ONLY)file path=kernel/drv/$(ARCH64)/ttymux group=sys
$(i386_ONLY)file path=kernel/drv/$(ARCH64)/tzmon group=sys
$(sparc_ONLY)file path=kernel/drv/$(ARCH64)/uata group=sys
$(i386_ONLY)file path=kernel/drv/$(ARCH64)/ucode group=sys
file path=kernel/drv/$(ARCH64)/udp group=sys
file path=kernel/drv/$(ARCH64)/udp6 group=sys
$(i386_ONLY)file path=kernel/drv/$(ARCH64)/vgatext group=sys
file path=kernel/drv/$(ARCH64)/vnic group=sys
file path=kernel/drv/$(ARCH64)/wc group=sys
$(i386_ONLY)file path=kernel/drv/acpi_drv group=sys
$(i386_ONLY)file path=kernel/drv/acpi_drv.conf group=sys
$(i386_ONLY)file path=kernel/drv/acpi_toshiba group=sys
$(i386_ONLY)file path=kernel/drv/aggr group=sys
file path=kernel/drv/aggr.conf group=sys
$(i386_ONLY)file path=kernel/drv/arp group=sys
file path=kernel/drv/arp.conf group=sys
$(i386_ONLY)file path=kernel/drv/bl group=sys
file path=kernel/drv/bl.conf group=sys
$(i386_ONLY)file path=kernel/drv/bridge group=sys
file path=kernel/drv/bridge.conf group=sys
$(i386_ONLY)file path=kernel/drv/bscbus group=sys
$(i386_ONLY)file path=kernel/drv/bscbus.conf group=sys
$(i386_ONLY)file path=kernel/drv/bscv group=sys
$(i386_ONLY)file path=kernel/drv/bscv.conf group=sys
$(i386_ONLY)file path=kernel/drv/clone group=sys
file path=kernel/drv/clone.conf group=sys
$(i386_ONLY)file path=kernel/drv/cn group=sys
file path=kernel/drv/cn.conf group=sys
$(i386_ONLY)file path=kernel/drv/conskbd group=sys
file path=kernel/drv/conskbd.conf group=sys
$(i386_ONLY)file path=kernel/drv/consms group=sys
file path=kernel/drv/consms.conf group=sys
$(i386_ONLY)file path=kernel/drv/cpuid group=sys
file path=kernel/drv/cpuid.conf group=sys
$(i386_ONLY)file path=kernel/drv/cpunex group=sys
$(i386_ONLY)file path=kernel/drv/crypto group=sys
file path=kernel/drv/crypto.conf group=sys
$(i386_ONLY)file path=kernel/drv/cryptoadm group=sys
file path=kernel/drv/cryptoadm.conf group=sys
$(sparc_ONLY)file path=kernel/drv/dad.conf group=sys
$(i386_ONLY)file path=kernel/drv/devinfo group=sys
file path=kernel/drv/devinfo.conf group=sys
$(i386_ONLY)file path=kernel/drv/dld group=sys
file path=kernel/drv/dld.conf group=sys
$(i386_ONLY)file path=kernel/drv/dlpistub group=sys
file path=kernel/drv/dlpistub.conf group=sys
$(i386_ONLY)file path=kernel/drv/i8042 group=sys
$(i386_ONLY)file path=kernel/drv/icmp group=sys
file path=kernel/drv/icmp.conf group=sys
$(i386_ONLY)file path=kernel/drv/icmp6 group=sys
file path=kernel/drv/icmp6.conf group=sys
```



```
$(i386_ONLY)file path=kernel/drv/intel_nb5000 group=sys
$(i386_ONLY)file path=kernel/drv/intel_nb5000.conf group=sys
$(i386_ONLY)file path=kernel/drv/intel_nhm group=sys
$(i386_ONLY)file path=kernel/drv/intel_nhm.conf group=sys
$(i386_ONLY)file path=kernel/drv/intel_nhmex group=sys
$(i386_ONLY)file path=kernel/drv/intel_nhmex.conf group=sys
$(i386_ONLY)file path=kernel/drv/ip group=sys
file path=kernel/drv/ip.conf group=sys
$(i386_ONLY)file path=kernel/drv/ip6 group=sys
file path=kernel/drv/ip6.conf group=sys
$(i386_ONLY)file path=kernel/drv/ipnet group=sys
file path=kernel/drv/ipnet.conf group=sys
$(i386_ONLY)file path=kernel/drv/ippctl group=sys
file path=kernel/drv/ippctl.conf group=sys
$(i386_ONLY)file path=kernel/drv/ipsecah group=sys
file path=kernel/drv/ipsecah.conf group=sys
$(i386_ONLY)file path=kernel/drv/ipsecesp group=sys
file path=kernel/drv/ipsecesp.conf group=sys
$(i386_ONLY)file path=kernel/drv/iptun group=sys
file path=kernel/drv/iptun.conf group=sys
$(i386_ONLY)file path=kernel/drv/iwscn group=sys
file path=kernel/drv/iwscn.conf group=sys
$(i386_ONLY)file path=kernel/drv/kb8042 group=sys
$(i386_ONLY)file path=kernel/drv/keysock group=sys
file path=kernel/drv/keysock.conf group=sys
$(i386_ONLY)file path=kernel/drv/kmdb group=sys
file path=kernel/drv/kmdb.conf group=sys
$(i386_ONLY)file path=kernel/drv/kssl group=sys
file path=kernel/drv/kssl.conf group=sys
$(i386_ONLY)file path=kernel/drv/llc1 group=sys
file path=kernel/drv/llc1.conf group=sys
$(i386_ONLY)file path=kernel/drv/lofi group=sys
file path=kernel/drv/lofi.conf group=sys
$(i386_ONLY)file path=kernel/drv/log group=sys
file path=kernel/drv/log.conf group=sys \
    original_name=SUNWckr:kernel/drv/log.conf preserve=true
$(i386_ONLY)file path=kernel/drv/mc-amd group=sys
$(i386_ONLY)file path=kernel/drv/mc-amd.conf group=sys
$(i386_ONLY)file path=kernel/drv/mm group=sys
file path=kernel/drv/mm.conf group=sys
$(i386_ONLY)file path=kernel/drv/mouse8042 group=sys
$(i386_ONLY)file path=kernel/drv/mpt group=sys
$(i386_ONLY)file path=kernel/drv/mpt.conf group=sys \
    original_name=SUNWckr:kernel/drv/mpt.conf preserve=true
$(i386_ONLY)file path=kernel/drv/nulldriver group=sys
$(i386_ONLY)file path=kernel/drv/openeepr group=sys
file path=kernel/drv/openeepr.conf group=sys
$(i386_ONLY)file path=kernel/drv/options group=sys
file path=kernel/drv/options.conf group=sys
$(i386_ONLY)file path=kernel/drv/pci_pci group=sys
$(i386_ONLY)file path=kernel/drv/pcieb group=sys
file path=kernel/drv/pcieb.conf group=sys
$(i386_ONLY)file path=kernel/drv/physmem group=sys
file path=kernel/drv/physmem.conf group=sys
$(i386_ONLY)file path=kernel/drv/poll group=sys
file path=kernel/drv/poll.conf group=sys
$(i386_ONLY)file path=kernel/drv/power group=sys
$(i386_ONLY)file path=kernel/drv/power.conf group=sys
$(i386_ONLY)file path=kernel/drv/pseudo group=sys
file path=kernel/drv/pseudo.conf group=sys
$(i386_ONLY)file path=kernel/drv/ptc group=sys
file path=kernel/drv/ptc.conf group=sys
$(i386_ONLY)file path=kernel/drv/ptsl group=sys
file path=kernel/drv/ptsl.conf group=sys
$(i386_ONLY)file path=kernel/drv/ramdisk group=sys
file path=kernel/drv/ramdisk.conf group=sys
```

```
$(i386_ONLY)file path=kernel/drv/random group=sys
file path=kernel/drv/random.conf group=sys
$(i386_ONLY)file path=kernel/drv/rts group=sys
file path=kernel/drv/rts.conf group=sys
$(i386_ONLY)file path=kernel/drv/sad group=sys
file path=kernel/drv/sad.conf group=sys
$(i386_ONLY)file path=kernel/drv/scsi_vhci group=sys
file path=kernel/drv/scsi_vhci.conf group=sys \
    original_name=SUNWckr:kernel/drv/scsi_vhci.conf preserve=true
$(sparc_ONLY)file path=kernel/drv/sd.conf group=sys \
    original_name=SUNWckr:kernel/drv/sd.conf preserve=true
$(i386_ONLY)file path=kernel/drv/sgen group=sys
file path=kernel/drv/sgen.conf group=sys \
    original_name=SUNWckr:kernel/drv/sgen.conf preserve=true
$(i386_ONLY)file path=kernel/drv/simnet group=sys
file path=kernel/drv/simnet.conf group=sys
$(i386_ONLY)file path=kernel/drv/smbios group=sys
$(i386_ONLY)file path=kernel/drv/smbios.conf group=sys
$(i386_ONLY)file path=kernel/drv/softmac group=sys
file path=kernel/drv/softmac.conf group=sys
$(i386_ONLY)file path=kernel/drv/spdsock group=sys
file path=kernel/drv/spdsock.conf group=sys
$(i386_ONLY)file path=kernel/drv/st group=sys
file path=kernel/drv/st.conf group=sys \
    original_name=SUNWckr:kernel/drv/st.conf preserve=true
$(i386_ONLY)file path=kernel/drv/sy group=sys
file path=kernel/drv/sy.conf group=sys
$(i386_ONLY)file path=kernel/drv/sysevent group=sys
file path=kernel/drv/sysevent.conf group=sys
$(i386_ONLY)file path=kernel/drv/sysmsg group=sys
file path=kernel/drv/sysmsg.conf group=sys
$(i386_ONLY)file path=kernel/drv/tcp group=sys
file path=kernel/drv/tcp.conf group=sys
$(i386_ONLY)file path=kernel/drv/tcp6 group=sys
file path=kernel/drv/tcp6.conf group=sys
$(i386_ONLY)file path=kernel/drv/tl group=sys
file path=kernel/drv/tl.conf group=sys
$(i386_ONLY)file path=kernel/drv/tzmon group=sys
$(i386_ONLY)file path=kernel/drv/tzmon.conf group=sys
$(sparc_ONLY)file path=kernel/drv/uata.conf group=sys \
    original_name=SUNWckr:kernel/drv/uata.conf preserve=true
$(i386_ONLY)file path=kernel/drv/ucode group=sys
$(i386_ONLY)file path=kernel/drv/ucode.conf group=sys
$(i386_ONLY)file path=kernel/drv/udp group=sys
file path=kernel/drv/udp.conf group=sys
$(i386_ONLY)file path=kernel/drv/udp6 group=sys
file path=kernel/drv/udp6.conf group=sys
$(i386_ONLY)file path=kernel/drv/vgatext group=sys
$(i386_ONLY)file path=kernel/drv/vnic group=sys
file path=kernel/drv/vnic.conf group=sys
$(i386_ONLY)file path=kernel/drv/wc group=sys
file path=kernel/drv/wc.conf group=sys
$(sparc_ONLY)file path=kernel/exec/$(ARCH64)/aoutexec group=sys mode=0755
file path=kernel/exec/$(ARCH64)/elfexec group=sys mode=0755
file path=kernel/exec/$(ARCH64)/intpexec group=sys mode=0755
$(i386_ONLY)file path=kernel/exec/elfexec group=sys mode=0755
$(i386_ONLY)file path=kernel/exec/intpexec group=sys mode=0755
file path=kernel/fs/$(ARCH64)/autofs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/cachefs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/ctfs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/dcfs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/dev group=sys mode=0755
file path=kernel/fs/$(ARCH64)/devfs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/fifofs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/hsfs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/lofs group=sys mode=0755
```



```
file path=kernel/fs/$(ARCH64)/mntfs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/namefs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/objfs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/procfs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/sharefs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/sockfs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/specfs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/tmpfs group=sys mode=0755
file path=kernel/fs/$(ARCH64)/ufs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/autofs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/cachefs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/ctfs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/dcfs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/dev group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/devfs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/fifofs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/hsfs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/lofs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/mntfs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/namefs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/objfs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/procfs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/sharefs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/sockfs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/specfs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/tmpfs group=sys mode=0755
$(i386_ONLY)file path=kernel/fs/ufs group=sys mode=0755
$(i386_ONLY)file path=kernel/genunix group=sys mode=0755
file path=kernel/ipp/$(ARCH64)/ipgpc group=sys mode=0755
$(i386_ONLY)file path=kernel/ipp/ipgpc group=sys mode=0755
file path=kernel/kiconv/$(ARCH64)/kiconv_emea group=sys mode=0755
file path=kernel/kiconv/$(ARCH64)/kiconv_ja group=sys mode=0755
file path=kernel/kiconv/$(ARCH64)/kiconv_ko group=sys mode=0755
file path=kernel/kiconv/$(ARCH64)/kiconv_sc group=sys mode=0755
file path=kernel/kiconv/$(ARCH64)/kiconv_tc group=sys mode=0755
$(i386_ONLY)file path=kernel/kiconv/kiconv_emea group=sys mode=0755
$(i386_ONLY)file path=kernel/kiconv/kiconv_ja group=sys mode=0755
$(i386_ONLY)file path=kernel/kiconv/kiconv_ko group=sys mode=0755
$(i386_ONLY)file path=kernel/kiconv/kiconv_sc group=sys mode=0755
$(i386_ONLY)file path=kernel/kiconv/kiconv_tc group=sys mode=0755
file path=kernel/mac/$(ARCH64)/mac_6to4 group=sys mode=0755
file path=kernel/mac/$(ARCH64)/mac_ether group=sys mode=0755
file path=kernel/mac/$(ARCH64)/mac_ib group=sys mode=0755
file path=kernel/mac/$(ARCH64)/mac_ipv4 group=sys mode=0755
file path=kernel/mac/$(ARCH64)/mac_ipv6 group=sys mode=0755
file path=kernel/mac/$(ARCH64)/mac_wifi group=sys mode=0755
$(i386_ONLY)file path=kernel/mac/mac_6to4 group=sys mode=0755
$(i386_ONLY)file path=kernel/mac/mac_ether group=sys mode=0755
$(i386_ONLY)file path=kernel/mac/mac_ib group=sys mode=0755
$(i386_ONLY)file path=kernel/mac/mac_ipv4 group=sys mode=0755
$(i386_ONLY)file path=kernel/mac/mac_ipv6 group=sys mode=0755
$(i386_ONLY)file path=kernel/mac/mac_wifi group=sys mode=0755
$(i386_ONLY)file path=kernel/misc/$(ARCH64)/acpica group=sys mode=0755
$(i386_ONLY)file path=kernel/misc/$(ARCH64)/agpmaster group=sys mode=0755
file path=kernel/misc/$(ARCH64)/bignum group=sys mode=0755
$(i386_ONLY)file path=kernel/misc/$(ARCH64)/bootdev group=sys mode=0755
file path=kernel/misc/$(ARCH64)/busra group=sys mode=0755
file path=kernel/misc/$(ARCH64)/cardbus group=sys mode=0755
file path=kernel/misc/$(ARCH64)/cmlb group=sys mode=0755
file path=kernel/misc/$(ARCH64)/consconfig group=sys mode=0755
file path=kernel/misc/$(ARCH64)/ctf group=sys mode=0755
$(sparc_ONLY)file path=kernel/misc/$(ARCH64)/dada group=sys mode=0755
file path=kernel/misc/$(ARCH64)/dls group=sys mode=0755
file path=kernel/misc/$(ARCH64)/fssnap_if group=sys mode=0755
file path=kernel/misc/$(ARCH64)/gld group=sys mode=0755
file path=kernel/misc/$(ARCH64)/hook group=sys mode=0755
```

```
656 file path=kernel/misc/$(ARCH64)/hpcsvc group=sys mode=0755
657 file path=kernel/misc/$(ARCH64)/idmap group=sys mode=0755
658 $(i386_ONLY)file path=kernel/misc/$(ARCH64)/iommulib group=sys mode=0755
659 file path=kernel/misc/$(ARCH64)/ipc group=sys mode=0755
660 file path=kernel/misc/$(ARCH64)/kbtrans group=sys mode=0755
661 file path=kernel/misc/$(ARCH64)/kcf group=sys mode=0755
662 $(i386_ONLY)file path=kernel/misc/$(ARCH64)/kmdbmod group=sys mode=0755
663 file path=kernel/misc/$(ARCH64)/ksocket group=sys mode=0755
664 file path=kernel/misc/$(ARCH64)/mac group=sys mode=0755
665 file path=kernel/misc/$(ARCH64)/mii group=sys mode=0755
666 $(i386_ONLY)file path=kernel/misc/$(ARCH64)/net80211 group=sys mode=0755
667 file path=kernel/misc/$(ARCH64)/neti group=sys mode=0755
668 $(i386_ONLY)file path=kernel/misc/$(ARCH64)/pci_autoconfig group=sys mode=0755
669 $(i386_ONLY)file path=kernel/misc/$(ARCH64)/pcicfg group=sys mode=0755
670 $(i386_ONLY)file path=kernel/misc/$(ARCH64)/pcie group=sys mode=0755
671 file path=kernel/misc/$(ARCH64)/pcihp group=sys mode=0755
672 file path=kernel/misc/$(ARCH64)/pcmcia group=sys mode=0755
673 file path=kernel/misc/$(ARCH64)/rpcsec group=sys mode=0755
674 $(i386_ONLY)file path=kernel/misc/$(ARCH64)/sata group=sys mode=0755
675 file path=kernel/misc/$(ARCH64)/scsi group=sys mode=0755
676 file path=kernel/misc/$(ARCH64)/strplumb group=sys mode=0755
677 $(sparc_ONLY)file path=kernel/misc/$(ARCH64)/swapgeneric group=sys mode=0755
678 file path=kernel/misc/$(ARCH64)/tem group=sys mode=0755
679 file path=kernel/misc/$(ARCH64)/tlimod group=sys mode=0755
680 $(i386_ONLY)file path=kernel/misc/acpica group=sys mode=0755
681 $(i386_ONLY)file path=kernel/misc/agpmaster group=sys mode=0755
682 $(i386_ONLY)file path=kernel/misc/bignum group=sys mode=0755
683 $(i386_ONLY)file path=kernel/misc/bootdev group=sys mode=0755
684 $(i386_ONLY)file path=kernel/misc/busra group=sys mode=0755
685 $(i386_ONLY)file path=kernel/misc/cardbus group=sys mode=0755
686 $(i386_ONLY)file path=kernel/misc/cmlb group=sys mode=0755
687 $(i386_ONLY)file path=kernel/misc/consconfig group=sys mode=0755
688 $(i386_ONLY)file path=kernel/misc/ctf group=sys mode=0755
689 $(i386_ONLY)file path=kernel/misc/dls group=sys mode=0755
690 $(i386_ONLY)file path=kernel/misc/fssnap_if group=sys mode=0755
691 $(i386_ONLY)file path=kernel/misc/gld group=sys mode=0755
692 $(i386_ONLY)file path=kernel/misc/hook group=sys mode=0755
693 $(i386_ONLY)file path=kernel/misc/hpcsvc group=sys mode=0755
694 $(i386_ONLY)file path=kernel/misc/idmap group=sys mode=0755
695 $(i386_ONLY)file path=kernel/misc/iommulib group=sys mode=0755
696 $(i386_ONLY)file path=kernel/misc/ipc group=sys mode=0755
697 $(i386_ONLY)file path=kernel/misc/kbtrans group=sys mode=0755
698 $(i386_ONLY)file path=kernel/misc/kcf group=sys mode=0755
699 $(i386_ONLY)file path=kernel/misc/kmdbmod group=sys mode=0755
700 $(i386_ONLY)file path=kernel/misc/ksocket group=sys mode=0755
701 $(i386_ONLY)file path=kernel/misc/mac group=sys mode=0755
702 $(i386_ONLY)file path=kernel/misc/mii group=sys mode=0755
703 $(i386_ONLY)file path=kernel/misc/net80211 group=sys mode=0755
704 $(i386_ONLY)file path=kernel/misc/neti group=sys mode=0755
705 $(i386_ONLY)file path=kernel/misc/pci_autoconfig group=sys mode=0755
706 $(i386_ONLY)file path=kernel/misc/pcicfg group=sys mode=0755
707 $(i386_ONLY)file path=kernel/misc/pcie group=sys mode=0755
708 $(i386_ONLY)file path=kernel/misc/pcihp group=sys mode=0755
709 $(i386_ONLY)file path=kernel/misc/pcmcia group=sys mode=0755
710 $(i386_ONLY)file path=kernel/misc/rpcsec group=sys mode=0755
711 $(i386_ONLY)file path=kernel/misc/sata group=sys mode=0755
712 $(i386_ONLY)file path=kernel/misc/scsi group=sys mode=0755
713 file path=kernel/misc/scsi_vhci/$(ARCH64)/scsi_vhci_f_asym_emc group=sys \
714     mode=0755
715 file path=kernel/misc/scsi_vhci/$(ARCH64)/scsi_vhci_f_asym_lsi group=sys \
716     mode=0755
717 file path=kernel/misc/scsi_vhci/$(ARCH64)/scsi_vhci_f_asym_sun group=sys \
718     mode=0755
719 file path=kernel/misc/scsi_vhci/$(ARCH64)/scsi_vhci_f_sym group=sys mode=0755
720 file path=kernel/misc/scsi_vhci/$(ARCH64)/scsi_vhci_f_sym_emc group=sys \
721     mode=0755
```



```
722 file path=kernel/misc/scsi_vhci/$(ARCH64)/scsi_vhci_f_sym_hds group=sys \
723     mode=0755
724 file path=kernel/misc/scsi_vhci/$(ARCH64)/scsi_vhci_f_tape group=sys mode=0755
725 file path=kernel/misc/scsi_vhci/$(ARCH64)/scsi_vhci_f_tpgs group=sys mode=0755
726 file path=kernel/misc/scsi_vhci/$(ARCH64)/scsi_vhci_f_tpgs_tape group=sys \
727     mode=0755
728 $(i386_ONLY)file path=kernel/misc/scsi_vhci/scsi_vhci_f_asym_emc group=sys \
729     mode=0755
730 $(i386_ONLY)file path=kernel/misc/scsi_vhci/scsi_vhci_f_asym_lsi group=sys \
731     mode=0755
732 $(i386_ONLY)file path=kernel/misc/scsi_vhci/scsi_vhci_f_asym_sun group=sys \
733     mode=0755
734 $(i386_ONLY)file path=kernel/misc/scsi_vhci/scsi_vhci_f_sym group=sys \
735     mode=0755
736 $(i386_ONLY)file path=kernel/misc/scsi_vhci/scsi_vhci_f_sym_emc group=sys \
737     mode=0755
738 $(i386_ONLY)file path=kernel/misc/scsi_vhci/scsi_vhci_f_sym_hds group=sys \
739     mode=0755
740 $(i386_ONLY)file path=kernel/misc/scsi_vhci/scsi_vhci_f_tape group=sys \
741     mode=0755
742 $(i386_ONLY)file path=kernel/misc/scsi_vhci/scsi_vhci_f_tpgs group=sys \
743     mode=0755
744 $(i386_ONLY)file path=kernel/misc/scsi_vhci/scsi_vhci_f_tpgs_tape group=sys \
745     mode=0755
746 $(i386_ONLY)file path=kernel/misc/strplumb group=sys mode=0755
747 $(i386_ONLY)file path=kernel/misc/tem group=sys mode=0755
748 $(i386_ONLY)file path=kernel/misc/tlimod group=sys mode=0755
749 file path=kernel/sched/$(ARCH64)/SDC group=sys mode=0755
750 file path=kernel/sched/$(ARCH64)/TS group=sys mode=0755
751 file path=kernel/sched/$(ARCH64)/TS_DPTBL group=sys mode=0755
752 $(i386_ONLY)file path=kernel/sched/SDC group=sys mode=0755
753 $(i386_ONLY)file path=kernel/sched/TS group=sys mode=0755
754 $(i386_ONLY)file path=kernel/sched/TS_DPTBL group=sys mode=0755
755 file path=kernel/socketmod/$(ARCH64)/ksslf group=sys mode=0755
756 file path=kernel/socketmod/$(ARCH64)/socksctp group=sys mode=0755
757 file path=kernel/socketmod/$(ARCH64)/trill group=sys mode=0755
758 $(i386_ONLY)file path=kernel/socketmod/ksslf group=sys mode=0755
759 $(i386_ONLY)file path=kernel/socketmod/socksctp group=sys mode=0755
760 $(i386_ONLY)file path=kernel/socketmod/trill group=sys mode=0755
761 file path=kernel/strmod/$(ARCH64)/bufmod group=sys mode=0755
762 file path=kernel/strmod/$(ARCH64)/connld group=sys mode=0755
763 file path=kernel/strmod/$(ARCH64)/dedump group=sys mode=0755
764 file path=kernel/strmod/$(ARCH64)/drcompat group=sys mode=0755
765 file path=kernel/strmod/$(ARCH64)/ldterm group=sys mode=0755
766 $(sparc_ONLY)file path=kernel/strmod/$(ARCH64)/ms group=sys mode=0755
767 file path=kernel/strmod/$(ARCH64)/pckt group=sys mode=0755
768 file path=kernel/strmod/$(ARCH64)/pfmod group=sys mode=0755
769 file path=kernel/strmod/$(ARCH64)/pipemod group=sys mode=0755
770 file path=kernel/strmod/$(ARCH64)/ptem group=sys mode=0755
771 file path=kernel/strmod/$(ARCH64)/redirmod group=sys mode=0755
772 file path=kernel/strmod/$(ARCH64)/rpcmod group=sys mode=0755
773 file path=kernel/strmod/$(ARCH64)/timod group=sys mode=0755
774 file path=kernel/strmod/$(ARCH64)/tirdwr group=sys mode=0755
775 file path=kernel/strmod/$(ARCH64)/ttcompat group=sys mode=0755
776 $(sparc_ONLY)file path=kernel/strmod/$(ARCH64)/vuid3ps2 group=sys mode=0755
777 $(i386_ONLY)file path=kernel/strmod/bufmod group=sys mode=0755
778 $(i386_ONLY)file path=kernel/strmod/connld group=sys mode=0755
779 $(i386_ONLY)file path=kernel/strmod/dedump group=sys mode=0755
780 $(i386_ONLY)file path=kernel/strmod/drcompat group=sys mode=0755
781 $(i386_ONLY)file path=kernel/strmod/ldterm group=sys mode=0755
782 $(i386_ONLY)file path=kernel/strmod/pckt group=sys mode=0755
783 $(i386_ONLY)file path=kernel/strmod/pfmod group=sys mode=0755
784 $(i386_ONLY)file path=kernel/strmod/pipemod group=sys mode=0755
785 $(i386_ONLY)file path=kernel/strmod/ptem group=sys mode=0755
786 $(i386_ONLY)file path=kernel/strmod/redirmod group=sys mode=0755
787 $(i386_ONLY)file path=kernel/strmod/rpcmod group=sys mode=0755
```

```
$(i386_ONLY)file path=kernel/strmod/timod group=sys mode=0755
$(i386_ONLY)file path=kernel/strmod/tirdwr group=sys mode=0755
$(i386_ONLY)file path=kernel/strmod/ttcompat group=sys mode=0755
file path=kernel/sys/$(ARCH64)/c2audit group=sys mode=0755
file path=kernel/sys/$(ARCH64)/doorfs group=sys mode=0755
file path=kernel/sys/$(ARCH64)/inst_sync group=sys mode=0755
file path=kernel/sys/$(ARCH64)/kaio group=sys mode=0755
file path=kernel/sys/$(ARCH64)/msgsys group=sys mode=0755
file path=kernel/sys/$(ARCH64)/pipe group=sys mode=0755
file path=kernel/sys/$(ARCH64)/portfs group=sys mode=0755
file path=kernel/sys/$(ARCH64)/pset group=sys mode=0755
file path=kernel/sys/$(ARCH64)/semsys group=sys mode=0755
file path=kernel/sys/$(ARCH64)/shmsys group=sys mode=0755
$(i386_ONLY)file path=kernel/sys/c2audit group=sys mode=0755
$(i386_ONLY)file path=kernel/sys/doorfs group=sys mode=0755
$(i386_ONLY)file path=kernel/sys/inst_sync group=sys mode=0755
$(i386_ONLY)file path=kernel/sys/kaio group=sys mode=0755
$(i386_ONLY)file path=kernel/sys/msgsys group=sys mode=0755
$(i386_ONLY)file path=kernel/sys/pipe group=sys mode=0755
$(i386_ONLY)file path=kernel/sys/portfs group=sys mode=0755
$(i386_ONLY)file path=kernel/sys/pset group=sys mode=0755
$(i386_ONLY)file path=kernel/sys/semsys group=sys mode=0755
$(i386_ONLY)file path=kernel/sys/shmsys group=sys mode=0755
file path=lib/svc/manifest/system/dumpadm.xml group=sys mode=0444
file path=lib/svc/manifest/system/intrd.xml group=sys mode=0444
file path=lib/svc/manifest/system/scheduler.xml group=sys mode=0444
file path=lib/svc/method/svc-dumpadm mode=0555
file path=lib/svc/method/svc-intrd mode=0555
file path=lib/svc/method/svc-scheduler mode=0555
$(sparc_ONLY)file path=usr/share/man/man1m/monitor.1m
$(sparc_ONLY)file path=usr/share/man/man1m/obpsym.1m
# On SPARC driver/bscv is Serverblade1 specific, and in system/kernel/platform
# We keep the manual page generic
$(sparc_ONLY)file path=usr/share/man/man7d/dad.7d
$(i386_ONLY)file path=usr/share/man/man7d/smbios.7d
# Sadly vuid mouse support is in different packages on different platforms
# While kstat(7D) is in SUNWcs, the structures are general
hardlink path=kernel/misc/$(ARCH64)/edonr \
    target=../../../kernel/crypto/$(ARCH64)/edonr
hardlink path=kernel/misc/$(ARCH64)/md5 \
    target=../../../kernel/crypto/$(ARCH64)/md5
hardlink path=kernel/misc/$(ARCH64)/sha1 \
    target=../../../kernel/crypto/$(ARCH64)/sha1
hardlink path=kernel/misc/$(ARCH64)/sha2 \
    target=../../../kernel/crypto/$(ARCH64)/sha2
hardlink path=kernel/misc/$(ARCH64)/skein \
    target=../../../kernel/crypto/$(ARCH64)/skein
$(i386_ONLY)hardlink path=kernel/misc/edonr target=../../kernel/crypto/edonr
$(i386_ONLY)hardlink path=kernel/misc/md5 target=../../kernel/crypto/md5
$(i386_ONLY)hardlink path=kernel/misc/sha1 target=../../kernel/crypto/sha1
$(i386_ONLY)hardlink path=kernel/misc/sha2 target=../../kernel/crypto/sha2
$(i386_ONLY)hardlink path=kernel/misc/skein target=../../kernel/crypto/skein
hardlink path=kernel/socketmod/$(ARCH64)/icmp \
    target=../../../kernel/drv/$(ARCH64)/icmp
hardlink path=kernel/socketmod/$(ARCH64)/rts \
    target=../../../kernel/drv/$(ARCH64)/rts
hardlink path=kernel/socketmod/$(ARCH64)/tcp \
    target=../../../kernel/drv/$(ARCH64)/tcp
hardlink path=kernel/socketmod/$(ARCH64)/udp \
    target=../../../kernel/drv/$(ARCH64)/udp
$(i386_ONLY)hardlink path=kernel/socketmod/icmp target=../../kernel/drv/icmp
$(i386_ONLY)hardlink path=kernel/socketmod/rts target=../../kernel/drv/rts
$(i386_ONLY)hardlink path=kernel/socketmod/tcp target=../../kernel/drv/tcp
$(i386_ONLY)hardlink path=kernel/socketmod/udp target=../../kernel/drv/udp
hardlink path=kernel/strmod/$(ARCH64)/arp \
    target=../../../kernel/drv/$(ARCH64)/arp
```



```
hardlink path=kernel/strmod/$(ARCH64)/icmp \
    target=../../../kernel/drv/$(ARCH64)/icmp
hardlink path=kernel/strmod/$(ARCH64)/ip \
    target=../../../kernel/drv/$(ARCH64)/ip
hardlink path=kernel/strmod/$(ARCH64)/ipsecah \
    target=../../../kernel/drv/$(ARCH64)/ipsecah
hardlink path=kernel/strmod/$(ARCH64)/ipsecesp \
    target=../../../kernel/drv/$(ARCH64)/ipsecesp
hardlink path=kernel/strmod/$(ARCH64)/keysock \
    target=../../../kernel/drv/$(ARCH64)/keysock
hardlink path=kernel/strmod/$(ARCH64)/tcp \
    target=../../../kernel/drv/$(ARCH64)/tcp
hardlink path=kernel/strmod/$(ARCH64)/udp \
    target=../../../kernel/drv/$(ARCH64)/udp
$(i386_ONLY)hardlink path=kernel/strmod/arp target=../../kernel/drv/arp
$(i386_ONLY)hardlink path=kernel/strmod/icmp target=../../kernel/drv/icmp
$(i386_ONLY)hardlink path=kernel/strmod/ip target=../../kernel/drv/ip
$(i386_ONLY)hardlink path=kernel/strmod/ipsecah \
    target=../../kernel/drv/ipsecah
$(i386_ONLY)hardlink path=kernel/strmod/ipsecesp \
    target=../../kernel/drv/ipsecesp
$(i386_ONLY)hardlink path=kernel/strmod/keysock \
    target=../../kernel/drv/keysock
$(i386_ONLY)hardlink path=kernel/strmod/tcp target=../../kernel/drv/tcp
$(i386_ONLY)hardlink path=kernel/strmod/udp target=../../kernel/drv/udp
hardlink path=kernel/sys/$(ARCH64)/autofs \
    target=../../../kernel/fs/$(ARCH64)/autofs
hardlink path=kernel/sys/$(ARCH64)/rpcmod \
    target=../../../kernel/strmod/$(ARCH64)/rpcmod
$(i386_ONLY)hardlink path=kernel/sys/autofs target=../../kernel/fs/autofs
$(i386_ONLY)hardlink path=kernel/sys/rpcmod target=../../kernel/strmod/rpcmod
legacy pkg=SUNWckr \
    desc="core kernel software for a specific instruction-set architecture" \
    name="Core Solaris Kernel (Root)"
license cr_Sun license=cr_Sun
license lic_CDDL license=lic_CDDL
license usr/src/cmd/mdb/common/libstand/THIRDPARTYLICENSE \
    license=usr/src/cmd/mdb/common/libstand/THIRDPARTYLICENSE
license usr/src/common/bzip2/LICENSE license=usr/src/common/bzip2/LICENSE
license usr/src/common/crypto/THIRDPARTYLICENSE.cryptogams \
    license=usr/src/common/crypto/THIRDPARTYLICENSE.cryptogams
$(i386_ONLY)license usr/src/common/crypto/aes/amd64/THIRDPARTYLICENSE.gladman \
    license=usr/src/common/crypto/aes/amd64/THIRDPARTYLICENSE.gladman
$(i386_ONLY)license usr/src/common/crypto/aes/amd64/THIRDPARTYLICENSE.openssl \
    license=usr/src/common/crypto/aes/amd64/THIRDPARTYLICENSE.openssl
license usr/src/common/crypto/ecc/THIRDPARTYLICENSE \
    license=usr/src/common/crypto/ecc/THIRDPARTYLICENSE
$(i386_ONLY)license usr/src/common/crypto/md5/amd64/THIRDPARTYLICENSE \
    license=usr/src/common/crypto/md5/amd64/THIRDPARTYLICENSE
license usr/src/common/mpi/THIRDPARTYLICENSE \
    license=usr/src/common/mpi/THIRDPARTYLICENSE
license usr/src/uts/common/inet/ip/THIRDPARTYLICENSE.rts \
    license=usr/src/uts/common/inet/ip/THIRDPARTYLICENSE.rts
license usr/src/uts/common/inet/tcp/THIRDPARTYLICENSE \
    license=usr/src/uts/common/inet/tcp/THIRDPARTYLICENSE
license usr/src/uts/common/io/THIRDPARTYLICENSE.etheraddr \
    license=usr/src/uts/common/io/THIRDPARTYLICENSE.etheraddr
license usr/src/uts/common/sys/THIRDPARTYLICENSE.icu \
    license=usr/src/uts/common/sys/THIRDPARTYLICENSE.icu
license usr/src/uts/common/sys/THIRDPARTYLICENSE.unicode \
    license=usr/src/uts/common/sys/THIRDPARTYLICENSE.unicode
$(i386_ONLY)license usr/src/uts/intel/io/acpica/THIRDPARTYLICENSE \
    license=usr/src/uts/intel/io/acpica/THIRDPARTYLICENSE
$(i386_ONLY)link path=boot/solaris/bin/root_archive \
    target=../../../usr/sbin/root_archive
link path=dev/dld target=../devices/pseudo/dld@0:ctl
```

```
920 link path=kernel/misc/$(ARCH64)/des \
921     target=../../../kernel/crypto/$(ARCH64)/des
922 $(i386_ONLY)link path=kernel/misc/des target=../../kernel/crypto/des
```

```
**********************************************************
   44337 Mon Oct  7 19:17:17 2013
new/usr/src/uts/common/Makefile.files
4185 New hash algorithm support
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright (c) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2012 Nexenta Systems, Inc. All rights reserved.
# Copyright (c) 2013 by Delphix. All rights reserved.
# Copyright (c) 2013 by Saso Kiselkov. All rights reserved.
#

#
# This Makefile defines all file modules for the directory uts/common
# and its children. These are the source files which may be considered
# common to all SunOS systems.

i386_CORE_OBJS += \
                atomic.o        \
                avintr.o        \
                pic.o

sparc_CORE_OBJS +=

COMMON_CORE_OBJS +=             \
                beep.o          \
                bitset.o        \
                bp_map.o        \
                brand.o         \
                cpucaps.o       \
                cmt.o           \
                cmt_policy.o    \
                cpu.o           \
                cpu_event.o     \
                cpu_intr.o      \
                cpu_pm.o        \
                cpupart.o       \
                cap_util.o      \
                disp.o          \
                group.o         \
                kstat_fr.o      \
                iscsiboot_prop.o        \
                lgrp.o          \
                lgrp_topo.o     \
                mmapobj.o       \
```



```
                mutex.o         \
                page_lock.o     \
                page_retire.o   \
                panic.o         \
                param.o         \
                pg.o            \
                pghw.o          \
                putnext.o       \
                rctl_proc.o     \
                rwlock.o        \
                seg_kmem.o      \
                softint.o       \
                string.o        \
                strtol.o        \
                strtoul.o       \
                strtoll.o       \
                strtoull.o      \
                thread_intr.o   \
                vm_page.o       \
                vm_pagelist.o   \
                zlib_obj.o      \
                clock_tick.o

CORE_OBJS +=    $(COMMON_CORE_OBJS) $($(MACH)_CORE_OBJS)

ZLIB_OBJS =     zutil.o zmod.o zmod_subr.o \
                adler32.o crc32.o deflate.o inffast.o \
                inflate.o inftrees.o trees.o

GENUNIX_OBJS += \
                access.o        \
                acl.o           \
                acl_common.o    \
                adjtime.o       \
                alarm.o         \
                aio_subr.o      \
                auditsys.o      \
                audit_core.o    \
                audit_zone.o    \
                audit_memory.o  \
                autoconf.o      \
                avl.o           \
                bdev_dsort.o    \
                bio.o           \
                bitmap.o        \
                blabel.o        \
                brandsys.o      \
                bz2blocksort.o  \
                bz2compress.o   \
                bz2decompress.o \
                bz2randtable.o  \
                bz2bzlib.o      \
                bz2crctable.o   \
                bz2huffman.o    \
                callb.o         \
                callout.o       \
                chdir.o         \
                chmod.o         \
                chown.o         \
                cladm.o         \
                class.o         \
                clock.o         \
                clock_highres.o \
                clock_realtime.o\
                close.o         \
                compress.o      \
```

```
128                 condvar.o       \
129                 conf.o          \
130                 console.o       \
131                 contract.o      \
132                 copyops.o       \
133                 core.o          \
134                 corectl.o       \
135                 cred.o          \
136                 cs_stubs.o      \
137                 dacf.o          \
138                 dacf_clnt.o     \
139                 damap.o \
140                 cyclic.o        \
141                 ddi.o           \
142                 ddifm.o         \
143                 ddi_hp_impl.o   \
144                 ddi_hp_ndi.o    \
145                 ddi_intr.o      \
146                 ddi_intr_impl.o \
147                 ddi_intr_irm.o  \
148                 ddi_nodeid.o    \
149                 ddi_periodic.o  \
150                 devcfg.o        \
151                 devcache.o      \
152                 device.o        \
153                 devid.o         \
154                 devid_cache.o   \
155                 devid_scsi.o    \
156                 devid_smp.o     \
157                 devpolicy.o     \
158                 disp_lock.o     \
159                 dnlc.o          \
160                 driver.o        \
161                 dumpsubr.o      \
162                 driver_lyr.o    \
163                 dtrace_subr.o   \
164                 errorq.o        \
165                 etheraddr.o     \
166                 evchannels.o    \
167                 exacct.o        \
168                 exacct_core.o   \
169                 exec.o          \
170                 exit.o          \
171                 fbio.o          \
172                 fcntl.o         \
173                 fdbuffer.o      \
174                 fdsync.o        \
175                 fem.o           \
176                 ffs.o           \
177                 fio.o           \
178                 flock.o         \
179                 fm.o            \
180                 fork.o          \
181                 vpm.o           \
182                 fs_reparse.o    \
183                 fs_subr.o       \
184                 fsflush.o       \
185                 ftrace.o        \
186                 getcwd.o        \
187                 getdents.o      \
188                 getloadavg.o    \
189                 getpagesizes.o  \
190                 getpid.o        \
191                 gfs.o           \
192                 rusagesys.o     \
193                 gid.o           \
```



```
194                 groups.o        \
195                 grow.o          \
196                 hat_refmod.o    \
197                 id32.o          \
198                 id_space.o      \
199                 inet_ntop.o     \
200                 instance.o      \
201                 ioctl.o         \
202                 ip_cksum.o      \
203                 issetugid.o     \
204                 ippconf.o       \
205                 kcpc.o          \
206                 kdi.o           \
207                 kiconv.o        \
208                 klpd.o          \
209                 kmem.o          \
210                 ksyms_snapshot.o        \
211                 l_strplumb.o    \
212                 labelsys.o      \
213                 link.o          \
214                 list.o          \
215                 lockstat_subr.o \
216                 log_sysevent.o  \
217                 logsubr.o       \
218                 lookup.o        \
219                 lseek.o         \
220                 ltos.o          \
221                 lwp.o           \
222                 lwp_create.o    \
223                 lwp_info.o      \
224                 lwp_self.o      \
225                 lwp_sobj.o      \
226                 lwp_timer.o     \
227                 lwpsys.o        \
228                 main.o          \
229                 mmapobjsys.o    \
230                 memcntl.o       \
231                 memstr.o        \
232                 lgrpsys.o       \
233                 mkdir.o         \
234                 mknod.o         \
235                 mount.o         \
236                 move.o          \
237                 msacct.o        \
238                 multidata.o     \
239                 nbmlock.o       \
240                 ndifm.o         \
241                 nice.o          \
242                 netstack.o      \
243                 ntptime.o       \
244                 nvpair.o        \
245                 nvpair_alloc_system.o   \
246                 nvpair_alloc_fixed.o    \
247                 fnvpair.o       \
248                 octet.o         \
249                 open.o          \
250                 p_online.o      \
251                 pathconf.o      \
252                 pathname.o      \
253                 pause.o         \
254                 serializer.o    \
255                 pci_intr_lib.o  \
256                 pci_cap.o       \
257                 pcifm.o         \
258                 pgrp.o          \
259                 pgrpsys.o       \
```

```
260                 pid.o           \
261                 pkp_hash.o      \
262                 policy.o        \
263                 poll.o          \
264                 pool.o          \
265                 pool_pset.o     \
266                 port_subr.o     \
267                 ppriv.o         \
268                 printf.o        \
269                 priocntl.o      \
270                 priv.o          \
271                 priv_const.o    \
272                 proc.o          \
273                 procset.o       \
274                 processor_bind.o        \
275                 processor_info.o        \
276                 profil.o        \
277                 project.o       \
278                 qsort.o         \
279                 rctl.o          \
280                 rctlsys.o       \
281                 readlink.o      \
282                 refstr.o        \
283                 rename.o        \
284                 resolvepath.o   \
285                 retire_store.o  \
286                 process.o       \
287                 rlimit.o        \
288                 rmap.o          \
289                 rw.o            \
290                 rwstlock.o      \
291                 sad_conf.o      \
292                 sid.o           \
293                 sidsys.o        \
294                 sched.o         \
295                 schedctl.o      \
296                 sctp_crc32.o    \
297                 seg_dev.o       \
298                 seg_kp.o        \
299                 seg_kpm.o       \
300                 seg_map.o       \
301                 seg_vn.o        \
302                 seg_spt.o       \
303                 semaphore.o     \
304                 sendfile.o      \
305                 session.o       \
306                 share.o         \
307                 shuttle.o       \
308                 sig.o           \
309                 sigaction.o     \
310                 sigaltstack.o   \
311                 signotify.o     \
312                 sigpending.o    \
313                 sigprocmask.o   \
314                 sigqueue.o      \
315                 sigsendset.o    \
316                 sigsuspend.o    \
317                 sigtimedwait.o  \
318                 sleepq.o        \
319                 sock_conf.o     \
320                 space.o         \
321                 sscanf.o        \
322                 stat.o          \
323                 statfs.o        \
324                 statvfs.o       \
325                 stol.o          \
```



```
326                 str_conf.o      \
327                 strcalls.o      \
328                 stream.o        \
329                 streamio.o      \
330                 strext.o        \
331                 strsubr.o       \
332                 strsun.o        \
333                 subr.o          \
334                 sunddi.o        \
335                 sunmdi.o        \
336                 sunndi.o        \
337                 sunpci.o        \
338                 sunpm.o         \
339                 sundlpi.o       \
340                 suntpi.o        \
341                 swap_subr.o     \
342                 swap_vnops.o    \
343                 symlink.o       \
344                 sync.o          \
345                 sysclass.o      \
346                 sysconfig.o     \
347                 sysent.o        \
348                 sysfs.o         \
349                 systeminfo.o    \
350                 task.o          \
351                 taskq.o         \
352                 tasksys.o       \
353                 time.o          \
354                 timer.o         \
355                 times.o         \
356                 timers.o        \
357                 thread.o        \
358                 tlabel.o        \
359                 tnf_res.o       \
360                 turnstile.o     \
361                 tty_common.o    \
362                 u8_textprep.o   \
363                 uadmin.o        \
364                 uconv.o         \
365                 ucredsys.o      \
366                 uid.o           \
367                 umask.o         \
368                 umount.o        \
369                 uname.o         \
370                 unix_bb.o       \
371                 unlink.o        \
372                 urw.o           \
373                 utime.o         \
374                 utssys.o        \
375                 uucopy.o        \
376                 vfs.o           \
377                 vfs_conf.o      \
378                 vmem.o          \
379                 vm_anon.o       \
380                 vm_as.o         \
381                 vm_meter.o      \
382                 vm_pageout.o    \
383                 vm_pvn.o        \
384                 vm_rm.o         \
385                 vm_seg.o        \
386                 vm_subr.o       \
387                 vm_swap.o       \
388                 vm_usage.o      \
389                 vnode.o         \
390                 vuid_queue.o    \
391                 vuid_store.o    \
```

```
392                 waitq.o         \
393                 watchpoint.o    \
394                 yield.o         \
395                 scsi_confdata.o \
396                 xattr.o         \
397                 xattr_common.o  \
398                 xdr_mblk.o      \
399                 xdr_mem.o       \
400                 xdr.o           \
401                 xdr_array.o     \
402                 xdr_refer.o     \
403                 xhat.o          \
404                 zone.o

406 #
407 #       Stubs for the stand-alone linker/loader
408 #
409 sparc_GENSTUBS_OBJS =   \
410         kobj_stubs.o

412 i386_GENSTUBS_OBJS =

414 COMMON_GENSTUBS_OBJS =

416 GENSTUBS_OBJS += $(COMMON_GENSTUBS_OBJS) $($(MACH)_GENSTUBS_OBJS)

418 #
419 #       DTrace and DTrace Providers
420 #
421 DTRACE_OBJS += dtrace.o dtrace_isa.o dtrace_asm.o

423 SDT_OBJS += sdt_subr.o

425 PROFILE_OBJS += profile.o

427 SYSTRACE_OBJS += systrace.o

429 LOCKSTAT_OBJS += lockstat.o

431 FASTTRAP_OBJS += fasttrap.o fasttrap_isa.o

433 DCPC_OBJS += dcpc.o

435 #
436 #       Driver (pseudo-driver) Modules
437 #
438 IPP_OBJS +=     ippctl.o

440 AUDIO_OBJS += audio_client.o audio_ddi.o audio_engine.o \
441             audio_fltdata.o audio_format.o audio_ctrl.o \
442             audio_grc3.o audio_output.o audio_input.o \
443             audio_oss.o audio_sun.o

445 AUDIOEMU10K_OBJS += audioemu10k.o

447 AUDIOENS_OBJS += audioens.o

449 AUDIOVIA823X_OBJS += audiovia823x.o

451 AUDIOVIA97_OBJS += audiovia97.o

453 AUDIO1575_OBJS += audio1575.o

455 AUDIO810_OBJS += audio810.o

457 AUDIOCMI_OBJS += audiocmi.o
```



```
459 AUDIOCMIHD_OBJS += audiocmihd.o

461 AUDIOHD_OBJS += audiohd.o

463 AUDIOIXP_OBJS += audioixp.o

465 AUDIOLS_OBJS += audiols.o

467 AUDIOP16X_OBJS += audiop16x.o

469 AUDIOPCI_OBJS += audiopci.o

471 AUDIOSOLO_OBJS += audiosolo.o

473 AUDIOTS_OBJS += audiots.o

475 AC97_OBJS += ac97.o ac97_ad.o ac97_alc.o ac97_cmi.o

477 BLKDEV_OBJS += blkdev.o

479 CARDBUS_OBJS += cardbus.o cardbus_hp.o cardbus_cfg.o

481 CONSKBD_OBJS += conskbd.o

483 CONSMS_OBJS +=  consms.o

485 OLDPTY_OBJS +=  tty_ptyconf.o

487 PTC_OBJS +=     tty_pty.o

489 PTSL_OBJS +=    tty_pts.o

491 PTM_OBJS +=     ptm.o

493 MII_OBJS +=     mii.o mii_cicada.o mii_natsemi.o mii_intel.o mii_qualsemi.o \
494                 mii_marvell.o mii_realtek.o mii_other.o

496 PTS_OBJS +=     pts.o

498 PTY_OBJS +=     ptms_conf.o

500 SAD_OBJS +=     sad.o

502 MD4_OBJS +=     md4.o md4_mod.o

504 MD5_OBJS +=     md5.o md5_mod.o

506 SHA1_OBJS +=    sha1.o sha1_mod.o

508 SHA2_OBJS +=    sha2.o sha2_mod.o

510 SKEIN_OBJS +=   skein.o skein_block.o skein_mod.o

512 EDONR_OBJS +=   edonr.o edonr_mod.o

514 IPGPC_OBJS +=   classifierddi.o classifier.o filters.o trie.o table.o \
515                 ba_table.o

517 DSCPMK_OBJS +=  dscpmk.o dscpmkddi.o

519 DLCOSMK_OBJS += dlcosmk.o dlcosmkddi.o

521 FLOWACCT_OBJS +=        flowacctddi.o flowacct.o

523 TOKENMT_OBJS += tokenmt.o tokenmtddi.o
```

```
525 TSWTCL_OBJS +=  tswtcl.o tswtclddi.o

527 ARP_OBJS +=     arpddi.o

529 ICMP_OBJS +=    icmpddi.o

531 ICMP6_OBJS +=   icmp6ddi.o

533 RTS_OBJS +=     rtsddi.o

535 IP_ICMP_OBJS =  icmp.o icmp_opt_data.o
536 IP_RTS_OBJS =   rts.o rts_opt_data.o
537 IP_TCP_OBJS =   tcp.o tcp_fusion.o tcp_opt_data.o tcp_sack.o tcp_stats.o \
538                 tcp_misc.o tcp_timers.o tcp_time_wait.o tcp_tpi.o tcp_output.o \
539                 tcp_input.o tcp_socket.o tcp_bind.o tcp_cluster.o tcp_tunables.o
540 IP_UDP_OBJS =   udp.o udp_opt_data.o udp_tunables.o udp_stats.o
541 IP_SCTP_OBJS =  sctp.o sctp_opt_data.o sctp_output.o \
542                 sctp_init.o sctp_input.o sctp_cookie.o \
543                 sctp_conn.o sctp_error.o sctp_snmp.o \
544                 sctp_tunables.o sctp_shutdown.o sctp_common.o \
545                 sctp_timer.o sctp_heartbeat.o sctp_hash.o \
546                 sctp_bind.o sctp_notify.o sctp_asconf.o \
547                 sctp_addr.o tn_ipopt.o tnet.o ip_netinfo.o \
548                 sctp_misc.o
549 IP_ILB_OBJS =   ilb.o ilb_nat.o ilb_conn.o ilb_alg_hash.o ilb_alg_rr.o

551 IP_OBJS +=      igmp.o ipmp.o ip.o ip6.o ip6_asp.o ip6_if.o ip6_ire.o \
552                 ip6_rts.o ip_if.o ip_ire.o ip_listutils.o ip_mroute.o \
553                 ip_multi.o ip2mac.o ip_ndp.o ip_rts.o ip_srcid.o \
554                 ipddi.o ipdrop.o mi.o nd.o tunables.o optcom.o snmpcom.o \
555                 ipsec_loader.o spd.o ipclassifier.o inet_common.o ip_squeue.o \
556                 squeue.o ip_sadb.o ip_ftable.o proto_set.o radix.o ip_dummy.o \
557                 ip_helper_stream.o ip_tunables.o \
558                 ip_output.o ip_input.o ip6_input.o ip6_output.o ip_arp.o \
559                 conn_opt.o ip_attr.o ip_dce.o \
560                 $(IP_ICMP_OBJS) \
561                 $(IP_RTS_OBJS) \
562                 $(IP_TCP_OBJS) \
563                 $(IP_UDP_OBJS) \
564                 $(IP_SCTP_OBJS) \
565                 $(IP_ILB_OBJS)

567 IP6_OBJS +=     ip6ddi.o

569 HOOK_OBJS +=    hook.o

571 NETI_OBJS +=    neti_impl.o neti_mod.o neti_stack.o

573 KEYSOCK_OBJS += keysockddi.o keysock.o keysock_opt_data.o

575 IPNET_OBJS +=   ipnet.o ipnet_bpf.o

577 SPDSOCK_OBJS += spdsockddi.o spdsock.o spdsock_opt_data.o

579 IPSECESP_OBJS += ipsecespddi.o ipsecesp.o

581 IPSECAH_OBJS += ipsecahddi.o ipsecah.o sadb.o

583 SPPP_OBJS +=    sppp.o sppp_dlpi.o sppp_mod.o s_common.o

585 SPPPTUN_OBJS += sppptun.o sppptun_mod.o

587 SPPPASYN_OBJS += spppasyn.o spppasyn_mod.o

589 SPPPCOMP_OBJS += spppcomp.o spppcomp_mod.o deflate.o bsd-comp.o vjcompress.o \
```



```
590                 zlib.o

592 TCP_OBJS +=     tcpddi.o

594 TCP6_OBJS +=    tcp6ddi.o

596 NCA_OBJS +=     ncaddi.o

598 SDP_SOCK_MOD_OBJS += sockmod_sdp.o socksdp.o socksdpsubr.o

600 SCTP_SOCK_MOD_OBJS += sockmod_sctp.o socksctp.o socksctpsubr.o

602 PFP_SOCK_MOD_OBJS += sockmod_pfp.o

604 RDS_SOCK_MOD_OBJS += sockmod_rds.o

606 RDS_OBJS +=     rdsddi.o rdssubr.o rds_opt.o rds_ioctl.o

608 RDSIB_OBJS +=   rdsib.o rdsib_ib.o rdsib_cm.o rdsib_ep.o rdsib_buf.o \
609                 rdsib_debug.o rdsib_sc.o

611 RDSV3_OBJS +=   af_rds.o rdsv3_ddi.o bind.o loop.o threads.o connection.o \
612                 transport.o cong.o sysctl.o message.o rds_recv.o send.o \
613                 stats.o info.o page.o rdma_transport.o ib_ring.o ib_rdma.o \
614                 ib_recv.o ib.o ib_send.o ib_sysctl.o ib_stats.o ib_cm.o \
615                 rdsv3_sc.o rdsv3_debug.o rdsv3_impl.o rdma.o rdsv3_af_thr.o

617 ISER_OBJS +=    iser.o iser_cm.o iser_cq.o iser_ib.o iser_idm.o \
618                 iser_resource.o iser_xfer.o

620 UDP_OBJS +=     udpddi.o

622 UDP6_OBJS +=    udp6ddi.o

624 SY_OBJS +=      gentty.o

626 TCO_OBJS +=     ticots.o

628 TCOO_OBJS +=    ticotsord.o

630 TCL_OBJS +=     ticlts.o

632 TL_OBJS +=      tl.o

634 DUMP_OBJS +=    dump.o

636 BPF_OBJS +=     bpf.o bpf_filter.o bpf_mod.o bpf_dlt.o bpf_mac.o

638 CLONE_OBJS +=   clone.o

640 CN_OBJS +=      cons.o

642 DLD_OBJS +=     dld_drv.o dld_proto.o dld_str.o dld_flow.o

644 DLS_OBJS +=     dls.o dls_link.o dls_mod.o dls_stat.o dls_mgmt.o

646 GLD_OBJS +=     gld.o gldutil.o

648 MAC_OBJS +=     mac.o  mac_bcast.o mac_client.o mac_datapath_setup.o mac_flow.o
649                 mac_hio.o mac_mod.o mac_ndd.o mac_provider.o mac_sched.o \
650                 mac_protect.o mac_soft_ring.o mac_stat.o mac_util.o

652 MAC_6TO4_OBJS +=        mac_6to4.o

654 MAC_ETHER_OBJS +=       mac_ether.o
```

```
656 MAC_IPV4_OBJS +=        mac_ipv4.o

658 MAC_IPV6_OBJS +=        mac_ipv6.o

660 MAC_WIFI_OBJS +=        mac_wifi.o

662 MAC_IB_OBJS +=          mac_ib.o

664 IPTUN_OBJS +=   iptun_dev.o iptun_ctl.o iptun.o

666 AGGR_OBJS +=    aggr_dev.o aggr_ctl.o aggr_grp.o aggr_port.o \
667                 aggr_send.o aggr_recv.o aggr_lacp.o

669 SOFTMAC_OBJS += softmac_main.o softmac_ctl.o softmac_capab.o \
670                 softmac_dev.o softmac_stat.o softmac_pkt.o softmac_fp.o

672 NET80211_OBJS += net80211.o net80211_proto.o net80211_input.o \
673                  net80211_output.o net80211_node.o net80211_crypto.o \
674                  net80211_crypto_none.o net80211_crypto_wep.o net80211_ioctl.o \
675                  net80211_crypto_tkip.o net80211_crypto_ccmp.o  \
676                  net80211_ht.o

678 VNIC_OBJS +=    vnic_ctl.o vnic_dev.o

680 SIMNET_OBJS +=  simnet.o

682 IB_OBJS +=      ibnex.o ibnex_ioctl.o ibnex_hca.o

684 IBCM_OBJS +=    ibcm_impl.o ibcm_sm.o ibcm_ti.o ibcm_utils.o ibcm_path.o \
685                 ibcm_arp.o ibcm_arp_link.o

687 IBDM_OBJS +=    ibdm.o

689 IBDMA_OBJS +=   ibdma.o

691 IBMF_OBJS +=    ibmf.o ibmf_impl.o ibmf_dr.o ibmf_wqe.o ibmf_ud_dest.o ibmf_mod.
692                 ibmf_send.o ibmf_recv.o ibmf_handlers.o ibmf_trans.o \
693                 ibmf_timers.o ibmf_msg.o ibmf_utils.o ibmf_rmpp.o \
694                 ibmf_saa.o ibmf_saa_impl.o ibmf_saa_utils.o ibmf_saa_events.o

696 IBTL_OBJS +=    ibtl_impl.o ibtl_util.o ibtl_mem.o ibtl_handlers.o ibtl_qp.o \
697                 ibtl_cq.o ibtl_wr.o ibtl_hca.o ibtl_chan.o ibtl_cm.o \
698                 ibtl_mcg.o ibtl_ibnex.o ibtl_srq.o ibtl_part.o

700 TAVOR_OBJS +=   tavor.o tavor_agents.o tavor_cfg.o tavor_ci.o tavor_cmd.o \
701                 tavor_cq.o tavor_event.o tavor_ioctl.o tavor_misc.o \
702                 tavor_mr.o tavor_qp.o tavor_qpmod.o tavor_rsrc.o \
703                 tavor_srq.o tavor_stats.o tavor_umap.o tavor_wr.o

705 HERMON_OBJS +=  hermon.o hermon_agents.o hermon_cfg.o hermon_ci.o hermon_cmd.o \
706                 hermon_cq.o hermon_event.o hermon_ioctl.o hermon_misc.o \
707                 hermon_mr.o hermon_qp.o hermon_qpmod.o hermon_rsrc.o \
708                 hermon_srq.o hermon_stats.o hermon_umap.o hermon_wr.o \
709                 hermon_fcoib.o hermon_fm.o

711 DAPLT_OBJS +=   daplt.o

713 SOL_OFS_OBJS += sol_cma.o sol_ib_cma.o sol_uobj.o \
714                 sol_ofs_debug_util.o sol_ofs_gen_util.o \
715                 sol_kverbs.o

717 SOL_UCMA_OBJS +=        sol_ucma.o

719 SOL_UVERBS_OBJS +=      sol_uverbs.o sol_uverbs_comp.o sol_uverbs_event.o \
720                         sol_uverbs_hca.o sol_uverbs_qp.o
```



```
722 SOL_UMAD_OBJS += sol_umad.o

724 KSTAT_OBJS +=   kstat.o

726 KSYMS_OBJS +=   ksyms.o

728 INSTANCE_OBJS += inst_sync.o

730 IWSCN_OBJS +=   iwscons.o

732 LOFI_OBJS +=    lofi.o LzmaDec.o

734 FSSNAP_OBJS +=  fssnap.o

736 FSSNAPIF_OBJS += fssnap_if.o

738 MM_OBJS +=      mem.o

740 PHYSMEM_OBJS += physmem.o

742 OPTIONS_OBJS += options.o

744 WINLOCK_OBJS += winlockio.o

746 PM_OBJS +=      pm.o
747 SRN_OBJS +=     srn.o

749 PSEUDO_OBJS +=  pseudonex.o

751 RAMDISK_OBJS += ramdisk.o

753 LLC1_OBJS += llc1.o

755 USBKBM_OBJS += usbkbm.o

757 USBWCM_OBJS += usbwcm.o

759 BOFI_OBJS += bofi.o

761 HID_OBJS += hid.o

763 HWA_RC_OBJS += hwarc.o

765 USBSKEL_OBJS += usbskel.o

767 USBVC_OBJS += usbvc.o usbvc_v4l2.o

769 HIDPARSER_OBJS += hidparser.o

771 USB_AC_OBJS += usb_ac.o

773 USB_AS_OBJS += usb_as.o

775 USB_AH_OBJS += usb_ah.o

777 USBMS_OBJS += usbms.o

779 USBPRN_OBJS += usbprn.o

781 UGEN_OBJS += ugen.o

783 USBSER_OBJS += usbser.o usbser_rseq.o

785 USBSACM_OBJS += usbsacm.o

787 USBSER_KEYSPAN_OBJS += usbser_keyspan.o keyspan_dsd.o keyspan_pipe.o
```

```
789 USBS49_FW_OBJS += keyspan_49fw.o

791 USBSPRL_OBJS += usbser_pl2303.o pl2303_dsd.o

793 WUSB_CA_OBJS += wusb_ca.o

795 USBFTDI_OBJS += usbser_uftdi.o uftdi_dsd.o

797 USBECM_OBJS += usbecm.o

799 WC_OBJS += wscons.o vcons.o

801 VCONS_CONF_OBJS += vcons_conf.o

803 SCSI_OBJS +=    scsi_capabilities.o scsi_confsubr.o scsi_control.o \
804                 scsi_data.o scsi_fm.o scsi_hba.o scsi_reset_notify.o \
805                 scsi_resource.o scsi_subr.o scsi_transport.o scsi_watch.o \
806                 smp_transport.o

808 SCSI_VHCI_OBJS +=               scsi_vhci.o mpapi_impl.o scsi_vhci_tpgs.o

810 SCSI_VHCI_F_SYM_OBJS +=         sym.o

812 SCSI_VHCI_F_TPGS_OBJS +=        tpgs.o

814 SCSI_VHCI_F_ASYM_SUN_OBJS +=    asym_sun.o

816 SCSI_VHCI_F_SYM_HDS_OBJS +=     sym_hds.o

818 SCSI_VHCI_F_TAPE_OBJS +=        tape.o

820 SCSI_VHCI_F_TPGS_TAPE_OBJS +=   tpgs_tape.o

822 SGEN_OBJS +=    sgen.o

824 SMP_OBJS +=     smp.o

826 SATA_OBJS +=    sata.o

828 USBA_OBJS +=    hcdi.o  usba.o  usbai.o hubdi.o parser.o genconsole.o \
829                 usbai_pipe_mgmt.o usbai_req.o usbai_util.o usbai_register.o \
830                 usba_devdb.o usba10_calls.o usba_ugen.o whcdi.o wa.o
831 USBA_WITHOUT_WUSB_OBJS +=       hcdi.o  usba.o  usbai.o hubdi.o parser.o gencons
832                 usbai_pipe_mgmt.o usbai_req.o usbai_util.o usbai_register.o \
833                 usba_devdb.o usba10_calls.o usba_ugen.o

835 USBA10_OBJS +=  usba10.o

837 RSM_OBJS +=     rsm.o   rsmka_pathmanager.o     rsmka_util.o

839 RSMOPS_OBJS +=  rsmops.o

841 S1394_OBJS +=   t1394.o t1394_errmsg.o s1394.o s1394_addr.o s1394_asynch.o \
842                 s1394_bus_reset.o s1394_cmp.o s1394_csr.o s1394_dev_disc.o \
843                 s1394_fa.o s1394_fcp.o \
844                 s1394_hotplug.o s1394_isoch.o s1394_misc.o h1394.o nx1394.o

846 HCI1394_OBJS += hci1394.o hci1394_async.o hci1394_attach.o hci1394_buf.o \
847                 hci1394_csr.o hci1394_detach.o hci1394_extern.o \
848                 hci1394_ioctl.o hci1394_isoch.o hci1394_isr.o \
849                 hci1394_ixl_comp.o hci1394_ixl_isr.o hci1394_ixl_misc.o \
850                 hci1394_ixl_update.o hci1394_misc.o hci1394_ohci.o \
851                 hci1394_q.o hci1394_s1394if.o hci1394_tlabel.o \
852                 hci1394_tlist.o hci1394_vendor.o
```



```
854 AV1394_OBJS +=  av1394.o av1394_as.o av1394_async.o av1394_cfgrom.o \
855                 av1394_cmp.o av1394_fcp.o av1394_isoch.o av1394_isoch_chan.o \
856                 av1394_isoch_recv.o av1394_isoch_xmit.o av1394_list.o \
857                 av1394_queue.o

859 DCAM1394_OBJS += dcam.o dcam_frame.o dcam_param.o dcam_reg.o \
860                 dcam_ring_buff.o

862 SCSA1394_OBJS += hba.o sbp2_driver.o sbp2_bus.o

864 SBP2_OBJS += cfgrom.o sbp2.o

866 PMODEM_OBJS += pmodem.o pmodem_cis.o cis.o cis_callout.o cis_handlers.o cis_para

868 DSW_OBJS +=     dsw.o dsw_dev.o ii_tree.o

870 NCALL_OBJS +=   ncall.o \
871                 ncall_stub.o

873 RDC_OBJS +=     rdc.o \
874                 rdc_dev.o \
875                 rdc_io.o \
876                 rdc_clnt.o \
877                 rdc_prot_xdr.o \
878                 rdc_svc.o \
879                 rdc_bitmap.o \
880                 rdc_health.o \
881                 rdc_subr.o \
882                 rdc_diskq.o

884 RDCSRV_OBJS +=  rdcsrv.o

886 RDCSTUB_OBJS += rdc_stub.o

888 SDBC_OBJS +=    sd_bcache.o \
889                 sd_bio.o \
890                 sd_conf.o \
891                 sd_ft.o \
892                 sd_hash.o \
893                 sd_io.o \
894                 sd_misc.o \
895                 sd_pcu.o \
896                 sd_tdaemon.o \
897                 sd_trace.o \
898                 sd_iob_impl0.o \
899                 sd_iob_impl1.o \
900                 sd_iob_impl2.o \
901                 sd_iob_impl3.o \
902                 sd_iob_impl4.o \
903                 sd_iob_impl5.o \
904                 sd_iob_impl6.o \
905                 sd_iob_impl7.o \
906                 safestore.o \
907                 safestore_ram.o

909 NSCTL_OBJS +=   nsctl.o \
910                 nsc_cache.o \
911                 nsc_disk.o \
912                 nsc_dev.o \
913                 nsc_freeze.o \
914                 nsc_gen.o \
915                 nsc_mem.o \
916                 nsc_ncallio.o \
917                 nsc_power.o \
918                 nsc_resv.o \
919                 nsc_rmspin.o \
```

```
920                 nsc_solaris.o \
921                 nsc_trap.o \
922                 nsc_list.o
923 UNISTAT_OBJS += spuni.o \
924                 spcs_s_k.o

926 NSKERN_OBJS +=  nsc_ddi.o \
927                 nsc_proc.o \
928                 nsc_raw.o \
929                 nsc_thread.o \
930                 nskernd.o

932 SV_OBJS +=      sv.o

934 PMCS_OBJS += pmcs_attach.o pmcs_ds.o pmcs_intr.o pmcs_nvram.o pmcs_sata.o \
935                 pmcs_scsa.o pmcs_smhba.o pmcs_subr.o pmcs_fwlog.o

937 PMCS8001FW_C_OBJS +=    pmcs_fw_hdr.o
938 PMCS8001FW_OBJS +=              $(PMCS8001FW_C_OBJS) SPCBoot.o ila.o firmware.o

940 #
941 #       Build up defines and paths.

943 ST_OBJS +=      st.o    st_conf.o

945 EMLXS_OBJS +=   emlxs_clock.o emlxs_dfc.o emlxs_dhchap.o emlxs_diag.o \
946                 emlxs_download.o emlxs_dump.o emlxs_els.o emlxs_event.o \
947                 emlxs_fcf.o emlxs_fcp.o emlxs_fct.o emlxs_hba.o emlxs_ip.o \
948                 emlxs_mbox.o emlxs_mem.o emlxs_msg.o emlxs_node.o \
949                 emlxs_pkt.o emlxs_sli3.o emlxs_sli4.o emlxs_solaris.o \
950                 emlxs_thread.o

952 EMLXS_FW_OBJS +=        emlxs_fw.o

954 OCE_OBJS +=     oce_buf.o oce_fm.o oce_gld.o oce_hw.o oce_intr.o oce_main.o \
955                 oce_mbx.o oce_mq.o oce_queue.o oce_rx.o oce_stat.o oce_tx.o \
956                 oce_utils.o

958 FCT_OBJS += discovery.o fct.o

960 QLT_OBJS += 2400.o 2500.o 8100.o qlt.o qlt_dma.o

962 SRPT_OBJS += srpt_mod.o srpt_ch.o srpt_cm.o srpt_ioc.o srpt_stp.o

964 FCOE_OBJS += fcoe.o fcoe_eth.o fcoe_fc.o

966 FCOET_OBJS += fcoet.o fcoet_eth.o fcoet_fc.o

968 FCOEI_OBJS += fcoei.o fcoei_eth.o fcoei_lv.o

970 ISCSIT_SHARED_OBJS += \
971                 iscsit_common.o

973 ISCSIT_OBJS +=  $(ISCSIT_SHARED_OBJS) \
974                 iscsit.o iscsit_tgt.o iscsit_sess.o iscsit_login.o \
975                 iscsit_text.o iscsit_isns.o iscsit_radiusauth.o \
976                 iscsit_radiuspacket.o iscsit_auth.o iscsit_authclient.o

978 PPPT_OBJS +=    alua_ic_if.o pppt.o pppt_msg.o pppt_tgt.o

980 STMF_OBJS += lun_map.o stmf.o

982 STMF_SBD_OBJS += sbd.o sbd_scsi.o sbd_pgr.o sbd_zvol.o

984 SYSMSG_OBJS +=  sysmsg.o
```



```
986 SES_OBJS +=     ses.o ses_sen.o ses_safte.o ses_ses.o

988 TNF_OBJS +=     tnf_buf.o       tnf_trace.o     tnf_writer.o    trace_init.o \
989                 trace_funcs.o   tnf_probe.o     tnf.o

991 LOGINDMUX_OBJS += logindmux.o

993 DEVINFO_OBJS += devinfo.o

995 DEVPOLL_OBJS += devpoll.o

997 DEVPOOL_OBJS += devpool.o

999 I8042_OBJS +=   i8042.o

1001 KB8042_OBJS +=  \
1002                at_keyprocess.o \
1003                kb8042.o        \
1004                kb8042_keytables.o

1006 MOUSE8042_OBJS += mouse8042.o

1008 FDC_OBJS +=     fdc.o

1010 ASY_OBJS +=     asy.o

1012 ECPP_OBJS +=    ecpp.o

1014 VUIDM3P_OBJS += vuidmice.o vuidm3p.o

1016 VUIDM4P_OBJS += vuidmice.o vuidm4p.o

1018 VUIDM5P_OBJS += vuidmice.o vuidm5p.o

1020 VUIDPS2_OBJS += vuidmice.o vuidps2.o

1022 HPCSVC_OBJS += hpcsvc.o

1024 PCIE_MISC_OBJS += pcie.o pcie_fault.o pcie_hp.o pciehpc.o pcishpc.o pcie_pwr.o p

1026 PCIHPNEXUS_OBJS += pcihp.o

1028 OPENEEPR_OBJS += openprom.o

1030 RANDOM_OBJS += random.o

1032 PSHOT_OBJS += pshot.o

1034 GEN_DRV_OBJS += gen_drv.o

1036 TCLIENT_OBJS += tclient.o

1038 TPHCI_OBJS += tphci.o

1040 TVHCI_OBJS += tvhci.o

1042 EMUL64_OBJS += emul64.o emul64_bsd.o

1044 FCP_OBJS += fcp.o

1046 FCIP_OBJS += fcip.o

1048 FCSM_OBJS += fcsm.o

1050 FCTL_OBJS += fctl.o
```

```
1052 FP_OBJS += fp.o

1054 QLC_OBJS += ql_api.o ql_debug.o ql_hba_fru.o ql_init.o ql_iocb.o ql_ioctl.o \
1055         ql_isr.o ql_mbx.o ql_nx.o ql_xioctl.o ql_fw_table.o

1057 QLC_FW_2200_OBJS += ql_fw_2200.o

1059 QLC_FW_2300_OBJS += ql_fw_2300.o

1061 QLC_FW_2400_OBJS += ql_fw_2400.o

1063 QLC_FW_2500_OBJS += ql_fw_2500.o

1065 QLC_FW_6322_OBJS += ql_fw_6322.o

1067 QLC_FW_8100_OBJS += ql_fw_8100.o

1069 QLGE_OBJS += qlge.o qlge_dbg.o qlge_flash.o qlge_fm.o qlge_gld.o qlge_mpi.o

1071 ZCONS_OBJS += zcons.o

1073 NV_SATA_OBJS += nv_sata.o

1075 SI3124_OBJS += si3124.o

1077 AHCI_OBJS += ahci.o

1079 PCIIDE_OBJS += pci-ide.o

1081 PCEPP_OBJS += pcepp.o

1083 CPC_OBJS += cpc.o

1085 CPUID_OBJS += cpuid_drv.o

1087 SYSEVENT_OBJS += sysevent.o

1089 BL_OBJS += bl.o

1091 DRM_OBJS += drm_sunmod.o drm_kstat.o drm_agpsupport.o \
1092             drm_auth.o drm_bufs.o drm_context.o drm_dma.o \
1093             drm_drawable.o drm_drv.o drm_fops.o drm_ioctl.o drm_irq.o \
1094             drm_lock.o drm_memory.o drm_msg.o drm_pci.o drm_scatter.o \
1095             drm_cache.o drm_gem.o drm_mm.o ati_pcigart.o

1097 FM_OBJS += devfm.o devfm_machdep.o

1099 RTLS_OBJS +=    rtls.o

1101 #
1102 #                       exec modules
1103 #
1104 AOUTEXEC_OBJS +=aout.o

1106 ELFEXEC_OBJS += elf.o elf_notes.o old_notes.o

1108 INTPEXEC_OBJS +=intp.o

1110 SHBINEXEC_OBJS +=shbin.o

1112 JAVAEXEC_OBJS +=java.o

1114 #
1115 #                       file system modules
1116 #
1117 AUTOFS_OBJS +=  auto_vfsops.o auto_vnops.o auto_subr.o auto_xdr.o auto_sys.o
```



```
1119 CACHEFS_OBJS += cachefs_cnode.o         cachefs_cod.o \
1120                 cachefs_dir.o           cachefs_dlog.o  cachefs_filegrp.o \
1121                 cachefs_fscache.o       cachefs_ioctl.o cachefs_log.o \
1122                 cachefs_module.o \
1123                 cachefs_noopc.o         cachefs_resource.o \
1124                 cachefs_strict.o \
1125                 cachefs_subr.o          cachefs_vfsops.o \
1126                 cachefs_vnops.o

1128 DCFS_OBJS +=    dc_vnops.o

1130 DEVFS_OBJS +=   devfs_subr.o    devfs_vfsops.o  devfs_vnops.o

1132 DEV_OBJS  +=    sdev_subr.o     sdev_vfsops.o   sdev_vnops.o    \
1133                 sdev_ptsops.o   sdev_zvolops.o  sdev_comm.o     \
1134                 sdev_profile.o  sdev_ncache.o   sdev_netops.o   \
1135                 sdev_ipnetops.o \
1136                 sdev_vtops.o

1138 CTFS_OBJS +=    ctfs_all.o ctfs_cdir.o ctfs_ctl.o ctfs_event.o \
1139                 ctfs_latest.o ctfs_root.o ctfs_sym.o ctfs_tdir.o ctfs_tmpl.o

1141 OBJFS_OBJS +=   objfs_vfs.o     objfs_root.o    objfs_common.o \
1142                 objfs_odir.o    objfs_data.o

1144 FDFS_OBJS +=    fdops.o

1146 FIFO_OBJS +=    fifosubr.o      fifovnops.o

1148 PIPE_OBJS +=    pipe.o

1150 HSFS_OBJS +=    hsfs_node.o     hsfs_subr.o     hsfs_vfsops.o   hsfs_vnops.o \
1151                 hsfs_susp.o     hsfs_rrip.o     hsfs_susp_subr.o

1153 LOFS_OBJS +=    lofs_subr.o     lofs_vfsops.o   lofs_vnops.o

1155 NAMEFS_OBJS +=  namevfs.o       namevno.o

1157 NFS_OBJS +=     nfs_client.o    nfs_common.o    nfs_dump.o \
1158                 nfs_subr.o      nfs_vfsops.o    nfs_vnops.o \
1159                 nfs_xdr.o       nfs_sys.o       nfs_strerror.o \
1160                 nfs3_vfsops.o   nfs3_vnops.o    nfs3_xdr.o \
1161                 nfs_acl_vnops.o nfs_acl_xdr.o   nfs4_vfsops.o \
1162                 nfs4_vnops.o    nfs4_xdr.o      nfs4_idmap.o \
1163                 nfs4_shadow.o   nfs4_subr.o \
1164                 nfs4_attr.o     nfs4_rnode.o    nfs4_client.o \
1165                 nfs4_acache.o   nfs4_common.o   nfs4_client_state.o \
1166                 nfs4_callback.o nfs4_recovery.o nfs4_client_secinfo.o \
1167                 nfs4_client_debug.o     nfs_stats.o \
1168                 nfs4_acl.o      nfs4_stub_vnops.o       nfs_cmd.o

1170 NFSSRV_OBJS +=  nfs_server.o    nfs_srv.o       nfs3_srv.o \
1171                 nfs_acl_srv.o   nfs_auth.o      nfs_auth_xdr.o \
1172                 nfs_export.o    nfs_log.o       nfs_log_xdr.o \
1173                 nfs4_srv.o      nfs4_state.o    nfs4_srv_attr.o \
1174                 nfs4_srv_ns.o   nfs4_db.o       nfs4_srv_deleg.o \
1175                 nfs4_deleg_ops.o nfs4_srv_readdir.o nfs4_dispatch.o

1177 SMBSRV_SHARED_OBJS += \
1178                 smb_inet.o \
1179                 smb_match.o \
1180                 smb_msgbuf.o \
1181                 smb_oem.o \
1182                 smb_string.o \
1183                 smb_utf8.o \
```

```
1184                    smb_door_legacy.o \
1185                    smb_xdr.o \
1186                    smb_token.o \
1187                    smb_token_xdr.o \
1188                    smb_sid.o \
1189                    smb_native.o \
1190                    smb_netbios_util.o

1192 SMBSRV_OBJS +=     $(SMBSRV_SHARED_OBJS)                   \
1193                    smb_acl.o                               \
1194                    smb_alloc.o                             \
1195                    smb_close.o                             \
1196                    smb_common_open.o                       \
1197                    smb_common_transact.o                   \
1198                    smb_create.o                            \
1199                    smb_delete.o                            \
1200                    smb_directory.o                         \
1201                    smb_dispatch.o                          \
1202                    smb_echo.o                              \
1203                    smb_fem.o                               \
1204                    smb_find.o                              \
1205                    smb_flush.o                             \
1206                    smb_fsinfo.o                            \
1207                    smb_fsops.o                             \
1208                    smb_init.o                              \
1209                    smb_kdoor.o                             \
1210                    smb_kshare.o                            \
1211                    smb_kutil.o                             \
1212                    smb_lock.o                              \
1213                    smb_lock_byte_range.o                   \
1214                    smb_locking_andx.o                      \
1215                    smb_logoff_andx.o                       \
1216                    smb_mangle_name.o                       \
1217                    smb_mbuf_marshaling.o                   \
1218                    smb_mbuf_util.o                         \
1219                    smb_negotiate.o                         \
1220                    smb_net.o                               \
1221                    smb_node.o                              \
1222                    smb_nt_cancel.o                         \
1223                    smb_nt_create_andx.o                    \
1224                    smb_nt_transact_create.o                \
1225                    smb_nt_transact_ioctl.o                 \
1226                    smb_nt_transact_notify_change.o         \
1227                    smb_nt_transact_quota.o                 \
1228                    smb_nt_transact_security.o              \
1229                    smb_odir.o                              \
1230                    smb_ofile.o                             \
1231                    smb_open_andx.o                         \
1232                    smb_opipe.o                             \
1233                    smb_oplock.o                            \
1234                    smb_pathname.o                          \
1235                    smb_print.o                             \
1236                    smb_process_exit.o                      \
1237                    smb_query_fileinfo.o                    \
1238                    smb_read.o                              \
1239                    smb_rename.o                            \
1240                    smb_sd.o                                \
1241                    smb_seek.o                              \
1242                    smb_server.o                            \
1243                    smb_session.o                           \
1244                    smb_session_setup_andx.o                \
1245                    smb_set_fileinfo.o                      \
1246                    smb_signing.o                           \
1247                    smb_tree.o                              \
1248                    smb_trans2_create_directory.o           \
1249                    smb_trans2_dfs.o                        \
```



```
1250                    smb_trans2_find.o                       \
1251                    smb_tree_connect.o                      \
1252                    smb_unlock_byte_range.o                 \
1253                    smb_user.o                              \
1254                    smb_vfs.o                               \
1255                    smb_vops.o                              \
1256                    smb_vss.o                               \
1257                    smb_write.o                             \
1258                    smb_write_raw.o

1260 PCFS_OBJS +=      pc_alloc.o      pc_dir.o        pc_node.o       pc_subr.o \
1261                    pc_vfsops.o     pc_vnops.o

1263 PROC_OBJS +=      prcontrol.o     prioctl.o       prsubr.o        prusrio.o \
1264                    prvfsops.o      prvnops.o

1266 MNTFS_OBJS +=     mntvfsops.o     mntvnops.o

1268 SHAREFS_OBJS +=   sharetab.o      sharefs_vfsops.o        sharefs_vnops.o

1270 SPEC_OBJS +=      specsubr.o      specvfsops.o    specvnops.o

1272 SOCK_OBJS +=      socksubr.o      sockvfsops.o    sockparams.o    \
1273                    socksyscalls.o  socktpi.o       sockstr.o \
1274                    sockcommon_vnops.o      sockcommon_subr.o \
1275                    sockcommon_sops.o       sockcommon.o    \
1276                    sock_notsupp.o  socknotify.o \
1277                    nl7c.o          nl7curi.o       nl7chttp.o      nl7clogd.o \
1278                    nl7cnca.o       sodirect.o      sockfilter.o

1280 TMPFS_OBJS +=     tmp_dir.o       tmp_subr.o      tmp_tnode.o     tmp_vfsops.o \
1281                    tmp_vnops.o

1283 UDFS_OBJS +=      udf_alloc.o     udf_bmap.o      udf_dir.o       \
1284                    udf_inode.o     udf_subr.o      udf_vfsops.o    \
1285                    udf_vnops.o

1287 UFS_OBJS +=       ufs_alloc.o     ufs_bmap.o      ufs_dir.o       ufs_xattr.o \
1288                    ufs_inode.o     ufs_subr.o      ufs_tables.o    ufs_vfsops.o \
1289                    ufs_vnops.o     quota.o         quotacalls.o    quota_ufs.o \
1290                    ufs_filio.o     ufs_lockfs.o    ufs_thread.o    ufs_trans.o \
1291                    ufs_acl.o       ufs_panic.o     ufs_directio.o  ufs_log.o \
1292                    ufs_extvnops.o  ufs_snap.o      lufs.o          lufs_thread.o \
1293                    lufs_log.o      lufs_map.o      lufs_top.o      lufs_debug.o
1294 VSCAN_OBJS +=     vscan_drv.o     vscan_svc.o vscan_door.o

1296 NSMB_OBJS +=      smb_conn.o      smb_dev.o       smb_iod.o       smb_pass.o \
1297                    smb_rq.o        smb_sign.o      smb_smb.o       smb_subrs.o \
1298                    smb_time.o      smb_tran.o      smb_trantcp.o   smb_usr.o \
1299                    subr_mchain.o

1301 SMBFS_COMMON_OBJS += smbfs_ntacl.o
1302 SMBFS_OBJS +=     smbfs_vfsops.o  smbfs_vnops.o   smbfs_node.o    \
1303                    smbfs_acl.o     smbfs_client.o  smbfs_smb.o     \
1304                    smbfs_subr.o    smbfs_subr2.o   \
1305                    smbfs_rwlock.o  smbfs_xattr.o   \
1306                    $(SMBFS_COMMON_OBJS)


1309 #
1310 #                       LVM modules
1311 #
1312 MD_OBJS += md.o md_error.o md_ioctl.o md_mddb.o md_names.o \
1313         md_med.o md_rename.o md_subr.o

1315 MD_COMMON_OBJS = md_convert.o md_crc.o md_revchk.o
```

```
1317 MD_DERIVED_OBJS = metamed_xdr.o meta_basic_xdr.o

1319 SOFTPART_OBJS += sp.o sp_ioctl.o

1321 STRIPE_OBJS += stripe.o stripe_ioctl.o

1323 HOTSPARES_OBJS += hotspares.o

1325 RAID_OBJS += raid.o raid_ioctl.o raid_replay.o raid_resync.o raid_hotspare.o

1327 MIRROR_OBJS += mirror.o mirror_ioctl.o mirror_resync.o

1329 NOTIFY_OBJS += md_notify.o

1331 TRANS_OBJS += mdtrans.o trans_ioctl.o trans_log.o

1333 ZFS_COMMON_OBJS +=              \
1334         arc.o                   \
1335         bplist.o                \
1336         bpobj.o                 \
1337         bptree.o                \
1338         dbuf.o                  \
1339         ddt.o                   \
1340         ddt_zap.o               \
1341         dmu.o                   \
1342         dmu_diff.o              \
1343         dmu_send.o              \
1344         dmu_object.o            \
1345         dmu_objset.o            \
1346         dmu_traverse.o          \
1347         dmu_tx.o                \
1348         dnode.o                 \
1349         dnode_sync.o            \
1350         dsl_dir.o               \
1351         dsl_dataset.o           \
1352         dsl_deadlist.o          \
1353         dsl_destroy.o           \
1354         dsl_pool.o              \
1355         dsl_synctask.o          \
1356         dsl_userhold.o          \
1357         dmu_zfetch.o            \
1358         dsl_deleg.o             \
1359         dsl_prop.o              \
1360         dsl_scan.o              \
1361         zfeature.o              \
1362         gzip.o                  \
1363         lz4.o                   \
1364         lzjb.o                  \
1365         metaslab.o              \
1366         range_tree.o            \
1367         refcount.o              \
1368         rrwlock.o               \
1369         sa.o                    \
1370         sha256.o                \
1371         edonr_zfs.o             \
1372         skein_zfs.o             \
1373         spa.o                   \
1374         spa_config.o            \
1375         spa_errlog.o            \
1376         spa_history.o           \
1377         spa_misc.o              \
1378         space_map.o             \
1379         space_reftree.o         \
1380         txg.o                   \
1381         uberblock.o             \
```



```
1382         unique.o                \
1383         vdev.o                  \
1384         vdev_cache.o            \
1385         vdev_file.o             \
1386         vdev_label.o            \
1387         vdev_mirror.o           \
1388         vdev_missing.o          \
1389         vdev_queue.o            \
1390         vdev_raidz.o            \
1391         vdev_root.o             \
1392         zap.o                   \
1393         zap_leaf.o              \
1394         zap_micro.o             \
1395         zfs_byteswap.o          \
1396         zfs_debug.o             \
1397         zfs_fm.o                \
1398         zfs_fuid.o              \
1399         zfs_sa.o                \
1400         zfs_znode.o             \
1401         zil.o                   \
1402         zio.o                   \
1403         zio_checksum.o          \
1404         zio_compress.o          \
1405         zio_inject.o            \
1406         zle.o                   \
1407         zrlock.o

1409 ZFS_SHARED_OBJS +=              \
1410         zfeature_common.o       \
1411         zfs_comutil.o           \
1412         zfs_deleg.o             \
1413         zfs_fletcher.o          \
1414         zfs_namecheck.o         \
1415         zfs_prop.o              \
1416         zpool_prop.o            \
1417         zprop_common.o

1419 ZFS_OBJS +=                     \
1420         $(ZFS_COMMON_OBJS)      \
1421         $(ZFS_SHARED_OBJS)      \
1422         vdev_disk.o             \
1423         zfs_acl.o               \
1424         zfs_ctldir.o            \
1425         zfs_dir.o               \
1426         zfs_ioctl.o             \
1427         zfs_log.o               \
1428         zfs_onexit.o            \
1429         zfs_replay.o            \
1430         zfs_rlock.o             \
1431         zfs_vfsops.o            \
1432         zfs_vnops.o             \
1433         zvol.o

1435 ZUT_OBJS +=                     \
1436         zut.o

1438 #
1439 #                       streams modules
1440 #
1441 BUFMOD_OBJS     +=      bufmod.o

1443 CONNLD_OBJS +=  connld.o

1445 DEDUMP_OBJS +=  dedump.o

1447 DRCOMPAT_OBJS +=        drcompat.o
```

```
1449 LDLINUX_OBJS += ldlinux.o

1451 LDTERM_OBJS +=  ldterm.o uwidth.o

1453 PCKT_OBJS +=    pckt.o

1455 PFMOD_OBJS +=   pfmod.o

1457 PTEM_OBJS +=    ptem.o

1459 REDIRMOD_OBJS += strredirm.o

1461 TIMOD_OBJS +=   timod.o

1463 TIRDWR_OBJS +=  tirdwr.o

1465 TTCOMPAT_OBJS +=ttcompat.o

1467 LOG_OBJS +=     log.o

1469 PIPEMOD_OBJS += pipemod.o

1471 RPCMOD_OBJS +=  rpcmod.o        clnt_cots.o     clnt_clts.o \
1472                 clnt_gen.o      clnt_perr.o     mt_rpcinit.o    rpc_calmsg.o \
1473                 rpc_prot.o      rpc_sztypes.o   rpc_subr.o      rpcb_prot.o \
1474                 svc.o           svc_clts.o      svc_gen.o       svc_cots.o \
1475                 rpcsys.o        xdr_sizeof.o    clnt_rdma.o     svc_rdma.o \
1476                 xdr_rdma.o      rdma_subr.o     xdrrdma_sizeof.o

1478 KLMMOD_OBJS +=  klmmod.o \
1479                 nlm_impl.o \
1480                 nlm_rpc_handle.o \
1481                 nlm_dispatch.o \
1482                 nlm_rpc_svc.o \
1483                 nlm_client.o \
1484                 nlm_service.o \
1485                 nlm_prot_clnt.o \
1486                 nlm_prot_xdr.o \
1487                 nlm_rpc_clnt.o \
1488                 nsm_addr_clnt.o \
1489                 nsm_addr_xdr.o \
1490                 sm_inter_clnt.o \
1491                 sm_inter_xdr.o

1493 KLMOPS_OBJS +=  klmops.o

1495 TLIMOD_OBJS +=  tlimod.o        t_kalloc.o      t_kbind.o       t_kclose.o \
1496                 t_kconnect.o    t_kfree.o       t_kgtstate.o    t_kopen.o \
1497                 t_krcvudat.o    t_ksndudat.o    t_kspoll.o      t_kunbind.o \
1498                 t_kutil.o

1500 RLMOD_OBJS += rlmod.o

1502 TELMOD_OBJS += telmod.o

1504 CRYPTMOD_OBJS += cryptmod.o

1506 KB_OBJS +=      kbd.o           keytables.o

1508 #
1509 #                       ID mapping module
1510 #
1511 IDMAP_OBJS +=   idmap_mod.o     idmap_kapi.o    idmap_xdr.o     idmap_cache.o

1513 #
```



```
1514 #                       scheduling class modules
1515 #
1516 SDC_OBJS +=             sysdc.o

1518 RT_OBJS +=              rt.o
1519 RT_DPTBL_OBJS +=        rt_dptbl.o

1521 TS_OBJS +=              ts.o
1522 TS_DPTBL_OBJS +=        ts_dptbl.o

1524 IA_OBJS +=              ia.o

1526 FSS_OBJS +=             fss.o

1528 FX_OBJS +=              fx.o
1529 FX_DPTBL_OBJS +=        fx_dptbl.o

1531 #
1532 #                       Inter-Process Communication (IPC) modules
1533 #
1534 IPC_OBJS +=     ipc.o

1536 IPCMSG_OBJS +=  msg.o

1538 IPCSEM_OBJS +=  sem.o

1540 IPCSHM_OBJS +=  shm.o

1542 #
1543 #                       bignum module
1544 #
1545 COMMON_BIGNUM_OBJS += bignum_mod.o bignumimpl.o

1547 BIGNUM_OBJS += $(COMMON_BIGNUM_OBJS) $(BIGNUM_PSR_OBJS)

1549 #
1550 #                       kernel cryptographic framework
1551 #
1552 KCF_OBJS +=     kcf.o kcf_callprov.o kcf_cbufcall.o kcf_cipher.o kcf_crypto.o \
1553                 kcf_cryptoadm.o kcf_ctxops.o kcf_digest.o kcf_dual.o \
1554                 kcf_keys.o kcf_mac.o kcf_mech_tabs.o kcf_miscapi.o \
1555                 kcf_object.o kcf_policy.o kcf_prov_lib.o kcf_prov_tabs.o \
1556                 kcf_sched.o kcf_session.o kcf_sign.o kcf_spi.o kcf_verify.o \
1557                 kcf_random.o modes.o ecb.o cbc.o ctr.o ccm.o gcm.o \
1558                 fips_random.o

1560 CRYPTOADM_OBJS += cryptoadm.o

1562 CRYPTO_OBJS +=  crypto.o

1564 DPROV_OBJS +=   dprov.o

1566 DCA_OBJS +=     dca.o dca_3des.o dca_debug.o dca_dsa.o dca_kstat.o dca_rng.o \
1567                 dca_rsa.o

1569 AESPROV_OBJS += aes.o aes_impl.o aes_modes.o

1571 ARCFOURPROV_OBJS += arcfour.o arcfour_crypt.o

1573 BLOWFISHPROV_OBJS += blowfish.o blowfish_impl.o

1575 ECCPROV_OBJS += ecc.o ec.o ec2_163.o ec2_mont.o ecdecode.o ecl_mult.o \
1576                 ecp_384.o ecp_jac.o ec2_193.o ecl.o ecp_192.o ecp_521.o \
1577                 ecp_jm.o ec2_233.o ecl_curve.o ecp_224.o ecp_aff.o \
1578                 ecp_mont.o ec2_aff.o ec_naf.o ecl_gf.o ecp_256.o mp_gf2m.o \
1579                 mpi.o mplogic.o mpmontg.o mpprime.o oid.o \
```

```
1580                 secitem.o ec2_test.o ecp_test.o

1582 RSAPROV_OBJS += rsa.o rsa_impl.o pkcs1.o

1584 SWRANDPROV_OBJS += swrand.o

1586 #
1587 #                       kernel SSL
1588 #
1589 KSSL_OBJS +=    kssl.o kssloctl.o

1591 KSSL_SOCKFIL_MOD_OBJS += ksslfilter.o ksslapi.o ksslrec.o

1593 #
1594 #                       misc. modules
1595 #

1597 C2AUDIT_OBJS += adr.o audit.o audit_event.o audit_io.o \
1598                 audit_path.o audit_start.o audit_syscalls.o audit_token.o \
1599                 audit_mem.o

1601 PCIC_OBJS +=    pcic.o

1603 RPCSEC_OBJS +=  secmod.o        sec_clnt.o      sec_svc.o       sec_gen.o \
1604                 auth_des.o      auth_kern.o     auth_none.o     auth_loopb.o\
1605                 authdesprt.o    authdesubr.o    authu_prot.o \
1606                 key_call.o      key_prot.o      svc_authu.o     svcauthdes.o

1608 RPCSEC_GSS_OBJS +=      rpcsec_gssmod.o rpcsec_gss.o rpcsec_gss_misc.o \
1609                 rpcsec_gss_utils.o svc_rpcsec_gss.o

1611 CONSCONFIG_OBJS += consconfig.o

1613 CONSCONFIG_DACF_OBJS  += consconfig_dacf.o consplat.o

1615 TEM_OBJS += tem.o tem_safe.o 6x10.o 7x14.o 12x22.o

1617 KBTRANS_OBJS +=                        \
1618                 kbtrans.o               \
1619                 kbtrans_keytables.o     \
1620                 kbtrans_polled.o        \
1621                 kbtrans_streams.o       \
1622                 usb_keytables.o

1624 KGSSD_OBJS +=   gssd_clnt_stubs.o gssd_handle.o gssd_prot.o \
1625                 gss_display_name.o gss_release_name.o gss_import_name.o \
1626                 gss_release_buffer.o gss_release_oid_set.o gen_oids.o gssdmod.o

1628 KGSSD_DERIVED_OBJS = gssd_xdr.o

1630 KGSS_DUMMY_OBJS += dmech.o

1632 KSOCKET_OBJS += ksocket.o ksocket_mod.o

1634 CRYPTO= cksumtypes.o decrypt.o encrypt.o encrypt_length.o etypes.o \
1635         nfold.o verify_checksum.o prng.o block_size.o make_checksum.o\
1636         checksum_length.o hmac.o default_state.o mandatory_sumtype.o

1638 # crypto/des
1639 CRYPTO_DES= f_cbc.o f_cksum.o f_parity.o weak_key.o d3_cbc.o ef_crypto.o

1641 CRYPTO_DK= checksum.o derive.o dk_decrypt.o dk_encrypt.o

1643 CRYPTO_ARCFOUR= k5_arcfour.o

1645 # crypto/enc_provider
```



```
1646 CRYPTO_ENC= des.o des3.o arcfour_provider.o aes_provider.o

1648 # crypto/hash_provider
1649 CRYPTO_HASH= hash_kef_generic.o hash_kmd5.o hash_crc32.o hash_ksha1.o

1651 # crypto/keyhash_provider
1652 CRYPTO_KEYHASH= descbc.o k5_kmd5des.o k_hmac_md5.o

1654 # crypto/crc32
1655 CRYPTO_CRC32= crc32.o

1657 # crypto/old
1658 CRYPTO_OLD= old_decrypt.o old_encrypt.o

1660 # crypto/raw
1661 CRYPTO_RAW= raw_decrypt.o raw_encrypt.o

1663 K5_KRB= kfree.o copy_key.o \
1664         parse.o init_ctx.o \
1665         ser_adata.o ser_addr.o \
1666         ser_auth.o ser_cksum.o \
1667         ser_key.o ser_princ.o \
1668         serialize.o unparse.o \
1669         ser_actx.o

1671 K5_OS=  timeofday.o toffset.o \
1672         init_os_ctx.o c_ustime.o

1674 SEAL=   seal.o unseal.o

1676 MECH=   delete_sec_context.o \
1677         import_sec_context.o \
1678         gssapi_krb5.o \
1679         k5seal.o k5unseal.o k5sealv3.o \
1680         ser_sctx.o \
1681         sign.o \
1682         util_crypt.o  \
1683         util_validate.o  util_ordering.o  \
1684         util_seqnum.o util_set.o util_seed.o \
1685         wrap_size_limit.o verify.o



1689 MECH_GEN= util_token.o


1692 KGSS_KRB5_OBJS += krb5mech.o \
1693         $(MECH) $(SEAL) $(MECH_GEN) \
1694         $(CRYPTO) $(CRYPTO_DES) $(CRYPTO_DK) $(CRYPTO_ARCFOUR) \
1695         $(CRYPTO_ENC) $(CRYPTO_HASH) \
1696         $(CRYPTO_KEYHASH) $(CRYPTO_CRC32) \
1697         $(CRYPTO_OLD) \
1698         $(CRYPTO_RAW) $(K5_KRB) $(K5_OS)

1700 DES_OBJS +=     des_crypt.o des_impl.o des_ks.o des_soft.o

1702 DLBOOT_OBJS +=  bootparam_xdr.o nfs_dlinet.o scan.o

1704 KRTLD_OBJS +=   kobj_bootflags.o getoptstr.o \
1705                 kobj.o kobj_kdi.o kobj_lm.o kobj_subr.o

1707 MOD_OBJS +=     modctl.o modsubr.o modsysfile.o modconf.o modhash.o

1709 STRPLUMB_OBJS += strplumb.o

1711 CPR_OBJS +=     cpr_driver.o cpr_dump.o \
```

```
1712                 cpr_main.o cpr_misc.o cpr_mod.o cpr_stat.o \
1713                 cpr_uthread.o

1715 PROF_OBJS +=    prf.o

1717 SE_OBJS += se_driver.o

1719 SYSACCT_OBJS += acct.o

1721 ACCTCTL_OBJS += acctctl.o

1723 EXACCTSYS_OBJS += exacctsys.o

1725 KAIO_OBJS += aio.o

1727 PCMCIA_OBJS += pcmcia.o cs.o cis.o cis_callout.o cis_handlers.o cis_params.o

1729 BUSRA_OBJS += busra.o

1731 PCS_OBJS += pcs.o

1733 PCAN_OBJS += pcan.o

1735 PCATA_OBJS += pcide.o pcdisk.o pclabel.o pcata.o

1737 PCSER_OBJS += pcser.o pcser_cis.o

1739 PCWL_OBJS += pcwl.o

1741 PSET_OBJS +=    pset.o

1743 OHCI_OBJS += ohci.o ohci_hub.o ohci_polled.o

1745 UHCI_OBJS += uhci.o uhciutil.o uhcitgt.o uhcihub.o uhcipolled.o

1747 EHCI_OBJS += ehci.o ehci_hub.o ehci_xfer.o ehci_intr.o ehci_util.o ehci_polled.o

1749 HUBD_OBJS += hubd.o

1751 USB_MID_OBJS += usb_mid.o

1753 USB_IA_OBJS += usb_ia.o

1755 UWBA_OBJS += uwba.o uwbai.o

1757 SCSA2USB_OBJS += scsa2usb.o usb_ms_bulkonly.o usb_ms_cbi.o

1759 HWAHC_OBJS += hwahc.o hwahc_util.o

1761 WUSB_DF_OBJS += wusb_df.o
1762 WUSB_FWMOD_OBJS += wusb_fwmod.o

1764 IPF_OBJS += ip_fil_solaris.o fil.o solaris.o ip_state.o ip_frag.o ip_nat.o \
1765             ip_proxy.o ip_auth.o ip_pool.o ip_htable.o ip_lookup.o \
1766             ip_log.o misc.o ip_compat.o ip_nat6.o drand48.o

1768 IBD_OBJS +=     ibd.o ibd_cm.o

1770 EIBNX_OBJS +=   enx_main.o enx_hdlrs.o enx_ibt.o enx_log.o enx_fip.o \
1771                 enx_misc.o enx_q.o enx_ctl.o

1773 EOIB_OBJS +=    eib_adm.o eib_chan.o eib_cmn.o eib_ctl.o eib_data.o \
1774                 eib_fip.o eib_ibt.o eib_log.o eib_mac.o eib_main.o \
1775                 eib_rsrc.o eib_svc.o eib_vnic.o

1777 DLPISTUB_OBJS += dlpistub.o
```



```
1779 SDP_OBJS +=     sdpddi.o

1781 TRILL_OBJS +=   trill.o

1783 CTF_OBJS += ctf_create.o ctf_decl.o ctf_error.o ctf_hash.o ctf_labels.o \
1784         ctf_lookup.o ctf_open.o ctf_types.o ctf_util.o ctf_subr.o ctf_mod.o

1786 SMBIOS_OBJS += smb_error.o smb_info.o smb_open.o smb_subr.o smb_dev.o

1788 RPCIB_OBJS += rpcib.o

1790 KMDB_OBJS += kdrv.o

1792 AFE_OBJS += afe.o

1794 BGE_OBJS += bge_main2.o bge_chip2.o bge_kstats.o bge_log.o bge_ndd.o \
1795                 bge_atomic.o bge_mii.o bge_send.o bge_recv2.o bge_mii_5906.o

1797 DMFE_OBJS += dmfe_log.o dmfe_main.o dmfe_mii.o

1799 EFE_OBJS += efe.o

1801 ELXL_OBJS += elxl.o

1803 HME_OBJS += hme.o

1805 IXGB_OBJS += ixgb.o ixgb_atomic.o ixgb_chip.o ixgb_gld.o ixgb_kstats.o \
1806                 ixgb_log.o ixgb_ndd.o ixgb_rx.o ixgb_tx.o ixgb_xmii.o

1808 NGE_OBJS += nge_main.o nge_atomic.o nge_chip.o nge_ndd.o nge_kstats.o \
1809                 nge_log.o nge_rx.o nge_tx.o nge_xmii.o

1811 PCN_OBJS += pcn.o

1813 RGE_OBJS += rge_main.o rge_chip.o rge_ndd.o rge_kstats.o rge_log.o rge_rxtx.o

1815 URTW_OBJS += urtw.o

1817 ARN_OBJS += arn_hw.o arn_eeprom.o arn_mac.o arn_calib.o arn_ani.o arn_phy.o arn_
1818                 arn_main.o arn_recv.o arn_xmit.o arn_rc.o

1820 ATH_OBJS += ath_aux.o ath_main.o ath_osdep.o ath_rate.o

1822 ATU_OBJS += atu.o

1824 IPW_OBJS += ipw2100_hw.o ipw2100.o

1826 IWI_OBJS += ipw2200_hw.o ipw2200.o

1828 IWH_OBJS += iwh.o

1830 IWK_OBJS += iwk2.o

1832 IWP_OBJS += iwp.o

1834 MWL_OBJS += mwl.o

1836 MWLFW_OBJS += mwlfw_mode.o

1838 WPI_OBJS += wpi.o

1840 RAL_OBJS += rt2560.o ral_rate.o

1842 RUM_OBJS += rum.o
```

```
1844 RWD_OBJS += rt2661.o

1846 RWN_OBJS += rt2860.o

1848 UATH_OBJS += uath.o

1850 UATHFW_OBJS += uathfw_mod.o

1852 URAL_OBJS += ural.o

1854 RTW_OBJS += rtw.o smc93cx6.o rtwphy.o rtwphyio.o

1856 ZYD_OBJS += zyd.o zyd_usb.o zyd_hw.o zyd_fw.o

1858 MXFE_OBJS += mxfe.o

1860 MPTSAS_OBJS += mptsas.o mptsas_impl.o mptsas_init.o mptsas_raid.o mptsas_smhba.o

1862 SFE_OBJS += sfe.o sfe_util.o

1864 BFE_OBJS += bfe.o

1866 BRIDGE_OBJS += bridge.o

1868 IDM_SHARED_OBJS += base64.o

1870 IDM_OBJS +=     $(IDM_SHARED_OBJS) \
1871                 idm.o idm_impl.o idm_text.o idm_conn_sm.o idm_so.o

1873 VR_OBJS += vr.o

1875 ATGE_OBJS += atge_main.o atge_l1e.o atge_mii.o atge_l1.o atge_l1c.o

1877 YGE_OBJS = yge.o

1879 #
1880 #       Build up defines and paths.
1881 #
1882 LINT_DEFS       += -Dunix

1884 #
1885 #       This duality can be removed when the native and target compilers
1886 #       are the same (or at least recognize the same command line syntax!)
1887 #       It is a bug in the current compilation system that the assember
1888 #       can't process the -Y I, flag.
1889 #
1890 NATIVE_INC_PATH += $(INC_PATH) $(CCYFLAG)$(UTSBASE)/common
1891 AS_INC_PATH     += $(INC_PATH) -I$(UTSBASE)/common
1892 INCLUDE_PATH    += $(INC_PATH) $(CCYFLAG)$(UTSBASE)/common

1894 PCIEB_OBJS += pcieb.o

1896 #       Chelsio N110 10G NIC driver module
1897 #
1898 CH_OBJS = ch.o glue.o pe.o sge.o

1900 CH_COM_OBJS =   ch_mac.o ch_subr.o cspi.o espi.o ixf1010.o mc3.o mc4.o mc5.o \
1901                 mv88e1xxx.o mv88x201x.o my3126.o pm3393.o tp.o ulp.o \
1902                 vsc7321.o vsc7326.o xpak.o

1904 #
1905 #       Chelsio Terminator 4 10G NIC nexus driver module
1906 #
1907 CXGBE_FW_OBJS  =        t4_fw.o t4_cfg.o
1908 CXGBE_COM_OBJS =        t4_hw.o common.o
1909 CXGBE_NEX_OBJS =        t4_nexus.o t4_sge.o t4_mac.o t4_ioctl.o shared.o \
```



```
1910                         t4_l2t.o adapter.o osdep.o

1912 #
1913 #       Chelsio Terminator 4 10G NIC driver module
1914 #
1915 CXGBE_OBJS =    cxgbe.o

1917 #
1918 #       PCI strings file
1919 #
1920 PCI_STRING_OBJS = pci_strings.o

1922 NET_DACF_OBJS += net_dacf.o

1924 #
1925 #       Xframe 10G NIC driver module
1926 #
1927 XGE_OBJS = xge.o xgell.o

1929 XGE_HAL_OBJS =  xgehal-channel.o xgehal-fifo.o xgehal-ring.o  xgehal-config.o \
1930                 xgehal-driver.o  xgehal-mm.o xgehal-stats.o  xgehal-device.o \
1931                 xge-queue.o  xgehal-mgmt.o xgehal-mgmtaux.o

1933 #
1934 #       e1000/igb common objs
1935 #
1936 #       Historically e1000g and igb had separate copies of all of the common
1937 #       code. At this time while they are now sharing the same copy of it, they
1938 #       are building it into their own modules which is due to the differences
1939 #       in the osdep and debug portions of their code.
1940 #
1941 E1000API_OBJS += e1000_80003es2lan.o e1000_82540.o e1000_82541.o e1000_82542.o \
1942                 e1000_82543.o e1000_82571.o e1000_api.o e1000_ich8lan.o \
1943                 e1000_mac.o e1000_manage.o e1000_nvm.o e1000_phy.o \
1944                 e1000_82575.o e1000_i210.o e1000_mbx.o e1000_vf.o

1946 #
1947 #       e1000g module
1948 #
1949 E1000G_OBJS +=  e1000g_debug.o e1000g_main.o e1000g_alloc.o \
1950                 e1000g_tx.o e1000g_rx.o e1000g_stat.o \
1951                 e1000g_osdep.o e1000g_workarounds.o
1952

1954 #
1955 #       Intel 82575 1G NIC driver module
1956 #
1957 IGB_OBJS =      igb_buf.o igb_debug.o igb_gld.o igb_log.o igb_main.o \
1958                 igb_rx.o igb_stat.o igb_tx.o igb_osdep.o

1960 #
1961 #       Intel Pro/100 NIC driver module
1962 #
1963 IPRB_OBJS =     iprb.o

1965 #
1966 #       Intel 10GbE PCIE NIC driver module
1967 #
1968 IXGBE_OBJS =    ixgbe_82598.o ixgbe_82599.o ixgbe_api.o         \
1969                 ixgbe_common.o ixgbe_phy.o                      \
1970                 ixgbe_buf.o ixgbe_debug.o ixgbe_gld.o           \
1971                 ixgbe_log.o ixgbe_main.o                        \
1972                 ixgbe_osdep.o ixgbe_rx.o ixgbe_stat.o           \
1973                 ixgbe_tx.o  ixgbe_x540.o ixgbe_mbx.o

1975 #
```

```
1976 #       NIU 10G/1G driver module
1977 #
1978 NXGE_OBJS =     nxge_mac.o nxge_ipp.o nxge_rxdma.o              \
1979                 nxge_txdma.o nxge_txc.o nxge_main.o             \
1980                 nxge_hw.o nxge_fzc.o nxge_virtual.o             \
1981                 nxge_send.o nxge_classify.o nxge_fflp.o         \
1982                 nxge_fflp_hash.o nxge_ndd.o nxge_kstats.o       \
1983                 nxge_zcp.o nxge_fm.o nxge_espc.o nxge_hv.o      \
1984                 nxge_hio.o nxge_hio_guest.o nxge_intr.o

1986 NXGE_NPI_OBJS = \
1987                 npi.o npi_mac.o npi_ipp.o                       \
1988                 npi_txdma.o npi_rxdma.o npi_txc.o               \
1989                 npi_zcp.o npi_espc.o npi_fflp.o                 \
1990                 npi_vir.o

1992 NXGE_HCALL_OBJS =       \
1993                 nxge_hcall.o

1995 #
1996 # Virtio modules
1997 #

1999 # Virtio core
2000 VIRTIO_OBJS = virtio.o

2002 # Virtio block driver
2003 VIOBLK_OBJS = vioblk.o

2005 #
2006 #       kiconv modules
2007 #
2008 KICONV_EMEA_OBJS += kiconv_emea.o

2010 KICONV_JA_OBJS += kiconv_ja.o

2012 KICONV_KO_OBJS += kiconv_cck_common.o kiconv_ko.o

2014 KICONV_SC_OBJS += kiconv_cck_common.o kiconv_sc.o

2016 KICONV_TC_OBJS += kiconv_cck_common.o kiconv_tc.o

2018 #
2019 #       AAC module
2020 #
2021 AAC_OBJS = aac.o aac_ioctl.o

2023 #
2024 #       sdcard modules
2025 #
2026 SDA_OBJS =      sda_cmd.o sda_host.o sda_init.o sda_mem.o sda_mod.o sda_slot.o
2027 SDHOST_OBJS =   sdhost.o

2029 #
2030 #       hxge 10G driver module
2031 #
2032 HXGE_OBJS =     hxge_main.o hxge_vmac.o hxge_send.o              \
2033                 hxge_txdma.o hxge_rxdma.o hxge_virtual.o        \
2034                 hxge_fm.o hxge_fzc.o hxge_hw.o hxge_kstats.o    \
2035                 hxge_ndd.o hxge_pfc.o                           \
2036                 hpi.o hpi_vmac.o hpi_rxdma.o hpi_txdma.o        \
2037                 hpi_vir.o hpi_pfc.o

2039 #
2040 #       MEGARAID_SAS module
2041 #
```



```
2042 MEGA_SAS_OBJS = megaraid_sas.o

2044 #
2045 #       MR_SAS module
2046 #
2047 MR_SAS_OBJS = ld_pd_map.o mr_sas.o mr_sas_tbolt.o mr_sas_list.o

2049 #
2050 #       CPQARY3 module
2051 #
2052 CPQARY3_OBJS =  cpqary3.o cpqary3_noe.o cpqary3_talk2ctlr.o     \
2053                 cpqary3_isr.o cpqary3_transport.o cpqary3_mem.o \
2054                 cpqary3_scsi.o cpqary3_util.o cpqary3_ioctl.o   \
2055                 cpqary3_bd.o

2057 #
2058 #       ISCSI_INITIATOR module
2059 #
2060 ISCSI_INITIATOR_OBJS =  chap.o iscsi_io.o iscsi_thread.o        \
2061                         iscsi_ioctl.o iscsid.o iscsi.o          \
2062                         iscsi_login.o isns_client.o iscsiAuthClient.o   \
2063                         iscsi_lun.o iscsiAuthClientGlue.o       \
2064                         iscsi_net.o nvfile.o iscsi_cmd.o        \
2065                         iscsi_queue.o persistent.o iscsi_conn.o \
2066                         iscsi_sess.o radius_auth.o iscsi_crc.o  \
2067                         iscsi_stats.o radius_packet.o iscsi_doorclt.o   \
2068                         iscsi_targetparam.o utils.o kifconf.o

2070 #
2071 #       ntxn 10Gb/1Gb NIC driver module
2072 #
2073 NTXN_OBJS =     unm_nic_init.o unm_gem.o unm_nic_hw.o unm_ndd.o \
2074                         unm_nic_main.o unm_nic_isr.o unm_nic_ctx.o niu.o

2076 #
2077 #       Myricom 10Gb NIC driver module
2078 #
2079 MYRI10GE_OBJS = myri10ge.o myri10ge_lro.o

2081 #       nulldriver module
2082 #
2083 NULLDRIVER_OBJS =       nulldriver.o

2085 TPM_OBJS =      tpm.o tpm_hcall.o
```

```
**********************************************************
   73780 Mon Oct  7 19:17:18 2013
new/usr/src/uts/common/Makefile.rules
4185 New hash algorithm support
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright (c) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2012 Nexenta Systems, Inc. All rights reserved.
# Copyright 2013 Saso Kiselkov. All rights reserved.
#

#
# uts/common/Makefile.rules
#
#       This Makefile defines all the file build rules for the directory
# uts/common and its children. These are the source files which may
# be considered common to all SunOS systems.
#
#       The following two-level ordering must be maintained in this file.
#         Lines are sorted first in order of decreasing specificity based on
#         the first directory component.  That is, sun4u rules come before
#         sparc rules come before common rules.
#
#         Lines whose initial directory components are equal are sorted
#         alphabetically by the remaining components.

#
#       Section 1a: C objects build rules
#
$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/aes/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/arcfour/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/blowfish/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/ecc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```



```
$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/modes/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/padding/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/rng/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/rsa/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/bignum/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/bignum/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/mpi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/acl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/avl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/ucode/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/brand/sn1/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/brand/solaris10/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/c2/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/conf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/contract/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/cpr/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ctf/%.c
        $(COMPILE.c) -o $@ $<
```

```
128         $(CTFCONVERT_O)

130 $(OBJS_DIR)/%.o:                $(COMMONBASE)/ctf/%.c
131         $(COMPILE.c) -o $@ $<
132         $(CTFCONVERT_O)

134 $(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/des/%.c
135         $(COMPILE.c) -o $@ $<
136         $(CTFCONVERT_O)

138 $(OBJS_DIR)/%.o:                $(COMMONBASE)/smbios/%.c
139         $(COMPILE.c) -o $@ $<
140         $(CTFCONVERT_O)

142 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/des/%.c
143         $(COMPILE.c) -o $@ $<
144         $(CTFCONVERT_O)

146 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/crypto/api/%.c
147         $(COMPILE.c) -o $@ $<
148         $(CTFCONVERT_O)

150 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/crypto/core/%.c
151         $(COMPILE.c) -o $@ $<
152         $(CTFCONVERT_O)

154 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/crypto/io/%.c
155         $(COMPILE.c) -o $@ $<
156         $(CTFCONVERT_O)

158 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/crypto/spi/%.c
159         $(COMPILE.c) -o $@ $<
160         $(CTFCONVERT_O)

162 $(OBJS_DIR)/%.o:                $(COMMONBASE)/pci/%.c
163         $(COMPILE.c) -o $@ $<
164         $(CTFCONVERT_O)

166 $(OBJS_DIR)/%.o:                $(COMMONBASE)/devid/%.c
167         $(COMPILE.c) -o $@ $<
168         $(CTFCONVERT_O)

170 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/disp/%.c
171         $(COMPILE.c) -o $@ $<
172         $(CTFCONVERT_O)

174 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/dtrace/%.c
175         $(COMPILE.c) -o $@ $<
176         $(CTFCONVERT_O)

178 $(OBJS_DIR)/%.o:                $(COMMONBASE)/exacct/%.c
179         $(COMPILE.c) -o $@ $<
180         $(CTFCONVERT_O)

182 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/exec/aout/%.c
183         $(COMPILE.c) -o $@ $<
184         $(CTFCONVERT_O)

186 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/exec/elf/%.c
187         $(COMPILE.c) -o $@ $<
188         $(CTFCONVERT_O)

190 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/exec/intp/%.c
191         $(COMPILE.c) -o $@ $<
192         $(CTFCONVERT_O)
```



```
194 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/exec/shbin/%.c
195         $(COMPILE.c) -o $@ $<
196         $(CTFCONVERT_O)

198 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/exec/java/%.c
199         $(COMPILE.c) -o $@ $<
200         $(CTFCONVERT_O)

202 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/%.c
203         $(COMPILE.c) -o $@ $<
204         $(CTFCONVERT_O)

206 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/autofs/%.c
207         $(COMPILE.c) -o $@ $<
208         $(CTFCONVERT_O)

210 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/cachefs/%.c
211         $(COMPILE.c) -o $@ $<
212         $(CTFCONVERT_O)

214 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/dcfs/%.c
215         $(COMPILE.c) -o $@ $<
216         $(CTFCONVERT_O)

218 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/devfs/%.c
219         $(COMPILE.c) -o $@ $<
220         $(CTFCONVERT_O)

222 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/ctfs/%.c
223         $(COMPILE.c) -o $@ $<
224         $(CTFCONVERT_O)

226 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/doorfs/%.c
227         $(COMPILE.c) -o $@ $<
228         $(CTFCONVERT_O)

230 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/dev/%.c
231         $(COMPILE.c) -o $@ $<
232         $(CTFCONVERT_O)

234 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/fd/%.c
235         $(COMPILE.c) -o $@ $<
236         $(CTFCONVERT_O)

238 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/fifofs/%.c
239         $(COMPILE.c) -o $@ $<
240         $(CTFCONVERT_O)

242 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/hsfs/%.c
243         $(COMPILE.c) -o $@ $<
244         $(CTFCONVERT_O)

246 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/lofs/%.c
247         $(COMPILE.c) -o $@ $<
248         $(CTFCONVERT_O)

250 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/mntfs/%.c
251         $(COMPILE.c) -o $@ $<
252         $(CTFCONVERT_O)

254 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/namefs/%.c
255         $(COMPILE.c) -o $@ $<
256         $(CTFCONVERT_O)

258 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/nfs/%.c
259         $(COMPILE.c) -o $@ $<
```

```
260         $(CTFCONVERT_O)

262 $(OBJS_DIR)/%.o:                $(COMMONBASE)/smbsrv/%.c
263         $(COMPILE.c) -o $@ $<
264         $(CTFCONVERT_O)

266 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/smbsrv/%.c
267         $(COMPILE.c) -o $@ $<
268         $(CTFCONVERT_O)

270 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/objfs/%.c
271         $(COMPILE.c) -o $@ $<
272         $(CTFCONVERT_O)

274 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/pcfs/%.c
275         $(COMPILE.c) -o $@ $<
276         $(CTFCONVERT_O)

278 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/portfs/%.c
279         $(COMPILE.c) -o $@ $<
280         $(CTFCONVERT_O)

282 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/proc/%.c
283         $(COMPILE.c) -o $@ $<
284         $(CTFCONVERT_O)

286 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/sharefs/%.c
287         $(COMPILE.c) -o $@ $<
288         $(CTFCONVERT_O)

290 $(OBJS_DIR)/%.o:                $(COMMONBASE)/smbclnt/%.c
291         $(COMPILE.c) -o $@ $<
292         $(CTFCONVERT_O)

294 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/smbclnt/netsmb/%.c
295         $(COMPILE.c) -o $@ $<
296         $(CTFCONVERT_O)

298 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/smbclnt/smbfs/%.c
299         $(COMPILE.c) -o $@ $<
300         $(CTFCONVERT_O)

302 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/sockfs/%.c
303         $(COMPILE.c) -o $@ $<
304         $(CTFCONVERT_O)

306 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/specfs/%.c
307         $(COMPILE.c) -o $@ $<
308         $(CTFCONVERT_O)

310 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/swapfs/%.c
311         $(COMPILE.c) -o $@ $<
312         $(CTFCONVERT_O)

314 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/tmpfs/%.c
315         $(COMPILE.c) -o $@ $<
316         $(CTFCONVERT_O)

318 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/udfs/%.c
319         $(COMPILE.c) -o $@ $<
320         $(CTFCONVERT_O)

322 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/ufs/%.c
323         $(COMPILE.c) -o $@ $<
324         $(CTFCONVERT_O)
```



```
326 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/vscan/%.c
327         $(COMPILE.c) -o $@ $<
328         $(CTFCONVERT_O)

330 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/zfs/%.c
331         $(COMPILE.c) -o $@ $<
332         $(CTFCONVERT_O)

334 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/zut/%.c
335         $(COMPILE.c) -o $@ $<
336         $(CTFCONVERT_O)

338 $(OBJS_DIR)/%.o:                $(COMMONBASE)/xattr/%.c
339         $(COMPILE.c) -o $@ $<
340         $(CTFCONVERT_O)

342 $(OBJS_DIR)/%.o:                $(COMMONBASE)/zfs/%.c
343         $(COMPILE.c) -o $@ $<
344         $(CTFCONVERT_O)

346 $(OBJS_DIR)/%.o:        $(UTSBASE)/common/io/scsi/adapters/pmcs/%.c
347         $(COMPILE.c) -o $@ $<
348         $(CTFCONVERT_O)

350 $(OBJS_DIR)/%.o:        $(UTSBASE)/common/io/scsi/adapters/pmcs/%.bin
351         $(COMPILE.b) -o $@ $<
352         $(CTFCONVERT_O)

354 $(OBJS_DIR)/%.o:                $(COMMONBASE)/fsreparse/%.c
355         $(COMPILE.c) -o $@ $<
356         $(CTFCONVERT_O)

358 KMECHKRB5_BASE=$(UTSBASE)/common/gssapi/mechs/krb5

360 KGSSDFLAGS=-I $(UTSBASE)/common/gssapi/include

362 # Note, KRB5_DEFS can be assigned various preprocessor flags,
363 # typically -D defines on the make invocation.  The standard compiler
364 # flags will not be overwritten.
365 KGSSDFLAGS += $(KRB5_DEFS)

367 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/gssapi/%.c
368         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
369         $(CTFCONVERT_O)

371 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/gssapi/mechs/dummy/%.c
372         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
373         $(CTFCONVERT_O)

375 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/%.c
376         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
377         $(CTFCONVERT_O)

379 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/%.c
380         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
381         $(CTFCONVERT_O)

383 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/des/%.c
384         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
385         $(CTFCONVERT_O)

387 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/arcfour/%.c
388         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
389         $(CTFCONVERT_O)

391 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/dk/%.c
```

```
392         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
393         $(CTFCONVERT_O)

395 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/enc_provider/%.c
396         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
397         $(CTFCONVERT_O)

399 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/hash_provider/%.c
400         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
401         $(CTFCONVERT_O)

403 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/keyhash_provider/%.c
404         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
405         $(CTFCONVERT_O)

407 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/raw/%.c
408         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
409         $(CTFCONVERT_O)

411 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/old/%.c
412         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
413         $(CTFCONVERT_O)

415 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/krb5/krb/%.c
416         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
417         $(CTFCONVERT_O)

419 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/krb5/os/%.c
420         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
421         $(CTFCONVERT_O)

423 $(OBJS_DIR)/ser_sctx.o := CPPFLAGS += -DPROVIDE_KERNEL_IMPORT=1

425 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/mech/%.c
426         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
427         $(CTFCONVERT_O)

429 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/profile/%.c
430         $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
431         $(CTFCONVERT_O)

433 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ncall/%.c
434         $(COMPILE.c) -o $@ $<
435         $(CTFCONVERT_O)

437 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ns/dsw/%.c
438         $(COMPILE.c) -o $@ $<
439         $(CTFCONVERT_O)

441 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ns/nsctl/%.c
442         $(COMPILE.c) -o $@ $<
443         $(CTFCONVERT_O)

445 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ns/rdc/%.c
446         $(COMPILE.c) -o $@ $<
447         $(CTFCONVERT_O)

449 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ns/sdbc/%.c
450         $(COMPILE.c) -o $@ $<
451         $(CTFCONVERT_O)

453 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ns/solaris/%.c
454         $(COMPILE.c) -o $@ $<
455         $(CTFCONVERT_O)

457 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ns/sv/%.c
```



```
458         $(COMPILE.c) -o $@ $<
459         $(CTFCONVERT_O)

461 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ns/unistat/%.c
462         $(COMPILE.c) -o $@ $<
463         $(CTFCONVERT_O)

465 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/idmap/%.c
466         $(COMPILE.c) -o $@ $<
467         $(CTFCONVERT_O)

469 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/%.c
470         $(COMPILE.c) -o $@ $<
471         $(CTFCONVERT_O)

473 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/arp/%.c
474         $(COMPILE.c) -o $@ $<
475         $(CTFCONVERT_O)

477 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/ip/%.c
478         $(COMPILE.c) -o $@ $<
479         $(CTFCONVERT_O)

481 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/ipnet/%.c
482         $(COMPILE.c) -o $@ $<
483         $(CTFCONVERT_O)

485 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/iptun/%.c
486         $(COMPILE.c) -o $@ $<
487         $(CTFCONVERT_O)

489 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/kssl/%.c
490         $(COMPILE.c) -o $@ $<
491         $(CTFCONVERT_O)

493 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/sctp/%.c
494         $(COMPILE.c) -o $@ $<
495         $(CTFCONVERT_O)

497 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/tcp/%.c
498         $(COMPILE.c) -o $@ $<
499         $(CTFCONVERT_O)

501 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/ilb/%.c
502         $(COMPILE.c) -o $@ $<
503         $(CTFCONVERT_O)

505 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/ipf/%.c
506         $(COMPILE.c) -o $@ $<
507         $(CTFCONVERT_O)

509 $(OBJS_DIR)/%.o:                $(COMMONBASE)/net/patricia/%.c
510         $(COMPILE.c) -o $@ $<
511         $(CTFCONVERT_O)

513 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/udp/%.c
514         $(COMPILE.c) -o $@ $<
515         $(CTFCONVERT_O)

517 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/nca/%.c
518         $(COMPILE.c) -o $@ $<
519         $(CTFCONVERT_O)

521 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/sockmods/%.c
522         $(COMPILE.c) -o $@ $<
523         $(CTFCONVERT_O)
```

```
525 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/dlpistub/%.c
526         $(COMPILE.c) -o $@ $<
527         $(CTFCONVERT_O)

529 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/%.c
530         $(COMPILE.c) -o $@ $<
531         $(CTFCONVERT_O)

533 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/1394/%.c
534         $(COMPILE.c) -o $@ $<
535         $(CTFCONVERT_O)

537 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/1394/adapters/%.c
538         $(COMPILE.c) -o $@ $<
539         $(CTFCONVERT_O)

541 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/1394/targets/av1394/%.c
542         $(COMPILE.c) -o $@ $<
543         $(CTFCONVERT_O)

545 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/1394/targets/dcam1394/%.c
546         $(COMPILE.c) -o $@ $<
547         $(CTFCONVERT_O)

549 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/1394/targets/scsa1394/%.c
550         $(COMPILE.c) -o $@ $<
551         $(CTFCONVERT_O)

553 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sbp2/%.c
554         $(COMPILE.c) -o $@ $<
555         $(CTFCONVERT_O)

557 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/aac/%.c
558         $(COMPILE.c) -o $@ $<
559         $(CTFCONVERT_O)

561 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/afe/%.c
562         $(COMPILE.c) -o $@ $<
563         $(CTFCONVERT_O)

565 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/atge/%.c
566         $(COMPILE.c) -o $@ $<
567         $(CTFCONVERT_O)

569 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/arn/%.c
570         $(COMPILE.c) -o $@ $<
571         $(CTFCONVERT_O)

573 $(OBJS_DIR)/%.o:               $(UTSBASE)/common/io/ath/%.c
574         $(COMPILE.c) -o $@ $<
575         $(CTFCONVERT_O)

577 $(OBJS_DIR)/%.o:               $(UTSBASE)/common/io/atu/%.c
578         $(COMPILE.c) -o $@ $<
579         $(CTFCONVERT_O)

581 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/impl/%.c
582         $(COMPILE.c) -o $@ $<
583         $(CTFCONVERT_O)

585 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/ac97/%.c
586         $(COMPILE.c) -o $@ $<
587         $(CTFCONVERT_O)

589 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audioens/%.c
```



```
590         $(COMPILE.c) -o $@ $<
591         $(CTFCONVERT_O)

593 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audioemu10k/%.c
594         $(COMPILE.c) -o $@ $<
595         $(CTFCONVERT_O)

597 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audio1575/%.c
598         $(COMPILE.c) -o $@ $<
599         $(CTFCONVERT_O)

601 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audio810/%.c
602         $(COMPILE.c) -o $@ $<
603         $(CTFCONVERT_O)

605 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiocmi/%.c
606         $(COMPILE.c) -o $@ $<
607         $(CTFCONVERT_O)

609 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiocmihd/%.c
610         $(COMPILE.c) -o $@ $<
611         $(CTFCONVERT_O)

613 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiohd/%.c
614         $(COMPILE.c) -o $@ $<
615         $(CTFCONVERT_O)

617 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audioixp/%.c
618         $(COMPILE.c) -o $@ $<
619         $(CTFCONVERT_O)

621 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiols/%.c
622         $(COMPILE.c) -o $@ $<
623         $(CTFCONVERT_O)

625 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiopci/%.c
626         $(COMPILE.c) -o $@ $<
627         $(CTFCONVERT_O)

629 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiop16x/%.c
630         $(COMPILE.c) -o $@ $<
631         $(CTFCONVERT_O)

633 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiosolo/%.c
634         $(COMPILE.c) -o $@ $<
635         $(CTFCONVERT_O)

637 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiots/%.c
638         $(COMPILE.c) -o $@ $<
639         $(CTFCONVERT_O)

641 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiovia823x/%.c
642         $(COMPILE.c) -o $@ $<
643         $(CTFCONVERT_O)

645 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiovia97/%.c
646         $(COMPILE.c) -o $@ $<
647         $(CTFCONVERT_O)

649 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bfe/%.c
650         $(COMPILE.c) -o $@ $<
651         $(CTFCONVERT_O)

653 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bge/%.c
654         $(COMPILE.c) -o $@ $<
655         $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/blkdev/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bpf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/cardbus/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/stmf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/port/fct/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/port/qlt/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/port/srpt/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/port/fcoet/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/iscsit/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/port/iscsit/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/port/pppt/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/lu/stmf_sbd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/cpqary3/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/dld/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/dls/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/dmfe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/drm/%.c
```



```
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/efe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/elxl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fcoe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/hme/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/pciex/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/hotplug/hpcsvc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/pciex/hotplug/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/hotplug/pcihp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/rds/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/rdsv3/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/iser/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/ibd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/eoib/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/of/sol_ofs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/of/sol_ucma/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/of/sol_umad/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
789 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/of/sol_uverbs/%.
790         $(COMPILE.c) -o $@ $<
791         $(CTFCONVERT_O)

793 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/sdp/%.c
794         $(COMPILE.c) -o $@ $<
795         $(CTFCONVERT_O)

797 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/mgt/ibcm/%.c
798         $(COMPILE.c) -o $@ $<
799         $(CTFCONVERT_O)

801 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/mgt/ibdm/%.c
802         $(COMPILE.c) -o $@ $<
803         $(CTFCONVERT_O)

805 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/mgt/ibdma/%.c
806         $(COMPILE.c) -o $@ $<
807         $(CTFCONVERT_O)

809 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/mgt/ibmf/%.c
810         $(COMPILE.c) -o $@ $<
811         $(CTFCONVERT_O)

813 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/ibnex/%.c
814         $(COMPILE.c) -o $@ $<
815         $(CTFCONVERT_O)

817 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/ibtl/%.c
818         $(COMPILE.c) -o $@ $<
819         $(CTFCONVERT_O)

821 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/adapters/tavor/%.c
822         $(COMPILE.c) -o $@ $<
823         $(CTFCONVERT_O)

825 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/adapters/hermon/%.c
826         $(COMPILE.c) -o $@ $<
827         $(CTFCONVERT_O)

829 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/daplt/%.c
830         $(COMPILE.c) -o $@ $<
831         $(CTFCONVERT_O)

833 $(OBJS_DIR)/%.o:                $(COMMONBASE)/iscsi/%.c
834         $(COMPILE.c) -o $@ $<
835         $(CTFCONVERT_O)

837 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/idm/%.c
838         $(COMPILE.c) -o $@ $<
839         $(CTFCONVERT_O)

841 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ipw/%.c
842         $(COMPILE.c) -o $@ $<
843         $(CTFCONVERT_O)

845 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/iwh/%.c
846         $(COMPILE.c) -o $@ $<
847         $(CTFCONVERT_O)

849 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/iwi/%.c
850         $(COMPILE.c) -o $@ $<
851         $(CTFCONVERT_O)

853 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/iwk/%.c
```



```
854         $(COMPILE.c) -o $@ $<
855         $(CTFCONVERT_O)

857 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/iwp/%.c
858         $(COMPILE.c) -o $@ $<
859         $(CTFCONVERT_O)

861 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/kb8042/%.c
862         $(COMPILE.c) -o $@ $<
863         $(CTFCONVERT_O)

865 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/kbtrans/%.c
866         $(COMPILE.c) -o $@ $<
867         $(CTFCONVERT_O)

869 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ksocket/%.c
870         $(COMPILE.c) -o $@ $<
871         $(CTFCONVERT_O)

873 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/aggr/%.c
874         $(COMPILE.c) -o $@ $<
875         $(CTFCONVERT_O)

877 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/lp/%.c
878         $(COMPILE.c) -o $@ $<
879         $(CTFCONVERT_O)

881 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/lvm/hotspares/%.c
882         $(COMPILE.c) -o $@ $<
883         $(CTFCONVERT_O)

885 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/lvm/md/%.c
886         $(COMPILE.c) -o $@ $<
887         $(CTFCONVERT_O)

889 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/lvm/mirror/%.c
890         $(COMPILE.c) -o $@ $<
891         $(CTFCONVERT_O)

893 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/lvm/notify/%.c
894         $(COMPILE.c) -o $@ $<
895         $(CTFCONVERT_O)

897 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/lvm/raid/%.c
898         $(COMPILE.c) -o $@ $<
899         $(CTFCONVERT_O)

901 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/lvm/softpart/%.c
902         $(COMPILE.c) -o $@ $<
903         $(CTFCONVERT_O)

905 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/lvm/stripe/%.c
906         $(COMPILE.c) -o $@ $<
907         $(CTFCONVERT_O)

909 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/lvm/trans/%.c
910         $(COMPILE.c) -o $@ $<
911         $(CTFCONVERT_O)

913 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mac/%.c
914         $(COMPILE.c) -o $@ $<
915         $(CTFCONVERT_O)

917 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mac/plugins/%.c
918         $(COMPILE.c) -o $@ $<
919         $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mega_sas/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mii/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mr_sas/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/adapters/mpt_sas/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mxfe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mwl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mwl/mwl_fw/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/net80211/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/nge/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/nxge/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/nxge/npi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/nxge/%.s
        $(COMPILE.s) -o $@ $<

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/pci-ide/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/pcmcia/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/pcan/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/pcn/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/pcwl/%.c
        $(COMPILE.c) -o $@ $<
```



```
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ppp/sppp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ppp/spppasyn/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ppp/sppptun/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ral/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/rge/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/rtls/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/rsm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/rtw/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/rum/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/rwd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/rwn/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sata/adapters/ahci/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sata/adapters/nv_sata/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sata/adapters/si3124/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sata/impl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/conf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/impl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/targets/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/adapters/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/adapters/blk2scsa/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/adapters/scsi_vhci/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/adapters/scsi_vhci/fop
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fibre-channel/ulp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fibre-channel/impl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fibre-channel/fca/qlc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fibre-channel/fca/qlge/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fibre-channel/fca/emlxs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fibre-channel/fca/oce/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fibre-channel/fca/fcoei/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sdcard/adapters/sdhost/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sdcard/impl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sdcard/targets/sdcard/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sfe/%.c
        $(COMPILE.c) -o $@ $<
```



```
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/simnet/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/softmac/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/uath/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/uath/uath_fw/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ural/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/urtw/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/audio/usb_ac/%.
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/audio/usb_as/%.
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/audio/usb_ah/%.
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbskel/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/video/usbvc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/hwarc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/hid/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/hidparser/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/printer/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbkbm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
1184 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbms/%.c
1185         $(COMPILE.c) -o $@ $<
1186         $(CTFCONVERT_O)

1188 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbinput/usbwcm
1189         $(COMPILE.c) -o $@ $<
1190         $(CTFCONVERT_O)

1192 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/ugen/%.c
1193         $(COMPILE.c) -o $@ $<
1194         $(CTFCONVERT_O)

1196 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbser/%.c
1197         $(COMPILE.c) -o $@ $<
1198         $(CTFCONVERT_O)

1200 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbser/usbsacm/
1201         $(COMPILE.c) -o $@ $<
1202         $(CTFCONVERT_O)

1204 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbser/usbftdi/
1205         $(COMPILE.c) -o $@ $<
1206         $(CTFCONVERT_O)

1208 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbser/usbser_k
1209         $(COMPILE.c) -o $@ $<
1210         $(CTFCONVERT_O)

1212 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbser/usbsprl/
1213         $(COMPILE.c) -o $@ $<
1214         $(CTFCONVERT_O)

1216 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/wusb_df/%.c
1217         $(COMPILE.c) -o $@ $<
1218         $(CTFCONVERT_O)

1220 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/hwa1480_fw/%.c
1221         $(COMPILE.c) -o $@ $<
1222         $(CTFCONVERT_O)

1224 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/wusb_ca/%.c
1225         $(COMPILE.c) -o $@ $<
1226         $(CTFCONVERT_O)

1228 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbecm/%.c
1229         $(COMPILE.c) -o $@ $<
1230         $(CTFCONVERT_O)

1232 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/hcd/openhci/%.c
1233         $(COMPILE.c) -o $@ $<
1234         $(CTFCONVERT_O)

1236 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/hcd/ehci/%.c
1237         $(COMPILE.c) -o $@ $<
1238         $(CTFCONVERT_O)

1240 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/hcd/uhci/%.c
1241         $(COMPILE.c) -I../../common -o $@ $<
1242         $(CTFCONVERT_O)

1244 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/hubd/%.c
1245         $(COMPILE.c) -o $@ $<
1246         $(CTFCONVERT_O)

1248 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/scsa2usb/%.c
1249         $(COMPILE.c) -o $@ $<
```



```
1250         $(CTFCONVERT_O)

1252 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/usb_mid/%.c
1253         $(COMPILE.c) -o $@ $<
1254         $(CTFCONVERT_O)

1256 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/usb_ia/%.c
1257         $(COMPILE.c) -o $@ $<
1258         $(CTFCONVERT_O)

1260 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/usba/%.c
1261         $(COMPILE.c) -o $@ $<
1262         $(CTFCONVERT_O)

1264 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/usba10/%.c
1265         $(COMPILE.c) -o $@ $<
1266         $(CTFCONVERT_O)

1268 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/hwa/hwahc/%.c
1269         $(COMPILE.c) -o $@ $<
1270         $(CTFCONVERT_O)

1272 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/uwb/uwba/%.c
1273         $(COMPILE.c) -o $@ $<
1274         $(CTFCONVERT_O)

1276 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/vuidmice/%.c
1277         $(COMPILE.c) -o $@ $<
1278         $(CTFCONVERT_O)

1280 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/vnic/%.c
1281         $(COMPILE.c) -o $@ $<
1282         $(CTFCONVERT_O)

1284 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/wpi/%.c
1285         $(COMPILE.c) -o $@ $<
1286         $(CTFCONVERT_O)

1288 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/zyd/%.c
1289         $(COMPILE.c) -o $@ $<
1290         $(CTFCONVERT_O)

1292 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/chxge/com/%.c
1293         $(COMPILE.c) -o $@ $<
1294         $(CTFCONVERT_O)

1296 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/chxge/%.c
1297         $(COMPILE.c) -o $@ $<
1298         $(CTFCONVERT_O)

1300 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/cxgbe/common/%.c
1301         $(COMPILE.c) -o $@ $<
1302         $(CTFCONVERT_O)

1304 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/cxgbe/shared/%.c
1305         $(COMPILE.c) -o $@ $<
1306         $(CTFCONVERT_O)

1308 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/cxgbe/firmware/%.c
1309         $(COMPILE.c) -o $@ $<
1310         $(CTFCONVERT_O)

1312 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/cxgbe/t4nex/%.c
1313         $(COMPILE.c) -o $@ $<
1314         $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/cxgbe/cxgbe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ixgb/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/xge/drv/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/xge/hal/xgehal/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/e1000api/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/e1000g/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/igb/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/iprb/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ixgbe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ntxn/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/myri10ge/drv/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ipp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ipp/ipgpc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ipp/dlcosmk/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ipp/flowacct/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ipp/dscpmk/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ipp/meters/%.c
        $(COMPILE.c) -o $@ $<
```



```
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/kiconv/kiconv_emea/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/kiconv/kiconv_ja/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/kiconv/kiconv_ko/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/kiconv/kiconv_sc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/kiconv/kiconv_tc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/klm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/kmdb/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ktli/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/adapters/iscsi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/iscsi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/kifconf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/vr/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/yge/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/virtio/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/vioblk/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

#
# krtld must refer to its own bzero/bcopy until the kernel is fully linked
#
$(OBJS_DIR)/bootrd.o            := CPPFLAGS += -DKOBJ_OVERRIDES
```

```
1448 $(OBJS_DIR)/doreloc.o           := CPPFLAGS += -DKOBJ_OVERRIDES
1449 $(OBJS_DIR)/kobj.o              := CPPFLAGS += -DKOBJ_OVERRIDES
1450 $(OBJS_DIR)/kobj_boot.o         := CPPFLAGS += -DKOBJ_OVERRIDES
1451 $(OBJS_DIR)/kobj_bootflags.o    := CPPFLAGS += -DKOBJ_OVERRIDES
1452 $(OBJS_DIR)/kobj_convrelstr.o   := CPPFLAGS += -DKOBJ_OVERRIDES
1453 $(OBJS_DIR)/kobj_isa.o          := CPPFLAGS += -DKOBJ_OVERRIDES
1454 $(OBJS_DIR)/kobj_kdi.o          := CPPFLAGS += -DKOBJ_OVERRIDES
1455 $(OBJS_DIR)/kobj_lm.o           := CPPFLAGS += -DKOBJ_OVERRIDES
1456 $(OBJS_DIR)/kobj_reloc.o        := CPPFLAGS += -DKOBJ_OVERRIDES
1457 $(OBJS_DIR)/kobj_stubs.o        := CPPFLAGS += -DKOBJ_OVERRIDES
1458 $(OBJS_DIR)/kobj_subr.o         := CPPFLAGS += -DKOBJ_OVERRIDES

1460 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/krtld/%.c
1461         $(COMPILE.c) -o $@ $<
1462         $(CTFCONVERT_O)

1464 $(OBJS_DIR)/%.o:                $(COMMONBASE)/list/%.c
1465         $(COMPILE.c) -o $@ $<
1466         $(CTFCONVERT_O)

1468 $(OBJS_DIR)/%.o:                $(COMMONBASE)/lvm/%.c
1469         $(COMPILE.c) -o $@ $<
1470         $(CTFCONVERT_O)

1472 $(OBJS_DIR)/%.o:                $(COMMONBASE)/lzma/%.c
1473         $(COMPILE.c) -o $@ $<
1474         $(CTFCONVERT_O)

1476 $(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/md4/%.c
1477         $(COMPILE.c) -o $@ $<
1478         $(CTFCONVERT_O)

1480 $(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/md5/%.c
1481         $(COMPILE.c) -o $@ $<
1482         $(CTFCONVERT_O)

1484 $(OBJS_DIR)/%.o:                $(COMMONBASE)/net/dhcp/%.c
1485         $(COMPILE.c) -o $@ $<
1486         $(CTFCONVERT_O)

1488 $(OBJS_DIR)/%.o:                $(COMMONBASE)/nvpair/%.c
1489         $(COMPILE.c) -o $@ $<
1490         $(CTFCONVERT_O)

1492 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/os/%.c
1493         $(COMPILE.c) -o $@ $<
1494         $(CTFCONVERT_O)

1496 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/pcmcia/cis/%.c
1497         $(COMPILE.c) -o $@ $<
1498         $(CTFCONVERT_O)

1500 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/pcmcia/cs/%.c
1501         $(COMPILE.c) -o $@ $<
1502         $(CTFCONVERT_O)

1504 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/pcmcia/nexus/%.c
1505         $(COMPILE.c) -o $@ $<
1506         $(CTFCONVERT_O)

1508 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/pcmcia/pcs/%.c
1509         $(COMPILE.c) -o $@ $<
1510         $(CTFCONVERT_O)

1512 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/rpc/%.c
1513         $(COMPILE.c) -o $@ $<
```



```
1514         $(CTFCONVERT_O)

1516 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/rpc/sec/%.c
1517         $(COMPILE.c) -o $@ $<
1518         $(CTFCONVERT_O)

1520 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/rpc/sec_gss/%.c
1521         $(COMPILE.c) -o $@ $<
1522         $(CTFCONVERT_O)

1524 $(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/edonr/%.c
1525         $(COMPILE.c) -o $@ $<
1526         $(CTFCONVERT_O)

1528 $(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/sha1/%.c
1529         $(COMPILE.c) -o $@ $<
1530         $(CTFCONVERT_O)

1532 $(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/sha2/%.c
1533         $(COMPILE.c) -o $@ $<
1534         $(CTFCONVERT_O)

1536 $(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/skein/%.c
1537         $(COMPILE.c) -o $@ $<
1538         $(CTFCONVERT_O)

1540 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/syscall/%.c
1541         $(COMPILE.c) -o $@ $<
1542         $(CTFCONVERT_O)

1544 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/tnf/%.c
1545         $(COMPILE.c) -o $@ $<
1546         $(CTFCONVERT_O)

1548 $(OBJS_DIR)/%.o:                $(COMMONBASE)/tsol/%.c
1549         $(COMPILE.c) -o $@ $<
1550         $(CTFCONVERT_O)

1552 $(OBJS_DIR)/%.o:                $(COMMONBASE)/util/%.c
1553         $(COMPILE.c) -o $@ $<
1554         $(CTFCONVERT_O)

1556 $(OBJS_DIR)/%.o:                $(COMMONBASE)/unicode/%.c
1557         $(COMPILE.c) -o $@ $<
1558         $(CTFCONVERT_O)

1560 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/vm/%.c
1561         $(COMPILE.c) -o $@ $<
1562         $(CTFCONVERT_O)

1564 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/zmod/%.c
1565         $(COMPILE.c) -o $@ $<
1566         $(CTFCONVERT_O)

1568 $(OBJS_DIR)/zlib_obj.o:         $(ZLIB_OBJS:%=$(OBJS_DIR)/%)
1569         $(LD) -r -Breduce -M$(UTSBASE)/common/zmod/mapfile -o $@ \
1570             $(ZLIB_OBJS:%=$(OBJS_DIR)/%)
1571         $(CTFMERGE) -t -f -L VERSION -o $@ $(ZLIB_OBJS:%=$(OBJS_DIR)/%)

1573 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/hxge/%.c
1574         $(COMPILE.c) -o $@ $<
1575         $(CTFCONVERT_O)

1577 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/tpm/%.c
1578         $(COMPILE.c) -o $@ $<
1579         $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/tpm/%.s
        $(COMPILE.s) -o $@ $<

$(OBJS_DIR)/bz2%.o:             $(COMMONBASE)/bzip2/%.c
        $(COMPILE.c) -o $@ -I$(COMMONBASE)/bzip2 $<
        $(CTFCONVERT_O)

BZ2LINT = -erroff=%all -I$(UTSBASE)/common/bzip2

$(LINTS_DIR)/bz2%.ln:           $(COMMONBASE)/bzip2/%.c
         @($(LHEAD) $(LINT.c) -C $(LINTS_DIR)/‘basename $@ .ln‘ $(BZ2LINT) $< $(

#
#       Section 1b:     Lint ‘objects’
#
$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/aes/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/arcfour/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/blowfish/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/ecc/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/modes/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/padding/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/rng/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/rsa/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/bignum/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/bignum/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/mpi/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/acl/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/avl/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/ucode/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/brand/sn1/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/brand/solaris10/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/c2/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```



```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/conf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/contract/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/cpr/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ctf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/ctf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/pci/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/devid/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/des/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/smbios/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/avs/ncall/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/avs/ns/dsw/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/avs/ns/nsctl/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/avs/ns/rdc/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/avs/ns/sdbc/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/avs/ns/solaris/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/avs/ns/sv/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/avs/ns/unistat/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/des/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/crypto/api/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/crypto/core/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/crypto/io/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/crypto/spi/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
1713 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/disp/%.c
1714         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1716 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/dtrace/%.c
1717         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1719 $(LINTS_DIR)/%.ln:             $(COMMONBASE)/exacct/%.c
1720         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1722 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/exec/aout/%.c
1723         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1725 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/exec/elf/%.c
1726         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1728 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/exec/intp/%.c
1729         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1731 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/exec/shbin/%.c
1732         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1734 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/exec/java/%.c
1735         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1737 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/%.c
1738         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1740 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/autofs/%.c
1741         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1743 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/cachefs/%.c
1744         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1746 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/ctfs/%.c
1747         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1749 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/doorfs/%.c
1750         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1752 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/dcfs/%.c
1753         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1755 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/devfs/%.c
1756         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1758 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/dev/%.c
1759         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1761 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/fd/%.c
1762         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1764 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/fifofs/%.c
1765         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1767 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/hsfs/%.c
1768         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1770 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/lofs/%.c
1771         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1773 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/mntfs/%.c
1774         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1776 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/namefs/%.c
1777         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```



```
1779 $(LINTS_DIR)/%.ln:             $(COMMONBASE)/smbsrv/%.c
1780         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1782 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/smbsrv/%.c
1783         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1785 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/nfs/%.c
1786         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1788 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/objfs/%.c
1789         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1791 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/pcfs/%.c
1792         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1794 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/portfs/%.c
1795         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1797 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/proc/%.c
1798         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1800 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/sharefs/%.c
1801         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1803 $(LINTS_DIR)/%.ln:             $(COMMONBASE)/smbclnt/%.c
1804         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1806 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/smbclnt/netsmb/%.c
1807         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1809 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/smbclnt/smbfs/%.c
1810         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1812 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/sockfs/%.c
1813         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1815 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/specfs/%.c
1816         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1818 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/swapfs/%.c
1819         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1821 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/tmpfs/%.c
1822         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1824 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/udfs/%.c
1825         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1827 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/ufs/%.c
1828         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1830 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/ufs_log/%.c
1831         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1833 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/io/vscan/%.c
1834         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1836 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/zfs/%.c
1837         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1839 $(LINTS_DIR)/%.ln:             $(UTSBASE)/common/fs/zut/%.c
1840         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1842 $(LINTS_DIR)/%.ln:             $(COMMONBASE)/xattr/%.c
1843         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(COMMONBASE)/zfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/gssapi/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/gssapi/mechs/dummy/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/des/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/dk/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/os/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/arcfour/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/enc_provider/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/hash_provider/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/keyhash_provider/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/raw/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/old/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/krb5/krb/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/krb5/os/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/mech/%.c
        @($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/idmap/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/sockmods/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/arp/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/ip/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```



```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/ipnet/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/iptun/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/ipf/%.c
        @($(LHEAD) $(LINT.c) $(IPFFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/kssl/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/net/patricia/%.c
        @($(LHEAD) $(LINT.c) $(IPFFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/udp/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/sctp/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/tcp/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/ilb/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/nca/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/dlpistub/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/1394/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/1394/adapters/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/1394/targets/av1394/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/1394/targets/dcam1394/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/1394/targets/scsa1394/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sbp2/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/aac/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/afe/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/atge/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/arn/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
1977 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ath/%.c
1978         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1980 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/atu/%.c
1981         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1983 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/impl/%.c
1984         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1986 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/ac97/%.c
1987         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1989 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audio1575/%.c
1990         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1992 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audio810/%.c
1993         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1995 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiocmi/%.c
1996         @($(LHEAD) $(LINT.c) $< $(LTAIL))

1998 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiocmihd/%.c
1999         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2001 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audioens/%.c
2002         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2004 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audioemu10k/%.c
2005         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2007 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiohd/%.c
2008         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2010 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audioixp/%.c
2011         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2013 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiols/%.c
2014         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2016 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiopci/%.c
2017         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2019 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiop16x/%.c
2020         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2022 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiosolo/%.c
2023         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2025 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiots/%.c
2026         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2028 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiovia823x/%.c
2029         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2031 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiovia97/%.c
2032         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2034 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/bfe/%.c
2035         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2037 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/bpf/%.c
2038         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2040 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/bge/%.c
2041         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```



```
2043 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/blkdev/%.c
2044         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2046 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/cardbus/%.c
2047         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2049 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/lu/stmf_sbd/%.c
2050         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2052 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/port/fct/%.c
2053         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2055 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/port/qlt/%.c
2056         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2058 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/port/srpt/%.c
2059         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2061 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/iscsit/%.c
2062         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2064 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/port/fcoet/%.c
2065         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2067 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/port/iscsit/%.c
2068         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2070 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/port/pppt/%.c
2071         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2073 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/stmf/%.c
2074         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2076 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/cpqary3/%.c
2077         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2079 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/dld/%.c
2080         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2082 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/dls/%.c
2083         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2085 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/dmfe/%.c
2086         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2088 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/drm/%.c
2089         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2091 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/efe/%.c
2092         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2094 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/elxl/%.c
2095         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2097 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fcoe/%.c
2098         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2100 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/hme/%.c
2101         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2103 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/pciex/%.c
2104         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2106 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/hotplug/hpcsvc/%.c
2107         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
2109 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/pciex/hotplug/%.c
2110         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2112 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/hotplug/pcihp/%.c
2113         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2115 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/rds/%.c
2116         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2118 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/rdsv3/%.c
2119         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2121 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/iser/%.c
2122         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2124 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/ibd/%.c
2125         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2127 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/eoib/%.c
2128         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2130 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/of/sol_ofs/%.c
2131         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2133 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/of/sol_ucma/%.c
2134         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2136 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/of/sol_umad/%.c
2137         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2139 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/of/sol_uverbs/%.
2140         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2142 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/sdp/%.c
2143         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2145 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/mgt/ibcm/%.c
2146         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2148 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/mgt/ibdm/%.c
2149         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2151 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/mgt/ibdma/%.c
2152         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2154 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/mgt/ibmf/%.c
2155         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2157 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/ibnex/%.c
2158         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2160 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/ibtl/%.c
2161         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2163 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/adapters/tavor/%.c
2164         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2166 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/adapters/hermon/%.c
2167         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2169 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/daplt/%.c
2170         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2172 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/iscsi/%.c
2173         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```



```
2175 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/idm/%.c
2176         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2178 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ipw/%.c
2179         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2181 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/iwh/%.c
2182         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2184 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/iwi/%.c
2185         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2187 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/iwk/%.c
2188         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2190 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/iwp/%.c
2191         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2193 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/kb8042/%.c
2194         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2196 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/kbtrans/%.c
2197         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2199 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ksocket/%.c
2200         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2202 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/aggr/%.c
2203         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2205 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/lp/%.c
2206         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2208 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/lvm/hotspares/%.c
2209         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2211 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/lvm/md/%.c
2212         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2214 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/lvm/mirror/%.c
2215         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2217 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/lvm/raid/%.c
2218         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2220 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/lvm/softpart/%.c
2221         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2223 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/lvm/stripe/%.c
2224         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2226 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/lvm/notify/%.c
2227         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2229 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/lvm/trans/%.c
2230         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2232 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/mac/%.c
2233         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2235 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/mac/plugins/%.c
2236         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2238 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/mega_sas/%.c
2239         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
2241 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/mii/%.c
2242         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2244 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/mr_sas/%.c
2245         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2247 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/scsi/adapters/mpt_sas/%.c
2248         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2250 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/mxfe/%.c
2251         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2253 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/mwl/%.c
2254         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2256 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/mwl/mwl_fw/%.c
2257         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2259 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/net80211/%.c
2260         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2262 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/nge/%.c
2263         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2265 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/nxge/%.c
2266         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2268 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/nxge/%.s
2269         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2271 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/nxge/npi/%.c
2272         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2274 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/pci-ide/%.c
2275         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2277 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/pcmcia/%.c
2278         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2280 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/pcan/%.c
2281         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2283 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/pcn/%.c
2284         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2286 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/pcwl/%.c
2287         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2289 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ppp/sppp/%.c
2290         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2292 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ppp/spppasyn/%.c
2293         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2295 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ppp/sppptun/%.c
2296         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2298 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ral/%.c
2299         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2301 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/rge/%.c
2302         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2304 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/rtls/%.c
2305         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```



```
2307 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/rsm/%.c
2308         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2310 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/rtw/%.c
2311         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2313 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/rum/%.c
2314         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2316 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/rwd/%.c
2317         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2319 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/rwn/%.c
2320         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2322 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sata/adapters/ahci/%.c
2323         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2325 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sata/adapters/nv_sata/%.c
2326         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2328 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sata/adapters/si3124/%.c
2329         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2331 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sata/impl/%.c
2332         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2334 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/scsi/adapters/%.c
2335         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2337 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/scsi/adapters/blk2scsa/%.c
2338         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2340 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/scsi/adapters/pmcs/%.c
2341         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2343 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/scsi/adapters/scsi_vhci/%.c
2344         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2346 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/scsi/adapters/scsi_vhci/fop
2347         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2349 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fibre-channel/ulp/%.c
2350         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2352 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fibre-channel/impl/%.c
2353         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2355 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fibre-channel/fca/qlc/%.c
2356         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2358 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fibre-channel/fca/qlge/%.c
2359         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2361 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fibre-channel/fca/emlxs/%.c
2362         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2364 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fibre-channel/fca/oce/%.c
2365         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2367 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fibre-channel/fca/fcoei/%.c
2368         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2370 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/scsi/conf/%.c
2371         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
2373 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/scsi/impl/%.c
2374         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2376 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/scsi/targets/%.c
2377         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2379 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sdcard/adapters/sdhost/%.c
2380         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2382 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sdcard/impl/%.c
2383         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2385 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sdcard/targets/sdcard/%.c
2386         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2388 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sfe/%.c
2389         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2391 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/simnet/%.c
2392         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2394 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/softmac/%.c
2395         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2397 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/uath/%.c
2398         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2400 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/uath/uath_fw/%.c
2401         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2403 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ural/%.c
2404         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2406 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/urtw/%.c
2407         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2409 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/audio/usb_ac/%.
2410         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2412 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/audio/usb_as/%.
2413         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2415 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/audio/usb_ah/%.
2416         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2418 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbskel/%.c
2419         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2421 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/video/usbvc/%.c
2422         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2424 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/hwarc/%.c
2425         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2427 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/hid/%.c
2428         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2430 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/hidparser/%.c
2431         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2433 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbkbm/%.c
2434         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2436 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbms/%.c
2437         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```



```
2439 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbinput/usbwcm
2440         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2442 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/ugen/%.c
2443         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2445 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/printer/%.c
2446         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2448 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbser/%.c
2449         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2451 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbser/usbsacm/
2452         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2454 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbser/usbftdi/
2455         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2457 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbser/usbser_k
2458         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2460 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbser/usbsprl/
2461         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2463 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/wusb_df/%.c
2464         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2466 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/hwa1480_fw/%.c
2467         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2469 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/wusb_ca/%.c
2470         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2472 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbecm/%.c
2473         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2475 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/hcd/openhci/%.c
2476         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2478 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/hcd/ehci/%.c
2479         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2481 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/hcd/uhci/%.c
2482         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2484 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/hubd/%.c
2485         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2487 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/scsa2usb/%.c
2488         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2490 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/usb_mid/%.c
2491         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2493 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/usb_ia/%.c
2494         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2496 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/usba/%.c
2497         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2499 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/usba10/%.c
2500         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2502 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/uwb/uwba/%.c
2503         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
2505 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/hwa/hwahc/%.c
2506         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2508 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/vuidmice/%.c
2509         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2511 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/vnic/%.c
2512         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2514 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/wpi/%.c
2515         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2517 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/zyd/%.c
2518         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2520 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/chxge/com/%.c
2521         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2523 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/chxge/%.c
2524         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2526 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/cxgbe/common/%.c
2527         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2529 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/cxgbe/shared/%.c
2530         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2532 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/cxgbe/firmware/%.c
2533         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2535 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/cxgbe/t4nex/%.c
2536         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2538 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/cxgbe/cxgbe/%.c
2539         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2541 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ixgb/%.c
2542         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2544 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/xge/drv/%.c
2545         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2547 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/xge/hal/xgehal/%.c
2548         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2550 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/e1000g/%.c
2551         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2553 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/e1000api/%.c
2554         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2556 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/igb/%.c
2557         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2559 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/iprb/%.c
2560         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2562 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ixgbe/%.c
2563         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2565 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ntxn/%.c
2566         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2568 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/myri10ge/drv/%.c
2569         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```



```
2571 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ipp/%.c
2572         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2574 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ipp/ipgpc/%.c
2575         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2577 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ipp/dlcosmk/%.c
2578         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2580 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ipp/flowacct/%.c
2581         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2583 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ipp/dscpmk/%.c
2584         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2586 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ipp/meters/%.c
2587         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2589 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/kiconv/kiconv_emea/%.c
2590         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2592 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/kiconv/kiconv_ja/%.c
2593         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2595 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/kiconv/kiconv_ko/%.c
2596         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2598 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/kiconv/kiconv_sc/%.c
2599         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2601 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/kiconv/kiconv_tc/%.c
2602         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2604 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/klm/%.c
2605         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2607 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/kmdb/%.c
2608         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2610 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/krtld/%.c
2611         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2613 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ktli/%.c
2614         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2616 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/list/%.c
2617         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2619 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/lvm/%.c
2620         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2622 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/lzma/%.c
2623         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2625 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/md4/%.c
2626         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2628 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/md5/%.c
2629         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2631 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/net/dhcp/%.c
2632         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2634 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/nvpair/%.c
2635         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
2637 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/os/%.c
2638         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2640 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/rpc/%.c
2641         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2643 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/pcmcia/cs/%.c
2644         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2646 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/pcmcia/cis/%.c
2647         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2649 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/pcmcia/nexus/%.c
2650         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2652 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/pcmcia/pcs/%.c
2653         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2655 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/rpc/%.c
2656         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2658 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/rpc/sec/%.c
2659         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2661 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/rpc/sec_gss/%.c
2662         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2664 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/edonr/%.c
2665         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2667 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/sha1/%.c
2668         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2670 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/sha2/%.c
2671         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2673 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/skein/%.c
2674         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2676 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/syscall/%.c
2677         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2679 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/tnf/%.c
2680         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2682 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/tsol/%.c
2683         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2685 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/util/%.c
2686         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2688 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/unicode/%.c
2689         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2691 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/vm/%.c
2692         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2694 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/scsi/adapters/iscsi/%.c
2695         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2697 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/iscsi/%.c
2698         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2700 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/kifconf/%.c
2701         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```



```
2703 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/virtio/%.c
2704         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2706 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/vioblk/%.c
2707         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2709 ZMODLINTFLAGS = -erroff=E_CONSTANT_CONDITION

2711 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/zmod/%.c
2712         @($(LHEAD) $(LINT.c) $(ZMODLINTFLAGS) $< $(LTAIL))

2714 $(LINTS_DIR)/zlib_obj.ln:       $(ZLIB_OBJS:%.o=$(LINTS_DIR)/%.ln) \
2715                                 $(UTSBASE)/common/zmod/zlib_lint.c
2716         @($(LHEAD) $(LINT.c) -C $(LINTS_DIR)/zlib_obj \
2717             $(UTSBASE)/common/zmod/zlib_lint.c $(LTAIL))

2719 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/hxge/%.c
2720         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2722 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/tpm/%.c
2723         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2725 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/tpm/%.s
2726         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2728 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/vr/%.c
2729         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2731 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/yge/%.c
2732         @($(LHEAD) $(LINT.c) $< $(LTAIL))

2734 $(LINTS_DIR)/%.ln:              $(COMMONBASE)/fsreparse/%.c
2735         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
**********************************************************
    1631 Mon Oct  7 19:17:18 2013
new/usr/src/uts/common/crypto/io/edonr_mod.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://opensource.org/licenses/CDDL-1.0.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2013 Saso Kiselkov. All rights reserved.
 */

#include <sys/modctl.h>
#include <sys/crypto/common.h>
#include <sys/crypto/spi.h>
#include <sys/sysmacros.h>
#include <sys/systm.h>
#include <sys/edonr.h>

/*
 * Unlike sha2 or skein, we won't expose edonr via the Kernel Cryptographic
 * Framework (KCF), because Edon-R is *NOT* suitable for general-purpose
 * cryptographic use. Users of Edon-R must interface directly to this module.
 */

static struct modlmisc modlmisc = {
        &mod_miscops,
        "Edon-R Message-Digest Algorithm"
};

static struct modlinkage modlinkage = {
        MODREV_1, &modlmisc, NULL
};

int
_init(void)
{
        int error;

        if ((error = mod_install(&modlinkage)) != 0)
                return (error);

        return (0);
}

int
_info(struct modinfo *modinfop)
{
```



```
        return (mod_info(&modlinkage, modinfop));
}
```

```
**********************************************************
   22784 Mon Oct  7 19:17:18 2013
new/usr/src/uts/common/crypto/io/skein_mod.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://opensource.org/licenses/CDDL-1.0.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2013 Saso Kiselkov. All rights reserved.
 */

#include <sys/modctl.h>
#include <sys/crypto/common.h>
#include <sys/crypto/spi.h>
#include <sys/strsun.h>
#include <sys/sysmacros.h>
#include <sys/systm.h>
#define SKEIN_MODULE_IMPL
#include <sys/skein.h>

/*
 * Like the sha2 module, we create the skein module with two modlinkages:
 * - modlmisc to allow direct calls to Skein_* API functions.
 * - modlcrypto to integrate well into the Kernel Crypto Framework (KCF).
 */
static struct modlmisc modlmisc = {
        &mod_miscops,
        "Skein Message-Digest Algorithm"
};

static struct modlcrypto modlcrypto = {
        &mod_cryptoops,
        "Skein Kernel SW Provider"
};

static struct modlinkage modlinkage = {
        MODREV_1, &modlmisc, &modlcrypto, NULL
};

static crypto_mech_info_t skein_mech_info_tab[] = {
        {CKM_SKEIN_256, SKEIN_256_MECH_INFO_TYPE,
            CRYPTO_FG_DIGEST | CRYPTO_FG_DIGEST_ATOMIC,
            0, 0, CRYPTO_KEYSIZE_UNIT_IN_BITS},
        {CKM_SKEIN_256_MAC, SKEIN_256_MAC_MECH_INFO_TYPE,
            CRYPTO_FG_MAC | CRYPTO_FG_MAC_ATOMIC, 1, INT_MAX,
            CRYPTO_KEYSIZE_UNIT_IN_BYTES},
        {CKM_SKEIN_512, SKEIN_512_MECH_INFO_TYPE,
```



```
            CRYPTO_FG_DIGEST | CRYPTO_FG_DIGEST_ATOMIC,
            0, 0, CRYPTO_KEYSIZE_UNIT_IN_BITS},
        {CKM_SKEIN_512_MAC, SKEIN_512_MAC_MECH_INFO_TYPE,
            CRYPTO_FG_MAC | CRYPTO_FG_MAC_ATOMIC, 1, INT_MAX,
            CRYPTO_KEYSIZE_UNIT_IN_BYTES},
        {CKM_SKEIN1024, SKEIN1024_MECH_INFO_TYPE,
            CRYPTO_FG_DIGEST | CRYPTO_FG_DIGEST_ATOMIC,
            0, 0, CRYPTO_KEYSIZE_UNIT_IN_BITS},
        {CKM_SKEIN1024_MAC, SKEIN1024_MAC_MECH_INFO_TYPE,
            CRYPTO_FG_MAC | CRYPTO_FG_MAC_ATOMIC, 1, INT_MAX,
            CRYPTO_KEYSIZE_UNIT_IN_BYTES}
};

static void skein_provider_status(crypto_provider_handle_t, uint_t *);

static crypto_control_ops_t skein_control_ops = {
        skein_provider_status
};

static int skein_digest_init(crypto_ctx_t *, crypto_mechanism_t *,
    crypto_req_handle_t);
static int skein_digest(crypto_ctx_t *, crypto_data_t *, crypto_data_t *,
    crypto_req_handle_t);
static int skein_update(crypto_ctx_t *, crypto_data_t *, crypto_req_handle_t);
static int skein_final(crypto_ctx_t *, crypto_data_t *, crypto_req_handle_t);
static int skein_digest_atomic(crypto_provider_handle_t, crypto_session_id_t,
    crypto_mechanism_t *, crypto_data_t *, crypto_data_t *,
    crypto_req_handle_t);

static crypto_digest_ops_t skein_digest_ops = {
        skein_digest_init,
        skein_digest,
        skein_update,
        NULL,
        skein_final,
        skein_digest_atomic
};

static int skein_mac_init(crypto_ctx_t *, crypto_mechanism_t *, crypto_key_t *,
    crypto_spi_ctx_template_t, crypto_req_handle_t);
static int skein_mac_atomic(crypto_provider_handle_t, crypto_session_id_t,
    crypto_mechanism_t *, crypto_key_t *, crypto_data_t *, crypto_data_t *,
    crypto_spi_ctx_template_t, crypto_req_handle_t);

static crypto_mac_ops_t skein_mac_ops = {
        skein_mac_init,
        NULL,
        skein_update,   /* using regular digest update is OK here */
        skein_final,    /* using regular digest final is OK here */
        skein_mac_atomic,
        NULL
};

static int skein_create_ctx_template(crypto_provider_handle_t,
    crypto_mechanism_t *, crypto_key_t *, crypto_spi_ctx_template_t *,
    size_t *, crypto_req_handle_t);
static int skein_free_context(crypto_ctx_t *);

static crypto_ctx_ops_t skein_ctx_ops = {
        skein_create_ctx_template,
        skein_free_context
};

static crypto_ops_t skein_crypto_ops = {
        &skein_control_ops,
        &skein_digest_ops,
```

```
128         NULL,
129         &skein_mac_ops,
130         NULL,
131         NULL,
132         NULL,
133         NULL,
134         NULL,
135         NULL,
136         NULL,
137         NULL,
138         NULL,
139         &skein_ctx_ops,
140         NULL,
141         NULL,
142         NULL
143 };

145 static crypto_provider_info_t skein_prov_info = {
146         CRYPTO_SPI_VERSION_4,
147         "Skein Software Provider",
148         CRYPTO_SW_PROVIDER,
149         {&modlinkage},
150         NULL,
151         &skein_crypto_ops,
152         sizeof (skein_mech_info_tab) / sizeof (crypto_mech_info_t),
153         skein_mech_info_tab
154 };

156 static crypto_kcf_provider_handle_t skein_prov_handle = NULL;

158 typedef struct skein_ctx {
159         skein_mech_type_t               sc_mech_type;
160         size_t                          sc_digest_bitlen;
161         /*LINTED(E_ANONYMOUS_UNION_DECL)*/
162         union {
163                 Skein_256_Ctxt_t        sc_256;
164                 Skein_512_Ctxt_t        sc_512;
165                 Skein1024_Ctxt_t        sc_1024;
166         };
167 } skein_ctx_t;
168 #define SKEIN_CTX(_ctx_)        ((skein_ctx_t *)((_ctx_)->cc_provider_private))
169 #define SKEIN_CTX_LVALUE(_ctx_) (_ctx_)->cc_provider_private
170 #define SKEIN_OP(_skein_ctx, _op, ...)                                  \
171         do {                                                            \
172                 skein_ctx_t     *sc = (_skein_ctx);                     \
173                 switch (sc->sc_mech_type) {                             \
174                 case SKEIN_256_MECH_INFO_TYPE:                          \
175                 case SKEIN_256_MAC_MECH_INFO_TYPE:                      \
176                         (void) Skein_256_ ## _op(&sc->sc_256, __VA_ARGS__);\
177                         break;                                          \
178                 case SKEIN_512_MECH_INFO_TYPE:                          \
179                 case SKEIN_512_MAC_MECH_INFO_TYPE:                      \
180                         (void) Skein_512_ ## _op(&sc->sc_512, __VA_ARGS__);\
181                         break;                                          \
182                 case SKEIN1024_MECH_INFO_TYPE:                          \
183                 case SKEIN1024_MAC_MECH_INFO_TYPE:                      \
184                         (void) Skein1024_ ## _op(&sc->sc_1024, __VA_ARGS__);\
185                         break;                                          \
186                 }                                                       \
187                 _NOTE(CONSTCOND)                                        \
188         } while (0)

190 static int
191 skein_get_digest_bitlen(const crypto_mechanism_t *mechanism, size_t *result)
192 {
193         if (mechanism->cm_param != NULL) {
```



```
194                 /*LINTED(E_BAD_PTR_CAST_ALIGN)*/
195                 skein_param_t   *param = (skein_param_t *)mechanism->cm_param;

197                 if (mechanism->cm_param_len != sizeof (*param) ||
198                     param->sp_digest_bitlen == 0) {
199                         return (CRYPTO_MECHANISM_PARAM_INVALID);
200                 }
201                 *result = param->sp_digest_bitlen;
202         } else {
203                 switch (mechanism->cm_type) {
204                 case SKEIN_256_MECH_INFO_TYPE:
205                         *result = 256;
206                         break;
207                 case SKEIN_512_MECH_INFO_TYPE:
208                         *result = 512;
209                         break;
210                 case SKEIN1024_MECH_INFO_TYPE:
211                         *result = 1024;
212                         break;
213                 default:
214                         return (CRYPTO_MECHANISM_INVALID);
215                 }
216         }
217         return (CRYPTO_SUCCESS);
218 }

220 int
221 _init(void)
222 {
223         int error;

225         if ((error = mod_install(&modlinkage)) != 0)
226                 return (error);

228         /*
229          * Try to register with KCF - failure shouldn't unload us, since we
230          * still may want to continue providing misc/skein functionality.
231          */
232         (void) crypto_register_provider(&skein_prov_info, &skein_prov_handle);

234         return (0);
235 }

237 int
238 _info(struct modinfo *modinfop)
239 {
240         return (mod_info(&modlinkage, modinfop));
241 }

243 /*
244  * KCF software provider control entry points.
245  */
246 /* ARGSUSED */
247 static void
248 skein_provider_status(crypto_provider_handle_t provider, uint_t *status)
249 {
250         *status = CRYPTO_PROVIDER_READY;
251 }

253 /*
254  * General Skein hashing helper functions.
255  */

257 /*
258  * Performs an Update on a context with uio input data.
259  */
```

```
260 static int
261 skein_digest_update_uio(skein_ctx_t *ctx, const crypto_data_t *data)
262 {
263         off_t           offset = data->cd_offset;
264         size_t          length = data->cd_length;
265         uint_t          vec_idx;
266         size_t          cur_len;
267         const uio_t     *uio = data->cd_uio;

269         /* we support only kernel buffer */
270         if (uio->uio_segflg != UIO_SYSSPACE)
271                 return (CRYPTO_ARGUMENTS_BAD);

273         /*
274          * Jump to the first iovec containing data to be
275          * digested.
276          */
277         for (vec_idx = 0; vec_idx < uio->uio_iovcnt &&
278             offset >= uio->uio_iov[vec_idx].iov_len;
279             offset -= uio->uio_iov[vec_idx++].iov_len)
280                 ;
281         if (vec_idx == uio->uio_iovcnt) {
282                 /*
283                  * The caller specified an offset that is larger than the
284                  * total size of the buffers it provided.
285                  */
286                 return (CRYPTO_DATA_LEN_RANGE);
287         }

289         /*
290          * Now do the digesting on the iovecs.
291          */
292         while (vec_idx < uio->uio_iovcnt && length > 0) {
293                 cur_len = MIN(uio->uio_iov[vec_idx].iov_len - offset, length);
294                 SKEIN_OP(ctx, Update, (uint8_t *)uio->uio_iov[vec_idx].iov_base
295                     + offset, cur_len);
296                 length -= cur_len;
297                 vec_idx++;
298                 offset = 0;
299         }

301         if (vec_idx == uio->uio_iovcnt && length > 0) {
302                 /*
303                  * The end of the specified iovec's was reached but
304                  * the length requested could not be processed, i.e.
305                  * The caller requested to digest more data than it provided.
306                  */
307                 return (CRYPTO_DATA_LEN_RANGE);
308         }

310         return (CRYPTO_SUCCESS);
311 }

313 /*
314  * Performs a Final on a context and writes to a uio digest output.
315  */
316 static int
317 skein_digest_final_uio(skein_ctx_t *ctx, crypto_data_t *digest)
318 {
319         off_t   offset = digest->cd_offset;
320         uint_t  vec_idx;
321         uio_t   *uio = digest->cd_uio;

323         /* we support only kernel buffer */
324         if (uio->uio_segflg != UIO_SYSSPACE)
325                 return (CRYPTO_ARGUMENTS_BAD);
```



```
327         /*
328          * Jump to the first iovec containing ptr to the digest to be returned.
329          */
330         for (vec_idx = 0; offset >= uio->uio_iov[vec_idx].iov_len &&
331             vec_idx < uio->uio_iovcnt;
332             offset -= uio->uio_iov[vec_idx++].iov_len)
333                 ;
334         if (vec_idx == uio->uio_iovcnt) {
335                 /*
336                  * The caller specified an offset that is larger than the
337                  * total size of the buffers it provided.
338                  */
339                 return (CRYPTO_DATA_LEN_RANGE);
340         }
341         if (offset + CRYPTO_BITS2BYTES(ctx->sc_digest_bitlen) <=
342             uio->uio_iov[vec_idx].iov_len) {
343                 /* The computed digest will fit in the current iovec. */
344                 SKEIN_OP(ctx, Final,
345                     (uchar_t *)uio->uio_iov[vec_idx].iov_base + offset);
346         } else {
347                 uint8_t *digest_tmp;
348                 off_t scratch_offset = 0;
349                 size_t length = CRYPTO_BITS2BYTES(ctx->sc_digest_bitlen);
350                 size_t cur_len;

352                 digest_tmp = kmem_alloc(CRYPTO_BITS2BYTES(
353                     ctx->sc_digest_bitlen), KM_NOSLEEP);
354                 if (digest_tmp == NULL)
355                         return (CRYPTO_HOST_MEMORY);
356                 SKEIN_OP(ctx, Final, digest_tmp);
357                 while (vec_idx < uio->uio_iovcnt && length > 0) {
358                         cur_len = MIN(uio->uio_iov[vec_idx].iov_len - offset,
359                             length);
360                         bcopy(digest_tmp + scratch_offset,
361                             uio->uio_iov[vec_idx].iov_base + offset, cur_len);

363                         length -= cur_len;
364                         vec_idx++;
365                         scratch_offset += cur_len;
366                         offset = 0;
367                 }
368                 kmem_free(digest_tmp, CRYPTO_BITS2BYTES(ctx->sc_digest_bitlen));

370                 if (vec_idx == uio->uio_iovcnt && length > 0) {
371                         /*
372                          * The end of the specified iovec's was reached but
373                          * the length requested could not be processed, i.e.
374                          * The caller requested to digest more data than it
375                          * provided.
376                          */
377                         return (CRYPTO_DATA_LEN_RANGE);
378                 }
379         }

381         return (CRYPTO_SUCCESS);
382 }

384 /*
385  * Performs an Update on a context with mblk input data.
386  */
387 static int
388 skein_digest_update_mblk(skein_ctx_t *ctx, crypto_data_t *data)
389 {
390         off_t offset = data->cd_offset;
391         size_t length = data->cd_length;
```

```
392         mblk_t *mp;
393         size_t cur_len;

395         /* Jump to the first mblk_t containing data to be digested. */
396         for (mp = data->cd_mp; mp != NULL && offset >= MBLKL(mp);
397             offset -= MBLKL(mp), mp = mp->b_cont)
398                 ;
399         if (mp == NULL) {
400                 /*
401                  * The caller specified an offset that is larger than the
402                  * total size of the buffers it provided.
403                  */
404                 return (CRYPTO_DATA_LEN_RANGE);
405         }

407         /* Now do the digesting on the mblk chain. */
408         while (mp != NULL && length > 0) {
409                 cur_len = MIN(MBLKL(mp) - offset, length);
410                 SKEIN_OP(ctx, Update, mp->b_rptr + offset, cur_len);
411                 length -= cur_len;
412                 offset = 0;
413                 mp = mp->b_cont;
414         }

416         if (mp == NULL && length > 0) {
417                 /*
418                  * The end of the mblk was reached but the length requested
419                  * could not be processed, i.e. The caller requested
420                  * to digest more data than it provided.
421                  */
422                 return (CRYPTO_DATA_LEN_RANGE);
423         }

425         return (CRYPTO_SUCCESS);
426 }

428 /*
429  * Performs a Final on a context and writes to an mblk digest output.
430  */
431 static int
432 skein_digest_final_mblk(skein_ctx_t *ctx, crypto_data_t *digest)
433 {
434         off_t   offset = digest->cd_offset;
435         mblk_t  *mp;

437         /* Jump to the first mblk_t that will be used to store the digest. */
438         for (mp = digest->cd_mp; mp != NULL && offset >= MBLKL(mp);
439             offset -= MBLKL(mp), mp = mp->b_cont)
440                 ;
441         if (mp == NULL) {
442                 /* caller specified offset is too large */
443                 return (CRYPTO_DATA_LEN_RANGE);
444         }

446         if (offset + CRYPTO_BITS2BYTES(ctx->sc_digest_bitlen) <= MBLKL(mp)) {
447                 /* The digest will fit in the current mblk. */
448                 SKEIN_OP(ctx, Final, mp->b_rptr + offset);
449         } else {
450                 /* Split the digest up between the individual buffers. */
451                 uint8_t *digest_tmp;
452                 off_t scratch_offset = 0;
453                 size_t length = CRYPTO_BITS2BYTES(ctx->sc_digest_bitlen);
454                 size_t cur_len;

456                 digest_tmp = kmem_alloc(CRYPTO_BITS2BYTES(
457                     ctx->sc_digest_bitlen), KM_NOSLEEP);
```



```
458                 if (digest_tmp == NULL)
459                         return (CRYPTO_HOST_MEMORY);
460                 SKEIN_OP(ctx, Final, digest_tmp);
461                 while (mp != NULL && length > 0) {
462                         cur_len = MIN(MBLKL(mp) - offset, length);
463                         bcopy(digest_tmp + scratch_offset,
464                             mp->b_rptr + offset, cur_len);
465                         length -= cur_len;
466                         mp = mp->b_cont;
467                         scratch_offset += cur_len;
468                         offset = 0;
469                 }
470                 kmem_free(digest_tmp, CRYPTO_BITS2BYTES(ctx->sc_digest_bitlen));
471                 if (mp == NULL && length > 0) {
472                         /* digest too long to fit in the mblk buffers */
473                         return (CRYPTO_DATA_LEN_RANGE);
474                 }
475         }

477         return (CRYPTO_SUCCESS);
478 }

480 /*
481  * KCF software provider digest entry points.
482  */

484 /*
485  * Initializes a skein digest context to the configuration in 'mechanism'.
486  * The mechanism cm_type must be one of SKEIN_*_MECH_INFO_TYPE. The cm_param
487  * field may contain a skein_param_t structure indicating the length of the
488  * digest the algorithm should produce. Otherwise the default output lengths
489  * are applied (32 bytes for Skein-256, 64 bytes for Skein-512 and 128 bytes
490  * for Skein-1024).
491  */
492 static int
493 skein_digest_init(crypto_ctx_t *ctx, crypto_mechanism_t *mechanism,
494     crypto_req_handle_t req)
495 {
496         int     error = CRYPTO_SUCCESS;

498         if (!VALID_SKEIN_DIGEST_MECH(mechanism->cm_type))
499                 return (CRYPTO_MECHANISM_INVALID);

501         SKEIN_CTX_LVALUE(ctx) = kmem_alloc(sizeof (*SKEIN_CTX(ctx)),
502             crypto_kmflag(req));
503         if (SKEIN_CTX(ctx) == NULL)
504                 return (CRYPTO_HOST_MEMORY);

506         SKEIN_CTX(ctx)->sc_mech_type = mechanism->cm_type;
507         error = skein_get_digest_bitlen(mechanism,
508             &SKEIN_CTX(ctx)->sc_digest_bitlen);
509         if (error != CRYPTO_SUCCESS)
510                 goto errout;
511         SKEIN_OP(SKEIN_CTX(ctx), Init, SKEIN_CTX(ctx)->sc_digest_bitlen);

513         return (CRYPTO_SUCCESS);
514 errout:
515         bzero(SKEIN_CTX(ctx), sizeof (*SKEIN_CTX(ctx)));
516         kmem_free(SKEIN_CTX(ctx), sizeof (*SKEIN_CTX(ctx)));
517         SKEIN_CTX_LVALUE(ctx) = NULL;
518         return (error);
519 }

521 /*
522  * Executes a skein_update and skein_digest on a pre-initialized crypto
523  * context in a single step. See the documentation to these functions to
```

```
524  * see what to pass here.
525  */
526 static int
527 skein_digest(crypto_ctx_t *ctx, crypto_data_t *data, crypto_data_t *digest,
528     crypto_req_handle_t req)
529 {
530         int error = CRYPTO_SUCCESS;

532         ASSERT(SKEIN_CTX(ctx) != NULL);

534         if (digest->cd_length <
535             CRYPTO_BITS2BYTES(SKEIN_CTX(ctx)->sc_digest_bitlen)) {
536                 digest->cd_length =
537                     CRYPTO_BITS2BYTES(SKEIN_CTX(ctx)->sc_digest_bitlen);
538                 return (CRYPTO_BUFFER_TOO_SMALL);
539         }

541         error = skein_update(ctx, data, req);
542         if (error != CRYPTO_SUCCESS) {
543                 bzero(SKEIN_CTX(ctx), sizeof (*SKEIN_CTX(ctx)));
544                 kmem_free(SKEIN_CTX(ctx), sizeof (*SKEIN_CTX(ctx)));
545                 SKEIN_CTX_LVALUE(ctx) = NULL;
546                 digest->cd_length = 0;
547                 return (error);
548         }
549         error = skein_final(ctx, digest, req);

551         return (error);
552 }

554 /*
555  * Performs a skein Update with the input message in 'data' (successive calls
556  * can push more data). This is used both for digest and MAC operation.
557  * Supported input data formats are raw, uio and mblk.
558  */
559 /*ARGSUSED*/
560 static int
561 skein_update(crypto_ctx_t *ctx, crypto_data_t *data, crypto_req_handle_t req)
562 {
563         int error = CRYPTO_SUCCESS;

565         ASSERT(SKEIN_CTX(ctx) != NULL);

567         switch (data->cd_format) {
568         case CRYPTO_DATA_RAW:
569                 SKEIN_OP(SKEIN_CTX(ctx), Update,
570                     (uint8_t *)data->cd_raw.iov_base + data->cd_offset,
571                     data->cd_length);
572                 break;
573         case CRYPTO_DATA_UIO:
574                 error = skein_digest_update_uio(SKEIN_CTX(ctx), data);
575                 break;
576         case CRYPTO_DATA_MBLK:
577                 error = skein_digest_update_mblk(SKEIN_CTX(ctx), data);
578                 break;
579         default:
580                 error = CRYPTO_ARGUMENTS_BAD;
581         }

583         return (error);
584 }

586 /*
587  * Performs a skein Final, writing the output to 'digest'. This is used both
588  * for digest and MAC operation.
589  * Supported output digest formats are raw, uio and mblk.
```



```
590  */
591 /*ARGSUSED*/
592 static int
593 skein_final(crypto_ctx_t *ctx, crypto_data_t *digest, crypto_req_handle_t req)
594 {
595         int error = CRYPTO_SUCCESS;

597         ASSERT(SKEIN_CTX(ctx) != NULL);

599         if (digest->cd_length <
600             CRYPTO_BITS2BYTES(SKEIN_CTX(ctx)->sc_digest_bitlen)) {
601                 digest->cd_length =
602                     CRYPTO_BITS2BYTES(SKEIN_CTX(ctx)->sc_digest_bitlen);
603                 return (CRYPTO_BUFFER_TOO_SMALL);
604         }

606         switch (digest->cd_format) {
607         case CRYPTO_DATA_RAW:
608                 SKEIN_OP(SKEIN_CTX(ctx), Final,
609                     (uint8_t *)digest->cd_raw.iov_base + digest->cd_offset);
610                 break;
611         case CRYPTO_DATA_UIO:
612                 error = skein_digest_final_uio(SKEIN_CTX(ctx), digest);
613                 break;
614         case CRYPTO_DATA_MBLK:
615                 error = skein_digest_final_mblk(SKEIN_CTX(ctx), digest);
616                 break;
617         default:
618                 error = CRYPTO_ARGUMENTS_BAD;
619         }

621         if (error == CRYPTO_SUCCESS)
622                 digest->cd_length =
623                     CRYPTO_BITS2BYTES(SKEIN_CTX(ctx)->sc_digest_bitlen);
624         else
625                 digest->cd_length = 0;

627         bzero(SKEIN_CTX(ctx), sizeof (*SKEIN_CTX(ctx)));
628         kmem_free(SKEIN_CTX(ctx), sizeof (*(SKEIN_CTX(ctx))));
629         SKEIN_CTX_LVALUE(ctx) = NULL;

631         return (error);
632 }

634 /*
635  * Performs a full skein digest computation in a single call, configuring the
636  * algorithm according to 'mechanism', reading the input to be digested from
637  * 'data' and writing the output to 'digest'.
638  * Supported input/output formats are raw, uio and mblk.
639  */
640 /*ARGSUSED*/
641 static int
642 skein_digest_atomic(crypto_provider_handle_t provider,
643     crypto_session_id_t session_id, crypto_mechanism_t *mechanism,
644     crypto_data_t *data, crypto_data_t *digest, crypto_req_handle_t req)
645 {
646         int             error;
647         skein_ctx_t     skein_ctx;
648         crypto_ctx_t    ctx;
649         SKEIN_CTX_LVALUE(&ctx) = &skein_ctx;

651         /* Init */
652         if (!VALID_SKEIN_DIGEST_MECH(mechanism->cm_type))
653                 return (CRYPTO_MECHANISM_INVALID);
654         skein_ctx.sc_mech_type = mechanism->cm_type;
655         error = skein_get_digest_bitlen(mechanism, &skein_ctx.sc_digest_bitlen);
```

```
656         if (error != CRYPTO_SUCCESS)
657                 goto out;
658         SKEIN_OP(&skein_ctx, Init, skein_ctx.sc_digest_bitlen);
659
660         if ((error = skein_update(&ctx, data, digest)) != CRYPTO_SUCCESS)
661                 goto out;
662         if ((error = skein_final(&ctx, data, digest)) != CRYPTO_SUCCESS)
663                 goto out;
664
665 out:
666         if (error == CRYPTO_SUCCESS)
667                 digest->cd_length =
668                     CRYPTO_BITS2BYTES(skein_ctx.sc_digest_bitlen);
669         else
670                 digest->cd_length = 0;
671         bzero(&skein_ctx, sizeof (skein_ctx));
672
673         return (error);
674 }
675
676 /*
677  * Helper function that builds a Skein MAC context from the provided
678  * mechanism and key.
679  */
680 static int
681 skein_mac_ctx_build(skein_ctx_t *ctx, crypto_mechanism_t *mechanism,
682     crypto_key_t *key)
683 {
684         int error;
685
686         if (!VALID_SKEIN_MAC_MECH(mechanism->cm_type))
687                 return (CRYPTO_MECHANISM_INVALID);
688         if (key->ck_format != CRYPTO_KEY_RAW)
689                 return (CRYPTO_ARGUMENTS_BAD);
690         ctx->sc_mech_type = mechanism->cm_type;
691         error = skein_get_digest_bitlen(mechanism, &ctx->sc_digest_bitlen);
692         if (error != CRYPTO_SUCCESS)
693                 return (error);
694         SKEIN_OP(ctx, InitExt, ctx->sc_digest_bitlen, 0, key->ck_data,
695             CRYPTO_BITS2BYTES(key->ck_length));
696
697         return (CRYPTO_SUCCESS);
698 }
699
700 /*
701  * KCF software provide mac entry points.
702  */
703 /*
704  * Initializes a skein MAC context. You may pass a ctx_template, in which
705  * case the template will be reused to make initialization more efficient.
706  * Otherwise a new context will be constructed. The mechanism cm_type must
707  * be one of SKEIN_*_MAC_MECH_INFO_TYPE. Same as in skein_digest_init, you
708  * may pass a skein_param_t in cm_param to configure the length of the
709  * digest. The key must be in raw format.
710  */
711 static int
712 skein_mac_init(crypto_ctx_t *ctx, crypto_mechanism_t *mechanism,
713     crypto_key_t *key, crypto_spi_ctx_template_t ctx_template,
714     crypto_req_handle_t req)
715 {
716         int     error;
717
718         SKEIN_CTX_LVALUE(ctx) = kmem_alloc(sizeof (*SKEIN_CTX(ctx)),
719             crypto_kmflag(req));
720         if (SKEIN_CTX(ctx) == NULL)
721                 return (CRYPTO_HOST_MEMORY);
```



```
722
723         if (ctx_template != NULL) {
724                 bcopy(ctx_template, SKEIN_CTX(ctx),
725                     sizeof (*SKEIN_CTX(ctx)));
726         } else {
727                 error = skein_mac_ctx_build(SKEIN_CTX(ctx), mechanism, key);
728                 if (error != CRYPTO_SUCCESS)
729                         goto errout;
730         }
731
732         return (CRYPTO_SUCCESS);
733 errout:
734         bzero(SKEIN_CTX(ctx), sizeof (*SKEIN_CTX(ctx)));
735         kmem_free(SKEIN_CTX(ctx), sizeof (*SKEIN_CTX(ctx)));
736         return (error);
737 }
738
739 /*
740  * The MAC update and final calls are reused from the regular digest code.
741  */
742
743 /*ARGSUSED*/
744 /*
745  * Same as skein_digest_atomic, performs an atomic Skein MAC operation in
746  * one step. All the same properties apply to the arguments of this
747  * function as to those of the partial operations above.
748  */
749 static int
750 skein_mac_atomic(crypto_provider_handle_t provider,
751     crypto_session_id_t session_id, crypto_mechanism_t *mechanism,
752     crypto_key_t *key, crypto_data_t *data, crypto_data_t *mac,
753     crypto_spi_ctx_template_t ctx_template, crypto_req_handle_t req)
754 {
755         /* faux crypto context just for skein_digest_{update,final} */
756         int             error;
757         crypto_ctx_t    ctx;
758         skein_ctx_t     skein_ctx;
759         SKEIN_CTX_LVALUE(&ctx) = &skein_ctx;
760
761         if (ctx_template != NULL) {
762                 bcopy(ctx_template, &skein_ctx, sizeof (skein_ctx));
763         } else {
764                 error = skein_mac_ctx_build(&skein_ctx, mechanism, key);
765                 if (error != CRYPTO_SUCCESS)
766                         goto errout;
767         }
768
769         if ((error = skein_update(&ctx, data, req)) != CRYPTO_SUCCESS)
770                 goto errout;
771         if ((error = skein_final(&ctx, mac, req)) != CRYPTO_SUCCESS)
772                 goto errout;
773
774         return (CRYPTO_SUCCESS);
775 errout:
776         bzero(&skein_ctx, sizeof (skein_ctx));
777         return (error);
778 }
779
780 /*
781  * KCF software provider context management entry points.
782  */
783
784 /*
785  * Constructs a context template for the Skein MAC algorithm. The same
786  * properties apply to the arguments of this function as to those of
787  * skein_mac_init.
```

```
 */
/*ARGSUSED*/
static int
skein_create_ctx_template(crypto_provider_handle_t provider,
    crypto_mechanism_t *mechanism, crypto_key_t *key,
    crypto_spi_ctx_template_t *ctx_template, size_t *ctx_template_size,
    crypto_req_handle_t req)
{
        int             error;
        skein_ctx_t     *ctx_tmpl;

        ctx_tmpl = kmem_alloc(sizeof (*ctx_tmpl), crypto_kmflag(req));
        if (ctx_tmpl == NULL)
                return (CRYPTO_HOST_MEMORY);
        error = skein_mac_ctx_build(ctx_tmpl, mechanism, key);
        if (error != CRYPTO_SUCCESS)
                goto errout;
        *ctx_template = ctx_tmpl;
        *ctx_template_size = sizeof (*ctx_tmpl);

        return (CRYPTO_SUCCESS);
errout:
        bzero(ctx_tmpl, sizeof (*ctx_tmpl));
        kmem_free(ctx_tmpl, sizeof (*ctx_tmpl));
        return (error);
}

/*
 * Frees a skein context in a parent crypto context.
 */
static int
skein_free_context(crypto_ctx_t *ctx)
{
        if (SKEIN_CTX(ctx) != NULL) {
                bzero(SKEIN_CTX(ctx), sizeof (*SKEIN_CTX(ctx)));
                kmem_free(SKEIN_CTX(ctx), sizeof (*SKEIN_CTX(ctx)));
                SKEIN_CTX_LVALUE(ctx) = NULL;
        }

        return (CRYPTO_SUCCESS);
}
```

****
148091 Mon Oct 7 19:17:18 2013
new/usr/src/uts/common/fs/zfs/arc.c
4185 New hash algorithm support
****
_____***unchanged_portion_omitted_***

```
 967 #define ARC_MINTIME     (hz>>4) /* 62 ms */

 969 static void
 970 arc_cksum_verify(arc_buf_t *buf)
 971 {
 972         zio_cksum_t zc;

 974         if (!(zfs_flags & ZFS_DEBUG_MODIFY))
 975                 return;

 977         mutex_enter(&buf->b_hdr->b_freeze_lock);
 978         if (buf->b_hdr->b_freeze_cksum == NULL ||
 979             (buf->b_hdr->b_flags & ARC_IO_ERROR)) {
 980                 mutex_exit(&buf->b_hdr->b_freeze_lock);
 981                 return;
 982         }
 983         fletcher_2_native(buf->b_data, buf->b_hdr->b_size, NULL, NULL, &zc);
 983         fletcher_2_native(buf->b_data, buf->b_hdr->b_size, &zc);
 984         if (!ZIO_CHECKSUM_EQUAL(*buf->b_hdr->b_freeze_cksum, zc))
 985                 panic("buffer modified while frozen!");
 986         mutex_exit(&buf->b_hdr->b_freeze_lock);
 987 }

 989 static int
 990 arc_cksum_equal(arc_buf_t *buf)
 991 {
 992         zio_cksum_t zc;
 993         int equal;

 995         mutex_enter(&buf->b_hdr->b_freeze_lock);
 996         fletcher_2_native(buf->b_data, buf->b_hdr->b_size, NULL, NULL, &zc);
 996         fletcher_2_native(buf->b_data, buf->b_hdr->b_size, &zc);
 997         equal = ZIO_CHECKSUM_EQUAL(*buf->b_hdr->b_freeze_cksum, zc);
 998         mutex_exit(&buf->b_hdr->b_freeze_lock);

1000         return (equal);
1001 }

1003 static void
1004 arc_cksum_compute(arc_buf_t *buf, boolean_t force)
1005 {
1006         if (!force && !(zfs_flags & ZFS_DEBUG_MODIFY))
1007                 return;

1009         mutex_enter(&buf->b_hdr->b_freeze_lock);
1010         if (buf->b_hdr->b_freeze_cksum != NULL) {
1011                 mutex_exit(&buf->b_hdr->b_freeze_lock);
1012                 return;
1013         }
1014         buf->b_hdr->b_freeze_cksum = kmem_alloc(sizeof (zio_cksum_t), KM_SLEEP);
1015         fletcher_2_native(buf->b_data, buf->b_hdr->b_size, NULL, NULL,
1015         fletcher_2_native(buf->b_data, buf->b_hdr->b_size,
1016             buf->b_hdr->b_freeze_cksum);
1017         mutex_exit(&buf->b_hdr->b_freeze_lock);
1018         arc_buf_watch(buf);
1019 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3083 Mon Oct  7 19:17:19 2013
new/usr/src/uts/common/fs/zfs/edonr_zfs.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://opensource.org/licenses/CDDL-1.0.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2013 Saso Kiselkov.  All rights reserved.
 * Use is subject to license terms.
 */
#include <sys/zfs_context.h>
#include <sys/zio.h>
#include <sys/edonr.h>

#define EDONR_MODE              512
#define EDONR_BLOCK_SIZE        EdonR512_BLOCK_SIZE

/*
 * Native zio_checksum interface for the Edon-R hash function.
 */
/*ARGSUSED*/
void
zio_checksum_edonr_native(const void *buf, uint64_t size,
    const zio_cksum_salt_t *salt, const void *ctx_template, zio_cksum_t *zcp)
{
        uint8_t         digest[EDONR_MODE / 8];
        EdonRState      ctx;

        ASSERT(ctx_template != NULL);
        bcopy(ctx_template, &ctx, sizeof (ctx));
        EdonRUpdate(&ctx, buf, size * 8);
        EdonRFinal(&ctx, digest);
        bcopy(digest, zcp->zc_word, sizeof (zcp->zc_word));
}

/*
 * Byteswapped zio_checksum interface for the Edon-R hash function.
 */
void
zio_checksum_edonr_byteswap(const void *buf, uint64_t size,
    const zio_cksum_salt_t *salt, const void *ctx_template, zio_cksum_t *zcp)
{
        zio_cksum_t     tmp;

        zio_checksum_edonr_native(buf, size, salt, ctx_template, &tmp);
        zcp->zc_word[0] = BSWAP_64(zcp->zc_word[0]);
        zcp->zc_word[1] = BSWAP_64(zcp->zc_word[1]);
```



```
        zcp->zc_word[2] = BSWAP_64(zcp->zc_word[2]);
        zcp->zc_word[3] = BSWAP_64(zcp->zc_word[3]);
}

void *
zio_checksum_edonr_tmpl_init(const zio_cksum_salt_t *salt)
{
        EdonRState      *ctx;
        uint8_t         salt_block[EDONR_BLOCK_SIZE];

        /*
         * Edon-R needs all but the last hash invocation to be on full-size
         * blocks, but the salt is too small. Rather than simply padding it
         * with zeros, we expand the salt into a new salt block of proper
         * size by double-hashing it (the new salt block will be composed of
         * H(salt) || H(H(salt))).
         */
        CTASSERT(EDONR_BLOCK_SIZE == 2 * (EDONR_MODE / 8));
        EdonRHash(EDONR_MODE, salt->zcs_bytes, sizeof (salt->zcs_bytes) * 8,
            salt_block);
        EdonRHash(EDONR_MODE, salt_block, EDONR_MODE, salt_block +
            EDONR_MODE / 8);

        /*
         * Feed the new salt block into the hash function - this will serve
         * as our MAC key.
         */
        ctx = kmem_zalloc(sizeof (*ctx), KM_SLEEP);
        EdonRInit(ctx, EDONR_MODE);
        EdonRUpdate(ctx, salt_block, sizeof (salt_block) * 8);
        return (ctx);
}

void
zio_checksum_edonr_tmpl_free(void *ctx_template)
{
        EdonRState      *ctx = ctx_template;

        bzero(ctx, sizeof (*ctx));
        kmem_free(ctx, sizeof (*ctx));
}
```

```
**********************************************************
    2431 Mon Oct  7 19:17:19 2013
new/usr/src/uts/common/fs/zfs/sha256.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2013 Saso Kiselkov. All rights reserved.
 */
#include <sys/zfs_context.h>
#include <sys/zio.h>
#include <sys/sha2.h>

/*ARGSUSED*/
void
zio_checksum_SHA256(const void *buf, uint64_t size,
    const zio_cksum_salt_t *salt, const void *ctx_template, zio_cksum_t *zcp)
zio_checksum_SHA256(const void *buf, uint64_t size, zio_cksum_t *zcp)
{
        SHA2_CTX ctx;
        zio_cksum_t tmp;

        SHA2Init(SHA256, &ctx);
        SHA2Update(&ctx, buf, size);
        SHA2Final(&tmp, &ctx);

        /*
         * A prior implementation of this function had a
         * private SHA256 implementation always wrote things out in
         * Big Endian and there wasn't a byteswap variant of it.
         * To preseve on disk compatibility we need to force that
         * behaviour.
         */
        zcp->zc_word[0] = BE_64(tmp.zc_word[0]);
        zcp->zc_word[1] = BE_64(tmp.zc_word[1]);
        zcp->zc_word[2] = BE_64(tmp.zc_word[2]);
        zcp->zc_word[3] = BE_64(tmp.zc_word[3]);
}

/*ARGSUSED*/
void
zio_checksum_SHA512_native(const void *buf, uint64_t size,
    const zio_cksum_salt_t *salt, const void *ctx_template, zio_cksum_t *zcp)
```



```
{
        SHA2_CTX        ctx;

        SHA2Init(SHA512_256, &ctx);
        SHA2Update(&ctx, buf, size);
        SHA2Final(zcp, &ctx);
}

/*ARGSUSED*/
void
zio_checksum_SHA512_byteswap(const void *buf, uint64_t size,
    const zio_cksum_salt_t *salt, const void *ctx_template, zio_cksum_t *zcp)
{
        zio_cksum_t     tmp;

        zio_checksum_SHA512_native(buf, size, salt, ctx_template, &tmp);
        zcp->zc_word[0] = BSWAP_64(tmp.zc_word[0]);
        zcp->zc_word[1] = BSWAP_64(tmp.zc_word[1]);
        zcp->zc_word[2] = BSWAP_64(tmp.zc_word[2]);
        zcp->zc_word[3] = BSWAP_64(tmp.zc_word[3]);
}
______unchanged_portion_omitted_
```

```
**********************************************************
    2785 Mon Oct  7 19:17:19 2013
new/usr/src/uts/common/fs/zfs/skein_zfs.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://opensource.org/licenses/CDDL-1.0.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2013 Saso Kiselkov.  All rights reserved.
 */
#include <sys/zfs_context.h>
#include <sys/zio.h>
#include <sys/skein.h>

/*
 * Computes a native 256-bit skein MAC checksum. Please note that this
 * function requires the presence of a ctx_template that should be allocated
 * using zio_checksum_skein_tmpl_init.
 */
/*ARGSUSED*/
void
zio_checksum_skein_native(const void *buf, uint64_t size,
    const zio_cksum_salt_t *salt, const void *ctx_template, zio_cksum_t *zcp)
{
        Skein_512_Ctxt_t        ctx;

        ASSERT(ctx_template != NULL);
        bcopy(ctx_template, &ctx, sizeof (ctx));
        (void) Skein_512_Update(&ctx, buf, size);
        (void) Skein_512_Final(&ctx, (uint8_t *)zcp);
        bzero(&ctx, sizeof (ctx));
}

/*
 * Byteswapped version of zio_checksum_skein_native. This just invokes
 * the native checksum function and byteswaps the resulting checksum (since
 * skein is internally endian-insensitive).
 */
void
zio_checksum_skein_byteswap(const void *buf, uint64_t size,
    const zio_cksum_salt_t *salt, const void *ctx_template, zio_cksum_t *zcp)
{
        zio_cksum_t     tmp;

        zio_checksum_skein_native(buf, size, salt, ctx_template, &tmp);
        zcp->zc_word[0] = BSWAP_64(tmp.zc_word[0]);
        zcp->zc_word[1] = BSWAP_64(tmp.zc_word[1]);
        zcp->zc_word[2] = BSWAP_64(tmp.zc_word[2]);
```



```
        zcp->zc_word[3] = BSWAP_64(tmp.zc_word[3]);
}

/*
 * Allocates a skein MAC template suitable for using in skein MAC checksum
 * computations and returns a pointer to it.
 */
void *
zio_checksum_skein_tmpl_init(const zio_cksum_salt_t *salt)
{
        Skein_512_Ctxt_t        *ctx;

        ctx = kmem_zalloc(sizeof (*ctx), KM_SLEEP);
        (void) Skein_512_InitExt(ctx, sizeof (zio_cksum_t) * 8, 0,
            salt->zcs_bytes, sizeof (salt->zcs_bytes));
        return (ctx);
}

/*
 * Frees a skein context template previously allocated using
 * zio_checksum_skein_tmpl_init.
 */
void
zio_checksum_skein_tmpl_free(void *ctx_template)
{
        Skein_512_Ctxt_t        *ctx = ctx_template;

        bzero(ctx, sizeof (*ctx));
        kmem_free(ctx, sizeof (*ctx));
}
```

```
**********************************************************
  177031 Mon Oct  7 19:17:19 2013
new/usr/src/uts/common/fs/zfs/spa.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2013 by Delphix. All rights reserved.
 * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
 * Copyright 2013 Saso Kiselkov. All rights reserved.
 */

/*
 * SPA: Storage Pool Allocator
 *
 * This file contains all the routines used when modifying on-disk SPA state.
 * This includes opening, importing, destroying, exporting a pool, and syncing a
 * pool.
 */

#include <sys/zfs_context.h>
#include <sys/fm/fs/zfs.h>
#include <sys/spa_impl.h>
#include <sys/zio.h>
#include <sys/zio_checksum.h>
#include <sys/dmu.h>
#include <sys/dmu_tx.h>
#include <sys/zap.h>
#include <sys/zil.h>
#include <sys/ddt.h>
#include <sys/vdev_impl.h>
#include <sys/metaslab.h>
#include <sys/metaslab_impl.h>
#include <sys/uberblock_impl.h>
#include <sys/txg.h>
#include <sys/avl.h>
#include <sys/dmu_traverse.h>
#include <sys/dmu_objset.h>
#include <sys/unique.h>
#include <sys/dsl_pool.h>
#include <sys/dsl_dataset.h>
#include <sys/dsl_dir.h>
#include <sys/dsl_prop.h>
#include <sys/dsl_synctask.h>
#include <sys/fs/zfs.h>
```



```
#include <sys/arc.h>
#include <sys/callb.h>
#include <sys/systeminfo.h>
#include <sys/spa_boot.h>
#include <sys/zfs_ioctl.h>
#include <sys/dsl_scan.h>
#include <sys/zfeature.h>
#include <sys/dsl_destroy.h>

#ifdef  _KERNEL
#include <sys/bootprops.h>
#include <sys/callb.h>
#include <sys/cpupart.h>
#include <sys/pool.h>
#include <sys/sysdc.h>
#include <sys/zone.h>
#endif  /* _KERNEL */

#include "zfs_prop.h"
#include "zfs_comutil.h"

/*
 * The interval, in seconds, at which failed configuration cache file writes
 * should be retried.
 */
static int zfs_ccw_retry_interval = 300;

typedef enum zti_modes {
        ZTI_MODE_FIXED,                 /* value is # of threads (min 1) */
        ZTI_MODE_BATCH,                 /* cpu-intensive; value is ignored */
        ZTI_MODE_NULL,                  /* don't create a taskq */
        ZTI_NMODES
} zti_modes_t;
```

*unchanged_portion_omitted_*

```
/*
 * Load an existing storage pool, using the pool's builtin spa_config as a
 * source of configuration information.
 */
static int
spa_load_impl(spa_t *spa, uint64_t pool_guid, nvlist_t *config,
    spa_load_state_t state, spa_import_type_t type, boolean_t mosconfig,
    char **ereport)
{
        int error = 0;
        nvlist_t *nvroot = NULL;
        nvlist_t *label;
        vdev_t *rvd;
        uberblock_t *ub = &spa->spa_uberblock;
        uint64_t children, config_cache_txg = spa->spa_config_txg;
        int orig_mode = spa->spa_mode;
        int parse;
        uint64_t obj;
        boolean_t missing_feat_write = B_FALSE;

        /*
         * If this is an untrusted config, access the pool in read-only mode.
         * This prevents things like resilvering recently removed devices.
         */
        if (!mosconfig)
                spa->spa_mode = FREAD;

        ASSERT(MUTEX_HELD(&spa_namespace_lock));

        spa->spa_load_state = state;
```

```
2102         if (nvlist_lookup_nvlist(config, ZPOOL_CONFIG_VDEV_TREE, &nvroot))
2103                 return (SET_ERROR(EINVAL));

2105         parse = (type == SPA_IMPORT_EXISTING ?
2106             VDEV_ALLOC_LOAD : VDEV_ALLOC_SPLIT);

2108         /*
2109          * Create "The Godfather" zio to hold all async IOs
2110          */
2111         spa->spa_async_zio_root = zio_root(spa, NULL, NULL,
2112             ZIO_FLAG_CANFAIL | ZIO_FLAG_SPECULATIVE | ZIO_FLAG_GODFATHER);

2114         /*
2115          * Parse the configuration into a vdev tree.  We explicitly set the
2116          * value that will be returned by spa_version() since parsing the
2117          * configuration requires knowing the version number.
2118          */
2119         spa_config_enter(spa, SCL_ALL, FTAG, RW_WRITER);
2120         error = spa_config_parse(spa, &rvd, nvroot, NULL, 0, parse);
2121         spa_config_exit(spa, SCL_ALL, FTAG);

2123         if (error != 0)
2124                 return (error);

2126         ASSERT(spa->spa_root_vdev == rvd);

2128         if (type != SPA_IMPORT_ASSEMBLE) {
2129                 ASSERT(spa_guid(spa) == pool_guid);
2130         }

2132         /*
2133          * Try to open all vdevs, loading each label in the process.
2134          */
2135         spa_config_enter(spa, SCL_ALL, FTAG, RW_WRITER);
2136         error = vdev_open(rvd);
2137         spa_config_exit(spa, SCL_ALL, FTAG);
2138         if (error != 0)
2139                 return (error);

2141         /*
2142          * We need to validate the vdev labels against the configuration that
2143          * we have in hand, which is dependent on the setting of mosconfig. If
2144          * mosconfig is true then we're validating the vdev labels based on
2145          * that config.  Otherwise, we're validating against the cached config
2146          * (zpool.cache) that was read when we loaded the zfs module, and then
2147          * later we will recursively call spa_load() and validate against
2148          * the vdev config.
2149          *
2150          * If we're assembling a new pool that's been split off from an
2151          * existing pool, the labels haven't yet been updated so we skip
2152          * validation for now.
2153          */
2154         if (type != SPA_IMPORT_ASSEMBLE) {
2155                 spa_config_enter(spa, SCL_ALL, FTAG, RW_WRITER);
2156                 error = vdev_validate(rvd, mosconfig);
2157                 spa_config_exit(spa, SCL_ALL, FTAG);

2159                 if (error != 0)
2160                         return (error);

2162                 if (rvd->vdev_state <= VDEV_STATE_CANT_OPEN)
2163                         return (SET_ERROR(ENXIO));
2164         }

2166         /*
2167          * Find the best uberblock.
```



```
2168          */
2169         vdev_uberblock_load(rvd, ub, &label);

2171         /*
2172          * If we weren't able to find a single valid uberblock, return failure.
2173          */
2174         if (ub->ub_txg == 0) {
2175                 nvlist_free(label);
2176                 return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, ENXIO));
2177         }

2179         /*
2180          * If the pool has an unsupported version we can't open it.
2181          */
2182         if (!SPA_VERSION_IS_SUPPORTED(ub->ub_version)) {
2183                 nvlist_free(label);
2184                 return (spa_vdev_err(rvd, VDEV_AUX_VERSION_NEWER, ENOTSUP));
2185         }

2187         if (ub->ub_version >= SPA_VERSION_FEATURES) {
2188                 nvlist_t *features;

2190                 /*
2191                  * If we weren't able to find what's necessary for reading the
2192                  * MOS in the label, return failure.
2193                  */
2194                 if (label == NULL || nvlist_lookup_nvlist(label,
2195                     ZPOOL_CONFIG_FEATURES_FOR_READ, &features) != 0) {
2196                         nvlist_free(label);
2197                         return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA,
2198                             ENXIO));
2199                 }

2201                 /*
2202                  * Update our in-core representation with the definitive values
2203                  * from the label.
2204                  */
2205                 nvlist_free(spa->spa_label_features);
2206                 VERIFY(nvlist_dup(features, &spa->spa_label_features, 0) == 0);
2207         }

2209         nvlist_free(label);

2211         /*
2212          * Look through entries in the label nvlist's features_for_read. If
2213          * there is a feature listed there which we don't understand then we
2214          * cannot open a pool.
2215          */
2216         if (ub->ub_version >= SPA_VERSION_FEATURES) {
2217                 nvlist_t *unsup_feat;

2219                 VERIFY(nvlist_alloc(&unsup_feat, NV_UNIQUE_NAME, KM_SLEEP) ==
2220                     0);

2222                 for (nvpair_t *nvp = nvlist_next_nvpair(spa->spa_label_features,
2223                     NULL); nvp != NULL;
2224                     nvp = nvlist_next_nvpair(spa->spa_label_features, nvp)) {
2225                         if (!zfeature_is_supported(nvpair_name(nvp))) {
2226                                 VERIFY(nvlist_add_string(unsup_feat,
2227                                     nvpair_name(nvp), "") == 0);
2228                         }
2229                 }

2231                 if (!nvlist_empty(unsup_feat)) {
2232                         VERIFY(nvlist_add_nvlist(spa->spa_load_info,
2233                             ZPOOL_CONFIG_UNSUP_FEAT, unsup_feat) == 0);
```

```
2234                        nvlist_free(unsup_feat);
2235                        return (spa_vdev_err(rvd, VDEV_AUX_UNSUP_FEAT,
2236                            ENOTSUP));
2237                }

2239                nvlist_free(unsup_feat);
2240        }

2242        /*
2243         * If the vdev guid sum doesn't match the uberblock, we have an
2244         * incomplete configuration.  We first check to see if the pool
2245         * is aware of the complete config (i.e ZPOOL_CONFIG_VDEV_CHILDREN).
2246         * If it is, defer the vdev_guid_sum check till later so we
2247         * can handle missing vdevs.
2248         */
2249        if (nvlist_lookup_uint64(config, ZPOOL_CONFIG_VDEV_CHILDREN,
2250            &children) != 0 && mosconfig && type != SPA_IMPORT_ASSEMBLE &&
2251            rvd->vdev_guid_sum != ub->ub_guid_sum)
2252                return (spa_vdev_err(rvd, VDEV_AUX_BAD_GUID_SUM, ENXIO));

2254        if (type != SPA_IMPORT_ASSEMBLE && spa->spa_config_splitting) {
2255                spa_config_enter(spa, SCL_ALL, FTAG, RW_WRITER);
2256                spa_try_repair(spa, config);
2257                spa_config_exit(spa, SCL_ALL, FTAG);
2258                nvlist_free(spa->spa_config_splitting);
2259                spa->spa_config_splitting = NULL;
2260        }

2262        /*
2263         * Initialize internal SPA structures.
2264         */
2265        spa->spa_state = POOL_STATE_ACTIVE;
2266        spa->spa_ubsync = spa->spa_uberblock;
2267        spa->spa_verify_min_txg = spa->spa_extreme_rewind ?
2268            TXG_INITIAL - 1 : spa_last_synced_txg(spa) - TXG_DEFER_SIZE - 1;
2269        spa->spa_first_txg = spa->spa_last_ubsync_txg ?
2270            spa->spa_last_ubsync_txg : spa_last_synced_txg(spa) + 1;
2271        spa->spa_claim_max_txg = spa->spa_first_txg;
2272        spa->spa_prev_software_version = ub->ub_software_version;

2274        error = dsl_pool_init(spa, spa->spa_first_txg, &spa->spa_dsl_pool);
2275        if (error)
2276                return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));
2277        spa->spa_meta_objset = spa->spa_dsl_pool->dp_meta_objset;

2279        if (spa_dir_prop(spa, DMU_POOL_CONFIG, &spa->spa_config_object) != 0)
2280                return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));

2282        if (spa_version(spa) >= SPA_VERSION_FEATURES) {
2283                boolean_t missing_feat_read = B_FALSE;
2284                nvlist_t *unsup_feat, *enabled_feat;

2286                if (spa_dir_prop(spa, DMU_POOL_FEATURES_FOR_READ,
2287                    &spa->spa_feat_for_read_obj) != 0) {
2288                        return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));
2289                }

2291                if (spa_dir_prop(spa, DMU_POOL_FEATURES_FOR_WRITE,
2292                    &spa->spa_feat_for_write_obj) != 0) {
2293                        return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));
2294                }

2296                if (spa_dir_prop(spa, DMU_POOL_FEATURE_DESCRIPTIONS,
2297                    &spa->spa_feat_desc_obj) != 0) {
2298                        return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));
2299                }
```



```
2301                enabled_feat = fnvlist_alloc();
2302                unsup_feat = fnvlist_alloc();

2304                if (!feature_is_supported(spa->spa_meta_objset,
2305                    spa->spa_feat_for_read_obj, spa->spa_feat_desc_obj,
2306                    unsup_feat, enabled_feat))
2307                        missing_feat_read = B_TRUE;

2309                if (spa_writeable(spa) || state == SPA_LOAD_TRYIMPORT) {
2310                        if (!feature_is_supported(spa->spa_meta_objset,
2311                            spa->spa_feat_for_write_obj, spa->spa_feat_desc_obj,
2312                            unsup_feat, enabled_feat)) {
2313                                missing_feat_write = B_TRUE;
2314                        }
2315                }

2317                fnvlist_add_nvlist(spa->spa_load_info,
2318                    ZPOOL_CONFIG_ENABLED_FEAT, enabled_feat);

2320                if (!nvlist_empty(unsup_feat)) {
2321                        fnvlist_add_nvlist(spa->spa_load_info,
2322                            ZPOOL_CONFIG_UNSUP_FEAT, unsup_feat);
2323                }

2325                fnvlist_free(enabled_feat);
2326                fnvlist_free(unsup_feat);

2328                if (!missing_feat_read) {
2329                        fnvlist_add_boolean(spa->spa_load_info,
2330                            ZPOOL_CONFIG_CAN_RDONLY);
2331                }

2333                /*
2334                 * If the state is SPA_LOAD_TRYIMPORT, our objective is
2335                 * twofold: to determine whether the pool is available for
2336                 * import in read-write mode and (if it is not) whether the
2337                 * pool is available for import in read-only mode. If the pool
2338                 * is available for import in read-write mode, it is displayed
2339                 * as available in userland; if it is not available for import
2340                 * in read-only mode, it is displayed as unavailable in
2341                 * userland. If the pool is available for import in read-only
2342                 * mode but not read-write mode, it is displayed as unavailable
2343                 * in userland with a special note that the pool is actually
2344                 * available for open in read-only mode.
2345                 *
2346                 * As a result, if the state is SPA_LOAD_TRYIMPORT and we are
2347                 * missing a feature for write, we must first determine whether
2348                 * the pool can be opened read-only before returning to
2349                 * userland in order to know whether to display the
2350                 * abovementioned note.
2351                 */
2352                if (missing_feat_read || (missing_feat_write &&
2353                    spa_writeable(spa))) {
2354                        return (spa_vdev_err(rvd, VDEV_AUX_UNSUP_FEAT,
2355                            ENOTSUP));
2356                }
2357        }

2359        spa->spa_is_initializing = B_TRUE;
2360        error = dsl_pool_open(spa->spa_dsl_pool);
2361        spa->spa_is_initializing = B_FALSE;
2362        if (error != 0)
2363                return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));

2365        if (!mosconfig) {
```

```
2366                uint64_t hostid;
2367                nvlist_t *policy = NULL, *nvconfig;

2369                if (load_nvlist(spa, spa->spa_config_object, &nvconfig) != 0)
2370                        return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));

2372                if (!spa_is_root(spa) && nvlist_lookup_uint64(nvconfig,
2373                    ZPOOL_CONFIG_HOSTID, &hostid) == 0) {
2374                        char *hostname;
2375                        unsigned long myhostid = 0;

2377                        VERIFY(nvlist_lookup_string(nvconfig,
2378                            ZPOOL_CONFIG_HOSTNAME, &hostname) == 0);

2380 #ifdef  _KERNEL
2381                        myhostid = zone_get_hostid(NULL);
2382 #else   /* _KERNEL */
2383                        /*
2384                         * We're emulating the system's hostid in userland, so
2385                         * we can't use zone_get_hostid().
2386                         */
2387                        (void) ddi_strtoul(hw_serial, NULL, 10, &myhostid);
2388 #endif  /* _KERNEL */
2389                        if (hostid != 0 && myhostid != 0 &&
2390                            hostid != myhostid) {
2391                                nvlist_free(nvconfig);
2392                                cmn_err(CE_WARN, "pool '%s' could not be "
2393                                    "loaded as it was last accessed by "
2394                                    "another system (host: %s hostid: 0x%lx). "
2395                                    "See: http://illumos.org/msg/ZFS-8000-EY",
2396                                    spa_name(spa), hostname,
2397                                    (unsigned long)hostid);
2398                                return (SET_ERROR(EBADF));
2399                        }
2400                }
2401                if (nvlist_lookup_nvlist(spa->spa_config,
2402                    ZPOOL_REWIND_POLICY, &policy) == 0)
2403                        VERIFY(nvlist_add_nvlist(nvconfig,
2404                            ZPOOL_REWIND_POLICY, policy) == 0);

2406                spa_config_set(spa, nvconfig);
2407                spa_unload(spa);
2408                spa_deactivate(spa);
2409                spa_activate(spa, orig_mode);

2411                return (spa_load(spa, state, SPA_IMPORT_EXISTING, B_TRUE));
2412        }

2414        /* Grab the secret checksum salt from the MOS. */
2415        if (spa_dir_prop(spa, DMU_POOL_CHECKSUM_SALT,
2416            &spa->spa_cksum_salt_obj) == 0) {
2417                if (zap_lookup(spa->spa_meta_objset, spa->spa_cksum_salt_obj,
2418                    DMU_POOL_CHECKSUM_SALT, 1,
2419                    sizeof (spa->spa_cksum_salt.zcs_bytes),
2420                    spa->spa_cksum_salt.zcs_bytes) != 0) {
2421                        /*
2422                         * MOS format is broken, the salt object is there but
2423                         * is missing the actual salt value.
2424                         */
2425                        return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));
2426                }
2427        } else {
2428                /* Generate a new salt for subsequent use */
2429                (void) random_get_pseudo_bytes(spa->spa_cksum_salt.zcs_bytes,
2430                    sizeof (spa->spa_cksum_salt.zcs_bytes));
2431        }
```



```
2433        if (spa_dir_prop(spa, DMU_POOL_SYNC_BPOBJ, &obj) != 0)
2434                return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));
2435        error = bpobj_open(&spa->spa_deferred_bpobj, spa->spa_meta_objset, obj);
2436        if (error != 0)
2437                return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));

2439        /*
2440         * Load the bit that tells us to use the new accounting function
2441         * (raid-z deflation).  If we have an older pool, this will not
2442         * be present.
2443         */
2444        error = spa_dir_prop(spa, DMU_POOL_DEFLATE, &spa->spa_deflate);
2445        if (error != 0 && error != ENOENT)
2446                return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));

2448        error = spa_dir_prop(spa, DMU_POOL_CREATION_VERSION,
2449            &spa->spa_creation_version);
2450        if (error != 0 && error != ENOENT)
2451                return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));

2453        /*
2454         * Load the persistent error log.  If we have an older pool, this will
2455         * not be present.
2456         */
2457        error = spa_dir_prop(spa, DMU_POOL_ERRLOG_LAST, &spa->spa_errlog_last);
2458        if (error != 0 && error != ENOENT)
2459                return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));

2461        error = spa_dir_prop(spa, DMU_POOL_ERRLOG_SCRUB,
2462            &spa->spa_errlog_scrub);
2463        if (error != 0 && error != ENOENT)
2464                return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));

2466        /*
2467         * Load the history object.  If we have an older pool, this
2468         * will not be present.
2469         */
2470        error = spa_dir_prop(spa, DMU_POOL_HISTORY, &spa->spa_history);
2471        if (error != 0 && error != ENOENT)
2472                return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));

2474        /*
2475         * If we're assembling the pool from the split-off vdevs of
2476         * an existing pool, we don't want to attach the spares & cache
2477         * devices.
2478         */

2480        /*
2481         * Load any hot spares for this pool.
2482         */
2483        error = spa_dir_prop(spa, DMU_POOL_SPARES, &spa->spa_spares.sav_object);
2484        if (error != 0 && error != ENOENT)
2485                return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));
2486        if (error == 0 && type != SPA_IMPORT_ASSEMBLE) {
2487                ASSERT(spa_version(spa) >= SPA_VERSION_SPARES);
2488                if (load_nvlist(spa, spa->spa_spares.sav_object,
2489                    &spa->spa_spares.sav_config) != 0)
2490                        return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));

2492                spa_config_enter(spa, SCL_ALL, FTAG, RW_WRITER);
2493                spa_load_spares(spa);
2494                spa_config_exit(spa, SCL_ALL, FTAG);
2495        } else if (error == 0) {
2496                spa->spa_spares.sav_sync = B_TRUE;
2497        }
```

```
2499         /*
2500          * Load any level 2 ARC devices for this pool.
2501          */
2502         error = spa_dir_prop(spa, DMU_POOL_L2CACHE,
2503             &spa->spa_l2cache.sav_object);
2504         if (error != 0 && error != ENOENT)
2505                 return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));
2506         if (error == 0 && type != SPA_IMPORT_ASSEMBLE) {
2507                 ASSERT(spa_version(spa) >= SPA_VERSION_L2CACHE);
2508                 if (load_nvlist(spa, spa->spa_l2cache.sav_object,
2509                     &spa->spa_l2cache.sav_config) != 0)
2510                         return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));

2512                 spa_config_enter(spa, SCL_ALL, FTAG, RW_WRITER);
2513                 spa_load_l2cache(spa);
2514                 spa_config_exit(spa, SCL_ALL, FTAG);
2515         } else if (error == 0) {
2516                 spa->spa_l2cache.sav_sync = B_TRUE;
2517         }

2519         spa->spa_delegation = zpool_prop_default_numeric(ZPOOL_PROP_DELEGATION);

2521         error = spa_dir_prop(spa, DMU_POOL_PROPS, &spa->spa_pool_props_object);
2522         if (error && error != ENOENT)
2523                 return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));

2525         if (error == 0) {
2526                 uint64_t autoreplace;

2528                 spa_prop_find(spa, ZPOOL_PROP_BOOTFS, &spa->spa_bootfs);
2529                 spa_prop_find(spa, ZPOOL_PROP_AUTOREPLACE, &autoreplace);
2530                 spa_prop_find(spa, ZPOOL_PROP_DELEGATION, &spa->spa_delegation);
2531                 spa_prop_find(spa, ZPOOL_PROP_FAILUREMODE, &spa->spa_failmode);
2532                 spa_prop_find(spa, ZPOOL_PROP_AUTOEXPAND, &spa->spa_autoexpand);
2533                 spa_prop_find(spa, ZPOOL_PROP_DEDUPDITTO,
2534                     &spa->spa_dedup_ditto);

2536                 spa->spa_autoreplace = (autoreplace != 0);
2537         }

2539         /*
2540          * If the 'autoreplace' property is set, then post a resource notifying
2541          * the ZFS DE that it should not issue any faults for unopenable
2542          * devices.  We also iterate over the vdevs, and post a sysevent for any
2543          * unopenable vdevs so that the normal autoreplace handler can take
2544          * over.
2545          */
2546         if (spa->spa_autoreplace && state != SPA_LOAD_TRYIMPORT) {
2547                 spa_check_removed(spa->spa_root_vdev);
2548                 /*
2549                  * For the import case, this is done in spa_import(), because
2550                  * at this point we're using the spare definitions from
2551                  * the MOS config, not necessarily from the userland config.
2552                  */
2553                 if (state != SPA_LOAD_IMPORT) {
2554                         spa_aux_check_removed(&spa->spa_spares);
2555                         spa_aux_check_removed(&spa->spa_l2cache);
2556                 }
2557         }

2559         /*
2560          * Load the vdev state for all toplevel vdevs.
2561          */
2562         vdev_load(rvd);
```



```
2564         /*
2565          * Propagate the leaf DTLs we just loaded all the way up the tree.
2566          */
2567         spa_config_enter(spa, SCL_ALL, FTAG, RW_WRITER);
2568         vdev_dtl_reassess(rvd, 0, 0, B_FALSE);
2569         spa_config_exit(spa, SCL_ALL, FTAG);

2571         /*
2572          * Load the DDTs (dedup tables).
2573          */
2574         error = ddt_load(spa);
2575         if (error != 0)
2576                 return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));

2578         spa_update_dspace(spa);

2580         /*
2581          * Validate the config, using the MOS config to fill in any
2582          * information which might be missing.  If we fail to validate
2583          * the config then declare the pool unfit for use. If we're
2584          * assembling a pool from a split, the log is not transferred
2585          * over.
2586          */
2587         if (type != SPA_IMPORT_ASSEMBLE) {
2588                 nvlist_t *nvconfig;

2590                 if (load_nvlist(spa, spa->spa_config_object, &nvconfig) != 0)
2591                         return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA, EIO));

2593                 if (!spa_config_valid(spa, nvconfig)) {
2594                         nvlist_free(nvconfig);
2595                         return (spa_vdev_err(rvd, VDEV_AUX_BAD_GUID_SUM,
2596                             ENXIO));
2597                 }
2598                 nvlist_free(nvconfig);

2600                 /*
2601                  * Now that we've validated the config, check the state of the
2602                  * root vdev.  If it can't be opened, it indicates one or
2603                  * more toplevel vdevs are faulted.
2604                  */
2605                 if (rvd->vdev_state <= VDEV_STATE_CANT_OPEN)
2606                         return (SET_ERROR(ENXIO));

2608                 if (spa_check_logs(spa)) {
2609                         *ereport = FM_EREPORT_ZFS_LOG_REPLAY;
2610                         return (spa_vdev_err(rvd, VDEV_AUX_BAD_LOG, ENXIO));
2611                 }
2612         }

2614         if (missing_feat_write) {
2615                 ASSERT(state == SPA_LOAD_TRYIMPORT);

2617                 /*
2618                  * At this point, we know that we can open the pool in
2619                  * read-only mode but not read-write mode. We now have enough
2620                  * information and can return to userland.
2621                  */
2622                 return (spa_vdev_err(rvd, VDEV_AUX_UNSUP_FEAT, ENOTSUP));
2623         }

2625         /*
2626          * We've successfully opened the pool, verify that we're ready
2627          * to start pushing transactions.
2628          */
2629         if (state != SPA_LOAD_TRYIMPORT) {
```

```
2630                 if (error = spa_load_verify(spa))
2631                         return (spa_vdev_err(rvd, VDEV_AUX_CORRUPT_DATA,
2632                             error));
2633         }

2635         if (spa_writeable(spa) && (state == SPA_LOAD_RECOVER ||
2636             spa->spa_load_max_txg == UINT64_MAX)) {
2637                 dmu_tx_t *tx;
2638                 int need_update = B_FALSE;

2640                 ASSERT(state != SPA_LOAD_TRYIMPORT);

2642                 /*
2643                  * Claim log blocks that haven't been committed yet.
2644                  * This must all happen in a single txg.
2645                  * Note: spa_claim_max_txg is updated by spa_claim_notify(),
2646                  * invoked from zil_claim_log_block()'s i/o done callback.
2647                  * Price of rollback is that we abandon the log.
2648                  */
2649                 spa->spa_claiming = B_TRUE;

2651                 tx = dmu_tx_create_assigned(spa_get_dsl(spa),
2652                     spa_first_txg(spa));
2653                 (void) dmu_objset_find(spa_name(spa),
2654                     zil_claim, tx, DS_FIND_CHILDREN);
2655                 dmu_tx_commit(tx);

2657                 spa->spa_claiming = B_FALSE;

2659                 spa_set_log_state(spa, SPA_LOG_GOOD);
2660                 spa->spa_sync_on = B_TRUE;
2661                 txg_sync_start(spa->spa_dsl_pool);

2663                 /*
2664                  * Wait for all claims to sync.  We sync up to the highest
2665                  * claimed log block birth time so that claimed log blocks
2666                  * don't appear to be from the future.  spa_claim_max_txg
2667                  * will have been set for us by either zil_check_log_chain()
2668                  * (invoked from spa_check_logs()) or zil_claim() above.
2669                  */
2670                 txg_wait_synced(spa->spa_dsl_pool, spa->spa_claim_max_txg);

2672                 /*
2673                  * If the config cache is stale, or we have uninitialized
2674                  * metaslabs (see spa_vdev_add()), then update the config.
2675                  *
2676                  * If this is a verbatim import, trust the current
2677                  * in-core spa_config and update the disk labels.
2678                  */
2679                 if (config_cache_txg != spa->spa_config_txg ||
2680                     state == SPA_LOAD_IMPORT ||
2681                     state == SPA_LOAD_RECOVER ||
2682                     (spa->spa_import_flags & ZFS_IMPORT_VERBATIM))
2683                         need_update = B_TRUE;

2685                 for (int c = 0; c < rvd->vdev_children; c++)
2686                         if (rvd->vdev_child[c]->vdev_ms_array == 0)
2687                                 need_update = B_TRUE;

2689                 /*
2690                  * Update the config cache asychronously in case we're the
2691                  * root pool, in which case the config cache isn't writable yet.
2692                  */
2693                 if (need_update)
2694                         spa_async_request(spa, SPA_ASYNC_CONFIG_UPDATE);
```



```
2696                 /*
2697                  * Check all DTLs to see if anything needs resilvering.
2698                  */
2699                 if (!dsl_scan_resilvering(spa->spa_dsl_pool) &&
2700                     vdev_resilver_needed(rvd, NULL, NULL))
2701                         spa_async_request(spa, SPA_ASYNC_RESILVER);

2703                 /*
2704                  * Log the fact that we booted up (so that we can detect if
2705                  * we rebooted in the middle of an operation).
2706                  */
2707                 spa_history_log_version(spa, "open");

2709                 /*
2710                  * Delete any inconsistent datasets.
2711                  */
2712                 (void) dmu_objset_find(spa_name(spa),
2713                     dsl_destroy_inconsistent, NULL, DS_FIND_CHILDREN);

2715                 /*
2716                  * Clean up any stale temporary dataset userrefs.
2717                  */
2718                 dsl_pool_clean_tmp_userrefs(spa->spa_dsl_pool);
2719         }

2721         return (0);
2722 }
______unchanged_portion_omitted_

3380 /*
3381  * Pool Creation
3382  */
3383 int
3384 spa_create(const char *pool, nvlist_t *nvroot, nvlist_t *props,
3385     nvlist_t *zplprops)
3386 {
3387         spa_t *spa;
3388         char *altroot = NULL;
3389         vdev_t *rvd;
3390         dsl_pool_t *dp;
3391         dmu_tx_t *tx;
3392         int error = 0;
3393         uint64_t txg = TXG_INITIAL;
3394         nvlist_t **spares, **l2cache;
3395         uint_t nspares, nl2cache;
3396         uint64_t version, obj;
3397         boolean_t has_features;

3399         /*
3400          * If this pool already exists, return failure.
3401          */
3402         mutex_enter(&spa_namespace_lock);
3403         if (spa_lookup(pool) != NULL) {
3404                 mutex_exit(&spa_namespace_lock);
3405                 return (SET_ERROR(EEXIST));
3406         }

3408         /*
3409          * Allocate a new spa_t structure.
3410          */
3411         (void) nvlist_lookup_string(props,
3412             zpool_prop_to_name(ZPOOL_PROP_ALTROOT), &altroot);
3413         spa = spa_add(pool, NULL, altroot);
3414         spa_activate(spa, spa_mode_global);

3416         if (props && (error = spa_prop_validate(spa, props))) {
```

```
3417                 spa_deactivate(spa);
3418                 spa_remove(spa);
3419                 mutex_exit(&spa_namespace_lock);
3420                 return (error);
3421         }

3423         has_features = B_FALSE;
3424         for (nvpair_t *elem = nvlist_next_nvpair(props, NULL);
3425             elem != NULL; elem = nvlist_next_nvpair(props, elem)) {
3426                 if (zpool_prop_feature(nvpair_name(elem)))
3427                         has_features = B_TRUE;
3428         }

3430         if (has_features || nvlist_lookup_uint64(props,
3431             zpool_prop_to_name(ZPOOL_PROP_VERSION), &version) != 0) {
3432                 version = SPA_VERSION;
3433         }
3434         ASSERT(SPA_VERSION_IS_SUPPORTED(version));

3436         spa->spa_first_txg = txg;
3437         spa->spa_uberblock.ub_txg = txg - 1;
3438         spa->spa_uberblock.ub_version = version;
3439         spa->spa_ubsync = spa->spa_uberblock;

3441         /*
3442          * Create "The Godfather" zio to hold all async IOs
3443          */
3444         spa->spa_async_zio_root = zio_root(spa, NULL, NULL,
3445             ZIO_FLAG_CANFAIL | ZIO_FLAG_SPECULATIVE | ZIO_FLAG_GODFATHER);

3447         /*
3448          * Create the root vdev.
3449          */
3450         spa_config_enter(spa, SCL_ALL, FTAG, RW_WRITER);

3452         error = spa_config_parse(spa, &rvd, nvroot, NULL, 0, VDEV_ALLOC_ADD);

3454         ASSERT(error != 0 || rvd != NULL);
3455         ASSERT(error != 0 || spa->spa_root_vdev == rvd);

3457         if (error == 0 && !zfs_allocatable_devs(nvroot))
3458                 error = SET_ERROR(EINVAL);

3460         if (error == 0 &&
3461             (error = vdev_create(rvd, txg, B_FALSE)) == 0 &&
3462             (error = spa_validate_aux(spa, nvroot, txg,
3463             VDEV_ALLOC_ADD)) == 0) {
3464                 for (int c = 0; c < rvd->vdev_children; c++) {
3465                         vdev_metaslab_set_size(rvd->vdev_child[c]);
3466                         vdev_expand(rvd->vdev_child[c], txg);
3467                 }
3468         }

3470         spa_config_exit(spa, SCL_ALL, FTAG);

3472         if (error != 0) {
3473                 spa_unload(spa);
3474                 spa_deactivate(spa);
3475                 spa_remove(spa);
3476                 mutex_exit(&spa_namespace_lock);
3477                 return (error);
3478         }

3480         /*
3481          * Get the list of spares, if specified.
3482          */
```



```
3483         if (nvlist_lookup_nvlist_array(nvroot, ZPOOL_CONFIG_SPARES,
3484             &spares, &nspares) == 0) {
3485                 VERIFY(nvlist_alloc(&spa->spa_spares.sav_config, NV_UNIQUE_NAME,
3486                     KM_SLEEP) == 0);
3487                 VERIFY(nvlist_add_nvlist_array(spa->spa_spares.sav_config,
3488                     ZPOOL_CONFIG_SPARES, spares, nspares) == 0);
3489                 spa_config_enter(spa, SCL_ALL, FTAG, RW_WRITER);
3490                 spa_load_spares(spa);
3491                 spa_config_exit(spa, SCL_ALL, FTAG);
3492                 spa->spa_spares.sav_sync = B_TRUE;
3493         }

3495         /*
3496          * Get the list of level 2 cache devices, if specified.
3497          */
3498         if (nvlist_lookup_nvlist_array(nvroot, ZPOOL_CONFIG_L2CACHE,
3499             &l2cache, &nl2cache) == 0) {
3500                 VERIFY(nvlist_alloc(&spa->spa_l2cache.sav_config,
3501                     NV_UNIQUE_NAME, KM_SLEEP) == 0);
3502                 VERIFY(nvlist_add_nvlist_array(spa->spa_l2cache.sav_config,
3503                     ZPOOL_CONFIG_L2CACHE, l2cache, nl2cache) == 0);
3504                 spa_config_enter(spa, SCL_ALL, FTAG, RW_WRITER);
3505                 spa_load_l2cache(spa);
3506                 spa_config_exit(spa, SCL_ALL, FTAG);
3507                 spa->spa_l2cache.sav_sync = B_TRUE;
3508         }

3510         spa->spa_is_initializing = B_TRUE;
3511         spa->spa_dsl_pool = dp = dsl_pool_create(spa, zplprops, txg);
3512         spa->spa_meta_objset = dp->dp_meta_objset;
3513         spa->spa_is_initializing = B_FALSE;

3515         /*
3516          * Create DDTs (dedup tables).
3517          */
3518         ddt_create(spa);

3520         spa_update_dspace(spa);

3522         tx = dmu_tx_create_assigned(dp, txg);

3524         /*
3525          * Create the pool config object.
3526          */
3527         spa->spa_config_object = dmu_object_alloc(spa->spa_meta_objset,
3528             DMU_OT_PACKED_NVLIST, SPA_CONFIG_BLOCKSIZE,
3529             DMU_OT_PACKED_NVLIST_SIZE, sizeof (uint64_t), tx);

3531         if (zap_add(spa->spa_meta_objset,
3532             DMU_POOL_DIRECTORY_OBJECT, DMU_POOL_CONFIG,
3533             sizeof (uint64_t), 1, &spa->spa_config_object, tx) != 0) {
3534                 cmn_err(CE_PANIC, "failed to add pool config");
3535         }

3537         if (spa_version(spa) >= SPA_VERSION_FEATURES)
3538                 spa_feature_create_zap_objects(spa, tx);

3540         if (zap_add(spa->spa_meta_objset,
3541             DMU_POOL_DIRECTORY_OBJECT, DMU_POOL_CREATION_VERSION,
3542             sizeof (uint64_t), 1, &version, tx) != 0) {
3543                 cmn_err(CE_PANIC, "failed to add pool version");
3544         }

3546         /* Newly created pools with the right version are always deflated. */
3547         if (version >= SPA_VERSION_RAIDZ_DEFLATE) {
3548                 spa->spa_deflate = TRUE;
```

```
3549                 if (zap_add(spa->spa_meta_objset,
3550                     DMU_POOL_DIRECTORY_OBJECT, DMU_POOL_DEFLATE,
3551                     sizeof (uint64_t), 1, &spa->spa_deflate, tx) != 0) {
3552                         cmn_err(CE_PANIC, "failed to add deflate");
3553                 }
3554         }

3556         /*
3557          * Create the deferred-free bpobj.  Turn off compression
3558          * because sync-to-convergence takes longer if the blocksize
3559          * keeps changing.
3560          */
3561         obj = bpobj_alloc(spa->spa_meta_objset, 1 << 14, tx);
3562         dmu_object_set_compress(spa->spa_meta_objset, obj,
3563             ZIO_COMPRESS_OFF, tx);
3564         if (zap_add(spa->spa_meta_objset,
3565             DMU_POOL_DIRECTORY_OBJECT, DMU_POOL_SYNC_BPOBJ,
3566             sizeof (uint64_t), 1, &obj, tx) != 0) {
3567                 cmn_err(CE_PANIC, "failed to add bpobj");
3568         }
3569         VERIFY3U(0, ==, bpobj_open(&spa->spa_deferred_bpobj,
3570             spa->spa_meta_objset, obj));

3572         /*
3573          * Create the pool's history object.
3574          */
3575         if (version >= SPA_VERSION_ZPOOL_HISTORY)
3576                 spa_history_create_obj(spa, tx);

3578         /*
3579          * Generate some random noise for salted checksums to operate on. As
3580          * soon as a salted checksum is used for the first time we will
3581          * generate the persistent MOS object to hold the salt (see
3582          * spa_activate_salted_cksum).
3583          */
3584         (void) random_get_pseudo_bytes(spa->spa_cksum_salt.zcs_bytes,
3585             sizeof (spa->spa_cksum_salt.zcs_bytes));

3587         /*
3588          * Set pool properties.
3589          */
3590         spa->spa_bootfs = zpool_prop_default_numeric(ZPOOL_PROP_BOOTFS);
3591         spa->spa_delegation = zpool_prop_default_numeric(ZPOOL_PROP_DELEGATION);
3592         spa->spa_failmode = zpool_prop_default_numeric(ZPOOL_PROP_FAILUREMODE);
3593         spa->spa_autoexpand = zpool_prop_default_numeric(ZPOOL_PROP_AUTOEXPAND);

3595         if (props != NULL) {
3596                 spa_configfile_set(spa, props, B_FALSE);
3597                 spa_sync_props(props, tx);
3598         }

3600         dmu_tx_commit(tx);

3602         spa->spa_sync_on = B_TRUE;
3603         txg_sync_start(spa->spa_dsl_pool);

3605         /*
3606          * We explicitly wait for the first transaction to complete so that our
3607          * bean counters are appropriately updated.
3608          */
3609         txg_wait_synced(spa->spa_dsl_pool, txg);

3611         spa_config_sync(spa, B_FALSE, B_TRUE);

3613         spa_history_log_version(spa, "create");
```



```
3615         spa->spa_minref = refcount_count(&spa->spa_refcount);

3617         mutex_exit(&spa_namespace_lock);

3619         return (0);
3620 }
```

*_____unchanged_portion_omitted_*

```
**********************************************************
   48069 Mon Oct  7 19:17:20 2013
new/usr/src/uts/common/fs/zfs/spa_misc.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2013 by Delphix. All rights reserved.
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 * Copyright 2013 Saso Kiselkov. All rights reserved.
 */

#include <sys/zfs_context.h>
#include <sys/spa_impl.h>
#include <sys/spa_boot.h>
#include <sys/zio.h>
#include <sys/zio_checksum.h>
#include <sys/zio_compress.h>
#include <sys/dmu.h>
#include <sys/dmu_tx.h>
#include <sys/zap.h>
#include <sys/zil.h>
#include <sys/vdev_impl.h>
#include <sys/metaslab.h>
#include <sys/uberblock_impl.h>
#include <sys/txg.h>
#include <sys/avl.h>
#include <sys/unique.h>
#include <sys/dsl_pool.h>
#include <sys/dsl_dir.h>
#include <sys/dsl_prop.h>
#include <sys/dsl_scan.h>
#include <sys/fs/zfs.h>
#include <sys/metaslab_impl.h>
#include <sys/arc.h>
#include <sys/ddt.h>
#include "zfs_prop.h"
#include <sys/zfeature.h>
#include "zfeature_common.h"

/*
 * SPA locking
 *
 * There are four basic locks for managing spa_t structures:
 *
 * spa_namespace_lock (global mutex)
```



```
 *
 *      This lock must be acquired to do any of the following:
 *
 *              - Lookup a spa_t by name
 *              - Add or remove a spa_t from the namespace
 *              - Increase spa_refcount from non-zero
 *              - Check if spa_refcount is zero
 *              - Rename a spa_t
 *              - add/remove/attach/detach devices
 *              - Held for the duration of create/destroy/import/export
 *
 *      It does not need to handle recursion.  A create or destroy may
 *      reference objects (files or zvols) in other pools, but by
 *      definition they must have an existing reference, and will never need
 *      to lookup a spa_t by name.
 *
 * spa_refcount (per-spa refcount_t protected by mutex)
 *
 *      This reference count keep track of any active users of the spa_t.  The
 *      spa_t cannot be destroyed or freed while this is non-zero.  Internally,
 *      the refcount is never really 'zero' - opening a pool implicitly keeps
 *      some references in the DMU.  Internally we check against spa_minref, but
 *      present the image of a zero/non-zero value to consumers.
 *
 * spa_config_lock[] (per-spa array of rwlocks)
 *
 *      This protects the spa_t from config changes, and must be held in
 *      the following circumstances:
 *
 *              - RW_READER to perform I/O to the spa
 *              - RW_WRITER to change the vdev config
 *
 * The locking order is fairly straightforward:
 *
 *              spa_namespace_lock      ->      spa_refcount
 *
 *      The namespace lock must be acquired to increase the refcount from 0
 *      or to check if it is zero.
 *
 *              spa_refcount            ->      spa_config_lock[]
 *
 *      There must be at least one valid reference on the spa_t to acquire
 *      the config lock.
 *
 *              spa_namespace_lock      ->      spa_config_lock[]
 *
 *      The namespace lock must always be taken before the config lock.
 *
 *
 * The spa_namespace_lock can be acquired directly and is globally visible.
 *
 * The namespace is manipulated using the following functions, all of which
 * require the spa_namespace_lock to be held.
 *
 *      spa_lookup()            Lookup a spa_t by name.
 *
 *      spa_add()               Create a new spa_t in the namespace.
 *
 *      spa_remove()            Remove a spa_t from the namespace.  This also
 *                              frees up any memory associated with the spa_t.
 *
 *      spa_next()              Returns the next spa_t in the system, or the
 *                              first if NULL is passed.
 *
 *      spa_evict_all()         Shutdown and remove all spa_t structures in
 *                              the system.
```

```
 127  *
 128  *      spa_guid_exists()       Determine whether a pool/device guid exists.
 129  *
 130  * The spa_refcount is manipulated using the following functions:
 131  *
 132  *      spa_open_ref()          Adds a reference to the given spa_t.  Must be
 133  *                              called with spa_namespace_lock held if the
 134  *                              refcount is currently zero.
 135  *
 136  *      spa_close()             Remove a reference from the spa_t.  This will
 137  *                              not free the spa_t or remove it from the
 138  *                              namespace.  No locking is required.
 139  *
 140  *      spa_refcount_zero()     Returns true if the refcount is currently
 141  *                              zero.  Must be called with spa_namespace_lock
 142  *                              held.
 143  *
 144  * The spa_config_lock[] is an array of rwlocks, ordered as follows:
 145  * SCL_CONFIG > SCL_STATE > SCL_ALLOC > SCL_ZIO > SCL_FREE > SCL_VDEV.
 146  * spa_config_lock[] is manipulated with spa_config_{enter,exit,held}().
 147  *
 148  * To read the configuration, it suffices to hold one of these locks as reader.
 149  * To modify the configuration, you must hold all locks as writer.  To modify
 150  * vdev state without altering the vdev tree's topology (e.g. online/offline),
 151  * you must hold SCL_STATE and SCL_ZIO as writer.
 152  *
 153  * We use these distinct config locks to avoid recursive lock entry.
 154  * For example, spa_sync() (which holds SCL_CONFIG as reader) induces
 155  * block allocations (SCL_ALLOC), which may require reading space maps
 156  * from disk (dmu_read() -> zio_read() -> SCL_ZIO).
 157  *
 158  * The spa config locks cannot be normal rwlocks because we need the
 159  * ability to hand off ownership.  For example, SCL_ZIO is acquired
 160  * by the issuing thread and later released by an interrupt thread.
 161  * They do, however, obey the usual write-wanted semantics to prevent
 162  * writer (i.e. system administrator) starvation.
 163  *
 164  * The lock acquisition rules are as follows:
 165  *
 166  * SCL_CONFIG
 167  *      Protects changes to the vdev tree topology, such as vdev
 168  *      add/remove/attach/detach.  Protects the dirty config list
 169  *      (spa_config_dirty_list) and the set of spares and l2arc devices.
 170  *
 171  * SCL_STATE
 172  *      Protects changes to pool state and vdev state, such as vdev
 173  *      online/offline/fault/degrade/clear.  Protects the dirty state list
 174  *      (spa_state_dirty_list) and global pool state (spa_state).
 175  *
 176  * SCL_ALLOC
 177  *      Protects changes to metaslab groups and classes.
 178  *      Held as reader by metaslab_alloc() and metaslab_claim().
 179  *
 180  * SCL_ZIO
 181  *      Held by bp-level zios (those which have no io_vd upon entry)
 182  *      to prevent changes to the vdev tree.  The bp-level zio implicitly
 183  *      protects all of its vdev child zios, which do not hold SCL_ZIO.
 184  *
 185  * SCL_FREE
 186  *      Protects changes to metaslab groups and classes.
 187  *      Held as reader by metaslab_free().  SCL_FREE is distinct from
 188  *      SCL_ALLOC, and lower than SCL_ZIO, so that we can safely free
 189  *      blocks in zio_done() while another i/o that holds either
 190  *      SCL_ALLOC or SCL_ZIO is waiting for this i/o to complete.
 191  *
 192  * SCL_VDEV
```



```
 193  *      Held as reader to prevent changes to the vdev tree during trivial
 194  *      inquiries such as bp_get_dsize().  SCL_VDEV is distinct from the
 195  *      other locks, and lower than all of them, to ensure that it's safe
 196  *      to acquire regardless of caller context.
 197  *
 198  * In addition, the following rules apply:
 199  *
 200  * (a)  spa_props_lock protects pool properties, spa_config and spa_config_list.
 201  *      The lock ordering is SCL_CONFIG > spa_props_lock.
 202  *
 203  * (b)  I/O operations on leaf vdevs.  For any zio operation that takes
 204  *      an explicit vdev_t argument -- such as zio_ioctl(), zio_read_phys(),
 205  *      or zio_write_phys() -- the caller must ensure that the config cannot
 206  *      cannot change in the interim, and that the vdev cannot be reopened.
 207  *      SCL_STATE as reader suffices for both.
 208  *
 209  * The vdev configuration is protected by spa_vdev_enter() / spa_vdev_exit().
 210  *
 211  *      spa_vdev_enter()        Acquire the namespace lock and the config lock
 212  *                              for writing.
 213  *
 214  *      spa_vdev_exit()         Release the config lock, wait for all I/O
 215  *                              to complete, sync the updated configs to the
 216  *                              cache, and release the namespace lock.
 217  *
 218  * vdev state is protected by spa_vdev_state_enter() / spa_vdev_state_exit().
 219  * Like spa_vdev_enter/exit, these are convenience wrappers -- the actual
 220  * locking is, always, based on spa_namespace_lock and spa_config_lock[].
 221  *
 222  * spa_rename() is also implemented within this file since it requires
 223  * manipulation of the namespace.
 224  */

 226 static avl_tree_t spa_namespace_avl;
 227 kmutex_t spa_namespace_lock;
 228 static kcondvar_t spa_namespace_cv;
 229 static int spa_active_count;
 230 int spa_max_replication_override = SPA_DVAS_PER_BP;

 232 static kmutex_t spa_spare_lock;
 233 static avl_tree_t spa_spare_avl;
 234 static kmutex_t spa_l2cache_lock;
 235 static avl_tree_t spa_l2cache_avl;

 237 kmem_cache_t *spa_buffer_pool;
 238 int spa_mode_global;

 240 #ifdef ZFS_DEBUG
 241 /* Everything except dprintf and spa is on by default in debug builds */
 242 int zfs_flags = ~(ZFS_DEBUG_DPRINTF | ZFS_DEBUG_SPA);
 243 #else
 244 int zfs_flags = 0;
 245 #endif

 247 /*
 248  * zfs_recover can be set to nonzero to attempt to recover from
 249  * otherwise-fatal errors, typically caused by on-disk corruption.  When
 250  * set, calls to zfs_panic_recover() will turn into warning messages.
 251  */
 252 int zfs_recover = 0;

 254 /*
 255  * Expiration time in milliseconds. This value has two meanings. First it is
 256  * used to determine when the spa_deadman() logic should fire. By default the
 257  * spa_deadman() will fire if spa_sync() has not completed in 1000 seconds.
 258  * Secondly, the value determines if an I/O is considered "hung". Any I/O that
```

```
 259  * has not completed in zfs_deadman_synctime_ms is considered "hung" resulting
 260  * in a system panic.
 261  */
 262 uint64_t zfs_deadman_synctime_ms = 1000000ULL;

 264 /*
 265  * Check time in milliseconds. This defines the frequency at which we check
 266  * for hung I/O.
 267  */
 268 uint64_t zfs_deadman_checktime_ms = 5000ULL;

 270 /*
 271  * Override the zfs deadman behavior via /etc/system. By default the
 272  * deadman is enabled except on VMware and sparc deployments.
 273  */
 274 int zfs_deadman_enabled = -1;

 276 /*
 277  * The worst case is single-sector max-parity RAID-Z blocks, in which
 278  * case the space requirement is exactly (VDEV_RAIDZ_MAXPARITY + 1)
 279  * times the size; so just assume that.  Add to this the fact that
 280  * we can have up to 3 DVAs per bp, and one more factor of 2 because
 281  * the block may be dittoed with up to 3 DVAs by ddt_sync().  All together,
 282  * the worst case is:
 283  *     (VDEV_RAIDZ_MAXPARITY + 1) * SPA_DVAS_PER_BP * 2 == 24
 284  */
 285 int spa_asize_inflation = 24;

 287 /*
 288  * ==========================================================================
 289  * SPA config locking
 290  * ==========================================================================
 291  */
 292 static void
 293 spa_config_lock_init(spa_t *spa)
 294 {
 295         for (int i = 0; i < SCL_LOCKS; i++) {
 296                 spa_config_lock_t *scl = &spa->spa_config_lock[i];
 297                 mutex_init(&scl->scl_lock, NULL, MUTEX_DEFAULT, NULL);
 298                 cv_init(&scl->scl_cv, NULL, CV_DEFAULT, NULL);
 299                 refcount_create_untracked(&scl->scl_count);
 300                 scl->scl_writer = NULL;
 301                 scl->scl_write_wanted = 0;
 302         }
 303 }
_____unchanged_portion_omitted_

 476 /*
 477  * Create an uninitialized spa_t with the given name.  Requires
 478  * spa_namespace_lock.  The caller must ensure that the spa_t doesn't already
 479  * exist by calling spa_lookup() first.
 480  */
 481 spa_t *
 482 spa_add(const char *name, nvlist_t *config, const char *altroot)
 483 {
 484         spa_t *spa;
 485         spa_config_dirent_t *dp;
 486         cyc_handler_t hdlr;
 487         cyc_time_t when;

 489         ASSERT(MUTEX_HELD(&spa_namespace_lock));

 491         spa = kmem_zalloc(sizeof (spa_t), KM_SLEEP);

 493         mutex_init(&spa->spa_async_lock, NULL, MUTEX_DEFAULT, NULL);
 494         mutex_init(&spa->spa_errlist_lock, NULL, MUTEX_DEFAULT, NULL);
```



```
 495         mutex_init(&spa->spa_errlog_lock, NULL, MUTEX_DEFAULT, NULL);
 496         mutex_init(&spa->spa_history_lock, NULL, MUTEX_DEFAULT, NULL);
 497         mutex_init(&spa->spa_proc_lock, NULL, MUTEX_DEFAULT, NULL);
 498         mutex_init(&spa->spa_props_lock, NULL, MUTEX_DEFAULT, NULL);
 499         mutex_init(&spa->spa_cksum_tmpls_lock, NULL, MUTEX_DEFAULT, NULL);
 500         mutex_init(&spa->spa_scrub_lock, NULL, MUTEX_DEFAULT, NULL);
 501         mutex_init(&spa->spa_suspend_lock, NULL, MUTEX_DEFAULT, NULL);
 502         mutex_init(&spa->spa_vdev_top_lock, NULL, MUTEX_DEFAULT, NULL);
 503         mutex_init(&spa->spa_iokstat_lock, NULL, MUTEX_DEFAULT, NULL);

 505         cv_init(&spa->spa_async_cv, NULL, CV_DEFAULT, NULL);
 506         cv_init(&spa->spa_proc_cv, NULL, CV_DEFAULT, NULL);
 507         cv_init(&spa->spa_scrub_io_cv, NULL, CV_DEFAULT, NULL);
 508         cv_init(&spa->spa_suspend_cv, NULL, CV_DEFAULT, NULL);

 510         for (int t = 0; t < TXG_SIZE; t++)
 511                 bplist_create(&spa->spa_free_bplist[t]);

 513         (void) strlcpy(spa->spa_name, name, sizeof (spa->spa_name));
 514         spa->spa_state = POOL_STATE_UNINITIALIZED;
 515         spa->spa_freeze_txg = UINT64_MAX;
 516         spa->spa_final_txg = UINT64_MAX;
 517         spa->spa_load_max_txg = UINT64_MAX;
 518         spa->spa_proc = &p0;
 519         spa->spa_proc_state = SPA_PROC_NONE;

 521         hdlr.cyh_func = spa_deadman;
 522         hdlr.cyh_arg = spa;
 523         hdlr.cyh_level = CY_LOW_LEVEL;

 525         spa->spa_deadman_synctime = MSEC2NSEC(zfs_deadman_synctime_ms);

 527         /*
 528          * This determines how often we need to check for hung I/Os after
 529          * the cyclic has already fired. Since checking for hung I/Os is
 530          * an expensive operation we don't want to check too frequently.
 531          * Instead wait for 5 seconds before checking again.
 532          */
 533         when.cyt_interval = MSEC2NSEC(zfs_deadman_checktime_ms);
 534         when.cyt_when = CY_INFINITY;
 535         mutex_enter(&cpu_lock);
 536         spa->spa_deadman_cycid = cyclic_add(&hdlr, &when);
 537         mutex_exit(&cpu_lock);

 539         refcount_create(&spa->spa_refcount);
 540         spa_config_lock_init(spa);

 542         avl_add(&spa_namespace_avl, spa);

 544         /*
 545          * Set the alternate root, if there is one.
 546          */
 547         if (altroot) {
 548                 spa->spa_root = spa_strdup(altroot);
 549                 spa_active_count++;
 550         }

 552         /*
 553          * Every pool starts with the default cachefile
 554          */
 555         list_create(&spa->spa_config_list, sizeof (spa_config_dirent_t),
 556             offsetof(spa_config_dirent_t, scd_link));

 558         dp = kmem_zalloc(sizeof (spa_config_dirent_t), KM_SLEEP);
 559         dp->scd_path = altroot ? NULL : spa_strdup(spa_config_path);
 560         list_insert_head(&spa->spa_config_list, dp);
```

```
562         VERIFY(nvlist_alloc(&spa->spa_load_info, NV_UNIQUE_NAME,
563             KM_SLEEP) == 0);
565         if (config != NULL) {
566                 nvlist_t *features;
568                 if (nvlist_lookup_nvlist(config, ZPOOL_CONFIG_FEATURES_FOR_READ,
569                     &features) == 0) {
570                         VERIFY(nvlist_dup(features, &spa->spa_label_features,
571                             0) == 0);
572                 }
574                 VERIFY(nvlist_dup(config, &spa->spa_config, 0) == 0);
575         }
577         if (spa->spa_label_features == NULL) {
578                 VERIFY(nvlist_alloc(&spa->spa_label_features, NV_UNIQUE_NAME,
579                     KM_SLEEP) == 0);
580         }
582         spa->spa_iokstat = kstat_create("zfs", 0, name,
583             "disk", KSTAT_TYPE_IO, 1, 0);
584         if (spa->spa_iokstat) {
585                 spa->spa_iokstat->ks_lock = &spa->spa_iokstat_lock;
586                 kstat_install(spa->spa_iokstat);
587         }
589         spa->spa_debug = ((zfs_flags & ZFS_DEBUG_SPA) != 0);
591         return (spa);
592 }
594 /*
595  * Removes a spa_t from the namespace, freeing up any memory used.  Requires
596  * spa_namespace_lock.  This is called only after the spa_t has been closed and
597  * deactivated.
598  */
599 void
600 spa_remove(spa_t *spa)
601 {
602         spa_config_dirent_t *dp;
604         ASSERT(MUTEX_HELD(&spa_namespace_lock));
605         ASSERT(spa->spa_state == POOL_STATE_UNINITIALIZED);
607         nvlist_free(spa->spa_config_splitting);
609         avl_remove(&spa_namespace_avl, spa);
610         cv_broadcast(&spa_namespace_cv);
612         if (spa->spa_root) {
613                 spa_strfree(spa->spa_root);
614                 spa_active_count--;
615         }
617         while ((dp = list_head(&spa->spa_config_list)) != NULL) {
618                 list_remove(&spa->spa_config_list, dp);
619                 if (dp->scd_path != NULL)
620                         spa_strfree(dp->scd_path);
621                 kmem_free(dp, sizeof (spa_config_dirent_t));
622         }
624         list_destroy(&spa->spa_config_list);
626         nvlist_free(spa->spa_label_features);
```



```
627         nvlist_free(spa->spa_load_info);
628         spa_config_set(spa, NULL);
630         mutex_enter(&cpu_lock);
631         if (spa->spa_deadman_cycid != CYCLIC_NONE)
632                 cyclic_remove(spa->spa_deadman_cycid);
633         mutex_exit(&cpu_lock);
634         spa->spa_deadman_cycid = CYCLIC_NONE;
636         refcount_destroy(&spa->spa_refcount);
638         spa_config_lock_destroy(spa);
640         kstat_delete(spa->spa_iokstat);
641         spa->spa_iokstat = NULL;
643         for (int t = 0; t < TXG_SIZE; t++)
644                 bplist_destroy(&spa->spa_free_bplist[t]);
646         zio_checksum_templates_free(spa);
648         cv_destroy(&spa->spa_async_cv);
649         cv_destroy(&spa->spa_proc_cv);
650         cv_destroy(&spa->spa_scrub_io_cv);
651         cv_destroy(&spa->spa_suspend_cv);
653         mutex_destroy(&spa->spa_async_lock);
654         mutex_destroy(&spa->spa_errlist_lock);
655         mutex_destroy(&spa->spa_errlog_lock);
656         mutex_destroy(&spa->spa_history_lock);
657         mutex_destroy(&spa->spa_proc_lock);
658         mutex_destroy(&spa->spa_props_lock);
659         mutex_destroy(&spa->spa_cksum_tmpls_lock);
660         mutex_destroy(&spa->spa_scrub_lock);
661         mutex_destroy(&spa->spa_suspend_lock);
662         mutex_destroy(&spa->spa_vdev_top_lock);
663         mutex_destroy(&spa->spa_iokstat_lock);
665         kmem_free(spa, sizeof (spa_t));
666 }
_____unchanged_portion_omitted_
1790 static int
1791 activate_salted_cksum_check(zfeature_info_t *feature, dmu_tx_t *tx)
1792 {
1793         spa_t   *spa = dmu_tx_pool(tx)->dp_spa;
1795         if (!spa_feature_is_active(spa, feature))
1796                 return (0);
1797         else
1798                 return (SET_ERROR(EBUSY));
1799 }
1801 static void
1802 activate_salted_cksum_sync(zfeature_info_t *feature, dmu_tx_t *tx)
1803 {
1804         spa_t   *spa = dmu_tx_pool(tx)->dp_spa;
1806         spa_feature_incr(spa, feature, tx);
1807         /*
1808          * This is the first salted checksum that's been activated, so
1809          * create the persistent checksum salt object now.
1810          */
1811         if (spa->spa_cksum_salt_obj == 0) {
1812                 spa->spa_cksum_salt_obj = zap_create_link(spa->spa_meta_objset,
1813                     DMU_OTN_ZAP_METADATA, DMU_POOL_DIRECTORY_OBJECT,
```

```
1814                     DMU_POOL_CHECKSUM_SALT, tx);
1815                 VERIFY3U(zap_add(spa->spa_meta_objset,
1816                     spa->spa_cksum_salt_obj, DMU_POOL_CHECKSUM_SALT, 1,
1817                     sizeof (spa->spa_cksum_salt.zcs_bytes),
1818                     spa->spa_cksum_salt.zcs_bytes, tx), ==, 0);
1819         }
1820 }

1822 /*
1823  * Activates a feature associated with a salted checksum. You must call this
1824  * function instead of calling spa_feature_incr() directly, because we may
1825  * also need to sync the MOS object holding the checksum salt.
1826  * Arguments:
1827  *      spa     Pool on which to activate the salted checksum feature.
1828  *      feature Salted checksum algorithm feature to activate (see
1829  *              spa_feature_table).
1830  */
1831 int
1832 spa_activate_salted_cksum(spa_t *spa, struct zfeature_info *feature)
1833 {
1834         int err;

1836         /* EBUSY here indicates that the feature is already active */
1837         err = dsl_sync_task(spa_name(spa),
1838             (dsl_checkfunc_t *)activate_salted_cksum_check,
1839             (dsl_syncfunc_t *)activate_salted_cksum_sync, feature, 2);

1841         if (err != 0 && err != EBUSY)
1842                 return (err);
1843         else
1844                 return (0);
1845 }

1847 int
1848 spa_mode(spa_t *spa)
1849 {
1850         return (spa->spa_mode);
1851 }
```

*unchanged_portion_omitted_*

```
**********************************************************
   29012 Mon Oct  7 19:17:20 2013
new/usr/src/uts/common/fs/zfs/sys/dmu.h
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2013 by Delphix. All rights reserved.
 * Copyright 2011 Nexenta Systems, Inc. All rights reserved.
 * Copyright (c) 2012, Joyent, Inc. All rights reserved.
 * Copyright 2013 Saso Kiselkov. All rights reserved.
 */

/* Portions Copyright 2010 Robert Milkowski */

#ifndef _SYS_DMU_H
#define _SYS_DMU_H

/*
 * This file describes the interface that the DMU provides for its
 * consumers.
 *
 * The DMU also interacts with the SPA.  That interface is described in
 * dmu_spa.h.
 */

#include <sys/inttypes.h>
#include <sys/types.h>
#include <sys/param.h>
#include <sys/cred.h>
#include <sys/time.h>
#include <sys/fs/zfs.h>

#ifdef	__cplusplus
extern "C" {
#endif

struct uio;
struct xuio;
struct page;
struct vnode;
struct spa;
struct zilog;
struct zio;
struct blkptr;
```



```
struct zap_cursor;
struct dsl_dataset;
struct dsl_pool;
struct dnode;
struct drr_begin;
struct drr_end;
struct zbookmark;
struct spa;
struct nvlist;
struct arc_buf;
struct zio_prop;
struct sa_handle;

typedef struct objset objset_t;
typedef struct dmu_tx dmu_tx_t;
typedef struct dsl_dir dsl_dir_t;

typedef enum dmu_object_byteswap {
	DMU_BSWAP_UINT8,
	DMU_BSWAP_UINT16,
	DMU_BSWAP_UINT32,
	DMU_BSWAP_UINT64,
	DMU_BSWAP_ZAP,
	DMU_BSWAP_DNODE,
	DMU_BSWAP_OBJSET,
	DMU_BSWAP_ZNODE,
	DMU_BSWAP_OLDACL,
	DMU_BSWAP_ACL,
	/*
	 * Allocating a new byteswap type number makes the on-disk format
	 * incompatible with any other format that uses the same number.
	 *
	 * Data can usually be structured to work with one of the
	 * DMU_BSWAP_UINT* or DMU_BSWAP_ZAP types.
	 */
	DMU_BSWAP_NUMFUNCS
} dmu_object_byteswap_t;
_____unchanged_portion_omitted_

typedef void dmu_buf_evict_func_t(struct dmu_buf *db, void *user_ptr);

/*
 * The names of zap entries in the DIRECTORY_OBJECT of the MOS.
 */
#define DMU_POOL_DIRECTORY_OBJECT       1
#define DMU_POOL_CONFIG                 "config"
#define DMU_POOL_FEATURES_FOR_WRITE     "features_for_write"
#define DMU_POOL_FEATURES_FOR_READ      "features_for_read"
#define DMU_POOL_FEATURE_DESCRIPTIONS   "feature_descriptions"
#define DMU_POOL_ROOT_DATASET           "root_dataset"
#define DMU_POOL_SYNC_BPOBJ             "sync_bplist"
#define DMU_POOL_ERRLOG_SCRUB           "errlog_scrub"
#define DMU_POOL_ERRLOG_LAST            "errlog_last"
#define DMU_POOL_SPARES                 "spares"
#define DMU_POOL_DEFLATE                "deflate"
#define DMU_POOL_HISTORY                "history"
#define DMU_POOL_PROPS                  "pool_props"
#define DMU_POOL_L2CACHE                "l2cache"
#define DMU_POOL_TMP_USERREFS           "tmp_userrefs"
#define DMU_POOL_DDT                    "DDT-%s-%s-%s"
#define DMU_POOL_DDT_STATS              "DDT-statistics"
#define DMU_POOL_CREATION_VERSION       "creation_version"
#define DMU_POOL_SCAN                   "scan"
#define DMU_POOL_FREE_BPOBJ             "free_bpobj"
#define DMU_POOL_BPTREE_OBJ             "bptree_obj"
#define DMU_POOL_EMPTY_BPOBJ            "empty_bpobj"
```

```
313 #define DMU_POOL_CHECKSUM_SALT          "checksum_salt"

315 /*
316  * Allocate an object from this objset.  The range of object numbers
317  * available is (0, DN_MAX_OBJECT).  Object 0 is the meta-dnode.
318  *
319  * The transaction must be assigned to a txg.  The newly allocated
320  * object will be "held" in the transaction (ie. you can modify the
321  * newly allocated object in this transaction).
322  *
323  * dmu_object_alloc() chooses an object and returns it in *objectp.
324  *
325  * dmu_object_claim() allocates a specific object number.  If that
326  * number is already allocated, it fails and returns EEXIST.
327  *
328  * Return 0 on success, or ENOSPC or EEXIST as specified above.
329  */
330 uint64_t dmu_object_alloc(objset_t *os, dmu_object_type_t ot,
331     int blocksize, dmu_object_type_t bonus_type, int bonus_len, dmu_tx_t *tx);
332 int dmu_object_claim(objset_t *os, uint64_t object, dmu_object_type_t ot,
333     int blocksize, dmu_object_type_t bonus_type, int bonus_len, dmu_tx_t *tx);
334 int dmu_object_reclaim(objset_t *os, uint64_t object, dmu_object_type_t ot,
335     int blocksize, dmu_object_type_t bonustype, int bonuslen);

337 /*
338  * Free an object from this objset.
339  *
340  * The object's data will be freed as well (ie. you don't need to call
341  * dmu_free(object, 0, -1, tx)).
342  *
343  * The object need not be held in the transaction.
344  *
345  * If there are any holds on this object's buffers (via dmu_buf_hold()),
346  * or tx holds on the object (via dmu_tx_hold_object()), you can not
347  * free it; it fails and returns EBUSY.
348  *
349  * If the object is not allocated, it fails and returns ENOENT.
350  *
351  * Return 0 on success, or EBUSY or ENOENT as specified above.
352  */
353 int dmu_object_free(objset_t *os, uint64_t object, dmu_tx_t *tx);

355 /*
356  * Find the next allocated or free object.
357  *
358  * The objectp parameter is in-out.  It will be updated to be the next
359  * object which is allocated.  Ignore objects which have not been
360  * modified since txg.
361  *
362  * XXX Can only be called on a objset with no dirty data.
363  *
364  * Returns 0 on success, or ENOENT if there are no more objects.
365  */
366 int dmu_object_next(objset_t *os, uint64_t *objectp,
367     boolean_t hole, uint64_t txg);

369 /*
370  * Set the data blocksize for an object.
371  *
372  * The object cannot have any blocks allcated beyond the first.  If
373  * the first block is allocated already, the new size must be greater
374  * than the current block size.  If these conditions are not met,
375  * ENOTSUP will be returned.
376  *
377  * Returns 0 on success, or EBUSY if there are any holds on the object
378  * contents, or ENOTSUP as described above.
```



```
379  */
380 int dmu_object_set_blocksize(objset_t *os, uint64_t object, uint64_t size,
381     int ibs, dmu_tx_t *tx);

383 /*
384  * Set the checksum property on a dnode.  The new checksum algorithm will
385  * apply to all newly written blocks; existing blocks will not be affected.
386  */
387 void dmu_object_set_checksum(objset_t *os, uint64_t object, uint8_t checksum,
388     dmu_tx_t *tx);

390 /*
391  * Set the compress property on a dnode.  The new compression algorithm will
392  * apply to all newly written blocks; existing blocks will not be affected.
393  */
394 void dmu_object_set_compress(objset_t *os, uint64_t object, uint8_t compress,
395     dmu_tx_t *tx);

397 /*
398  * Decide how to write a block: checksum, compression, number of copies, etc.
399  */
400 #define WP_NOFILL       0x1
401 #define WP_DMU_SYNC     0x2
402 #define WP_SPILL        0x4

404 void dmu_write_policy(objset_t *os, struct dnode *dn, int level, int wp,
405     struct zio_prop *zp);
406 /*
407  * The bonus data is accessed more or less like a regular buffer.
408  * You must dmu_bonus_hold() to get the buffer, which will give you a
409  * dmu_buf_t with db_offset==-1ULL, and db_size = the size of the bonus
410  * data.  As with any normal buffer, you must call dmu_buf_read() to
411  * read db_data, dmu_buf_will_dirty() before modifying it, and the
412  * object must be held in an assigned transaction before calling
413  * dmu_buf_will_dirty.  You may use dmu_buf_set_user() on the bonus
414  * buffer as well.  You must release your hold with dmu_buf_rele().
415  *
416  * Returns ENOENT, EIO, or 0.
417  */
418 int dmu_bonus_hold(objset_t *os, uint64_t object, void *tag, dmu_buf_t **);
419 int dmu_bonus_max(void);
420 int dmu_set_bonus(dmu_buf_t *, int, dmu_tx_t *);
421 int dmu_set_bonustype(dmu_buf_t *, dmu_object_type_t, dmu_tx_t *);
422 dmu_object_type_t dmu_get_bonustype(dmu_buf_t *);
423 int dmu_rm_spill(objset_t *, uint64_t, dmu_tx_t *);

425 /*
426  * Special spill buffer support used by "SA" framework
427  */

429 int dmu_spill_hold_by_bonus(dmu_buf_t *bonus, void *tag, dmu_buf_t **dbp);
430 int dmu_spill_hold_by_dnode(struct dnode *dn, uint32_t flags,
431     void *tag, dmu_buf_t **dbp);
432 int dmu_spill_hold_existing(dmu_buf_t *bonus, void *tag, dmu_buf_t **dbp);

434 /*
435  * Obtain the DMU buffer from the specified object which contains the
436  * specified offset.  dmu_buf_hold() puts a "hold" on the buffer, so
437  * that it will remain in memory.  You must release the hold with
438  * dmu_buf_rele().  You musn't access the dmu_buf_t after releasing your
439  * hold.  You must have a hold on any dmu_buf_t* you pass to the DMU.
440  *
441  * You must call dmu_buf_read, dmu_buf_will_dirty, or dmu_buf_will_fill
442  * on the returned buffer before reading or writing the buffer's
443  * db_data.  The comments for those routines describe what particular
444  * operations are valid after calling them.
```

```
 *
 * The object number must be a valid, allocated object number.
 */
int dmu_buf_hold(objset_t *os, uint64_t object, uint64_t offset,
    void *tag, dmu_buf_t **, int flags);
void dmu_buf_add_ref(dmu_buf_t *db, void* tag);
void dmu_buf_rele(dmu_buf_t *db, void *tag);
uint64_t dmu_buf_refcount(dmu_buf_t *db);

/*
 * dmu_buf_hold_array holds the DMU buffers which contain all bytes in a
 * range of an object.  A pointer to an array of dmu_buf_t*'s is
 * returned (in *dbpp).
 *
 * dmu_buf_rele_array releases the hold on an array of dmu_buf_t*'s, and
 * frees the array.  The hold on the array of buffers MUST be released
 * with dmu_buf_rele_array.  You can NOT release the hold on each buffer
 * individually with dmu_buf_rele.
 */
int dmu_buf_hold_array_by_bonus(dmu_buf_t *db, uint64_t offset,
    uint64_t length, int read, void *tag, int *numbufsp, dmu_buf_t ***dbpp);
void dmu_buf_rele_array(dmu_buf_t **, int numbufs, void *tag);

/*
 * Returns NULL on success, or the existing user ptr if it's already
 * been set.
 *
 * user_ptr is for use by the user and can be obtained via dmu_buf_get_user().
 *
 * user_data_ptr_ptr should be NULL, or a pointer to a pointer which
 * will be set to db->db_data when you are allowed to access it.  Note
 * that db->db_data (the pointer) can change when you do dmu_buf_read(),
 * dmu_buf_tryupgrade(), dmu_buf_will_dirty(), or dmu_buf_will_fill().
 * *user_data_ptr_ptr will be set to the new value when it changes.
 *
 * If non-NULL, pageout func will be called when this buffer is being
 * excised from the cache, so that you can clean up the data structure
 * pointed to by user_ptr.
 *
 * dmu_evict_user() will call the pageout func for all buffers in a
 * objset with a given pageout func.
 */
void *dmu_buf_set_user(dmu_buf_t *db, void *user_ptr, void *user_data_ptr_ptr,
    dmu_buf_evict_func_t *pageout_func);
/*
 * set_user_ie is the same as set_user, but request immediate eviction
 * when hold count goes to zero.
 */
void *dmu_buf_set_user_ie(dmu_buf_t *db, void *user_ptr,
    void *user_data_ptr_ptr, dmu_buf_evict_func_t *pageout_func);
void *dmu_buf_update_user(dmu_buf_t *db_fake, void *old_user_ptr,
    void *user_ptr, void *user_data_ptr_ptr,
    dmu_buf_evict_func_t *pageout_func);
void dmu_evict_user(objset_t *os, dmu_buf_evict_func_t *func);

/*
 * Returns the user_ptr set with dmu_buf_set_user(), or NULL if not set.
 */
void *dmu_buf_get_user(dmu_buf_t *db);

/*
 * Returns the blkptr associated with this dbuf, or NULL if not set.
 */
struct blkptr *dmu_buf_get_blkptr(dmu_buf_t *db);

/*
```



```
 * Indicate that you are going to modify the buffer's data (db_data).
 *
 * The transaction (tx) must be assigned to a txg (ie. you've called
 * dmu_tx_assign()).  The buffer's object must be held in the tx
 * (ie. you've called dmu_tx_hold_object(tx, db->db_object)).
 */
void dmu_buf_will_dirty(dmu_buf_t *db, dmu_tx_t *tx);

/*
 * Tells if the given dbuf is freeable.
 */
boolean_t dmu_buf_freeable(dmu_buf_t *);

/*
 * You must create a transaction, then hold the objects which you will
 * (or might) modify as part of this transaction.  Then you must assign
 * the transaction to a transaction group.  Once the transaction has
 * been assigned, you can modify buffers which belong to held objects as
 * part of this transaction.  You can't modify buffers before the
 * transaction has been assigned; you can't modify buffers which don't
 * belong to objects which this transaction holds; you can't hold
 * objects once the transaction has been assigned.  You may hold an
 * object which you are going to free (with dmu_object_free()), but you
 * don't have to.
 *
 * You can abort the transaction before it has been assigned.
 *
 * Note that you may hold buffers (with dmu_buf_hold) at any time,
 * regardless of transaction state.
 */

#define DMU_NEW_OBJECT  (-1ULL)
#define DMU_OBJECT_END  (-1ULL)

dmu_tx_t *dmu_tx_create(objset_t *os);
void dmu_tx_hold_write(dmu_tx_t *tx, uint64_t object, uint64_t off, int len);
void dmu_tx_hold_free(dmu_tx_t *tx, uint64_t object, uint64_t off,
    uint64_t len);
void dmu_tx_hold_zap(dmu_tx_t *tx, uint64_t object, int add, const char *name);
void dmu_tx_hold_bonus(dmu_tx_t *tx, uint64_t object);
void dmu_tx_hold_spill(dmu_tx_t *tx, uint64_t object);
void dmu_tx_hold_sa(dmu_tx_t *tx, struct sa_handle *hdl, boolean_t may_grow);
void dmu_tx_hold_sa_create(dmu_tx_t *tx, int total_size);
void dmu_tx_abort(dmu_tx_t *tx);
int dmu_tx_assign(dmu_tx_t *tx, enum txg_how txg_how);
void dmu_tx_wait(dmu_tx_t *tx);
void dmu_tx_commit(dmu_tx_t *tx);

/*
 * To register a commit callback, dmu_tx_callback_register() must be called.
 *
 * dcb_data is a pointer to caller private data that is passed on as a
 * callback parameter. The caller is responsible for properly allocating and
 * freeing it.
 *
 * When registering a callback, the transaction must be already created, but
 * it cannot be committed or aborted. It can be assigned to a txg or not.
 *
 * The callback will be called after the transaction has been safely written
 * to stable storage and will also be called if the dmu_tx is aborted.
 * If there is any error which prevents the transaction from being committed to
 * disk, the callback will be called with a value of error != 0.
 */
typedef void dmu_tx_callback_func_t(void *dcb_data, int error);

void dmu_tx_callback_register(dmu_tx_t *tx, dmu_tx_callback_func_t *dcb_func,
```

```
577     void *dcb_data);

579 /*
580  * Free up the data blocks for a defined range of a file.  If size is
581  * -1, the range from offset to end-of-file is freed.
582  */
583 int dmu_free_range(objset_t *os, uint64_t object, uint64_t offset,
584         uint64_t size, dmu_tx_t *tx);
585 int dmu_free_long_range(objset_t *os, uint64_t object, uint64_t offset,
586         uint64_t size);
587 int dmu_free_long_object(objset_t *os, uint64_t object);

589 /*
590  * Convenience functions.
591  *
592  * Canfail routines will return 0 on success, or an errno if there is a
593  * nonrecoverable I/O error.
594  */
595 #define DMU_READ_PREFETCH       0 /* prefetch */
596 #define DMU_READ_NO_PREFETCH    1 /* don't prefetch */
597 int dmu_read(objset_t *os, uint64_t object, uint64_t offset, uint64_t size,
598         void *buf, uint32_t flags);
599 void dmu_write(objset_t *os, uint64_t object, uint64_t offset, uint64_t size,
600         const void *buf, dmu_tx_t *tx);
601 void dmu_prealloc(objset_t *os, uint64_t object, uint64_t offset, uint64_t size,
602         dmu_tx_t *tx);
603 int dmu_read_uio(objset_t *os, uint64_t object, struct uio *uio, uint64_t size);
604 int dmu_write_uio(objset_t *os, uint64_t object, struct uio *uio, uint64_t size,
605     dmu_tx_t *tx);
606 int dmu_write_uio_dbuf(dmu_buf_t *zdb, struct uio *uio, uint64_t size,
607     dmu_tx_t *tx);
608 int dmu_write_pages(objset_t *os, uint64_t object, uint64_t offset,
609     uint64_t size, struct page *pp, dmu_tx_t *tx);
610 struct arc_buf *dmu_request_arcbuf(dmu_buf_t *handle, int size);
611 void dmu_return_arcbuf(struct arc_buf *buf);
612 void dmu_assign_arcbuf(dmu_buf_t *handle, uint64_t offset, struct arc_buf *buf,
613     dmu_tx_t *tx);
614 int dmu_xuio_init(struct xuio *uio, int niov);
615 void dmu_xuio_fini(struct xuio *uio);
616 int dmu_xuio_add(struct xuio *uio, struct arc_buf *abuf, offset_t off,
617     size_t n);
618 int dmu_xuio_cnt(struct xuio *uio);
619 struct arc_buf *dmu_xuio_arcbuf(struct xuio *uio, int i);
620 void dmu_xuio_clear(struct xuio *uio, int i);
621 void xuio_stat_wbuf_copied();
622 void xuio_stat_wbuf_nocopy();

624 extern int zfs_prefetch_disable;

626 /*
627  * Asynchronously try to read in the data.
628  */
629 void dmu_prefetch(objset_t *os, uint64_t object, uint64_t offset,
630     uint64_t len);

632 typedef struct dmu_object_info {
633         /* All sizes are in bytes unless otherwise indicated. */
634         uint32_t doi_data_block_size;
635         uint32_t doi_metadata_block_size;
636         dmu_object_type_t doi_type;
637         dmu_object_type_t doi_bonus_type;
638         uint64_t doi_bonus_size;
639         uint8_t doi_indirection;                /* 2 = dnode->indirect->data */
640         uint8_t doi_checksum;
641         uint8_t doi_compress;
642         uint8_t doi_pad[5];
```



```
643         uint64_t doi_physical_blocks_512;       /* data + metadata, 512b blks */
644         uint64_t doi_max_offset;
645         uint64_t doi_fill_count;                /* number of non-empty blocks */
646 } dmu_object_info_t;
```

unchanged_portion_omitted_

```
**********************************************************
   26016 Mon Oct  7 19:17:20 2013
new/usr/src/uts/common/fs/zfs/sys/spa.h
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2012 by Delphix. All rights reserved.
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 * Copyright 2013 Saso Kiselkov. All rights reserved.
 */

#ifndef _SYS_SPA_H
#define _SYS_SPA_H

#include <sys/avl.h>
#include <sys/zfs_context.h>
#include <sys/nvpair.h>
#include <sys/sysmacros.h>
#include <sys/types.h>
#include <sys/fs/zfs.h>

#ifdef  __cplusplus
extern "C" {
#endif

/*
 * Forward references that lots of things need.
 */
typedef struct spa spa_t;
typedef struct vdev vdev_t;
typedef struct metaslab metaslab_t;
typedef struct metaslab_group metaslab_group_t;
typedef struct metaslab_class metaslab_class_t;
typedef struct zio zio_t;
typedef struct zilog zilog_t;
typedef struct spa_aux_vdev spa_aux_vdev_t;
typedef struct ddt ddt_t;
typedef struct ddt_entry ddt_entry_t;
struct dsl_pool;
struct dsl_dataset;
struct zfeature_info;

/*
 * General-purpose 32-bit and 64-bit bitfield encodings.
 */
```



```
#define BF32_DECODE(x, low, len)        P2PHASE((x) >> (low), 1U << (len))
#define BF64_DECODE(x, low, len)        P2PHASE((x) >> (low), 1ULL << (len))
#define BF32_ENCODE(x, low, len)        (P2PHASE((x), 1U << (len)) << (low))
#define BF64_ENCODE(x, low, len)        (P2PHASE((x), 1ULL << (len)) << (low))

#define BF32_GET(x, low, len)           BF32_DECODE(x, low, len)
#define BF64_GET(x, low, len)           BF64_DECODE(x, low, len)

#define BF32_SET(x, low, len, val)      \
        ((x) ^= BF32_ENCODE((x >> low) ^ (val), low, len))
#define BF64_SET(x, low, len, val)      \
        ((x) ^= BF64_ENCODE((x >> low) ^ (val), low, len))

#define BF32_GET_SB(x, low, len, shift, bias)   \
        ((BF32_GET(x, low, len) + (bias)) << (shift))
#define BF64_GET_SB(x, low, len, shift, bias)   \
        ((BF64_GET(x, low, len) + (bias)) << (shift))

#define BF32_SET_SB(x, low, len, shift, bias, val)      \
        BF32_SET(x, low, len, ((val) >> (shift)) - (bias))
#define BF64_SET_SB(x, low, len, shift, bias, val)      \
        BF64_SET(x, low, len, ((val) >> (shift)) - (bias))

/*
 * We currently support nine block sizes, from 512 bytes to 128K.
 * We could go higher, but the benefits are near-zero and the cost
 * of COWing a giant block to modify one byte would become excessive.
 */
#define SPA_MINBLOCKSHIFT       9
#define SPA_MAXBLOCKSHIFT       17
#define SPA_MINBLOCKSIZE        (1ULL << SPA_MINBLOCKSHIFT)
#define SPA_MAXBLOCKSIZE        (1ULL << SPA_MAXBLOCKSHIFT)

#define SPA_BLOCKSIZES          (SPA_MAXBLOCKSHIFT - SPA_MINBLOCKSHIFT + 1)

/*
 * Size of block to hold the configuration data (a packed nvlist)
 */
#define SPA_CONFIG_BLOCKSIZE    (1ULL << 14)

/*
 * The DVA size encodings for LSIZE and PSIZE support blocks up to 32MB.
 * The ASIZE encoding should be at least 64 times larger (6 more bits)
 * to support up to 4-way RAID-Z mirror mode with worst-case gang block
 * overhead, three DVAs per bp, plus one more bit in case we do anything
 * else that expands the ASIZE.
 */
#define SPA_LSIZEBITS           16      /* LSIZE up to 32M (2^16 * 512) */
#define SPA_PSIZEBITS           16      /* PSIZE up to 32M (2^16 * 512) */
#define SPA_ASIZEBITS           24      /* ASIZE up to 64 times larger  */

/*
 * All SPA data is represented by 128-bit data virtual addresses (DVAs).
 * The members of the dva_t should be considered opaque outside the SPA.
 */
typedef struct dva {
        uint64_t        dva_word[2];
} dva_t;
_____unchanged_portion_omitted_

/*
 * Some checksums/hashes need a 256-bit initialization salt. This salt is kept
 * secret and is suitable for use in MAC algorithms as the key.
 */
typedef struct zio_cksum_salt {
        uint8_t         zcs_bytes[32];
```

```
} zio_cksum_salt_t;

/*
 * Each block is described by its DVAs, time of birth, checksum, etc.
 * The word-by-word, bit-by-bit layout of the blkptr is as follows:
 *
 *       64      56      48      40      32      24      16      8       0
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * 0     |               vdev1           | GRID  |         ASIZE         |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * 1     |G|                      offset1                                |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * 2     |               vdev2           | GRID  |         ASIZE         |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * 3     |G|                      offset2                                |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * 4     |               vdev3           | GRID  |         ASIZE         |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * 5     |G|                      offset3                                |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * 6     |BDX|lvl| type  | cksum | comp  |     PSIZE     |     LSIZE     |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * 7     |                       padding                                 |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * 8     |                       padding                                 |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * 9     |                       physical birth txg                      |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * a     |                       logical birth txg                       |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * b     |                       fill count                              |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * c     |                       checksum[0]                             |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * d     |                       checksum[1]                             |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * e     |                       checksum[2]                             |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 * f     |                       checksum[3]                             |
 *       +-------+-------+-------+-------+-------+-------+-------+-------+
 *
 * Legend:
 *
 * vdev         virtual device ID
 * offset       offset into virtual device
 * LSIZE        logical size
 * PSIZE        physical size (after compression)
 * ASIZE        allocated size (including RAID-Z parity and gang block headers)
 * GRID         RAID-Z layout information (reserved for future use)
 * cksum        checksum function
 * comp         compression function
 * G            gang block indicator
 * B            byteorder (endianness)
 * D            dedup
 * X            unused
 * lvl          level of indirection
 * type         DMU object type
 * phys birth   txg of block allocation; zero if same as logical birth txg
 * log. birth   transaction group in which the block was logically born
 * fill count   number of non-zero blocks under this bp
 * checksum[4]  256-bit checksum of the data this bp describes
 */
#define SPA_BLKPTRSHIFT 7               /* blkptr_t is 128 bytes        */
#define SPA_DVAS_PER_BP 3               /* Number of DVAs in a bp       */

typedef struct blkptr {
```



```
        dva_t           blk_dva[SPA_DVAS_PER_BP]; /* Data Virtual Addresses */
        uint64_t        blk_prop;       /* size, compression, type, etc     */
        uint64_t        blk_pad[2];     /* Extra space for the future       */
        uint64_t        blk_phys_birth; /* txg when block was allocated     */
        uint64_t        blk_birth;      /* transaction group at birth       */
        uint64_t        blk_fill;       /* fill count                       */
        zio_cksum_t     blk_cksum;      /* 256-bit checksum                 */
} blkptr_t;
______unchanged_portion_omitted_

extern spa_log_state_t spa_get_log_state(spa_t *spa);
extern void spa_set_log_state(spa_t *spa, spa_log_state_t state);
extern int spa_offline_log(spa_t *spa);

/* Log claim callback */
extern void spa_claim_notify(zio_t *zio);

/* Accessor functions */
extern boolean_t spa_shutting_down(spa_t *spa);
extern struct dsl_pool *spa_get_dsl(spa_t *spa);
extern boolean_t spa_is_initializing(spa_t *spa);
extern blkptr_t *spa_get_rootblkptr(spa_t *spa);
extern void spa_set_rootblkptr(spa_t *spa, const blkptr_t *bp);
extern void spa_altroot(spa_t *, char *, size_t);
extern int spa_sync_pass(spa_t *spa);
extern char *spa_name(spa_t *spa);
extern uint64_t spa_guid(spa_t *spa);
extern uint64_t spa_load_guid(spa_t *spa);
extern uint64_t spa_last_synced_txg(spa_t *spa);
extern uint64_t spa_first_txg(spa_t *spa);
extern uint64_t spa_syncing_txg(spa_t *spa);
extern uint64_t spa_version(spa_t *spa);
extern pool_state_t spa_state(spa_t *spa);
extern spa_load_state_t spa_load_state(spa_t *spa);
extern uint64_t spa_freeze_txg(spa_t *spa);
extern uint64_t spa_get_asize(spa_t *spa, uint64_t lsize);
extern uint64_t spa_get_dspace(spa_t *spa);
extern void spa_update_dspace(spa_t *spa);
extern uint64_t spa_version(spa_t *spa);
extern boolean_t spa_deflate(spa_t *spa);
extern metaslab_class_t *spa_normal_class(spa_t *spa);
extern metaslab_class_t *spa_log_class(spa_t *spa);
extern int spa_max_replication(spa_t *spa);
extern int spa_prev_software_version(spa_t *spa);
extern int spa_busy(void);
extern uint8_t spa_get_failmode(spa_t *spa);
extern boolean_t spa_suspended(spa_t *spa);
extern uint64_t spa_bootfs(spa_t *spa);
extern uint64_t spa_delegation(spa_t *spa);
extern objset_t *spa_meta_objset(spa_t *spa);
extern uint64_t spa_deadman_synctime(spa_t *spa);

/* Miscellaneous support routines */
extern void spa_activate_mos_feature(spa_t *spa, const char *feature);
extern void spa_deactivate_mos_feature(spa_t *spa, const char *feature);
extern int spa_rename(const char *oldname, const char *newname);
extern spa_t *spa_by_guid(uint64_t pool_guid, uint64_t device_guid);
extern boolean_t spa_guid_exists(uint64_t pool_guid, uint64_t device_guid);
extern char *spa_strdup(const char *);
extern void spa_strfree(char *);
extern uint64_t spa_get_random(uint64_t range);
extern uint64_t spa_generate_guid(spa_t *spa);
extern void sprintf_blkptr(char *buf, const blkptr_t *bp);
extern void spa_freeze(spa_t *spa);
extern int spa_change_guid(spa_t *spa);
extern void spa_upgrade(spa_t *spa, uint64_t version);
```

```
625 extern void spa_evict_all(void);
626 extern vdev_t *spa_lookup_by_guid(spa_t *spa, uint64_t guid,
627     boolean_t l2cache);
628 extern boolean_t spa_has_spare(spa_t *, uint64_t guid);
629 extern uint64_t dva_get_dsize_sync(spa_t *spa, const dva_t *dva);
630 extern uint64_t bp_get_dsize_sync(spa_t *spa, const blkptr_t *bp);
631 extern uint64_t bp_get_dsize(spa_t *spa, const blkptr_t *bp);
632 extern boolean_t spa_has_slogs(spa_t *spa);
633 extern boolean_t spa_is_root(spa_t *spa);
634 extern boolean_t spa_writeable(spa_t *spa);

636 /* Salted checksum handling */
637 extern int spa_activate_salted_cksum(spa_t *spa, struct zfeature_info *feature);

639 extern int spa_mode(spa_t *spa);
640 extern uint64_t strtonum(const char *str, char **nptr);

642 extern char *spa_his_ievent_table[];

644 extern void spa_history_create_obj(spa_t *spa, dmu_tx_t *tx);
645 extern int spa_history_get(spa_t *spa, uint64_t *offset, uint64_t *len_read,
646     char *his_buf);
647 extern int spa_history_log(spa_t *spa, const char *his_buf);
648 extern int spa_history_log_nvl(spa_t *spa, nvlist_t *nvl);
649 extern void spa_history_log_version(spa_t *spa, const char *operation);
650 extern void spa_history_log_internal(spa_t *spa, const char *operation,
651     dmu_tx_t *tx, const char *fmt, ...);
652 extern void spa_history_log_internal_ds(struct dsl_dataset *ds, const char *op,
653     dmu_tx_t *tx, const char *fmt, ...);
654 extern void spa_history_log_internal_dd(dsl_dir_t *dd, const char *operation,
655     dmu_tx_t *tx, const char *fmt, ...);

657 /* error handling */
658 struct zbookmark;
659 extern void spa_log_error(spa_t *spa, zio_t *zio);
660 extern void zfs_ereport_post(const char *class, spa_t *spa, vdev_t *vd,
661     zio_t *zio, uint64_t stateoroffset, uint64_t length);
662 extern void zfs_post_remove(spa_t *spa, vdev_t *vd);
663 extern void zfs_post_state_change(spa_t *spa, vdev_t *vd);
664 extern void zfs_post_autoreplace(spa_t *spa, vdev_t *vd);
665 extern uint64_t spa_get_errlog_size(spa_t *spa);
666 extern int spa_get_errlog(spa_t *spa, void *uaddr, size_t *count);
667 extern void spa_errlog_rotate(spa_t *spa);
668 extern void spa_errlog_drain(spa_t *spa);
669 extern void spa_errlog_sync(spa_t *spa, uint64_t txg);
670 extern void spa_get_errlists(spa_t *spa, avl_tree_t *last, avl_tree_t *scrub);

672 /* vdev cache */
673 extern void vdev_cache_stat_init(void);
674 extern void vdev_cache_stat_fini(void);

676 /* Initialization and termination */
677 extern void spa_init(int flags);
678 extern void spa_fini(void);
679 extern void spa_boot_init();

681 /* properties */
682 extern int spa_prop_set(spa_t *spa, nvlist_t *nvp);
683 extern int spa_prop_get(spa_t *spa, nvlist_t **nvp);
684 extern void spa_prop_clear_bootfs(spa_t *spa, uint64_t obj, dmu_tx_t *tx);
685 extern void spa_configfile_set(spa_t *, nvlist_t *, boolean_t);

687 /* asynchronous event notification */
688 extern void spa_event_notify(spa_t *spa, vdev_t *vdev, const char *name);

690 #ifdef ZFS_DEBUG
```



```
691 #define dprintf_bp(bp, fmt, ...) do {                           \
692         if (zfs_flags & ZFS_DEBUG_DPRINTF) {                    \
693         char *__blkbuf = kmem_alloc(BP_SPRINTF_LEN, KM_SLEEP);  \
694         sprintf_blkptr(__blkbuf, (bp));                         \
695         dprintf(fmt " %s\n", __VA_ARGS__, __blkbuf);            \
696         kmem_free(__blkbuf, BP_SPRINTF_LEN);                    \
697         } \
698 _NOTE(CONSTCOND) } while (0)
699 #else
700 #define dprintf_bp(bp, fmt, ...)
701 #endif

703 extern boolean_t spa_debug_enabled(spa_t *spa);
704 #define spa_dbgmsg(spa, ...)                    \
705 {                                               \
706         if (spa_debug_enabled(spa))             \
707                 zfs_dbgmsg(__VA_ARGS__);        \
708 }
______unchanged_portion_omitted_
```

```
**********************************************************
   11367 Mon Oct  7 19:17:20 2013
new/usr/src/uts/common/fs/zfs/sys/spa_impl.h
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2013 by Delphix. All rights reserved.
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 * Copyright 2013 Saso Kiselkov. All rights reserved.
 */

#ifndef _SYS_SPA_IMPL_H
#define _SYS_SPA_IMPL_H

#include <sys/spa.h>
#include <sys/vdev.h>
#include <sys/metaslab.h>
#include <sys/dmu.h>
#include <sys/dsl_pool.h>
#include <sys/uberblock_impl.h>
#include <sys/zfs_context.h>
#include <sys/avl.h>
#include <sys/refcount.h>
#include <sys/bplist.h>
#include <sys/bpobj.h>

#ifdef  __cplusplus
extern "C" {
#endif

typedef struct spa_error_entry {
        zbookmark_t     se_bookmark;
        char            *se_name;
        avl_node_t      se_avl;
} spa_error_entry_t;
```

*unchanged_portion_omitted_*

```
struct spa {
        /*
         * Fields protected by spa_namespace_lock.
         */
        char            spa_name[MAXNAMELEN];   /* pool name */
        char            *spa_comment;           /* comment */
        avl_node_t      spa_avl;                /* node in spa_namespace_avl */
        nvlist_t        *spa_config;            /* last synced config */
```



```
        nvlist_t        *spa_config_syncing;    /* currently syncing config */
        nvlist_t        *spa_config_splitting;  /* config for splitting */
        nvlist_t        *spa_load_info;         /* info and errors from load */
        uint64_t        spa_config_txg;         /* txg of last config change */
        int             spa_sync_pass;          /* iterate-to-convergence */
        pool_state_t    spa_state;              /* pool state */
        int             spa_inject_ref;         /* injection references */
        uint8_t         spa_sync_on;            /* sync threads are running */
        spa_load_state_t spa_load_state;        /* current load operation */
        uint64_t        spa_import_flags;       /* import specific flags */
        spa_taskqs_t    spa_zio_taskq[ZIO_TYPES][ZIO_TASKQ_TYPES];
        dsl_pool_t      *spa_dsl_pool;
        boolean_t       spa_is_initializing;    /* true while opening pool */
        metaslab_class_t *spa_normal_class;     /* normal data class */
        metaslab_class_t *spa_log_class;        /* intent log data class */
        uint64_t        spa_first_txg;          /* first txg after spa_open() */
        uint64_t        spa_final_txg;          /* txg of export/destroy */
        uint64_t        spa_freeze_txg;         /* freeze pool at this txg */
        uint64_t        spa_load_max_txg;       /* best initial ub_txg */
        uint64_t        spa_claim_max_txg;      /* highest claimed birth txg */
        timespec_t      spa_loaded_ts;          /* 1st successful open time */
        objset_t        *spa_meta_objset;       /* copy of dp->dp_meta_objset */
        txg_list_t      spa_vdev_txg_list;      /* per-txg dirty vdev list */
        vdev_t          *spa_root_vdev;         /* top-level vdev container */
        uint64_t        spa_config_guid;        /* config pool guid */
        uint64_t        spa_load_guid;          /* spa_load initialized guid */
        uint64_t        spa_last_synced_guid;   /* last synced guid */
        list_t          spa_config_dirty_list;  /* vdevs with dirty config */
        list_t          spa_state_dirty_list;   /* vdevs with dirty state */
        spa_aux_vdev_t  spa_spares;             /* hot spares */
        spa_aux_vdev_t  spa_l2cache;            /* L2ARC cache devices */
        nvlist_t        *spa_label_features;    /* Features for reading MOS */
        uint64_t        spa_config_object;      /* MOS object for pool config */
        uint64_t        spa_config_generation;  /* config generation number */
        uint64_t        spa_syncing_txg;        /* txg currently syncing */
        bpobj_t         spa_deferred_bpobj;     /* deferred-free bplist */
        bplist_t        spa_free_bplist[TXG_SIZE]; /* bplist of stuff to free */
        zio_cksum_salt_t spa_cksum_salt;        /* secret salt for cksum */
        uint64_t        spa_cksum_salt_obj;     /* persistent salt object */
        /* checksum context templates */
        kmutex_t        spa_cksum_tmpls_lock;
        void            *spa_cksum_tmpls[ZIO_CHECKSUM_FUNCTIONS];
        uberblock_t     spa_ubsync;             /* last synced uberblock */
        uberblock_t     spa_uberblock;          /* current uberblock */
        boolean_t       spa_extreme_rewind;     /* rewind past deferred frees */
        uint64_t        spa_last_io;            /* lbolt of last non-scan I/O */
        kmutex_t        spa_scrub_lock;         /* resilver/scrub lock */
        uint64_t        spa_scrub_inflight;     /* in-flight scrub I/Os */
        kcondvar_t      spa_scrub_io_cv;        /* scrub I/O completion */
        uint8_t         spa_scrub_active;       /* active or suspended? */
        uint8_t         spa_scrub_type;         /* type of scrub we're doing */
        uint8_t         spa_scrub_finished;     /* indicator to rotate logs */
        uint8_t         spa_scrub_started;      /* started since last boot */
        uint8_t         spa_scrub_reopen;       /* scrub doing vdev_reopen */
        uint64_t        spa_scan_pass_start;    /* start time per pass/reboot */
        uint64_t        spa_scan_pass_exam;     /* examined bytes per pass */
        kmutex_t        spa_async_lock;         /* protect async state */
        kthread_t       *spa_async_thread;      /* thread doing async task */
        int             spa_async_suspended;    /* async tasks suspended */
        kcondvar_t      spa_async_cv;           /* wait for thread_exit() */
        uint16_t        spa_async_tasks;        /* async task mask */
        char            *spa_root;              /* alternate root directory */
        uint64_t        spa_ena;                /* spa-wide ereport ENA */
        int             spa_last_open_failed;   /* error if last open failed */
        uint64_t        spa_last_ubsync_txg;    /* "best" uberblock txg */
        uint64_t        spa_last_ubsync_txg_ts; /* timestamp from that ub */
```

```
190         uint64_t        spa_load_txg;           /* ub txg that loaded */
191         uint64_t        spa_load_txg_ts;        /* timestamp from that ub */
192         uint64_t        spa_load_meta_errors;   /* verify metadata err count */
193         uint64_t        spa_load_data_errors;   /* verify data err count */
194         uint64_t        spa_verify_min_txg;     /* start txg of verify scrub */
195         kmutex_t        spa_errlog_lock;        /* error log lock */
196         uint64_t        spa_errlog_last;        /* last error log object */
197         uint64_t        spa_errlog_scrub;       /* scrub error log object */
198         kmutex_t        spa_errlist_lock;       /* error list/ereport lock */
199         avl_tree_t      spa_errlist_last;       /* last error list */
200         avl_tree_t      spa_errlist_scrub;      /* scrub error list */
201         uint64_t        spa_deflate;            /* should we deflate? */
202         uint64_t        spa_history;            /* history object */
203         kmutex_t        spa_history_lock;       /* history lock */
204         vdev_t          *spa_pending_vdev;      /* pending vdev additions */
205         kmutex_t        spa_props_lock;         /* property lock */
206         uint64_t        spa_pool_props_object;  /* object for properties */
207         uint64_t        spa_bootfs;             /* default boot filesystem */
208         uint64_t        spa_failmode;           /* failure mode for the pool */
209         uint64_t        spa_delegation;         /* delegation on/off */
210         list_t          spa_config_list;        /* previous cache file(s) */
211         zio_t           *spa_async_zio_root;    /* root of all async I/O */
212         zio_t           *spa_suspend_zio_root;  /* root of all suspended I/O */
213         kmutex_t        spa_suspend_lock;       /* protects suspend_zio_root */
214         kcondvar_t      spa_suspend_cv;         /* notification of resume */
215         uint8_t         spa_suspended;          /* pool is suspended */
216         uint8_t         spa_claiming;           /* pool is doing zil_claim() */
217         boolean_t       spa_debug;              /* debug enabled? */
218         boolean_t       spa_is_root;            /* pool is root */
219         int             spa_minref;             /* num refs when first opened */
220         int             spa_mode;               /* FREAD | FWRITE */
221         spa_log_state_t spa_log_state;          /* log state */
222         uint64_t        spa_autoexpand;         /* lun expansion on/off */
223         ddt_t           *spa_ddt[ZIO_CHECKSUM_FUNCTIONS]; /* in-core DDTs */
224         uint64_t        spa_ddt_stat_object;    /* DDT statistics */
225         uint64_t        spa_dedup_ditto;        /* dedup ditto threshold */
226         uint64_t        spa_dedup_checksum;     /* default dedup checksum */
227         uint64_t        spa_dspace;             /* dspace in normal class */
228         kmutex_t        spa_vdev_top_lock;      /* dueling offline/remove */
229         kmutex_t        spa_proc_lock;          /* protects spa_proc* */
230         kcondvar_t      spa_proc_cv;            /* spa_proc_state transitions */
231         spa_proc_state_t spa_proc_state;        /* see definition */
232         struct proc     *spa_proc;              /* "zpool-poolname" process */
233         uint64_t        spa_did;                /* if procp != p0, did of t1 */
234         boolean_t       spa_autoreplace;        /* autoreplace set in open */
235         int             spa_vdev_locks;         /* locks grabbed */
236         uint64_t        spa_creation_version;   /* version at pool creation */
237         uint64_t        spa_prev_software_version; /* See ub_software_version */
238         uint64_t        spa_feat_for_write_obj; /* required to write to pool */
239         uint64_t        spa_feat_for_read_obj;  /* required to read from pool */
240         uint64_t        spa_feat_desc_obj;      /* Feature descriptions */
241         cyclic_id_t     spa_deadman_cycid;      /* cyclic id */
242         uint64_t        spa_deadman_calls;      /* number of deadman calls */
243         hrtime_t        spa_sync_starttime;     /* starting time fo spa_sync */
244         uint64_t        spa_deadman_synctime;   /* deadman expiration timer */

246         /*
247          * spa_iokstat_lock protects spa_iokstat and
248          * spa_queue_stats[].
249          */
250         kmutex_t        spa_iokstat_lock;
251         struct kstat    *spa_iokstat;           /* kstat of io to this pool */
252         struct {
253                 int spa_active;
254                 int spa_queued;
255         } spa_queue_stats[ZIO_PRIORITY_NUM_QUEUEABLE];
```



```
257         hrtime_t        spa_ccw_fail_time;      /* Conf cache write fail time */

259         /*
260          * spa_refcount & spa_config_lock must be the last elements
261          * because refcount_t changes size based on compilation options.
262          * In order for the MDB module to function correctly, the other
263          * fields must remain in the same location.
264          */
265         spa_config_lock_t spa_config_lock[SCL_LOCKS]; /* config changes */
266         refcount_t      spa_refcount;           /* number of opens */
267 };
```

_____***unchanged_portion_omitted_***

****
17813 Mon Oct 7 19:17:21 2013
new/usr/src/uts/common/fs/zfs/sys/zio.h
4185 New hash algorithm support
****
_____***unchanged_portion_omitted_***

```
 72 enum zio_checksum {
 73         ZIO_CHECKSUM_INHERIT = 0,
 74         ZIO_CHECKSUM_ON,
 75         ZIO_CHECKSUM_OFF,
 76         ZIO_CHECKSUM_LABEL,
 77         ZIO_CHECKSUM_GANG_HEADER,
 78         ZIO_CHECKSUM_ZILOG,
 79         ZIO_CHECKSUM_FLETCHER_2,
 80         ZIO_CHECKSUM_FLETCHER_4,
 81         ZIO_CHECKSUM_SHA256,
 82         ZIO_CHECKSUM_ZILOG2,
 83         ZIO_CHECKSUM_NOPARITY,
 84         ZIO_CHECKSUM_SHA512,
 85         ZIO_CHECKSUM_SKEIN,
 86         ZIO_CHECKSUM_EDONR,
 87         ZIO_CHECKSUM_FUNCTIONS
 88 };
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3219 Mon Oct  7 19:17:21 2013
new/usr/src/uts/common/fs/zfs/sys/zio_checksum.h
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright Saso Kiselkov 2013, All rights reserved.
 */

#ifndef _SYS_ZIO_CHECKSUM_H
#define _SYS_ZIO_CHECKSUM_H

#include <sys/zio.h>

#ifdef  __cplusplus
extern "C" {
#endif

/*
 * Signature for checksum functions.
 */
typedef void zio_checksum_t(const void *data, uint64_t size,
    const zio_cksum_salt_t *salt, const void *ctx_template, zio_cksum_t *zcp);
typedef void *zio_checksum_tmpl_init_t(const zio_cksum_salt_t *salt);
typedef void zio_checksum_tmpl_free_t(void *ctx_template);
typedef zio_checksum_t(const void *data, uint64_t size, zio_cksum_t *zcp);

/*
 * Information about each checksum function.
 */
typedef struct zio_checksum_info {
        /* checksum function for each byteorder */
        zio_checksum_t                  *ci_func[2];
        zio_checksum_tmpl_init_t        *ci_tmpl_init;
        zio_checksum_tmpl_free_t        *ci_tmpl_free;
        zio_checksum_t  *ci_func[2]; /* checksum function for each byteorder */
        int             ci_correctable; /* number of correctable bits */
        int             ci_eck;         /* uses zio embedded checksum? */
        int             ci_dedup;       /* strong enough for dedup? */
        int             ci_salted;      /* does the checksum support salts */
        char            *ci_name;       /* descriptive name */
} zio_checksum_info_t;
```

*unchanged_portion_omitted_*

```
extern zio_checksum_info_t zio_checksum_table[ZIO_CHECKSUM_FUNCTIONS];
```



```
/*
 * Checksum routines.
 */
extern zio_checksum_t zio_checksum_SHA256;
extern zio_checksum_t zio_checksum_SHA512_native;
extern zio_checksum_t zio_checksum_SHA512_byteswap;

/* Skein */
extern zio_checksum_t zio_checksum_skein_native;
extern zio_checksum_t zio_checksum_skein_byteswap;
extern zio_checksum_tmpl_init_t zio_checksum_skein_tmpl_init;
extern zio_checksum_tmpl_free_t zio_checksum_skein_tmpl_free;

/* Edon-R */
extern zio_checksum_t zio_checksum_edonr_native;
extern zio_checksum_t zio_checksum_edonr_byteswap;
extern zio_checksum_tmpl_init_t zio_checksum_edonr_tmpl_init;
extern zio_checksum_tmpl_free_t zio_checksum_edonr_tmpl_free;

extern void zio_checksum_compute(zio_t *zio, enum zio_checksum checksum,
    void *data, uint64_t size);
extern int zio_checksum_error(zio_t *zio, zio_bad_cksum_t *out);
extern enum zio_checksum spa_dedup_checksum(spa_t *spa);
extern void zio_checksum_templates_free(spa_t *spa);

#ifdef  __cplusplus
}
```

*unchanged_portion_omitted_*

```
**********************************************************
  146559 Mon Oct  7 19:17:21 2013
new/usr/src/uts/common/fs/zfs/zfs_ioctl.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
 * Portions Copyright 2011 Martin Matuska
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 * Copyright (c) 2012, Joyent, Inc. All rights reserved.
 * Copyright (c) 2013 by Delphix. All rights reserved.
 * Copyright (c) 2013 by Saso Kiselkov. All rights reserved.
 * Copyright (c) 2013 Steven Hartland. All rights reserved.
 */

/*
 * ZFS ioctls.
 *
 * This file handles the ioctls to /dev/zfs, used for configuring ZFS storage
 * pools and filesystems, e.g. with /sbin/zfs and /sbin/zpool.
 *
 * There are two ways that we handle ioctls: the legacy way where almost
 * all of the logic is in the ioctl callback, and the new way where most
 * of the marshalling is handled in the common entry point, zfsdev_ioctl().
 *
 * Non-legacy ioctls should be registered by calling
 * zfs_ioctl_register() from zfs_ioctl_init().  The ioctl is invoked
 * from userland by lzc_ioctl().
 *
 * The registration arguments are as follows:
 *
 * const char *name
 *   The name of the ioctl.  This is used for history logging.  If the
 *   ioctl returns successfully (the callback returns 0), and allow_log
 *   is true, then a history log entry will be recorded with the input &
 *   output nvlists.  The log entry can be printed with "zpool history -i".
 *
 * zfs_ioc_t ioc
 *   The ioctl request number, which userland will pass to ioctl(2).
 *   The ioctl numbers can change from release to release, because
 *   the caller (libzfs) must be matched to the kernel.
 *
 * zfs_secpolicy_func_t *secpolicy
 *   This function will be called before the zfs_ioc_func_t, to
 *   determine if this operation is permitted.  It should return EPERM
```



```
 *   on failure, and 0 on success.  Checks include determining if the
 *   dataset is visible in this zone, and if the user has either all
 *   zfs privileges in the zone (SYS_MOUNT), or has been granted permission
 *   to do this operation on this dataset with "zfs allow".
 *
 * zfs_ioc_namecheck_t namecheck
 *   This specifies what to expect in the zfs_cmd_t:zc_name -- a pool
 *   name, a dataset name, or nothing.  If the name is not well-formed,
 *   the ioctl will fail and the callback will not be called.
 *   Therefore, the callback can assume that the name is well-formed
 *   (e.g. is null-terminated, doesn't have more than one '@' character,
 *   doesn't have invalid characters).
 *
 * zfs_ioc_poolcheck_t pool_check
 *   This specifies requirements on the pool state.  If the pool does
 *   not meet them (is suspended or is readonly), the ioctl will fail
 *   and the callback will not be called.  If any checks are specified
 *   (i.e. it is not POOL_CHECK_NONE), namecheck must not be NO_NAME.
 *   Multiple checks can be or-ed together (e.g. POOL_CHECK_SUSPENDED |
 *   POOL_CHECK_READONLY).
 *
 * boolean_t smush_outnvlist
 *   If smush_outnvlist is true, then the output is presumed to be a
 *   list of errors, and it will be "smushed" down to fit into the
 *   caller's buffer, by removing some entries and replacing them with a
 *   single "N_MORE_ERRORS" entry indicating how many were removed.  See
 *   nvlist_smush() for details.  If smush_outnvlist is false, and the
 *   outnvlist does not fit into the userland-provided buffer, then the
 *   ioctl will fail with ENOMEM.
 *
 * zfs_ioc_func_t *func
 *   The callback function that will perform the operation.
 *
 *   The callback should return 0 on success, or an error number on
 *   failure.  If the function fails, the userland ioctl will return -1,
 *   and errno will be set to the callback's return value.  The callback
 *   will be called with the following arguments:
 *
 *   const char *name
 *     The name of the pool or dataset to operate on, from
 *     zfs_cmd_t:zc_name.  The 'namecheck' argument specifies the
 *     expected type (pool, dataset, or none).
 *
 *   nvlist_t *innvl
 *     The input nvlist, deserialized from zfs_cmd_t:zc_nvlist_src.  Or
 *     NULL if no input nvlist was provided.  Changes to this nvlist are
 *     ignored.  If the input nvlist could not be deserialized, the
 *     ioctl will fail and the callback will not be called.
 *
 *   nvlist_t *outnvl
 *     The output nvlist, initially empty.  The callback can fill it in,
 *     and it will be returned to userland by serializing it into
 *     zfs_cmd_t:zc_nvlist_dst.  If it is non-empty, and serialization
 *     fails (e.g. because the caller didn't supply a large enough
 *     buffer), then the overall ioctl will fail.  See the
 *     'smush_nvlist' argument above for additional behaviors.
 *
 *     There are two typical uses of the output nvlist:
 *       - To return state, e.g. property values.  In this case,
 *         smush_outnvlist should be false.  If the buffer was not large
 *         enough, the caller will reallocate a larger buffer and try
 *         the ioctl again.
 *
 *       - To return multiple errors from an ioctl which makes on-disk
 *         changes.  In this case, smush_outnvlist should be true.
 *         Ioctls which make on-disk modifications should generally not
```

```
 128  *         use the outnvl if they succeed, because the caller can not
 129  *         distinguish between the operation failing, and
 130  *         deserialization failing.
 131  */

 133 #include <sys/types.h>
 134 #include <sys/param.h>
 135 #include <sys/errno.h>
 136 #include <sys/uio.h>
 137 #include <sys/buf.h>
 138 #include <sys/modctl.h>
 139 #include <sys/open.h>
 140 #include <sys/file.h>
 141 #include <sys/kmem.h>
 142 #include <sys/conf.h>
 143 #include <sys/cmn_err.h>
 144 #include <sys/stat.h>
 145 #include <sys/zfs_ioctl.h>
 146 #include <sys/zfs_vfsops.h>
 147 #include <sys/zfs_znode.h>
 148 #include <sys/zap.h>
 149 #include <sys/spa.h>
 150 #include <sys/spa_impl.h>
 151 #include <sys/vdev.h>
 152 #include <sys/priv_impl.h>
 153 #include <sys/dmu.h>
 154 #include <sys/dsl_dir.h>
 155 #include <sys/dsl_dataset.h>
 156 #include <sys/dsl_prop.h>
 157 #include <sys/dsl_deleg.h>
 158 #include <sys/dmu_objset.h>
 159 #include <sys/dmu_impl.h>
 160 #include <sys/dmu_tx.h>
 161 #include <sys/ddi.h>
 162 #include <sys/sunddi.h>
 163 #include <sys/sunldi.h>
 164 #include <sys/policy.h>
 165 #include <sys/zone.h>
 166 #include <sys/nvpair.h>
 167 #include <sys/pathname.h>
 168 #include <sys/mount.h>
 169 #include <sys/sdt.h>
 170 #include <sys/fs/zfs.h>
 171 #include <sys/zfs_ctldir.h>
 172 #include <sys/zfs_dir.h>
 173 #include <sys/zfs_onexit.h>
 174 #include <sys/zvol.h>
 175 #include <sys/dsl_scan.h>
 176 #include <sharefs/share.h>
 177 #include <sys/dmu_objset.h>
 178 #include <sys/dmu_send.h>
 179 #include <sys/dsl_destroy.h>
 180 #include <sys/dsl_userhold.h>
 181 #include <sys/zfeature.h>
 182 #include <sys/zio_checksum.h>

 184 #include "zfs_namecheck.h"
 185 #include "zfs_prop.h"
 186 #include "zfs_deleg.h"
 187 #include "zfs_comutil.h"

 189 extern struct modlfs zfs_modlfs;

 191 extern void zfs_init(void);
 192 extern void zfs_fini(void);
```



```
 194 ldi_ident_t zfs_li = NULL;
 195 dev_info_t *zfs_dip;

 197 uint_t zfs_fsyncer_key;
 198 extern uint_t rrw_tsd_key;
 199 static uint_t zfs_allow_log_key;

 201 typedef int zfs_ioc_legacy_func_t(zfs_cmd_t *);
 202 typedef int zfs_ioc_func_t(const char *, nvlist_t *, nvlist_t *);
 203 typedef int zfs_secpolicy_func_t(zfs_cmd_t *, nvlist_t *, cred_t *);

 205 typedef enum {
 206         NO_NAME,
 207         POOL_NAME,
 208         DATASET_NAME
 209 } zfs_ioc_namecheck_t;
_____unchanged_portion_omitted_

2312 /*
2313  * If the named property is one that has a special function to set its value,
2314  * return 0 on success and a positive error code on failure; otherwise if it is
2315  * not one of the special properties handled by this function, return -1.
2316  *
2317  * XXX: It would be better for callers of the property interface if we handled
2318  * these special cases in dsl_prop.c (in the dsl layer).
2319  */
2320 static int
2321 zfs_prop_set_special(const char *dsname, zprop_source_t source,
2322     nvpair_t *pair)
2323 {
2324         const char *propname = nvpair_name(pair);
2325         zfs_prop_t prop = zfs_name_to_prop(propname);
2326         uint64_t intval;
2327         int err;

2329         if (prop == ZPROP_INVAL) {
2330                 if (zfs_prop_userquota(propname))
2331                         return (zfs_prop_set_userquota(dsname, pair));
2332                 return (-1);
2333         }

2335         if (nvpair_type(pair) == DATA_TYPE_NVLIST) {
2336                 nvlist_t *attrs;
2337                 VERIFY(nvpair_value_nvlist(pair, &attrs) == 0);
2338                 VERIFY(nvlist_lookup_nvpair(attrs, ZPROP_VALUE,
2339                     &pair) == 0);
2340         }

2342         if (zfs_prop_get_type(prop) == PROP_TYPE_STRING)
2343                 return (-1);

2345         VERIFY(0 == nvpair_value_uint64(pair, &intval));

2347         switch (prop) {
2348         case ZFS_PROP_QUOTA:
2349                 err = dsl_dir_set_quota(dsname, source, intval);
2350                 break;
2351         case ZFS_PROP_REFQUOTA:
2352                 err = dsl_dataset_set_refquota(dsname, source, intval);
2353                 break;
2354         case ZFS_PROP_RESERVATION:
2355                 err = dsl_dir_set_reservation(dsname, source, intval);
2356                 break;
2357         case ZFS_PROP_REFRESERVATION:
2358                 err = dsl_dataset_set_refreservation(dsname, source, intval);
2359                 break;
```

```
2360         case ZFS_PROP_VOLSIZE:
2361                 err = zvol_set_volsize(dsname, intval);
2362                 break;
2363         case ZFS_PROP_VERSION:
2364         {
2365                 zfsvfs_t *zfsvfs;

2367                 if ((err = zfsvfs_hold(dsname, FTAG, &zfsvfs, B_TRUE)) != 0)
2368                         break;

2370                 err = zfs_set_version(zfsvfs, intval);
2371                 zfsvfs_rele(zfsvfs, FTAG);

2373                 if (err == 0 && intval >= ZPL_VERSION_USERSPACE) {
2374                         zfs_cmd_t *zc;

2376                         zc = kmem_zalloc(sizeof (zfs_cmd_t), KM_SLEEP);
2377                         (void) strcpy(zc->zc_name, dsname);
2378                         (void) zfs_ioc_userspace_upgrade(zc);
2379                         kmem_free(zc, sizeof (zfs_cmd_t));
2380                 }
2381                 break;
2382         }
2383         case ZFS_PROP_COMPRESSION:
2384         {
2385                 if (intval == ZIO_COMPRESS_LZ4) {
2386                         zfeature_info_t *feature =
2387                             &spa_feature_table[SPA_FEATURE_LZ4_COMPRESS];
2388                         spa_t *spa;

2390                         if ((err = spa_open(dsname, &spa, FTAG)) != 0)
2391                                 return (err);

2393                         /*
2394                          * Setting the LZ4 compression algorithm activates
2395                          * the feature.
2396                          */
2397                         if (!spa_feature_is_active(spa, feature)) {
2398                                 if ((err = zfs_prop_activate_feature(spa,
2399                                     feature)) != 0) {
2400                                         spa_close(spa, FTAG);
2401                                         return (err);
2402                                 }
2403                         }

2405                         spa_close(spa, FTAG);
2406                 }
2407                 /*
2408                  * We still want the default set action to be performed in the
2409                  * caller, we only performed zfeature settings here.
2410                  */
2411                 err = -1;
2412                 break;
2413         }
2414         case ZFS_PROP_CHECKSUM:
2415         case ZFS_PROP_DEDUP:
2416         {
2417                 zfeature_info_t *feature = NULL;
2418                 spa_t           *spa;

2420                 if (intval == ZIO_CHECKSUM_SHA512)
2421                         feature = &spa_feature_table[SPA_FEATURE_SHA512];
2422                 else if (intval == ZIO_CHECKSUM_SKEIN)
2423                         feature = &spa_feature_table[SPA_FEATURE_SKEIN];
2424                 else if (intval == ZIO_CHECKSUM_EDONR)
2425                         feature = &spa_feature_table[SPA_FEATURE_EDONR];
```



```
2426                 if (feature == NULL) {
2427                         /* No features need to be activated for this cksum */
2428                         err = -1;
2429                         break;
2430                 }
2431                 if ((err = spa_open(dsname, &spa, FTAG)) != 0)
2432                         return (err);
2433                 if (!spa_feature_is_active(spa, feature)) {
2434                         /* Salted checksums must store the salt in the MOS */
2435                         if (zio_checksum_table[intval].ci_salted)
2436                                 err = spa_activate_salted_cksum(spa, feature);
2437                         else
2438                                 err = zfs_prop_activate_feature(spa, feature);
2439                 }
2440                 spa_close(spa, FTAG);
2441                 if (err == 0) {
2442                         /*
2443                          * We want the default set action to be performed in
2444                          * the caller, we only perform feature checks here.
2445                          */
2446                         err = -1;
2447                 }
2448                 break;
2449         }

2451         default:
2452                 err = -1;
2453         }

2455         return (err);
2456 }
_____unchanged_portion_omitted_

3621 static int
3622 zfs_check_settable(const char *dsname, nvpair_t *pair, cred_t *cr)
3623 {
3624         const char *propname = nvpair_name(pair);
3625         boolean_t issnap = (strchr(dsname, '@') != NULL);
3626         zfs_prop_t prop = zfs_name_to_prop(propname);
3627         uint64_t intval;
3628         int err;

3630         if (prop == ZPROP_INVAL) {
3631                 if (zfs_prop_user(propname)) {
3632                         if (err = zfs_secpolicy_write_perms(dsname,
3633                             ZFS_DELEG_PERM_USERPROP, cr))
3634                                 return (err);
3635                         return (0);
3636                 }

3638                 if (!issnap && zfs_prop_userquota(propname)) {
3639                         const char *perm = NULL;
3640                         const char *uq_prefix =
3641                             zfs_userquota_prop_prefixes[ZFS_PROP_USERQUOTA];
3642                         const char *gq_prefix =
3643                             zfs_userquota_prop_prefixes[ZFS_PROP_GROUPQUOTA];

3645                         if (strncmp(propname, uq_prefix,
3646                             strlen(uq_prefix)) == 0) {
3647                                 perm = ZFS_DELEG_PERM_USERQUOTA;
3648                         } else if (strncmp(propname, gq_prefix,
3649                             strlen(gq_prefix)) == 0) {
3650                                 perm = ZFS_DELEG_PERM_GROUPQUOTA;
3651                         } else {
3652                                 /* USERUSED and GROUPUSED are read-only */
3653                                 return (SET_ERROR(EINVAL));
```

```
3654                        }
3655
3656                        if (err = zfs_secpolicy_write_perms(dsname, perm, cr))
3657                                return (err);
3658                        return (0);
3659                }
3660
3661                return (SET_ERROR(EINVAL));
3662        }
3663
3664        if (issnap)
3665                return (SET_ERROR(EINVAL));
3666
3667        if (nvpair_type(pair) == DATA_TYPE_NVLIST) {
3668                /*
3669                 * dsl_prop_get_all_impl() returns properties in this
3670                 * format.
3671                 */
3672                nvlist_t *attrs;
3673                VERIFY(nvpair_value_nvlist(pair, &attrs) == 0);
3674                VERIFY(nvlist_lookup_nvpair(attrs, ZPROP_VALUE,
3675                    &pair) == 0);
3676        }
3677
3678        /*
3679         * Check that this value is valid for this pool version
3680         */
3681        switch (prop) {
3682        case ZFS_PROP_COMPRESSION:
3683                /*
3684                 * If the user specified gzip compression, make sure
3685                 * the SPA supports it. We ignore any errors here since
3686                 * we'll catch them later.
3687                 */
3688                if (nvpair_type(pair) == DATA_TYPE_UINT64 &&
3689                    nvpair_value_uint64(pair, &intval) == 0) {
3690                        if (intval >= ZIO_COMPRESS_GZIP_1 &&
3691                            intval <= ZIO_COMPRESS_GZIP_9 &&
3692                            zfs_earlier_version(dsname,
3693                            SPA_VERSION_GZIP_COMPRESSION)) {
3694                                return (SET_ERROR(ENOTSUP));
3695                        }
3696
3697                        if (intval == ZIO_COMPRESS_ZLE &&
3698                            zfs_earlier_version(dsname,
3699                            SPA_VERSION_ZLE_COMPRESSION))
3700                                return (SET_ERROR(ENOTSUP));
3701
3702                        if (intval == ZIO_COMPRESS_LZ4) {
3703                                zfeature_info_t *feature =
3704                                    &spa_feature_table[
3705                                    SPA_FEATURE_LZ4_COMPRESS];
3706                                spa_t *spa;
3707
3708                                if ((err = spa_open(dsname, &spa, FTAG)) != 0)
3709                                        return (err);
3710
3711                                if (!spa_feature_is_enabled(spa, feature)) {
3712                                        spa_close(spa, FTAG);
3713                                        return (SET_ERROR(ENOTSUP));
3714                                }
3715                                spa_close(spa, FTAG);
3716                        }
3717
3718                        /*
3719                         * If this is a bootable dataset then
```



```
3720                         * verify that the compression algorithm
3721                         * is supported for booting. We must return
3722                         * something other than ENOTSUP since it
3723                         * implies a downrev pool version.
3724                         */
3725                        if (zfs_is_bootfs(dsname) &&
3726                            !BOOTFS_COMPRESS_VALID(intval)) {
3727                                return (SET_ERROR(ERANGE));
3728                        }
3729                }
3730                break;
3731
3732        case ZFS_PROP_COPIES:
3733                if (zfs_earlier_version(dsname, SPA_VERSION_DITTO_BLOCKS))
3734                        return (SET_ERROR(ENOTSUP));
3735                break;
3736
3737        case ZFS_PROP_DEDUP:
3738                if (zfs_earlier_version(dsname, SPA_VERSION_DEDUP))
3739                        return (SET_ERROR(ENOTSUP));
3740                break;
3741
3742        case ZFS_PROP_SHARESMB:
3743                if (zpl_earlier_version(dsname, ZPL_VERSION_FUID))
3744                        return (SET_ERROR(ENOTSUP));
3745                break;
3746
3747        case ZFS_PROP_ACLINHERIT:
3748                if (nvpair_type(pair) == DATA_TYPE_UINT64 &&
3749                    nvpair_value_uint64(pair, &intval) == 0) {
3750                        if (intval == ZFS_ACL_PASSTHROUGH_X &&
3751                            zfs_earlier_version(dsname,
3752                            SPA_VERSION_PASSTHROUGH_X))
3753                                return (SET_ERROR(ENOTSUP));
3754                }
3755                break;
3756
3757        case ZFS_PROP_CHECKSUM:
3758        case ZFS_PROP_DEDUP: {
3759                zfeature_info_t *feature = NULL;
3760
3761                /* dedup feature version checks */
3762                if (prop == ZFS_PROP_DEDUP &&
3763                    zfs_earlier_version(dsname, SPA_VERSION_DEDUP))
3764                        return (SET_ERROR(ENOTSUP));
3765
3766                if (nvpair_value_uint64(pair, &intval) != 0)
3767                        return (SET_ERROR(EINVAL));
3768
3769                /* check prop value is enabled in features */
3770                if (intval == ZIO_CHECKSUM_SHA512)
3771                        feature = &spa_feature_table[SPA_FEATURE_SHA512];
3772                else if (intval == ZIO_CHECKSUM_SKEIN)
3773                        feature = &spa_feature_table[SPA_FEATURE_SKEIN];
3774                else if (intval == ZIO_CHECKSUM_EDONR)
3775                        feature = &spa_feature_table[SPA_FEATURE_EDONR];
3776
3777                if (feature != NULL) {
3778                        spa_t *spa;
3779
3780                        if ((err = spa_open(dsname, &spa, FTAG)) != 0)
3781                                return (err);
3782                        /*
3783                         * Salted checksums are not supported on root pools.
3784                         */
3785                        if (spa_bootfs(spa) != 0 &&
```

```
3781                            intval < ZIO_CHECKSUM_FUNCTIONS &&
3782                            zio_checksum_table[intval].ci_salted != 0) {
3783                                spa_close(spa, FTAG);
3784                                return (SET_ERROR(ERANGE));
3785                        }
3786                        if (!spa_feature_is_enabled(spa, feature)) {
3787                                spa_close(spa, FTAG);
3788                                return (SET_ERROR(ENOTSUP));
3789                        }
3790                        spa_close(spa, FTAG);
3791                }
3792                break;
3793        }
3794        }

3796        return (zfs_secpolicy_setprop(dsname, prop, pair, CRED()));
3797 }
```

*unchanged_portion_omitted_*

```
**********************************************************
   12610 Mon Oct  7 19:17:22 2013
new/usr/src/uts/common/fs/zfs/zio_checksum.c
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2013 by Delphix. All rights reserved.
 * Copyright (c) 2013, Joyent, Inc. All rights reserved.
 * Copyright 2013 Saso Kiselkov. All rights reserved.
 */

#include <sys/zfs_context.h>
#include <sys/spa.h>
#include <sys/spa_impl.h>
#include <sys/zio.h>
#include <sys/zio_checksum.h>
#include <sys/zil.h>
#include <zfs_fletcher.h>

/*
 * Checksum vectors.
 *
 * In the SPA, everything is checksummed.  We support checksum vectors
 * for three distinct reasons:
 *
 *   1. Different kinds of data need different levels of protection.
 *      For SPA metadata, we always want a very strong checksum.
 *      For user data, we let users make the trade-off between speed
 *      and checksum strength.
 *
 *   2. Cryptographic hash and MAC algorithms are an area of active research.
 *      It is likely that in future hash functions will be at least as strong
 *      as current best-of-breed, and may be substantially faster as well.
 *      We want the ability to take advantage of these new hashes as soon as
 *      they become available.
 *
 *   3. If someone develops hardware that can compute a strong hash quickly,
 *      we want the ability to take advantage of that hardware.
 *
 * Of course, we don't want a checksum upgrade to invalidate existing
 * data, so we store the checksum *function* in eight bits of the bp.
 * This gives us room for up to 256 different checksum functions.
 *
 * When writing a block, we always checksum it with the latest-and-greatest
 * checksum function of the appropriate strength.  When reading a block,
```



```
 * we compare the expected checksum against the actual checksum, which we
 * compute via the checksum function specified by BP_GET_CHECKSUM(bp).
 *
 * SALTED CHECKSUMS
 *
 * To enable the use of non-cryptographically secure hash algorithms in
 * dedup we introduce the notion of salted checksums (MACs, really). A salted
 * checksum is fed both a random 256-bit value (the salt) and the data to be
 * checksummed. This salt is kept secret (stored on the pool, but never shown
 * to the user), thus even if an attacker knew of collision weaknesses in the
 * hash algorithm, they won't be able to mount a known plaintext attack on
 * the DDT, since the actual hash value cannot be known ahead of time. How
 * the salt is used is algorithm-specific (some might simply prefix it to the
 * data block, others might need to utilize a full-blown HMAC). On disk the
 * salt is stored in a ZAP object in the MOS (DMU_POOL_CHECKSUM_SALT).
 *
 * CONTEXT TEMPLATES
 *
 * Some hashing algorithms need to perform a substantial amount of
 * initialization work (e.g. salted checksums above may need to pre-hash the
 * salt) before being able to process data. Performing this redundant work
 * for each block would be very wasteful, so we instead allow a checksum
 * algorithm to do the work once (the first time it's used) and then keep
 * this pre-initialized context as a template inside the spa_t
 * (spa_cksum_tmpls). If the zio_checksum_info_t contains non-NULL
 * ci_tmpl_init and ci_tmpl_free callbacks, they are used to construct and
 * destruct the pre-initialized checksum context. The pre-initialized
 * context is then reused during each checksum invocation and passed to the
 * checksum function.
 */

/*ARGSUSED*/
static void
zio_checksum_off(const void *buf, uint64_t size, const zio_cksum_salt_t *salt,
    const void *ctx_template, zio_cksum_t *zcp)
zio_checksum_off(const void *buf, uint64_t size, zio_cksum_t *zcp)
{
        ZIO_SET_CHECKSUM(zcp, 0, 0, 0, 0);
}

zio_checksum_info_t zio_checksum_table[ZIO_CHECKSUM_FUNCTIONS] = {
        {{NULL, NULL}, NULL, NULL, 0, 0, 0, 0, "inherit"},
        {{NULL, NULL}, NULL, NULL, 0, 0, 0, 0, "on"},
        {{zio_checksum_off,             zio_checksum_off},
            NULL, NULL, 0, 0, 0, 0, "off"},
        {{zio_checksum_SHA256,          zio_checksum_SHA256},
            NULL, NULL, 1, 1, 0, 0, "label"},
        {{zio_checksum_SHA256,          zio_checksum_SHA256},
            NULL, NULL, 1, 1, 0, 0, "gang_header"},
        {{fletcher_2_native,            fletcher_2_byteswap},
            NULL, NULL, 0, 1, 0, 0, "zilog"},
        {{fletcher_2_native,            fletcher_2_byteswap},
            NULL, NULL, 0, 0, 0, 0, "fletcher2"},
        {{fletcher_4_native,            fletcher_4_byteswap},
            NULL, NULL, 1, 0, 0, 0, "fletcher4"},
        {{zio_checksum_SHA256,          zio_checksum_SHA256},
            NULL, NULL, 1, 0, 1, 0, "sha256"},
        {{fletcher_4_native,            fletcher_4_byteswap},
            NULL, NULL, 0, 1, 0, 0, "zilog2"},
        {{zio_checksum_off,             zio_checksum_off},
            NULL, NULL, 0, 0, 0, 0, "noparity"},
        {{zio_checksum_SHA512_native, zio_checksum_SHA512_byteswap},
            NULL, NULL, 1, 0, 1, 0, "sha512"},
        {{zio_checksum_skein_native,    zio_checksum_skein_byteswap},
            zio_checksum_skein_tmpl_init, zio_checksum_skein_tmpl_free,
            1, 0, 1, 1, "skein"},
```

```
        {{zio_checksum_edonr_native,    zio_checksum_edonr_byteswap},
            zio_checksum_edonr_tmpl_init, zio_checksum_edonr_tmpl_free,
            1, 0, 1, 1, "edonr"}
        {{NULL,                 NULL},                  0, 0, 0, "inherit"},
        {{NULL,                 NULL},                  0, 0, 0, "on"},
        {{zio_checksum_off,     zio_checksum_off},      0, 0, 0, "off"},
        {{zio_checksum_SHA256,  zio_checksum_SHA256},   1, 1, 0, "label"},
        {{zio_checksum_SHA256,  zio_checksum_SHA256},   1, 1, 0, "gang_header"},
        {{fletcher_2_native,    fletcher_2_byteswap},   0, 1, 0, "zilog"},
        {{fletcher_2_native,    fletcher_2_byteswap},   0, 0, 0, "fletcher2"},
        {{fletcher_4_native,    fletcher_4_byteswap},   1, 0, 0, "fletcher4"},
        {{zio_checksum_SHA256,  zio_checksum_SHA256},   1, 0, 1, "sha256"},
        {{fletcher_4_native,    fletcher_4_byteswap},   0, 1, 0, "zilog2"},
        {{zio_checksum_off,     zio_checksum_off},      0, 0, 0, "noparity"},
};
```

*unchanged_portion_omitted_*

```
/*
 * Calls the template init function of a checksum which supports context
 * templates and installs the template into the spa_t.
 */
static void
zio_checksum_template_init(enum zio_checksum checksum, spa_t *spa)
{
        zio_checksum_info_t *ci = &zio_checksum_table[checksum];

        VERIFY(ci->ci_tmpl_init != NULL && ci->ci_tmpl_free != NULL);
        mutex_enter(&spa->spa_cksum_tmpls_lock);
        if (spa->spa_cksum_tmpls[checksum] == NULL) {
                spa->spa_cksum_tmpls[checksum] =
                    ci->ci_tmpl_init(&spa->spa_cksum_salt);
                VERIFY(spa->spa_cksum_tmpls[checksum] != NULL);
        }
        mutex_exit(&spa->spa_cksum_tmpls_lock);
}

/*
 * Generate the checksum.
 */
void
zio_checksum_compute(zio_t *zio, enum zio_checksum checksum,
        void *data, uint64_t size)
{
        blkptr_t *bp = zio->io_bp;
        uint64_t offset = zio->io_offset;
        zio_checksum_info_t *ci = &zio_checksum_table[checksum];
        zio_cksum_t cksum;
        spa_t *spa = zio->io_spa;

        ASSERT((uint_t)checksum < ZIO_CHECKSUM_FUNCTIONS);
        ASSERT(ci->ci_func[0] != NULL);

        if (ci->ci_tmpl_init != NULL && spa->spa_cksum_tmpls[checksum] == NULL)
                zio_checksum_template_init(checksum, spa);

        if (ci->ci_eck) {
                zio_eck_t *eck;

                if (checksum == ZIO_CHECKSUM_ZILOG2) {
                        zil_chain_t *zilc = data;

                        size = P2ROUNDUP_TYPED(zilc->zc_nused, ZIL_MIN_BLKSZ,
                            uint64_t);
                        eck = &zilc->zc_eck;
                } else {
                        eck = (zio_eck_t *)((char *)data + size) - 1;
```



```
                }
                if (checksum == ZIO_CHECKSUM_GANG_HEADER)
                        zio_checksum_gang_verifier(&eck->zec_cksum, bp);
                else if (checksum == ZIO_CHECKSUM_LABEL)
                        zio_checksum_label_verifier(&eck->zec_cksum, offset);
                else
                        bp->blk_cksum = eck->zec_cksum;
                eck->zec_magic = ZEC_MAGIC;
                ci->ci_func[0](data, size, &spa->spa_cksum_salt,
                    spa->spa_cksum_tmpls[checksum], &cksum);
                ci->ci_func[0](data, size, &cksum);
                eck->zec_cksum = cksum;
        } else {
                ci->ci_func[0](data, size, &spa->spa_cksum_salt,
                    spa->spa_cksum_tmpls[checksum], &bp->blk_cksum);
                ci->ci_func[0](data, size, &bp->blk_cksum);
        }
}

int
zio_checksum_error(zio_t *zio, zio_bad_cksum_t *info)
{
        blkptr_t *bp = zio->io_bp;
        uint_t checksum = (bp == NULL ? zio->io_prop.zp_checksum :
            (BP_IS_GANG(bp) ? ZIO_CHECKSUM_GANG_HEADER : BP_GET_CHECKSUM(bp)));
        int byteswap;
        int error;
        uint64_t size = (bp == NULL ? zio->io_size :
            (BP_IS_GANG(bp) ? SPA_GANGBLOCKSIZE : BP_GET_PSIZE(bp)));
        uint64_t offset = zio->io_offset;
        void *data = zio->io_data;
        zio_checksum_info_t *ci = &zio_checksum_table[checksum];
        zio_cksum_t actual_cksum, expected_cksum, verifier;
        spa_t *spa = zio->io_spa;

        if (checksum >= ZIO_CHECKSUM_FUNCTIONS || ci->ci_func[0] == NULL)
                return (SET_ERROR(EINVAL));

        if (ci->ci_tmpl_init != NULL && spa->spa_cksum_tmpls[checksum] == NULL)
                zio_checksum_template_init(checksum, spa);

        if (ci->ci_eck) {
                zio_eck_t *eck;

                if (checksum == ZIO_CHECKSUM_ZILOG2) {
                        zil_chain_t *zilc = data;
                        uint64_t nused;

                        eck = &zilc->zc_eck;
                        if (eck->zec_magic == ZEC_MAGIC)
                                nused = zilc->zc_nused;
                        else if (eck->zec_magic == BSWAP_64(ZEC_MAGIC))
                                nused = BSWAP_64(zilc->zc_nused);
                        else
                                return (SET_ERROR(ECKSUM));

                        if (nused > size)
                                return (SET_ERROR(ECKSUM));

                        size = P2ROUNDUP_TYPED(nused, ZIL_MIN_BLKSZ, uint64_t);
                } else {
                        eck = (zio_eck_t *)((char *)data + size) - 1;
                }

                if (checksum == ZIO_CHECKSUM_GANG_HEADER)
                        zio_checksum_gang_verifier(&verifier, bp);
```

```
310                 else if (checksum == ZIO_CHECKSUM_LABEL)
311                         zio_checksum_label_verifier(&verifier, offset);
312                 else
313                         verifier = bp->blk_cksum;

315                 byteswap = (eck->zec_magic == BSWAP_64(ZEC_MAGIC));

317                 if (byteswap)
318                         byteswap_uint64_array(&verifier, sizeof (zio_cksum_t));

320                 expected_cksum = eck->zec_cksum;
321                 eck->zec_cksum = verifier;
322                 ci->ci_func[byteswap](data, size, &spa->spa_cksum_salt,
323                     spa->spa_cksum_tmpls[checksum], &actual_cksum);
246                 ci->ci_func[byteswap](data, size, &actual_cksum);
324                 eck->zec_cksum = expected_cksum;

326                 if (byteswap)
327                         byteswap_uint64_array(&expected_cksum,
328                             sizeof (zio_cksum_t));
329         } else {
330                 ASSERT(!BP_IS_GANG(bp));
331                 byteswap = BP_SHOULD_BYTESWAP(bp);
332                 expected_cksum = bp->blk_cksum;
333                 ci->ci_func[byteswap](data, size, &spa->spa_cksum_salt,
334                     spa->spa_cksum_tmpls[checksum], &actual_cksum);
256                 ci->ci_func[byteswap](data, size, &actual_cksum);
335         }

337         info->zbc_expected = expected_cksum;
338         info->zbc_actual = actual_cksum;
339         info->zbc_checksum_name = ci->ci_name;
340         info->zbc_byteswapped = byteswap;
341         info->zbc_injected = 0;
342         info->zbc_has_cksum = 1;

344         if (!ZIO_CHECKSUM_EQUAL(actual_cksum, expected_cksum))
345                 return (SET_ERROR(ECKSUM));

347         if (zio_injection_enabled && !zio->io_error &&
348             (error = zio_handle_fault_injection(zio, ECKSUM)) != 0) {

350                 info->zbc_injected = 1;
351                 return (error);
352         }

354         return (0);
355 }

357 /*
358  * Called by a spa_t that's about to be deallocated. This steps through
359  * all of the checksum context templates and deallocates any that were
360  * initialized using the algorithm-specific template init function.
361  */
362 void
363 zio_checksum_templates_free(spa_t *spa)
364 {
365         for (int checksum = 0; checksum < ZIO_CHECKSUM_FUNCTIONS; checksum++) {
366                 if (spa->spa_cksum_tmpls[checksum] != NULL) {
367                         zio_checksum_info_t *ci = &zio_checksum_table[checksum];

369                         VERIFY(ci->ci_tmpl_free != NULL);
370                         ci->ci_tmpl_free(spa->spa_cksum_tmpls[checksum]);
371                         spa->spa_cksum_tmpls[checksum] = NULL;
372                 }
373         }
```



```
374 }
```

*unchanged_portion_omitted_*

```
**********************************************************
   22521 Mon Oct  7 19:17:22 2013
new/usr/src/uts/common/sys/Makefile
4185 New hash algorithm support
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2013, Joyent, Inc. All rights reserved.
# Copyright 2013 Saso Kiselkov. All rights reserved.
#

include $(SRC)/uts/Makefile.uts

FILEMODE=644

#
#       Note that the following headers are present in the kernel but
#       neither installed or shipped as part of the product:
#               cpuid_drv.h:            Private interface for cpuid consumers
#               unix_bb_info.h:         Private interface to kcov
#

i386_HDRS=                      \
        agp/agpamd64gart_io.h   \
        agp/agpdefs.h           \
        agp/agpgart_impl.h      \
        agp/agpmaster_io.h      \
        agp/agptarget_io.h      \
        agpgart.h               \
        asy.h                   \
        fd_debug.h              \
        fdc.h                   \
        fdmedia.h               \
        mouse.h                 \
        ucode.h

sparc_HDRS=                     \
        mouse.h                 \
        scsi/targets/ssddef.h   \
        $(MDESCHDRS)

# Generated headers
GENHDRS=                        \
        priv_const.h            \
        priv_names.h            \
        usb/usbdevs.h
```



```
CHKHDRS=                        \
        acpi_drv.h              \
        acct.h                  \
        acctctl.h               \
        acl.h                   \
        acl_impl.h              \
        aggr.h                  \
        aggr_impl.h             \
        aio.h                   \
        aio_impl.h              \
        aio_req.h               \
        aiocb.h                 \
        ascii.h                 \
        asynch.h                \
        atomic.h                \
        attr.h                  \
        audio.h                 \
        audioio.h               \
        autoconf.h              \
        auxv.h                  \
        auxv_386.h              \
        auxv_SPARC.h            \
        avl.h                   \
        avl_impl.h              \
        bitmap.h                \
        bitset.h                \
        bl.h                    \
        blkdev.h                \
        bofi.h                  \
        bofi_impl.h             \
        bpp_io.h                \
        bootstat.h              \
        brand.h                 \
        buf.h                   \
        bufmod.h                \
        bustypes.h              \
        byteorder.h             \
        callb.h                 \
        callo.h                 \
        cap_util.h              \
        cpucaps.h               \
        cpucaps_impl.h          \
        ccompile.h              \
        cdio.h                  \
        cladm.h                 \
        class.h                 \
        clconf.h                \
        clock_impl.h            \
        cmlb.h                  \
        cmn_err.h               \
        compress.h              \
        condvar.h               \
        condvar_impl.h          \
        conf.h                  \
        consdev.h               \
        console.h               \
        consplat.h              \
        vt.h                    \
        vtdaemon.h              \
        kd.h                    \
        contract.h              \
        contract_impl.h         \
        copyops.h               \
        core.h                  \
        corectl.h               \
```

```
128         cpc_impl.h              \
129         cpc_pcbe.h              \
130         cpr.h                   \
131         cpupart.h               \
132         cpuvar.h                \
133         crc32.h                 \
134         cred.h                  \
135         cred_impl.h             \
136         crtctl.h                \
137         cryptmod.h              \
138         csiioctl.h              \
139         ctf.h                   \
140         ctfs.h                  \
141         ctfs_impl.h             \
142         ctf_api.h               \
143         ctype.h                 \
144         cyclic.h                \
145         cyclic_impl.h           \
146         dacf.h                  \
147         dacf_impl.h             \
148         damap.h                 \
149         damap_impl.h            \
150         dc_ki.h                 \
151         ddi.h                   \
152         ddifm.h                 \
153         ddifm_impl.h            \
154         ddi_hp.h                \
155         ddi_hp_impl.h           \
156         ddi_intr.h              \
157         ddi_intr_impl.h         \
158         ddi_impldefs.h          \
159         ddi_implfuncs.h         \
160         ddi_obsolete.h          \
161         ddi_periodic.h          \
162         ddidevmap.h             \
163         ddidmareq.h             \
164         ddimapreq.h             \
165         ddipropdefs.h           \
166         dditypes.h              \
167         debug.h                 \
168         des.h                   \
169         devctl.h                \
170         devcache.h              \
171         devcache_impl.h         \
172         devfm.h                 \
173         devid_cache.h           \
174         devinfo_impl.h          \
175         devops.h                \
176         devpolicy.h             \
177         devpoll.h               \
178         dirent.h                \
179         disp.h                  \
180         dkbad.h                 \
181         dkio.h                  \
182         dklabel.h               \
183         dl.h                    \
184         dlpi.h                  \
185         dld.h                   \
186         dld_impl.h              \
187         dld_ioc.h               \
188         dls.h                   \
189         dls_mgmt.h              \
190         dls_impl.h              \
191         dma_i8237A.h            \
192         dnlc.h                  \
193         door.h                  \
```



```
194         door_data.h             \
195         door_impl.h             \
196         dtrace.h                \
197         dtrace_impl.h           \
198         dumpadm.h               \
199         dumphdr.h               \
200         ecppsys.h               \
201         ecppio.h                \
202         ecppreg.h               \
203         ecppvar.h               \
204         edonr.h                 \
205         efi_partition.h         \
206         elf.h                   \
207         elf_386.h               \
208         elf_SPARC.h             \
209         elf_notes.h             \
210         elf_amd64.h             \
211         elftypes.h              \
212         emul64.h                \
213         emul64cmd.h             \
214         emul64var.h             \
215         epm.h                   \
216         errno.h                 \
217         errorq.h                \
218         errorq_impl.h           \
219         esunddi.h               \
220         ethernet.h              \
221         euc.h                   \
222         eucioctl.h              \
223         exacct.h                \
224         exacct_catalog.h        \
225         exacct_impl.h           \
226         exec.h                  \
227         exechdr.h               \
228         extdirent.h             \
229         fault.h                 \
230         fasttrap.h              \
231         fasttrap_impl.h         \
232         fbio.h                  \
233         fbuf.h                  \
234         fcntl.h                 \
235         fct.h                   \
236         fct_defines.h           \
237         fctio.h                 \
238         fdbuffer.h              \
239         fdio.h                  \
240         feature_tests.h         \
241         fem.h                   \
242         file.h                  \
243         filio.h                 \
244         flock.h                 \
245         flock_impl.h            \
246         fork.h                  \
247         fss.h                   \
248         fsspriocntl.h           \
249         fsid.h                  \
250         fssnap.h                \
251         fssnap_if.h             \
252         fstyp.h                 \
253         ftrace.h                \
254         fx.h                    \
255         fxpriocntl.h            \
256         gfs.h                   \
257         gld.h                   \
258         gldpriv.h               \
259         group.h                 \
```

```
260         hdio.h                  \
261         hook.h                  \
262         hook_event.h            \
263         hook_impl.h             \
264         hwconf.h                \
265         ia.h                    \
266         iapriocntl.h            \
267         ibpart.h                \
268         id32.h                  \
269         idmap.h                 \
270         ieeefp.h                \
271         id_space.h              \
272         instance.h              \
273         int_const.h             \
274         int_fmtio.h             \
275         int_limits.h            \
276         int_types.h             \
277         inttypes.h              \
278         ioccom.h                \
279         ioctl.h                 \
280         ipc.h                   \
281         ipc_impl.h              \
282         ipc_rctl.h              \
283         ipmi.h                  \
284         isa_defs.h              \
285         iscsi_authclient.h      \
286         iscsi_authclientglue.h  \
287         iscsi_protocol.h        \
288         jioctl.h                \
289         kbd.h                   \
290         kbdreg.h                \
291         kbio.h                  \
292         kcpc.h                  \
293         kdi.h                   \
294         kdi_impl.h              \
295         kiconv.h                \
296         kiconv_big5_utf8.h      \
297         kiconv_cck_common.h     \
298         kiconv_cp950hkscs_utf8.h        \
299         kiconv_emea1.h          \
300         kiconv_emea2.h          \
301         kiconv_euckr_utf8.h     \
302         kiconv_euctw_utf8.h     \
303         kiconv_gb18030_utf8.h   \
304         kiconv_gb2312_utf8.h    \
305         kiconv_hkscs_utf8.h     \
306         kiconv_ja.h             \
307         kiconv_ja_jis_to_unicode.h      \
308         kiconv_ja_unicode_to_jis.h      \
309         kiconv_ko.h             \
310         kiconv_latin1.h         \
311         kiconv_sc.h             \
312         kiconv_tc.h             \
313         kiconv_uhc_utf8.h       \
314         kiconv_utf8_big5.h      \
315         kiconv_utf8_cp950hkscs.h        \
316         kiconv_utf8_euckr.h     \
317         kiconv_utf8_euctw.h     \
318         kiconv_utf8_gb18030.h   \
319         kiconv_utf8_gb2312.h    \
320         kiconv_utf8_hkscs.h     \
321         kiconv_utf8_uhc.h       \
322         kidmap.h                \
323         klpd.h                  \
324         klwp.h                  \
325         kmdb.h                  \
```



```
326         kmem.h                  \
327         kmem_impl.h             \
328         kobj.h                  \
329         kobj_impl.h             \
330         ksocket.h               \
331         kstat.h                 \
332         kstr.h                  \
333         ksyms.h                 \
334         ksynch.h                \
335         ldterm.h                \
336         lgrp.h                  \
337         lgrp_user.h             \
338         libc_kernel.h           \
339         link.h                  \
340         list.h                  \
341         list_impl.h             \
342         llc1.h                  \
343         loadavg.h               \
344         lock.h                  \
345         lockfs.h                \
346         lockstat.h              \
347         lofi.h                  \
348         log.h                   \
349         logindmux.h             \
350         logindmux_impl.h        \
351         lwp.h                   \
352         lwp_timer_impl.h        \
353         lwp_upimutex_impl.h     \
354         lpif.h                  \
355         mac.h                   \
356         mac_client.h            \
357         mac_client_impl.h       \
358         mac_ether.h             \
359         mac_flow.h              \
360         mac_flow_impl.h         \
361         mac_impl.h              \
362         mac_provider.h          \
363         mac_soft_ring.h         \
364         mac_stat.h              \
365         machelf.h               \
366         map.h                   \
367         md4.h                   \
368         md5.h                   \
369         md5_consts.h            \
370         mdi_impldefs.h          \
371         mem.h                   \
372         mem_config.h            \
373         memlist.h               \
374         mkdev.h                 \
375         mhd.h                   \
376         mii.h                   \
377         miiregs.h               \
378         mixer.h                 \
379         mman.h                  \
380         mmapobj.h               \
381         mntent.h                \
382         mntio.h                 \
383         mnttab.h                \
384         modctl.h                \
385         mode.h                  \
386         model.h                 \
387         modhash.h               \
388         modhash_impl.h          \
389         mount.h                 \
390         mouse.h                 \
391         msacct.h                \
```

```
392         msg.h                   \
393         msg_impl.h              \
394         msio.h                  \
395         msreg.h                 \
396         mtio.h                  \
397         multidata.h             \
398         multidata_impl.h        \
399         mutex.h                 \
400         nbmlock.h               \
401         ndifm.h                 \
402         ndi_impldefs.h          \
403         net80211.h              \
404         net80211_crypto.h       \
405         net80211_ht.h           \
406         net80211_proto.h        \
407         netconfig.h             \
408         neti.h                  \
409         netstack.h              \
410         nexusdefs.h             \
411         note.h                  \
412         nvpair.h                \
413         nvpair_impl.h           \
414         objfs.h                 \
415         objfs_impl.h            \
416         ontrap.h                \
417         open.h                  \
418         openpromio.h            \
419         panic.h                 \
420         param.h                 \
421         pathconf.h              \
422         pathname.h              \
423         pattr.h                 \
424         queue.h                 \
425         serializer.h            \
426         pbio.h                  \
427         pccard.h                \
428         pci.h                   \
429         pcie.h                  \
430         pci_impl.h              \
431         pci_tools.h             \
432         pcmcia.h                \
433         pctypes.h               \
434         pfmod.h                 \
435         pg.h                    \
436         pghw.h                  \
437         physmem.h               \
438         pkp_hash.h              \
439         pm.h                    \
440         policy.h                \
441         poll.h                  \
442         poll_impl.h             \
443         pool.h                  \
444         pool_impl.h             \
445         pool_pset.h             \
446         port.h                  \
447         port_impl.h             \
448         port_kernel.h           \
449         portif.h                \
450         ppmio.h                 \
451         pppt_ic_if.h            \
452         pppt_ioctl.h            \
453         priocntl.h              \
454         priv.h                  \
455         priv_impl.h             \
456         prnio.h                 \
457         proc.h                  \
```



```
458         processor.h             \
459         procfs.h                \
460         procset.h               \
461         project.h               \
462         protosw.h               \
463         prsystm.h               \
464         pset.h                  \
465         pshot.h                 \
466         ptem.h                  \
467         ptms.h                  \
468         ptyvar.h                \
469         raidioctl.h             \
470         ramdisk.h               \
471         random.h                \
472         rctl.h                  \
473         rctl_impl.h             \
474         rds.h                   \
475         reboot.h                \
476         refstr.h                \
477         refstr_impl.h           \
478         resource.h              \
479         rlioctl.h               \
480         rt.h                    \
481         rtpriocntl.h            \
482         rwlock.h                \
483         rwlock_impl.h           \
484         rwstlock.h              \
485         sad.h                   \
486         schedctl.h              \
487         sdt.h                   \
488         select.h                \
489         sem.h                   \
490         sem_impl.h              \
491         sema_impl.h             \
492         semaphore.h             \
493         sendfile.h              \
494         ser_sync.h              \
495         session.h               \
496         sha1.h                  \
497         sha1_consts.h           \
498         sha2.h                  \
499         sha2_consts.h           \
500         share.h                 \
501         shm.h                   \
502         shm_impl.h              \
503         sid.h                   \
504         siginfo.h               \
505         signal.h                \
506         skein.h                 \
507         sleepq.h                \
508         smbios.h                \
509         smbios_impl.h           \
510         sobject.h               \
511         socket.h                \
512         socket_impl.h           \
513         socket_proto.h          \
514         socketvar.h             \
515         sockfilter.h            \
516         sockio.h                \
517         soundcard.h             \
518         squeue.h                \
519         squeue_impl.h           \
520         srn.h                   \
521         sservice.h              \
522         stat.h                  \
523         statfs.h                \
```

```
524         statvfs.h                   \
525         stdbool.h                   \
526         stdint.h                    \
527         stermio.h                   \
528         stmf.h                      \
529         stmf_defines.h              \
530         stmf_ioctl.h                \
531         stmf_sbd_ioctl.h            \
532         stream.h                    \
533         strft.h                     \
534         strlog.h                    \
535         strmdep.h                   \
536         stropts.h                   \
537         strredir.h                  \
538         strstat.h                   \
539         strsubr.h                   \
540         strsun.h                    \
541         strtty.h                    \
542         sunddi.h                    \
543         sunldi.h                    \
544         sunldi_impl.h               \
545         sunmdi.h                    \
546         sunndi.h                    \
547         sunos_dhcp_class.h          \
548         sunpm.h                     \
549         suntpi.h                    \
550         suntty.h                    \
551         swap.h                      \
552         synch.h                     \
553         sysdc.h                     \
554         sysdc_impl.h                \
555         syscall.h                   \
556         sysconf.h                   \
557         sysconfig.h                 \
558         sysevent.h                  \
559         sysevent_impl.h             \
560         sysinfo.h                   \
561         syslog.h                    \
562         sysmacros.h                 \
563         sysmsg_impl.h               \
564         systeminfo.h                \
565         systm.h                     \
566         task.h                      \
567         taskq.h                     \
568         taskq_impl.h                \
569         t_kuser.h                   \
570         t_lock.h                    \
571         telioctl.h                  \
572         termio.h                    \
573         termios.h                   \
574         termiox.h                   \
575         thread.h                    \
576         ticlts.h                    \
577         ticots.h                    \
578         ticotsord.h                 \
579         tihdr.h                     \
580         time.h                      \
581         time_impl.h                 \
582         time_std_impl.h             \
583         timeb.h                     \
584         timer.h                     \
585         times.h                     \
586         timex.h                     \
587         timod.h                     \
588         tirdwr.h                    \
589         tiuser.h                    \
```



```
590         tl.h                        \
591         tnf.h                       \
592         tnf_com.h                   \
593         tnf_probe.h                 \
594         tnf_writer.h                \
595         todio.h                     \
596         tpicommon.h                 \
597         ts.h                        \
598         tspriocntl.h                \
599         ttcompat.h                  \
600         ttold.h                     \
601         tty.h                       \
602         ttychars.h                  \
603         ttydev.h                    \
604         tuneable.h                  \
605         turnstile.h                 \
606         types.h                     \
607         types32.h                   \
608         tzfile.h                    \
609         u8_textprep.h               \
610         u8_textprep_data.h          \
611         uadmin.h                    \
612         ucred.h                     \
613         uio.h                       \
614         ulimit.h                    \
615         un.h                        \
616         unistd.h                    \
617         user.h                      \
618         ustat.h                     \
619         utime.h                     \
620         utsname.h                   \
621         utssys.h                    \
622         uuid.h                      \
623         va_impl.h                   \
624         va_list.h                   \
625         var.h                       \
626         varargs.h                   \
627         vfs.h                       \
628         vfs_opreg.h                 \
629         vfstab.h                    \
630         vgareg.h                    \
631         videodev2.h                 \
632         visual_io.h                 \
633         vlan.h                      \
634         vm.h                        \
635         vm_usage.h                  \
636         vmem.h                      \
637         vmem_impl.h                 \
638         vmsystm.h                   \
639         vnic.h                      \
640         vnic_impl.h                 \
641         vnode.h                     \
642         vscan.h                     \
643         vtoc.h                      \
644         vtrace.h                    \
645         vuid_event.h                \
646         vuid_wheel.h                \
647         vuid_queue.h                \
648         vuid_state.h                \
649         vuid_store.h                \
650         wait.h                      \
651         waitq.h                     \
652         wanboot_impl.h              \
653         watchpoint.h                \
654         winlockio.h                 \
655         zcons.h                     \
```

```
656         zone.h                  \
657         xti_inet.h              \
658         xti_osi.h               \
659         xti_xtiopt.h            \
660         zmod.h

662 HDRS=                           \
663         $(GENHDRS)              \
664         $(CHKHDRS)

666 AUDIOHDRS=                      \
667         ac97.h                  \
668         audio_common.h          \
669         audio_driver.h          \
670         audio_oss.h             \
671         g711.h

673 AVHDRS=                        \
674         iec61883.h

676 BSCHDRS=                       \
677         bscbus.h                \
678         bscv_impl.h             \
679         lom_ebuscodes.h         \
680         lom_io.h                \
681         lom_priv.h              \
682         lombus.h

684 MDESCHDRS=                     \
685         mdesc.h                 \
686         mdesc_impl.h

688 CPUDRVHDRS=                    \
689         cpudrv.h

691 CRYPTOHDRS=                    \
692         elfsign.h               \
693         ioctl.h                 \
694         ioctladmin.h            \
695         common.h                \
696         impl.h                  \
697         spi.h                   \
698         api.h                   \
699         ops_impl.h              \
700         sched_impl.h

702 DCAMHDRS=                      \
703         dcam1394_io.h

705 IBHDRS=                \
706         ib_types.h      \
707         ib_pkt_hdrs.h

709 IBTLHDRS=               \
710         ibtl_types.h    \
711         ibtl_status.h   \
712         ibti.h          \
713         ibti_cm.h       \
714         ibci.h          \
715         ibti_common.h   \
716         ibvti.h         \
717         ibtl_ci_types.h

719 IBTLIMPLHDRS=           \
720         ibtl_util.h
```



```
722 IBNEXHDRS=              \
723         ibnex_devctl.h

725 IBMFHDRS=               \
726         ibmf.h          \
727         ibmf_msg.h      \
728         ibmf_saa.h      \
729         ibmf_utils.h

731 IBMGTHDRS=              \
732         ib_dm_attr.h    \
733         ib_mad.h        \
734         sm_attr.h       \
735         sa_recs.h

737 IBDHDRS=                \
738         ibd.h

740 OFHDRS=                 \
741         ofa_solaris.h   \
742         ofed_kernel.h

744 RDMAHDRS=               \
745         ib_addr.h       \
746         ib_user_mad.h   \
747         ib_user_sa.h    \
748         ib_user_verbs.h \
749         ib_verbs.h      \
750         rdma_cm.h       \
751         rdma_user_cm.h

753 SOL_UVERBSHDRS=         \
754         sol_uverbs.h    \
755         sol_uverbs2ucma.h       \
756         sol_uverbs_comp.h       \
757         sol_uverbs_hca.h        \
758         sol_uverbs_qp.h         \
759         sol_uverbs_event.h

761 SOL_UMADHDRS=           \
762         sol_umad.h

764 SOL_UCMAHDRS=           \
765         sol_ucma.h      \
766         sol_rdma_user_cm.h

768 SOL_OFSHDRS=            \
769         sol_cma.h       \
770         sol_ib_cma.h    \
771         sol_ofs_common.h        \
772         sol_kverb_impl.h

774 TAVORHDRS=              \
775         tavor_ioctl.h

777 HERMONHDRS=             \
778         hermon_ioctl.h

780 MLNXHDRS=              \
781         mlnx_umap.h

783 IDMHDRS=                \
784         idm.h           \
785         idm_impl.h      \
786         idm_so.h        \
787         idm_text.h      \
```

```
788         idm_transport.h \
789         idm_conn_sm.h

791 ISCSITHDRS=             \
792         radius_packet.h         \
793         radius_protocol.h       \
794         chap.h                  \
795         isns_protocol.h         \
796         iscsi_if.h              \
797         iscsit_common.h

799 ISOHDRS=                       \
800         signal_iso.h

802 DERIVED_LVMHDRS=        \
803         md_mdiox.h              \
804         md_basic.h              \
805         mdmed.h                 \
806         md_mhdx.h               \
807         mdmn_commd.h

809 LVMHDRS=                       \
810         md_convert.h            \
811         md_crc.h                \
812         md_hotspares.h          \
813         md_mddb.h               \
814         md_mirror.h             \
815         md_mirror_shared.h      \
816         md_names.h              \
817         md_notify.h             \
818         md_raid.h               \
819         md_rename.h             \
820         md_sp.h                 \
821         md_stripe.h             \
822         md_trans.h              \
823         mdio.h                  \
824         mdvar.h

826 ALL_LVMHDRS=                   \
827         $(LVMHDRS)              \
828         $(DERIVED_LVMHDRS)

830 FMHDRS=                        \
831         protocol.h              \
832         util.h

834 FMFSHDRS=                      \
835         zfs.h

837 FMIOHDRS=                      \
838         ddi.h                   \
839         disk.h                  \
840         pci.h                   \
841         scsi.h                  \
842         sun4upci.h              \
843         opl_mc_fm.h

845 FSHDRS=                        \
846         autofs.h                \
847         cachefs_dir.h           \
848         cachefs_dlog.h          \
849         cachefs_filegrp.h       \
850         cachefs_fs.h            \
851         cachefs_fscache.h       \
852         cachefs_ioctl.h         \
853         cachefs_log.h           \
```



```
854         decomp.h                \
855         dv_node.h               \
856         sdev_impl.h             \
857         fifonode.h              \
858         hsfs_isospec.h          \
859         hsfs_node.h             \
860         hsfs_rrip.h             \
861         hsfs_spec.h             \
862         hsfs_susp.h             \
863         lofs_info.h             \
864         lofs_node.h             \
865         mntdata.h               \
866         namenode.h              \
867         pc_dir.h                \
868         pc_fs.h                 \
869         pc_label.h              \
870         pc_node.h               \
871         pxfs_ki.h               \
872         snode.h                 \
873         swapnode.h              \
874         tmp.h                   \
875         tmpnode.h               \
876         udf_inode.h             \
877         udf_volume.h            \
878         ufs_acl.h               \
879         ufs_bio.h               \
880         ufs_filio.h             \
881         ufs_fs.h                \
882         ufs_fsdir.h             \
883         ufs_inode.h             \
884         ufs_lockfs.h            \
885         ufs_log.h               \
886         ufs_mount.h             \
887         ufs_panic.h             \
888         ufs_prot.h              \
889         ufs_quota.h             \
890         ufs_snap.h              \
891         ufs_trans.h             \
892         zfs.h                   \
893         zut.h

895 PCMCIAHDRS=            \
896         pcata.h         \
897         pcser_conf.h    \
898         pcser_io.h      \
899         pcser_reg.h     \
900         pcser_manuspec.h \
901         pcser_var.h

903 SCSIHDRS=              \
904         scsi.h          \
905         scsi_address.h  \
906         scsi_ctl.h      \
907         scsi_fm.h       \
908         scsi_params.h   \
909         scsi_pkt.h      \
910         scsi_resource.h \
911         scsi_types.h    \
912         scsi_watch.h

914 SCSICONFHDRS=           \
915         autoconf.h      \
916         device.h

918 SCSIGENHDRS=            \
919         commands.h      \
```

```
920         dad_mode.h      \
921         inquiry.h       \
922         message.h       \
923         mode.h          \
924         persist.h       \
925         sense.h         \
926         sff_frames.h    \
927         smp_frames.h    \
928         status.h

930 SCSIIMPLHDRS=          \
931         commands.h      \
932         inquiry.h       \
933         mode.h          \
934         scsi_reset_notify.h \
935         scsi_sas.h      \
936         sense.h         \
937         services.h      \
938         smp_transport.h \
939         spc3_types.h    \
940         status.h        \
941         transport.h     \
942         types.h         \
943         uscsi.h         \
944         usmp.h

946 SCSITARGETSHDRS=        \
947         ses.h           \
948         sesio.h         \
949         sgendef.h       \
950         stdef.h         \
951         sddef.h         \
952         smp.h

954 SCSIADHDRS=

956 SCSICADHDRS=

958 SCSIISCSIHDRS=          \
959         iscsi_door.h    \
960         iscsi_if.h

962 SCSIVHCIHDRS=          \
963         scsi_vhci.h     \
964         mpapi_impl.h    \
965         mpapi_scsi_vhci.h

967 SDCARDHDRS=            \
968         sda.h           \
969         sda_impl.h      \
970         sda_ioctl.h

972 FC4HDRS=                       \
973         fc_transport.h  \
974         linkapp.h       \
975         fc.h            \
976         fcp.h           \
977         fcal_transport.h \
978         fcal.h          \
979         fcal_linkapp.h  \
980         fcio.h

982 FCHDRS=                \
983         fc.h            \
984         fcio.h          \
985         fc_types.h      \
```



```
986         fc_appif.h

988 FCIMPLHDRS=            \
989         fc_error.h      \
990         fcph.h

992 FCULPHDRS=            \
993         fcp_util.h      \
994         fcsm.h

996 SATAGENHDRS=           \
997         sata_hba.h      \
998         sata_defs.h     \
999         sata_cfgadm.h

1001 SYSEVENTHDRS=          \
1002         ap_driver.h     \
1003         dev.h           \
1004         domain.h        \
1005         dr.h            \
1006         env.h           \
1007         eventdefs.h     \
1008         ipmp.h          \
1009         pwrctl.h        \
1010         svm.h           \
1011         vrrp.h

1013 CONTRACTHDRS=          \
1014         process.h       \
1015         process_impl.h  \
1016         device.h        \
1017         device_impl.h

1019 USBHDRS=        \
1020         usba.h  \
1021         usbai.h

1023 UWBHDRS=        \
1024         uwb.h   \
1025         uwbai.h

1027 UWBAHDRS=       \
1028         uwba.h

1030 USBAUDHDRS=           \
1031         usb_audio.h

1033 USBHUBDHDRS=           \
1034         hub.h           \
1035         hubd_impl.h

1037 USBHIDHDRS=            \
1038         hid.h

1040 USBHWARCHDRS=          \
1041         hwarc.h

1043 USBMSHDRS=            \
1044         usb_bulkonly.h  \
1045         usb_cbi.h

1047 USBPRNHDRS=           \
1048         usb_printer.h

1050 USBCDCHDRS=           \
1051         usb_cdc.h
```

```
1053 USBVIDHDRS=           \
1054         usbvc.h

1056 USBWCMHDRS=             \
1057         usbwcm.h

1059 UGENHDRS=           \
1060         usb_ugen.h

1062 HOTPLUGHDRS=    \
1063         hpcsvc.h        \
1064         hpctrl.h

1066 HOTPLUGPCIHDRS= \
1067         pcicfg.h        \
1068         pcihp.h

1070 RSMHDRS= \
1071         rsm.h   \
1072         rsm_common.h    \
1073         rsmapi_common.h \
1074         rsmpi.h \
1075         rsmpi_driver.h  \
1076         rsmka_path_int.h

1078 TSOLHDRS=               \
1079         label.h         \
1080         label_macro.h   \
1081         priv.h          \
1082         tndb.h          \
1083         tsyscall.h

1085 I1394HDRS=              \
1086         cmd1394.h       \
1087         id1394.h        \
1088         ieee1212.h      \
1089         ieee1394.h      \
1090         ixl1394.h       \
1091         s1394_impl.h    \
1092         t1394.h

1094 # "cmdk" headers used on sparc
1095 SDKTPHDRS=              \
1096         dadkio.h        \
1097         fdisk.h

1099 # "cmdk" headers used on i386
1100 DKTPHDRS=               \
1101         altsctr.h       \
1102         bbh.h           \
1103         cm.h            \
1104         cmdev.h         \
1105         cmdk.h          \
1106         cmpkt.h         \
1107         controller.h    \
1108         dadev.h         \
1109         dadk.h          \
1110         dadkio.h        \
1111         fctypes.h       \
1112         fdisk.h         \
1113         flowctrl.h      \
1114         gda.h           \
1115         quetypes.h      \
1116         queue.h         \
1117         tgcom.h         \
```



```
1118         tgdk.h

1120 # "pc" header files used on i386
1121 PCHDRS=                 \
1122         avintr.h        \
1123         dma_engine.h    \
1124         i8272A.h        \
1125         pcic_reg.h      \
1126         pcic_var.h      \
1127         pic.h           \
1128         pit.h           \
1129         rtc.h

1131 NXGEHDRS=                       \
1132         nxge.h                  \
1133         nxge_common.h           \
1134         nxge_common_impl.h      \
1135         nxge_defs.h             \
1136         nxge_hw.h               \
1137         nxge_impl.h             \
1138         nxge_ipp.h              \
1139         nxge_ipp_hw.h           \
1140         nxge_mac.h              \
1141         nxge_mac_hw.h           \
1142         nxge_fflp.h             \
1143         nxge_fflp_hw.h          \
1144         nxge_mii.h              \
1145         nxge_rxdma.h            \
1146         nxge_rxdma_hw.h         \
1147         nxge_txc.h              \
1148         nxge_txc_hw.h           \
1149         nxge_txdma.h            \
1150         nxge_txdma_hw.h         \
1151         nxge_virtual.h          \
1152         nxge_espc.h

1154 include Makefile.syshdrs

1156 dcam/%.check:   dcam/%.h
1157         $(DOT_H_CHECK)

1159 CHECKHDRS=                                                      \
1160         $($(MACH)_HDRS:%.h=%.check)                             \
1161         $(AUDIOHDRS:%.h=audio/%.check)                          \
1162         $(AVHDRS:%.h=av/%.check)                                \
1163         $(BSCHDRS:%.h=%.check)                                  \
1164         $(CHKHDRS:%.h=%.check)                                  \
1165         $(CPUDRVHDRS:%.h=%.check)                               \
1166         $(CRYPTOHDRS:%.h=crypto/%.check)                        \
1167         $(DCAMHDRS:%.h=dcam/%.check)                            \
1168         $(FC4HDRS:%.h=fc4/%.check)                              \
1169         $(FCHDRS:%.h=fibre-channel/%.check)                     \
1170         $(FCIMPLHDRS:%.h=fibre-channel/impl/%.check)            \
1171         $(FCULPHDRS:%.h=fibre-channel/ulp/%.check)              \
1172         $(IBHDRS:%.h=ib/%.check)                                \
1173         $(IBDHDRS:%.h=ib/clients/ibd/%.check)                   \
1174         $(IBTLHDRS:%.h=ib/ibtl/%.check)                         \
1175         $(IBTLIMPLHDRS:%.h=ib/ibtl/impl/%.check)                \
1176         $(IBNEXHDRS:%.h=ib/ibnex/%.check)               \
1177         $(IBMGTHDRS:%.h=ib/mgt/%.check)                         \
1178         $(IBMFHDRS:%.h=ib/mgt/ibmf/%.check)                     \
1179         $(OFHDRS:%.h=ib/clients/of/%.check)                             \
1180         $(RDMAHDRS:%.h=ib/clients/of/rdma/%.check)                      \
1181         $(SOL_UVERBSHDRS:%.h=ib/clients/of/sol_uverbs/%.check)  \
1182         $(SOL_UCMAHDRS:%.h=ib/clients/of/sol_ucma/%.check)      \
1183         $(SOL_OFSHDRS:%.h=ib/clients/of/sol_ofs/%.check)        \
```

```
1184         $(TAVORHDRS:%.h=ib/adapters/tavor/%.check)      \
1185         $(HERMONHDRS:%.h=ib/adapters/hermon/%.check)    \
1186         $(MLNXHDRS:%.h=ib/adapters/%.check)             \
1187         $(IDMHDRS:%.h=idm/%.check)                      \
1188         $(ISCSIHDRS:%.h=iscsi/%.check)                  \
1189         $(ISCSITHDRS:%.h=iscsit/%.check)                        \
1190         $(ISOHDRS:%.h=iso/%.check)                      \
1191         $(FMHDRS:%.h=fm/%.check)                        \
1192         $(FMFSHDRS:%.h=fm/fs/%.check)                   \
1193         $(FMIOHDRS:%.h=fm/io/%.check)                   \
1194         $(FSHDRS:%.h=fs/%.check)                        \
1195         $(LVMHDRS:%.h=lvm/%.check)                      \
1196         $(PCMCIAHDRS:%.h=pcmcia/%.check)                \
1197         $(SCSIHDRS:%.h=scsi/%.check)                    \
1198         $(SCSIADHDRS:%.h=scsi/adapters/%.check)         \
1199         $(SCSICONFHDRS:%.h=scsi/conf/%.check)           \
1200         $(SCSIIMPLHDRS:%.h=scsi/impl/%.check)           \
1201         $(SCSIISCSIHDRS:%.h=scsi/adapters/%.check)      \
1202         $(SCSIGENHDRS:%.h=scsi/generic/%.check)         \
1203         $(SCSITARGETSHDRS:%.h=scsi/targets/%.check)     \
1204         $(SCSIVHCIHDRS:%.h=scsi/adapters/%.check)       \
1205         $(SATAGENHDRS:%.h=sata/%.check)                 \
1206         $(SDCARDHDRS:%.h=sdcard/%.check)                \
1207         $(SYSEVENTHDRS:%.h=sysevent/%.check)            \
1208         $(CONTRACTHDRS:%.h=contract/%.check)            \
1209         $(USBAUDHDRS:%.h=usb/clients/audio/%.check)   \
1210         $(USBHUBDHDRS:%.h=usb/hubd/%.check)             \
1211         $(USBHIDHDRS:%.h=usb/clients/hid/%.check)       \
1212         $(USBHWARCHDRS:%.h=usb/clients/hwarc/%.check)   \
1213         $(USBMSHDRS:%.h=usb/clients/mass_storage/%.check)       \
1214         $(USBPRNHDRS:%.h=usb/clients/printer/%.check)   \
1215         $(USBCDCHDRS:%.h=usb/clients/usbcdc/%.check)    \
1216         $(USBVIDHDRS:%.h=usb/clients/video/usbvc/%.check)   \
1217         $(USBWCMHDRS:%.h=usb/clients/usbinput/usbwcm/%.check)   \
1218         $(UGENHDRS:%.h=usb/clients/ugen/%.check)   \
1219         $(USBHDRS:%.h=usb/%.check)                      \
1220         $(UWBHDRS:%.h=uwb/%.check)                      \
1221         $(UWBAHDRS:%.h=uwb/uwba/%.check)                    \
1222         $(I1394HDRS:%.h=1394/%.check)                   \
1223         $(RSMHDRS:%.h=rsm/%.check)                      \
1224         $(TSOLHDRS:%.h=tsol/%.check)                    \
1225         $(NXGEHDRS:%.h=nxge/%.check)


1228 .KEEP_STATE:

1230 .PARALLEL:                      \
1231         $(CHECKHDRS)            \
1232         $(ROOTHDRS)             \
1233         $(ROOTAUDHDRS)          \
1234         $(ROOTAVHDRS)           \
1235         $(ROOTCRYPTOHDRS)       \
1236         $(ROOTDCAMHDRS)         \
1237         $(ROOTISOHDRS)          \
1238         $(ROOTIDMHDRS)          \
1239         $(ROOTISCSIHDRS)        \
1240         $(ROOTISCSITHDRS)       \
1241         $(ROOTFC4HDRS)          \
1242         $(ROOTFCHDRS)           \
1243         $(ROOTFCIMPLHDRS)       \
1244         $(ROOTFCULPHDRS)        \
1245         $(ROOTFMHDRS)           \
1246         $(ROOTFMIOHDRS)         \
1247         $(ROOTFMFSHDRS)         \
1248         $(ROOTFSHDRS)           \
1249         $(ROOTIBDHDRS)          \
```



```
1250         $(ROOTIBHDRS)           \
1251         $(ROOTIBTLHDRS)         \
1252         $(ROOTIBTLIMPLHDRS)     \
1253         $(ROOTIBNEXHDRS)        \
1254         $(ROOTIBMGTHDRS)        \
1255         $(ROOTIBMFHDRS)         \
1256         $(ROOTOFHDRS)           \
1257         $(ROOTRDMAHDRS)         \
1258         $(ROOTSOL_OFSHDRS)      \
1259         $(ROOTSOL_UMADHDRS)     \
1260         $(ROOTSOL_UVERBSHDRS)   \
1261         $(ROOTSOL_UCMAHDRS)     \
1262         $(ROOTTAVORHDRS)        \
1263         $(ROOTHERMONHDRS)       \
1264         $(ROOTMLNXHDRS)         \
1265         $(ROOTLVMHDRS)          \
1266         $(ROOTPCMCIAHDRS)       \
1267         $(ROOTSCSIHDRS)         \
1268         $(ROOTSCSIADHDRS)       \
1269         $(ROOTSCSICONFHDRS)     \
1270         $(ROOTSCSIISCSIHDRS)    \
1271         $(ROOTSCSIGENHDRS)      \
1272         $(ROOTSCSIIMPLHDRS)     \
1273         $(ROOTSCSIVHCIHDRS)     \
1274         $(ROOTSDCARDHDRS)       \
1275         $(ROOTSYSEVENTHDRS)     \
1276         $(ROOTCONTRACTHDRS)     \
1277         $(ROOTUSBHDRS)          \
1278         $(ROOTUWBHDRS)          \
1279         $(ROOTUWBAHDRS)         \
1280         $(ROOTUSBAUDHDRS)       \
1281         $(ROOTUSBHUBDHDRS)      \
1282         $(ROOTUSBHIDHDRS)       \
1283         $(ROOTUSBHRCHDRS)       \
1284         $(ROOTUSBMSHDRS)        \
1285         $(ROOTUSBPRNHDRS)       \
1286         $(ROOTUSBCDCHDRS)       \
1287         $(ROOTUSBVIDHDRS)       \
1288         $(ROOTUSBWCMHDRS)       \
1289         $(ROOTUGENHDRS)         \
1290         $(ROOT1394HDRS)         \
1291         $(ROOTHOTPLUGHDRS)      \
1292         $(ROOTHOTPLUGPCIHDRS)   \
1293         $(ROOTRSMHDRS)          \
1294         $(ROOTTSOLHDRS)         \
1295         $($(MACH)_ROOTHDRS)


1298 install_h:                      \
1299         $(ROOTDIRS)             \
1300         LVMDERIVED_H            \
1301         .WAIT                   \
1302         $(ROOTHDRS)             \
1303         $(ROOTAUDHDRS)          \
1304         $(ROOTAVHDRS)           \
1305         $(ROOTCRYPTOHDRS)       \
1306         $(ROOTDCAMHDRS)         \
1307         $(ROOTISOHDRS)          \
1308         $(ROOTIDMHDRS)          \
1309         $(ROOTISCSIHDRS)        \
1310         $(ROOTISCSITHDRS)       \
1311         $(ROOTFC4HDRS)          \
1312         $(ROOTFCHDRS)           \
1313         $(ROOTFCIMPLHDRS)       \
1314         $(ROOTFCULPHDRS)        \
1315         $(ROOTFMHDRS)           \
```

```
1316         $(ROOTFMFSHDRS)         \
1317         $(ROOTFMIOHDRS)         \
1318         $(ROOTFSHDRS)           \
1319         $(ROOTIBDHDRS)          \
1320         $(ROOTIBHDRS)           \
1321         $(ROOTIBTLHDRS)         \
1322         $(ROOTIBTLIMPLHDRS)     \
1323         $(ROOTIBNEXHDRS)        \
1324         $(ROOTIBMGTHDRS)        \
1325         $(ROOTIBMFHDRS)         \
1326         $(ROOTOFHDRS)           \
1327         $(ROOTRDMAHDRS)         \
1328         $(ROOTSOL_OFSHDRS)      \
1329         $(ROOTSOL_UMADHDRS)     \
1330         $(ROOTSOL_UVERBSHDRS)   \
1331         $(ROOTSOL_UCMAHDRS)     \
1332         $(ROOTTAVORHDRS)        \
1333         $(ROOTHERMONHDRS)       \
1334         $(ROOTMLNXHDRS)         \
1335         $(ROOTLVMHDRS)          \
1336         $(ROOTPCMCIAHDRS)       \
1337         $(ROOTSCSIHDRS)         \
1338         $(ROOTSCSIADHDRS)       \
1339         $(ROOTSCSIISCSIHDRS)    \
1340         $(ROOTSCSICONFHDRS)     \
1341         $(ROOTSCSIGENHDRS)      \
1342         $(ROOTSCSIIMPLHDRS)     \
1343         $(ROOTSCSIVHCIHDRS)     \
1344         $(ROOTSDCARDHDRS)       \
1345         $(ROOTSYSEVENTHDRS)     \
1346         $(ROOTCONTRACTHDRS)     \
1347         $(ROOTUWBHDRS)          \
1348         $(ROOTUWBAHDRS)         \
1349         $(ROOTUSBHDRS)          \
1350         $(ROOTUSBAUDHDRS)       \
1351         $(ROOTUSBHUBDHDRS)      \
1352         $(ROOTUSBHIDHDRS)       \
1353         $(ROOTUSBHRCHDRS)       \
1354         $(ROOTUSBMSHDRS)        \
1355         $(ROOTUSBPRNHDRS)       \
1356         $(ROOTUSBCDCHDRS)       \
1357         $(ROOTUSBVIDHDRS)       \
1358         $(ROOTUSBWCMHDRS)       \
1359         $(ROOTUGENHDRS)         \
1360         $(ROOT1394HDRS)         \
1361         $(ROOTHOTPLUGHDRS)      \
1362         $(ROOTHOTPLUGPCIHDRS)   \
1363         $(ROOTRSMHDRS)          \
1364         $(ROOTTSOLHDRS)         \
1365         $($(MACH)_ROOTHDRS)

1367 all_h: $(GENHDRS)

1369 priv_const.h: $(PRIVS_AWK) $(PRIVS_DEF)
1370         $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v privhfile=$@

1372 priv_names.h: $(PRIVS_AWK) $(PRIVS_DEF)
1373         $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v pubhfile=$@

1375 usb/usbdevs.h: $(USBDEVS_AWK) $(USBDEVS_DATA)
1376         $(NAWK) -f $(USBDEVS_AWK) $(USBDEVS_DATA) -H > $@

1378 LVMDERIVED_H:
1379         cd $(SRC)/uts/common/sys/lvm; pwd; $(MAKE) all_h

1381 clean:
```



```
1382         $(RM) $(GENHDRS)

1384 clobber: clean
1385         cd $(SRC)/uts/common/sys/lvm; pwd; $(MAKE) clobber

1387 check:  $(CHECKHDRS)

1389 FRC:
```

```
**********************************************************
   18689 Mon Oct  7 19:17:22 2013
new/usr/src/uts/common/sys/crypto/common.h
4185 New hash algorithm support
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright (c) 2003, 2010, Oracle and/or its affiliates. All rights reserved.
  23  */
  24 /*
  25  * Copyright 2013 Saso Kiselkov.  All rights reserved.
  26  */

  28 #ifndef _SYS_CRYPTO_COMMON_H
  29 #define _SYS_CRYPTO_COMMON_H

  31 /*
  32  * Header file for the common data structures of the cryptographic framework
  33  */

  35 #ifdef __cplusplus
  36 extern "C" {
  37 #endif

  39 #include <sys/types.h>
  40 #include <sys/uio.h>
  41 #include <sys/stream.h>
  42 #include <sys/mutex.h>
  43 #include <sys/condvar.h>


  46 /* Cryptographic Mechanisms */

  48 #define CRYPTO_MAX_MECH_NAME 32
  49 typedef char crypto_mech_name_t[CRYPTO_MAX_MECH_NAME];

  51 typedef uint64_t crypto_mech_type_t;

  53 typedef struct crypto_mechanism {
  54         crypto_mech_type_t      cm_type;        /* mechanism type */
  55         caddr_t                 cm_param;       /* mech. parameter */
  56         size_t                  cm_param_len;   /* mech. parameter len */
  57 } crypto_mechanism_t;
_____unchanged_portion_omitted_

 162 #endif  /* _SYSCALL32 */
 163 #endif /* _KERNEL */
```



```
 165 /*
 166  * The measurement unit bit flag for a mechanism's minimum or maximum key size.
 167  * The unit are mechanism dependent.  It can be in bits or in bytes.
 168  */
 169 typedef uint32_t crypto_keysize_unit_t;

 171 /*
 172  * The following bit flags are valid in cm_mech_flags field in
 173  * the crypto_mech_info_t structure of the SPI.
 174  *
 175  * Only the first two bit flags are valid in mi_keysize_unit
 176  * field in the crypto_mechanism_info_t structure of the API.
 177  */
 178 #define CRYPTO_KEYSIZE_UNIT_IN_BITS     0x00000001
 179 #define CRYPTO_KEYSIZE_UNIT_IN_BYTES    0x00000002
 180 #define CRYPTO_CAN_SHARE_OPSTATE        0x00000004 /* supports sharing */


 183 /* Mechanisms supported out-of-the-box */
 184 #define SUN_CKM_MD4                     "CKM_MD4"
 185 #define SUN_CKM_MD5                     "CKM_MD5"
 186 #define SUN_CKM_MD5_HMAC                "CKM_MD5_HMAC"
 187 #define SUN_CKM_MD5_HMAC_GENERAL        "CKM_MD5_HMAC_GENERAL"
 188 #define SUN_CKM_SHA1                    "CKM_SHA_1"
 189 #define SUN_CKM_SHA1_HMAC               "CKM_SHA_1_HMAC"
 190 #define SUN_CKM_SHA1_HMAC_GENERAL       "CKM_SHA_1_HMAC_GENERAL"
 191 #define SUN_CKM_SHA256                  "CKM_SHA256"
 192 #define SUN_CKM_SHA256_HMAC             "CKM_SHA256_HMAC"
 193 #define SUN_CKM_SHA256_HMAC_GENERAL     "CKM_SHA256_HMAC_GENERAL"
 194 #define SUN_CKM_SHA384                  "CKM_SHA384"
 195 #define SUN_CKM_SHA384_HMAC             "CKM_SHA384_HMAC"
 196 #define SUN_CKM_SHA384_HMAC_GENERAL     "CKM_SHA384_HMAC_GENERAL"
 197 #define SUN_CKM_SHA512                  "CKM_SHA512"
 198 #define SUN_CKM_SHA512_HMAC             "CKM_SHA512_HMAC"
 199 #define SUN_CKM_SHA512_HMAC_GENERAL     "CKM_SHA512_HMAC_GENERAL"
 200 #define SUN_CKM_SHA512_224              "CKM_SHA512_224"
 201 #define SUN_CKM_SHA512_256              "CKM_SHA512_256"
 202 #define SUN_CKM_DES_CBC                 "CKM_DES_CBC"
 203 #define SUN_CKM_DES3_CBC                "CKM_DES3_CBC"
 204 #define SUN_CKM_DES_ECB                 "CKM_DES_ECB"
 205 #define SUN_CKM_DES3_ECB                "CKM_DES3_ECB"
 206 #define SUN_CKM_BLOWFISH_CBC            "CKM_BLOWFISH_CBC"
 207 #define SUN_CKM_BLOWFISH_ECB            "CKM_BLOWFISH_ECB"
 208 #define SUN_CKM_AES_CBC                 "CKM_AES_CBC"
 209 #define SUN_CKM_AES_ECB                 "CKM_AES_ECB"
 210 #define SUN_CKM_AES_CTR                 "CKM_AES_CTR"
 211 #define SUN_CKM_AES_CCM                 "CKM_AES_CCM"
 212 #define SUN_CKM_AES_GCM                 "CKM_AES_GCM"
 213 #define SUN_CKM_AES_GMAC                "CKM_AES_GMAC"
 214 #define SUN_CKM_AES_CFB128              "CKM_AES_CFB128"
 215 #define SUN_CKM_RC4                     "CKM_RC4"
 216 #define SUN_CKM_RSA_PKCS                "CKM_RSA_PKCS"
 217 #define SUN_CKM_RSA_X_509               "CKM_RSA_X_509"
 218 #define SUN_CKM_MD5_RSA_PKCS            "CKM_MD5_RSA_PKCS"
 219 #define SUN_CKM_SHA1_RSA_PKCS           "CKM_SHA1_RSA_PKCS"
 220 #define SUN_CKM_SHA256_RSA_PKCS         "CKM_SHA256_RSA_PKCS"
 221 #define SUN_CKM_SHA384_RSA_PKCS         "CKM_SHA384_RSA_PKCS"
 222 #define SUN_CKM_SHA512_RSA_PKCS         "CKM_SHA512_RSA_PKCS"
 223 #define SUN_CKM_EC_KEY_PAIR_GEN         "CKM_EC_KEY_PAIR_GEN"
 224 #define SUN_CKM_ECDH1_DERIVE            "CKM_ECDH1_DERIVE"
 225 #define SUN_CKM_ECDSA_SHA1              "CKM_ECDSA_SHA1"
 226 #define SUN_CKM_ECDSA                   "CKM_ECDSA"

 228 /* Shared operation context format for CKM_RC4 */
 229 typedef struct {
```

```
230 #if defined(__amd64)
231         uint32_t        i, j;
232         uint32_t        arr[256];
233         uint32_t        flag;
234 #else
235         uchar_t         arr[256];
236         uchar_t         i, j;
237 #endif /* __amd64 */
238         uint64_t        pad;            /* For 64-bit alignment */
239 } arcfour_state_t;
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    4730 Mon Oct  7 19:17:22 2013
new/usr/src/uts/common/sys/debug.h
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright (c) 2012 by Delphix. All rights reserved.
 * Copyright 2013 Saso Kiselkov. All rights reserved.
 */

/*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
/*        All Rights Reserved   */

#ifndef _SYS_DEBUG_H
#define _SYS_DEBUG_H

#include <sys/isa_defs.h>
#include <sys/types.h>
#include <sys/note.h>

#ifdef  __cplusplus
extern "C" {
#endif

/*
 * ASSERT(ex) causes a panic or debugger entry if expression ex is not
 * true.  ASSERT() is included only for debugging, and is a no-op in
 * production kernels.  VERIFY(ex), on the other hand, behaves like
 * ASSERT and is evaluated on both debug and non-debug kernels.
 */

#if defined(__STDC__)
extern int assfail(const char *, const char *, int);
#define VERIFY(EX) ((void)((EX) || assfail(#EX, __FILE__, __LINE__)))
#if DEBUG
#define ASSERT(EX) ((void)((EX) || assfail(#EX, __FILE__, __LINE__)))
#else
#define ASSERT(x)  ((void)0)
#endif
#else   /* defined(__STDC__) */
extern int assfail();
```



```
#define VERIFY(EX) ((void)((EX) || assfail("EX", __FILE__, __LINE__)))
#if DEBUG
#define ASSERT(EX) ((void)((EX) || assfail("EX", __FILE__, __LINE__)))
#else
#define ASSERT(x)  ((void)0)
#endif
#endif  /* defined(__STDC__) */

/*
 * Assertion variants sensitive to the compilation data model
 */
#if defined(_LP64)
#define ASSERT64(x)     ASSERT(x)
#define ASSERT32(x)
#else
#define ASSERT64(x)
#define ASSERT32(x)     ASSERT(x)
#endif

/*
 * IMPLY and EQUIV are assertions of the form:
 *
 *      if (a) then (b)
 * and
 *      if (a) then (b) *AND* if (b) then (a)
 */
#if DEBUG
#define IMPLY(A, B) \
        ((void)((((!(A)) || (B)) || \
            assfail("(" #A ") implies (" #B ")", __FILE__, __LINE__)))
#define EQUIV(A, B) \
        ((void)((!!(A) == !!(B)) || \
            assfail("(" #A ") is equivalent to (" #B ")", __FILE__, __LINE__)))
#else
#define IMPLY(A, B) ((void)0)
#define EQUIV(A, B) ((void)0)
#endif

/*
 * ASSERT3() behaves like ASSERT() except that it is an explicit conditional,
 * and prints out the values of the left and right hand expressions as part of
 * the panic message to ease debugging.  The three variants imply the type
 * of their arguments.  ASSERT3S() is for signed data types, ASSERT3U() is
 * for unsigned, and ASSERT3P() is for pointers.  The VERIFY3*() macros
 * have the same relationship as above.
 */
extern void assfail3(const char *, uintmax_t, const char *, uintmax_t,
    const char *, int);
#define VERIFY3_IMPL(LEFT, OP, RIGHT, TYPE) do { \
        const TYPE __left = (TYPE)(LEFT); \
        const TYPE __right = (TYPE)(RIGHT); \
        if (!(__left OP __right)) \
                assfail3(#LEFT " " #OP " " #RIGHT, \
                        (uintmax_t)__left, #OP, (uintmax_t)__right, \
                        __FILE__, __LINE__); \
_NOTE(CONSTCOND) } while (0)

#define VERIFY3S(x, y, z)       VERIFY3_IMPL(x, y, z, int64_t)
#define VERIFY3U(x, y, z)       VERIFY3_IMPL(x, y, z, uint64_t)
#define VERIFY3P(x, y, z)       VERIFY3_IMPL(x, y, z, uintptr_t)
#define VERIFY0(x)              VERIFY3_IMPL(x, ==, 0, uintmax_t)

#if DEBUG
#define ASSERT3S(x, y, z)       VERIFY3_IMPL(x, y, z, int64_t)
#define ASSERT3U(x, y, z)       VERIFY3_IMPL(x, y, z, uint64_t)
#define ASSERT3P(x, y, z)       VERIFY3_IMPL(x, y, z, uintptr_t)
```

```
#define ASSERT0(x)              VERIFY3_IMPL(x, ==, 0, uintmax_t)
#else
#define ASSERT3S(x, y, z)       ((void)0)
#define ASSERT3U(x, y, z)       ((void)0)
#define ASSERT3P(x, y, z)       ((void)0)
#define ASSERT0(x)              ((void)0)
#endif

/*
 * Compile-time assertion. The condition 'x' must be constant.
 */
#define CTASSERT(x)             _CTASSERT(x, __LINE__)
#define _CTASSERT(x, y)         __CTASSERT(x, y)
#define __CTASSERT(x, y) \
        typedef char __compile_time_assertion__ ## y [(x) ? 1 : -1]

#ifdef  _KERNEL

extern void abort_sequence_enter(char *);
extern void debug_enter(char *);

#endif  /* _KERNEL */

#if defined(DEBUG) && !defined(__sun)
/* CSTYLED */
#define STATIC
#else
/* CSTYLED */
#define STATIC static
#endif

#ifdef  __cplusplus
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    2605 Mon Oct  7 19:17:22 2013
new/usr/src/uts/common/sys/edonr.h
4185 New hash algorithm support
**********************************************************
```

```
/*
 * IDI,NTNU
 *
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://opensource.org/licenses/CDDL-1.0.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 *
 * Copyright (C) 2009, 2010, Jorn Amundsen <jorn.amundsen@ntnu.no>
 *
 * Tweaked Edon-R implementation for SUPERCOP, based on NIST API.
 *
 * $Id: edonr.h 517 2013-02-17 20:34:39Z joern $
 */
/*
 * Portions copyright (c) 2013, Saso Kiselkov, All rights reserved
 */

#ifndef _SYS_EDONR_H_
#define _SYS_EDONR_H_

#ifdef  __cplusplus
extern "C" {
#endif

#include <sys/types.h>

/*
 * EdonR allows to call EdonRUpdate() consecutively only if the total length
 * of stored unprocessed data and the new supplied data is less than or equal
 * to the BLOCK_SIZE on which the compression functions operates.
 * Otherwise an assertion failure is invoked.
 */

/* Specific algorithm definitions */
#define EdonR224_DIGEST_SIZE    28
#define EdonR224_BLOCK_SIZE     64
#define EdonR256_DIGEST_SIZE    32
#define EdonR256_BLOCK_SIZE     64
#define EdonR384_DIGEST_SIZE    48
#define EdonR384_BLOCK_SIZE     128
#define EdonR512_DIGEST_SIZE    64
#define EdonR512_BLOCK_SIZE     128

#define EdonR256_BLOCK_BITSIZE  512
#define EdonR512_BLOCK_BITSIZE  1024
```



```
typedef struct {
        uint32_t DoublePipe[16];
        uint8_t LastPart[EdonR256_BLOCK_SIZE * 2];
} EdonRData256;
typedef struct {
        uint64_t DoublePipe[16];
        uint8_t LastPart[EdonR512_BLOCK_SIZE * 2];
} EdonRData512;

typedef struct {
        size_t hashbitlen;

        /* + algorithm specific parameters */
        int unprocessed_bits;
        uint64_t bits_processed;
        union {
                EdonRData256 p256[1];
                EdonRData512 p512[1];
        } pipe[1];
} EdonRState;

void EdonRInit(EdonRState *state, size_t hashbitlen);
void EdonRUpdate(EdonRState *state, const uint8_t *data, size_t databitlen);
void EdonRFinal(EdonRState *state, uint8_t *hashval);
void EdonRHash(size_t hashbitlen, const uint8_t *data, size_t databitlen,
    uint8_t *hashval);

#ifdef  __cplusplus
}
#endif

#endif  /* _SYS_EDONR_H_ */
```

```
**********************************************************
    4699 Mon Oct  7 19:17:22 2013
new/usr/src/uts/common/sys/sha2.h
4185 New hash algorithm support
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/* Copyright 2013 Saso Kiselkov.  All rights reserved. */

#ifndef _SYS_SHA2_H
#define _SYS_SHA2_H

#include <sys/types.h>          /* for uint_* */

#ifdef  __cplusplus
extern "C" {
#endif

#define SHA2_HMAC_MIN_KEY_LEN   1       /* SHA2-HMAC min key length in bytes */
#define SHA2_HMAC_MAX_KEY_LEN   INT_MAX /* SHA2-HMAC max key length in bytes */

#define SHA256_DIGEST_LENGTH    32      /* SHA256 digest length in bytes */
#define SHA384_DIGEST_LENGTH    48      /* SHA384 digest length in bytes */
#define SHA512_DIGEST_LENGTH    64      /* SHA512 digest length in bytes */

/* Truncated versions of SHA-512 according to FIPS-180-4, section 5.3.6 */
#define SHA512_224_DIGEST_LENGTH        28      /* SHA512/224 digest length */
#define SHA512_256_DIGEST_LENGTH        32      /* SHA512/256 digest length */

#define SHA256_HMAC_BLOCK_SIZE  64      /* SHA256-HMAC block size */
#define SHA512_HMAC_BLOCK_SIZE  128     /* SHA512-HMAC block size */

#define SHA256                  0
#define SHA256_HMAC             1
#define SHA256_HMAC_GEN         2
#define SHA384                  3
#define SHA384_HMAC             4
#define SHA384_HMAC_GEN         5
#define SHA512                  6
#define SHA512_HMAC             7
#define SHA512_HMAC_GEN         8
#define SHA512_224              9
#define SHA512_256              10
```



```
/*
 * SHA2 context.
 * The contents of this structure are a private interface between the
 * Init/Update/Final calls of the functions defined below.
 * Callers must never attempt to read or write any of the fields
 * in this structure directly.
 */
typedef struct  {
        uint32_t algotype;              /* Algorithm Type */

        /* state (ABCDEFGH) */
        union {
                uint32_t s32[8];        /* for SHA256 */
                uint64_t s64[8];        /* for SHA384/512 */
        } state;
        /* number of bits */
        union {
                uint32_t c32[2];        /* for SHA256 , modulo 2^64 */
                uint64_t c64[2];        /* for SHA384/512, modulo 2^128 */
        } count;
        union {
                uint8_t         buf8[128];      /* undigested input */
                uint32_t        buf32[32];      /* realigned input */
                uint64_t        buf64[16];      /* realigned input */
        } buf_un;
} SHA2_CTX;

typedef SHA2_CTX SHA256_CTX;
typedef SHA2_CTX SHA384_CTX;
typedef SHA2_CTX SHA512_CTX;

extern void SHA2Init(uint64_t mech, SHA2_CTX *);

extern void SHA2Update(SHA2_CTX *, const void *, size_t);

extern void SHA2Final(void *, SHA2_CTX *);

extern void SHA256Init(SHA256_CTX *);

extern void SHA256Update(SHA256_CTX *, const void *, size_t);

extern void SHA256Final(void *, SHA256_CTX *);

extern void SHA384Init(SHA384_CTX *);

extern void SHA384Update(SHA384_CTX *, const void *, size_t);

extern void SHA384Final(void *, SHA384_CTX *);

extern void SHA512Init(SHA512_CTX *);

extern void SHA512Update(SHA512_CTX *, const void *, size_t);

extern void SHA512Final(void *, SHA512_CTX *);

#ifdef _SHA2_IMPL
/*
 * The following types/functions are all private to the implementation
 * of the SHA2 functions and must not be used by consumers of the interface
 */

/*
 * List of support mechanisms in this module.
 *
 * It is important to note that in the module, division or modulus calculations
 * are used on the enumerated type to determine which mechanism is being used;
```

```
 128  * therefore, changing the order or additional mechanisms should be done
 129  * carefully
 130  */
 131 typedef enum sha2_mech_type {
 132         SHA256_MECH_INFO_TYPE,          /* SUN_CKM_SHA256 */
 133         SHA256_HMAC_MECH_INFO_TYPE,     /* SUN_CKM_SHA256_HMAC */
 134         SHA256_HMAC_GEN_MECH_INFO_TYPE, /* SUN_CKM_SHA256_HMAC_GENERAL */
 135         SHA384_MECH_INFO_TYPE,          /* SUN_CKM_SHA384 */
 136         SHA384_HMAC_MECH_INFO_TYPE,     /* SUN_CKM_SHA384_HMAC */
 137         SHA384_HMAC_GEN_MECH_INFO_TYPE, /* SUN_CKM_SHA384_HMAC_GENERAL */
 138         SHA512_MECH_INFO_TYPE,          /* SUN_CKM_SHA512 */
 139         SHA512_HMAC_MECH_INFO_TYPE,     /* SUN_CKM_SHA512_HMAC */
 140         SHA512_HMAC_GEN_MECH_INFO_TYPE, /* SUN_CKM_SHA512_HMAC_GENERAL */
 141         SHA512_224_MECH_INFO_TYPE,      /* SUN_CKM_SHA512_224 */
 142         SHA512_256_MECH_INFO_TYPE       /* SUN_CKM_SHA512_256 */
 133         SHA512_HMAC_GEN_MECH_INFO_TYPE  /* SUN_CKM_SHA512_HMAC_GENERAL */
 143 } sha2_mech_type_t;
```

*unchanged_portion_omitted_*

```
**********************************************************
    6786 Mon Oct  7 19:17:23 2013
new/usr/src/uts/common/sys/skein.h
4185 New hash algorithm support
**********************************************************
/*
 * Interface declarations for Skein hashing.
 * Source code author: Doug Whiting, 2008.
 * This algorithm and source code is released to the public domain.
 *
 * The following compile-time switches may be defined to control some
 * tradeoffs between speed, code size, error checking, and security.
 *
 * The "default" note explains what happens when the switch is not defined.
 *
 *  SKEIN_DEBUG            -- make callouts from inside Skein code
 *                            to examine/display intermediate values.
 *                            [default: no callouts (no overhead)]
 *
 *  SKEIN_ERR_CHECK        -- how error checking is handled inside Skein
 *                            code. If not defined, most error checking
 *                            is disabled (for performance). Otherwise,
 *                            the switch value is interpreted as:
 *                                0: use assert()      to flag errors
 *                                1: return SKEIN_FAIL to flag errors
 */
/* Copyright 2013 Doug Whiting. This code is released to the public domain. */
#ifndef _SYS_SKEIN_H_
#define _SYS_SKEIN_H_

#include <sys/types.h>          /* get size_t definition */

#ifdef  __cplusplus
extern "C" {
#endif

enum {
        SKEIN_SUCCESS = 0,      /* return codes from Skein calls */
        SKEIN_FAIL = 1,
        SKEIN_BAD_HASHLEN = 2
};

#define SKEIN_MODIFIER_WORDS    (2)     /* number of modifier (tweak) words */

#define SKEIN_256_STATE_WORDS   (4)
#define SKEIN_512_STATE_WORDS   (8)
#define SKEIN1024_STATE_WORDS   (16)
#define SKEIN_MAX_STATE_WORDS   (16)

#define SKEIN_256_STATE_BYTES   (8 * SKEIN_256_STATE_WORDS)
#define SKEIN_512_STATE_BYTES   (8 * SKEIN_512_STATE_WORDS)
#define SKEIN1024_STATE_BYTES   (8 * SKEIN1024_STATE_WORDS)

#define SKEIN_256_STATE_BITS    (64 * SKEIN_256_STATE_WORDS)
#define SKEIN_512_STATE_BITS    (64 * SKEIN_512_STATE_WORDS)
#define SKEIN1024_STATE_BITS    (64 * SKEIN1024_STATE_WORDS)

#define SKEIN_256_BLOCK_BYTES   (8 * SKEIN_256_STATE_WORDS)
#define SKEIN_512_BLOCK_BYTES   (8 * SKEIN_512_STATE_WORDS)
#define SKEIN1024_BLOCK_BYTES   (8 * SKEIN1024_STATE_WORDS)

typedef struct {
        size_t hashBitLen;      /* size of hash result, in bits */
        size_t bCnt;            /* current byte count in buffer b[] */
        /* tweak words: T[0]=byte cnt, T[1]=flags */
        uint64_t T[SKEIN_MODIFIER_WORDS];
```



```
} Skein_Ctxt_Hdr_t;

typedef struct {                /*  256-bit Skein hash context structure */
        Skein_Ctxt_Hdr_t h;     /* common header context variables */
        uint64_t X[SKEIN_256_STATE_WORDS];      /* chaining variables */
        /* partial block buffer (8-byte aligned) */
        uint8_t b[SKEIN_256_BLOCK_BYTES];
} Skein_256_Ctxt_t;

typedef struct {                /*  512-bit Skein hash context structure */
        Skein_Ctxt_Hdr_t h;     /* common header context variables */
        uint64_t X[SKEIN_512_STATE_WORDS];      /* chaining variables */
        /* partial block buffer (8-byte aligned) */
        uint8_t b[SKEIN_512_BLOCK_BYTES];
} Skein_512_Ctxt_t;

typedef struct {                /* 1024-bit Skein hash context structure */
        Skein_Ctxt_Hdr_t h;     /* common header context variables */
        uint64_t X[SKEIN1024_STATE_WORDS];      /* chaining variables */
        /* partial block buffer (8-byte aligned) */
        uint8_t b[SKEIN1024_BLOCK_BYTES];
} Skein1024_Ctxt_t;

/*   Skein APIs for (incremental) "straight hashing" */
int Skein_256_Init(Skein_256_Ctxt_t *ctx, size_t hashBitLen);
int Skein_512_Init(Skein_512_Ctxt_t *ctx, size_t hashBitLen);
int Skein1024_Init(Skein1024_Ctxt_t *ctx, size_t hashBitLen);

int Skein_256_Update(Skein_256_Ctxt_t *ctx, const uint8_t *msg,
    size_t msgByteCnt);
int Skein_512_Update(Skein_512_Ctxt_t *ctx, const uint8_t *msg,
    size_t msgByteCnt);
int Skein1024_Update(Skein1024_Ctxt_t *ctx, const uint8_t *msg,
    size_t msgByteCnt);

int Skein_256_Final(Skein_256_Ctxt_t *ctx, uint8_t *hashVal);
int Skein_512_Final(Skein_512_Ctxt_t *ctx, uint8_t *hashVal);
int Skein1024_Final(Skein1024_Ctxt_t *ctx, uint8_t *hashVal);

/*
 * Skein APIs for "extended" initialization: MAC keys, tree hashing.
 * After an InitExt() call, just use Update/Final calls as with Init().
 *
 * Notes: Same parameters as _Init() calls, plus treeInfo/key/keyBytes.
 *          When keyBytes == 0 and treeInfo == SKEIN_SEQUENTIAL,
 *              the results of InitExt() are identical to calling Init().
 *          The function Init() may be called once to "precompute" the IV for
 *              a given hashBitLen value, then by saving a copy of the context
 *              the IV computation may be avoided in later calls.
 *          Similarly, the function InitExt() may be called once per MAC key
 *              to precompute the MAC IV, then a copy of the context saved and
 *              reused for each new MAC computation.
 */
int Skein_256_InitExt(Skein_256_Ctxt_t *ctx, size_t hashBitLen,
    uint64_t treeInfo, const uint8_t *key, size_t keyBytes);
int Skein_512_InitExt(Skein_512_Ctxt_t *ctx, size_t hashBitLen,
    uint64_t treeInfo, const uint8_t *key, size_t keyBytes);
int Skein1024_InitExt(Skein1024_Ctxt_t *ctx, size_t hashBitLen,
    uint64_t treeInfo, const uint8_t *key, size_t keyBytes);

/*
 * Skein APIs for MAC and tree hash:
 *      Final_Pad: pad, do final block, but no OUTPUT type
 *      Output:    do just the output stage
 */
int Skein_256_Final_Pad(Skein_256_Ctxt_t *ctx, uint8_t *hashVal);
```

```
128 int Skein_512_Final_Pad(Skein_512_Ctxt_t *ctx, uint8_t *hashVal);
129 int Skein1024_Final_Pad(Skein1024_Ctxt_t *ctx, uint8_t *hashVal);

131 #ifndef SKEIN_TREE_HASH
132 #define SKEIN_TREE_HASH (1)
133 #endif
134 #if     SKEIN_TREE_HASH
135 int Skein_256_Output(Skein_256_Ctxt_t *ctx, uint8_t *hashVal);
136 int Skein_512_Output(Skein_512_Ctxt_t *ctx, uint8_t *hashVal);
137 int Skein1024_Output(Skein1024_Ctxt_t *ctx, uint8_t *hashVal);
138 #endif

140 /*
141  * When you initialize a Skein KCF hashing method you can pass this param
142  * structure in cm_param to fine-tune the algorithm's defaults.
143  */
144 typedef struct skein_param {
145         size_t  sp_digest_bitlen;               /* length of digest in bits */
146 } skein_param_t;

148 /* Module definitions */
149 #ifdef  SKEIN_MODULE_IMPL
150 #define CKM_SKEIN_256                           "CKM_SKEIN_256"
151 #define CKM_SKEIN_512                           "CKM_SKEIN_512"
152 #define CKM_SKEIN1024                           "CKM_SKEIN1024"
153 #define CKM_SKEIN_256_MAC                       "CKM_SKEIN_256_MAC"
154 #define CKM_SKEIN_512_MAC                       "CKM_SKEIN_512_MAC"
155 #define CKM_SKEIN1024_MAC                       "CKM_SKEIN1024_MAC"

157 typedef enum skein_mech_type {
158         SKEIN_256_MECH_INFO_TYPE,
159         SKEIN_512_MECH_INFO_TYPE,
160         SKEIN1024_MECH_INFO_TYPE,
161         SKEIN_256_MAC_MECH_INFO_TYPE,
162         SKEIN_512_MAC_MECH_INFO_TYPE,
163         SKEIN1024_MAC_MECH_INFO_TYPE
164 } skein_mech_type_t;

166 #define VALID_SKEIN_DIGEST_MECH(__mech)                         \
167         ((int)(__mech) >= SKEIN_256_MECH_INFO_TYPE &&           \
168         (__mech) <= SKEIN1024_MECH_INFO_TYPE)
169 #define VALID_SKEIN_MAC_MECH(__mech)                            \
170         ((int)(__mech) >= SKEIN_256_MAC_MECH_INFO_TYPE &&       \
171         (__mech) <= SKEIN1024_MAC_MECH_INFO_TYPE)
172 #endif  /* SKEIN_MODULE_IMPL */

174 #ifdef  __cplusplus
175 }
176 #endif

178 #endif  /* _SYS_SKEIN_H_ */
```

```
**********************************************************
   17042 Mon Oct  7 19:17:23 2013
new/usr/src/uts/intel/Makefile.intel.shared
4185 New hash algorithm support
**********************************************************
```

```
   1 # CDDL HEADER START
   2 #
   3 # The contents of this file are subject to the terms of the
   4 # Common Development and Distribution License (the "License").
   5 # You may not use this file except in compliance with the License.
   6 #
   7 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   8 # or http://www.opensolaris.org/os/licensing.
   9 # See the License for the specific language governing permissions
  10 # and limitations under the License.
  11 #
  12 # When distributing Covered Code, include this CDDL HEADER in each
  13 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  14 # If applicable, add the following below this CDDL HEADER, with the
  15 # fields enclosed by brackets "[]" replaced with your own identifying
  16 # information: Portions Copyright [yyyy] [name of copyright owner]
  17 #
  18 # CDDL HEADER END
  19 #

  21 # Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
  22 # Copyright (c) 2012 Nexenta Systems, Inc. All rights reserved.
  23 # Copyright (c) 2013 Andrew Stormont.  All rights reserved.

  25 #
  26 #       This makefile contains the common definitions for all intel
  27 #       implementation architecture independent modules.
  28 #

  30 #
  31 #       Machine type (implementation architecture):
  32 #
  33 PLATFORM         = i86pc

  35 #
  36 #       Everybody needs to know how to build modstubs.o and to locate unix.o.
  37 #       Note that unix.o must currently be selected from among the possible
  38 #       "implementation architectures". Note further, that unix.o is only
  39 #       used as an optional error check for undefines so (theoretically)
  40 #       any "implementation architectures" could be used. We choose i86pc
  41 #       because it is the reference port.
  42 #
  43 UNIX_DIR         = $(UTSBASE)/i86pc/unix
  44 GENLIB_DIR       = $(UTSBASE)/intel/genunix
  45 IPDRV_DIR        = $(UTSBASE)/intel/ip
  46 MODSTUBS_DIR     = $(UNIX_DIR)
  47 DSF_DIR          = $(UTSBASE)/$(PLATFORM)/genassym
  48 LINTS_DIR        = $(OBJS_DIR)
  49 LINT_LIB_DIR     = $(UTSBASE)/intel/lint-libs/$(OBJS_DIR)

  51 UNIX_O           = $(UNIX_DIR)/$(OBJS_DIR)/unix.o
  52 GENLIB           = $(GENLIB_DIR)/$(OBJS_DIR)/libgenunix.so
  53 MODSTUBS_O       = $(MODSTUBS_DIR)/$(OBJS_DIR)/modstubs.o
  54 LINT_LIB         = $(UTSBASE)/i86pc/lint-libs/$(OBJS_DIR)/llib-lunix.ln
  55 GEN_LINT_LIB     = $(UTSBASE)/intel/lint-libs/$(OBJS_DIR)/llib-lgenunix.ln

  57 #
  58 #       Include the makefiles which define build rule templates, the
  59 #       collection of files per module, and a few specific flags. Note
  60 #       that order is significant, just as with an include path. The
  61 #       first build rule template which matches the files name will be
```



```
  62 #       used. By including these in order from most machine dependent
  63 #       to most machine independent, we allow a machine dependent file
  64 #       to be used in preference over a machine independent version
  65 #       (Such as a machine specific optimization, which preserves the
  66 #       interfaces.)
  67 #
  68 include $(UTSTREE)/intel/Makefile.files
  69 include $(UTSTREE)/common/Makefile.files

  71 #
  72 # ----- TRANSITIONAL SECTION --------------------------------------------------
  73 #

  75 #
  76 #       Not everything which *should* be a module is a module yet. The
  77 #       following is a list of such objects which are currently part of
  78 #       genunix but which might someday become kmods.  This must be
  79 #       defined before we include Makefile.uts, or else genunix's build
  80 #       won't be as parallel as we might like.
  81 #
  82 NOT_YET_KMODS    = $(OLDPTY_OBJS) $(PTY_OBJS) $(VCONS_CONF_OBJS) $(MOD_OBJS)

  84 #
  85 # ----- END OF TRANSITIONAL SECTION -------------------------------------------
  86 #
  87 #       Include machine independent rules. Note that this does not imply
  88 #       that the resulting module from rules in Makefile.uts is machine
  89 #       independent. Only that the build rules are machine independent.
  90 #
  91 include $(UTSBASE)/Makefile.uts

  93 #
  94 #       The following must be defined for all implementations:
  95 #
  96 MODSTUBS               = $(UTSBASE)/intel/ia32/ml/modstubs.s

  98 #
  99 #       Define supported builds
 100 #
 101 DEF_BUILDS             = $(DEF_BUILDS64) $(DEF_BUILDS32)
 102 ALL_BUILDS             = $(ALL_BUILDS64) $(ALL_BUILDS32)

 104 #
 105 #       x86 or amd64 inline templates
 106 #
 107 INLINES_32             = $(UTSBASE)/intel/ia32/ml/ia32.il
 108 INLINES_64             = $(UTSBASE)/intel/amd64/ml/amd64.il
 109 INLINES                += $(INLINES_$(CLASS))

 111 #
 112 #       kernel-specific optimizations; override default in Makefile.master
 113 #

 115 CFLAGS_XARCH_32         = $(i386_CFLAGS)
 116 CFLAGS_XARCH_64         = $(amd64_CFLAGS)
 117 CFLAGS_XARCH            = $(CFLAGS_XARCH_$(CLASS))

 119 COPTFLAG_32             = $(COPTFLAG)
 120 COPTFLAG_64             = $(COPTFLAG64)
 121 COPTIMIZE               = $(COPTFLAG_$(CLASS))

 123 CFLAGS                  = $(CFLAGS_XARCH)
 124 CFLAGS                  += $(COPTIMIZE)
 125 CFLAGS                  += $(INLINES) -D_ASM_INLINES
 126 CFLAGS                  += $(CCMODE)
 127 CFLAGS                  += $(SPACEFLAG)
```

```
128 CFLAGS                  += $(CCUNBOUND)
129 CFLAGS                  += $(CFLAGS_uts)
130 CFLAGS                  += -xstrconst

132 ASFLAGS_XARCH_32        = $(i386_ASFLAGS)
133 ASFLAGS_XARCH_64        = $(amd64_ASFLAGS)
134 ASFLAGS_XARCH           = $(ASFLAGS_XARCH_$(CLASS))

136 ASFLAGS                 += $(ASFLAGS_XARCH)

138 #
139 #       Define the base directory for installation.
140 #
141 BASE_INS_DIR    = $(ROOT)

143 #
144 #       Debugging level
145 #
146 #       Special knowledge of which special debugging options affect which
147 #       file is used to optimize the build if these flags are changed.
148 #
149 DEBUG_DEFS_OBJ32        =
150 DEBUG_DEFS_DBG32        = -DDEBUG
151 DEBUG_DEFS_OBJ64        =
152 DEBUG_DEFS_DBG64        = -DDEBUG
153 DEBUG_DEFS              = $(DEBUG_DEFS_$(BUILD_TYPE))

155 DEBUG_COND_OBJ32        = $(POUND_SIGN)
156 DEBUG_COND_DBG32        =
157 DEBUG_COND_OBJ64        = $(POUND_SIGN)
158 DEBUG_COND_DBG64        =
159 IF_DEBUG_OBJ            = $(DEBUG_COND_$(BUILD_TYPE))$(OBJS_DIR)/

161 $(IF_DEBUG_OBJ)syscall.o        :=      DEBUG_DEFS      += -DSYSCALLTRACE
162 $(IF_DEBUG_OBJ)clock.o          :=      DEBUG_DEFS      += -DKSLICE=1

164 #
165 #       Collect the preprocessor definitions to be associated with *all*
166 #       files.
167 #
168 ALL_DEFS         = $(DEBUG_DEFS) $(OPTION_DEFS)

170 #
171 #       The kernels modules which are "implementation architecture"
172 #       specific for this machine are enumerated below. Note that most
173 #       of these modules must exist (in one form or another) for each
174 #       architecture.
175 #
176 #       Common Drivers (usually pseudo drivers) (/kernel/drv)
177 #       DRV_KMODS are built both 32-bit and 64-bit
178 #       DRV_KMODS_32 are built only 32-bit
179 #       DRV_KMODS_64 are built only 64-bit
180 #
181 DRV_KMODS       += aac
182 DRV_KMODS       += aggr
183 DRV_KMODS       += ahci
184 DRV_KMODS       += amd64_gart
185 DRV_KMODS       += amr
186 DRV_KMODS       += agpgart
187 DRV_KMODS       += srn
188 DRV_KMODS       += agptarget
189 DRV_KMODS       += arn
190 DRV_KMODS       += arp
191 DRV_KMODS       += asy
192 DRV_KMODS       += ata
193 DRV_KMODS       += ath
```



```
194 DRV_KMODS       += atu
195 DRV_KMODS       += audio
196 DRV_KMODS       += audio1575
197 DRV_KMODS       += audio810
198 DRV_KMODS       += audiocmi
199 DRV_KMODS       += audiocmihd
200 DRV_KMODS       += audioemu10k
201 DRV_KMODS       += audioens
202 DRV_KMODS       += audiohd
203 DRV_KMODS       += audioixp
204 DRV_KMODS       += audiols
205 DRV_KMODS       += audiop16x
206 DRV_KMODS       += audiopci
207 DRV_KMODS       += audiosolo
208 DRV_KMODS       += audiots
209 DRV_KMODS       += audiovia823x
210 DRV_KMODS_32    += audiovia97
211 DRV_KMODS       += bl
212 DRV_KMODS       += blkdev
213 DRV_KMODS       += bge
214 DRV_KMODS       += bofi
215 DRV_KMODS       += bpf
216 DRV_KMODS       += bridge
217 DRV_KMODS       += bscbus
218 DRV_KMODS       += bscv
219 DRV_KMODS       += chxge
220 DRV_KMODS       += cxgbe
221 DRV_KMODS       += ntxn
222 DRV_KMODS       += myri10ge
223 DRV_KMODS       += clone
224 DRV_KMODS       += cmdk
225 DRV_KMODS       += cn
226 DRV_KMODS       += conskbd
227 DRV_KMODS       += consms
228 DRV_KMODS       += cpqary3
229 DRV_KMODS       += cpuid
230 DRV_KMODS       += cpunex
231 DRV_KMODS       += crypto
232 DRV_KMODS       += cryptoadm
233 DRV_KMODS       += dca
234 DRV_KMODS       += devinfo
235 DRV_KMODS       += dld
236 DRV_KMODS       += dlpistub
237 DRV_KMODS_32    += dnet
238 DRV_KMODS       += dump
239 DRV_KMODS       += ecpp
240 DRV_KMODS       += emlxs
241 DRV_KMODS       += fd
242 DRV_KMODS       += fdc
243 DRV_KMODS       += fm
244 DRV_KMODS       += fssnap
245 DRV_KMODS       += hxge
246 DRV_KMODS       += i8042
247 DRV_KMODS       += i915
248 DRV_KMODS       += icmp
249 DRV_KMODS       += icmp6
250 DRV_KMODS       += intel_nb5000
251 DRV_KMODS       += intel_nhm
252 DRV_KMODS       += ip
253 DRV_KMODS       += ip6
254 DRV_KMODS       += ipf
255 DRV_KMODS       += ipnet
256 DRV_KMODS       += ippctl
257 DRV_KMODS       += ipsecah
258 DRV_KMODS       += ipsecesp
259 DRV_KMODS       += ipw
```

```
260 DRV_KMODS       += iwh
261 DRV_KMODS       += iwi
262 DRV_KMODS       += iwk
263 DRV_KMODS       += iwp
264 DRV_KMODS       += iwscn
265 DRV_KMODS       += kb8042
266 DRV_KMODS       += keysock
267 DRV_KMODS       += kssl
268 DRV_KMODS       += kstat
269 DRV_KMODS       += ksyms
270 DRV_KMODS       += kmdb
271 DRV_KMODS       += llc1
272 DRV_KMODS       += lofi
273 DRV_KMODS       += log
274 DRV_KMODS       += logindmux
275 DRV_KMODS       += mega_sas
276 DRV_KMODS       += mc-amd
277 DRV_KMODS       += mm
278 DRV_KMODS       += mouse8042
279 DRV_KMODS       += mpt_sas
280 DRV_KMODS       += mr_sas
281 DRV_KMODS       += mwl
282 DRV_KMODS       += nca
283 DRV_KMODS       += nsmb
284 DRV_KMODS       += nulldriver
285 DRV_KMODS       += nv_sata
286 DRV_KMODS       += nxge
287 DRV_KMODS       += oce
288 DRV_KMODS       += openeepr
289 DRV_KMODS       += pci_pci
290 DRV_KMODS       += pcic
291 DRV_KMODS       += pcieb
292 DRV_KMODS       += physmem
293 DRV_KMODS       += pcan
294 DRV_KMODS       += pcwl
295 DRV_KMODS       += pit_beep
296 DRV_KMODS       += pm
297 DRV_KMODS       += poll
298 DRV_KMODS       += pool
299 DRV_KMODS       += power
300 DRV_KMODS       += pseudo
301 DRV_KMODS       += ptc
302 DRV_KMODS       += ptm
303 DRV_KMODS       += pts
304 DRV_KMODS       += ptsl
305 DRV_KMODS       += qlge
306 DRV_KMODS       += radeon
307 DRV_KMODS       += ral
308 DRV_KMODS       += ramdisk
309 DRV_KMODS       += random
310 DRV_KMODS       += rds
311 DRV_KMODS       += rdsv3
312 DRV_KMODS       += rpcib
313 DRV_KMODS       += rsm
314 DRV_KMODS       += rts
315 DRV_KMODS       += rtw
316 DRV_KMODS       += rum
317 DRV_KMODS       += rwd
318 DRV_KMODS       += rwn
319 DRV_KMODS       += sad
320 DRV_KMODS       += sd
321 DRV_KMODS       += sdhost
322 DRV_KMODS       += sgen
323 DRV_KMODS       += si3124
324 DRV_KMODS       += smbios
325 DRV_KMODS       += softmac
```



```
326 DRV_KMODS       += spdsock
327 DRV_KMODS       += smbsrv
328 DRV_KMODS       += smp
329 DRV_KMODS       += sppp
330 DRV_KMODS       += sppptun
331 DRV_KMODS       += srpt
332 DRV_KMODS       += st
333 DRV_KMODS       += sy
334 DRV_KMODS       += sysevent
335 DRV_KMODS       += sysmsg
336 DRV_KMODS       += tcp
337 DRV_KMODS       += tcp6
338 DRV_KMODS       += tl
339 DRV_KMODS       += tnf
340 DRV_KMODS       += tpm
341 DRV_KMODS       += trill
342 DRV_KMODS       += udp
343 DRV_KMODS       += udp6
344 DRV_KMODS       += ucode
345 DRV_KMODS       += ural
346 DRV_KMODS       += uath
347 DRV_KMODS       += urtw
348 DRV_KMODS       += vgatext
349 DRV_KMODS       += heci
350 DRV_KMODS       += vnic
351 DRV_KMODS       += vscan
352 DRV_KMODS       += wc
353 DRV_KMODS       += winlock
354 DRV_KMODS       += wpi
355 DRV_KMODS       += xge
356 DRV_KMODS       += yge
357 DRV_KMODS       += zcons
358 DRV_KMODS       += zyd
359 DRV_KMODS       += simnet
360 DRV_KMODS       += stmf
361 DRV_KMODS       += stmf_sbd
362 DRV_KMODS       += fct
363 DRV_KMODS       += fcoe
364 DRV_KMODS       += fcoet
365 DRV_KMODS       += fcoei
366 DRV_KMODS       += qlt
367 DRV_KMODS       += iscsit
368 DRV_KMODS       += pppt
369 DRV_KMODS       += ncall nsctl sdbc nskern sv
370 DRV_KMODS       += ii rdc rdcsrv rdcstub
371 DRV_KMODS       += iptun

373 $(CLOSED_BUILD)CLOSED_DRV_KMODS         += bmc
374 $(CLOSED_BUILD)CLOSED_DRV_KMODS         += glm
375 $(CLOSED_BUILD)CLOSED_DRV_KMODS         += intel_nhmex
376 $(CLOSED_BUILD)CLOSED_DRV_KMODS         += cpqary3
377 $(CLOSED_BUILD)CLOSED_DRV_KMODS         += marvell88sx
378 $(CLOSED_BUILD)CLOSED_DRV_KMODS         += bcm_sata
379 $(CLOSED_BUILD)CLOSED_DRV_KMODS         += memtest
380 $(CLOSED_BUILD)CLOSED_DRV_KMODS         += mpt
381 $(CLOSED_BUILD)CLOSED_DRV_KMODS         += atiatom
382 $(CLOSED_BUILD)CLOSED_DRV_KMODS         += acpi_toshiba

384 #
385 # Common code drivers
386 #

388 DRV_KMODS       += afe
389 DRV_KMODS       += atge
390 DRV_KMODS       += bfe
391 DRV_KMODS       += dmfe
```

```
392 DRV_KMODS       += e1000g
393 DRV_KMODS       += efe
394 DRV_KMODS       += elxl
395 DRV_KMODS       += hme
396 DRV_KMODS       += mxfe
397 DRV_KMODS       += nge
398 DRV_KMODS       += pcn
399 DRV_KMODS       += rge
400 DRV_KMODS       += rtls
401 DRV_KMODS       += sfe
402 DRV_KMODS       += amd8111s
403 DRV_KMODS       += igb
404 DRV_KMODS       += ipmi
405 DRV_KMODS       += iprb
406 DRV_KMODS       += ixgbe
407 DRV_KMODS       += vr
408 $(CLOSED_BUILD)CLOSED_DRV_KMODS += ixgb

410 #
411 # Virtio drivers
412 #

414 # Virtio core
415 DRV_KMODS       += virtio

417 # Virtio block driver
418 DRV_KMODS       += vioblk

420 #
421 #       DTrace and DTrace Providers
422 #
423 DRV_KMODS       += dtrace
424 DRV_KMODS       += fbt
425 DRV_KMODS       += lockstat
426 DRV_KMODS       += profile
427 DRV_KMODS       += sdt
428 DRV_KMODS       += systrace
429 DRV_KMODS       += fasttrap
430 DRV_KMODS       += dcpc

432 #
433 #       I/O framework test drivers
434 #
435 DRV_KMODS       += pshot
436 DRV_KMODS       += gen_drv
437 DRV_KMODS       += tvhci tphci tclient
438 DRV_KMODS       += emul64

440 #
441 #       Machine Specific Driver Modules (/kernel/drv):
442 #
443 DRV_KMODS       += options
444 DRV_KMODS       += scsi_vhci
445 DRV_KMODS       += pmcs
446 DRV_KMODS       += pmcs8001fw
447 DRV_KMODS       += arcmsr
448 DRV_KMODS       += fcp
449 DRV_KMODS       += fcip
450 DRV_KMODS       += fcsm
451 DRV_KMODS       += fp
452 DRV_KMODS       += qlc
453 DRV_KMODS       += iscsi

455 #
456 #       PCMCIA specific module(s)
457 #
```



```
458 DRV_KMODS       += pcs
459 DRV_KMODS       += pcata
460 MISC_KMODS      += cardbus
461 $(CLOSED_BUILD)CLOSED_DRV_KMODS += pcser

463 #
464 #       SCSI Enclosure Services driver
465 #
466 DRV_KMODS       += ses

468 #
469 #       USB specific modules
470 #
471 DRV_KMODS       += hid
472 DRV_KMODS       += hwarc hwahc
473 DRV_KMODS       += hubd
474 DRV_KMODS       += uhci
475 DRV_KMODS       += ehci
476 DRV_KMODS       += ohci
477 DRV_KMODS       += usb_mid
478 DRV_KMODS       += usb_ia
479 DRV_KMODS       += scsa2usb
480 DRV_KMODS       += usbprn
481 DRV_KMODS       += ugen
482 DRV_KMODS       += usbser
483 DRV_KMODS       += usbsacm
484 DRV_KMODS       += usbsksp
485 DRV_KMODS       += usbsprl
486 DRV_KMODS       += usb_ac
487 DRV_KMODS       += usb_as
488 DRV_KMODS       += usbskel
489 DRV_KMODS       += usbvc
490 DRV_KMODS       += usbftdi
491 DRV_KMODS       += wusb_df
492 DRV_KMODS       += wusb_ca
493 DRV_KMODS       += usbecm

495 $(CLOSED_BUILD)CLOSED_DRV_KMODS += usbser_edge

497 #
498 #       1394 modules
499 #
500 MISC_KMODS      += s1394 sbp2
501 DRV_KMODS       += hci1394 scsa1394
502 DRV_KMODS       += av1394
503 DRV_KMODS       += dcam1394

505 #
506 #       InfiniBand pseudo drivers
507 #
508 DRV_KMODS       += ib ibp eibnx eoib rdsib sdp iser daplt hermon tavor sol_ucma
509 DRV_KMODS       += sol_umad

511 #
512 #       LVM modules
513 #
514 DRV_KMODS       += md
515 MISC_KMODS      += md_stripe md_hotspares md_mirror md_raid md_trans md_notify
516 MISC_KMODS      += md_sp

518 #
519 #       Brand modules
520 #
521 BRAND_KMODS     += sn1_brand s10_brand

523 #
```

```
524 #       Exec Class Modules (/kernel/exec):
525 #
526 EXEC_KMODS      += elfexec intpexec shbinexec javaexec

528 #
529 #       Scheduling Class Modules (/kernel/sched):
530 #
531 SCHED_KMODS     += IA RT TS RT_DPTBL TS_DPTBL FSS FX FX_DPTBL SDC

533 #
534 #       File System Modules (/kernel/fs):
535 #
536 FS_KMODS        += autofs cachefs ctfs dcfs dev devfs fdfs fifofs hsfs lofs
537 FS_KMODS        += mntfs namefs nfs objfs zfs zut
538 FS_KMODS        += pcfs procfs sockfs specfs tmpfs udfs ufs sharefs
539 FS_KMODS        += smbfs

541 #
542 #       Streams Modules (/kernel/strmod):
543 #
544 STRMOD_KMODS    += bufmod connld dedump ldterm pckt pfmod pipemod
545 STRMOD_KMODS    += ptem redirmod rpcmod rlmod telmod timod
546 STRMOD_KMODS    += spppasyn spppcomp
547 STRMOD_KMODS    += tirdwr ttcompat
548 STRMOD_KMODS    += usbkbm
549 STRMOD_KMODS    += usbms
550 STRMOD_KMODS    += usbwcm
551 STRMOD_KMODS    += usb_ah
552 STRMOD_KMODS    += drcompat
553 STRMOD_KMODS    += cryptmod
554 STRMOD_KMODS    += vuid2ps2
555 STRMOD_KMODS    += vuid3ps2
556 STRMOD_KMODS    += vuidm3p
557 STRMOD_KMODS    += vuidm4p
558 STRMOD_KMODS    += vuidm5p

560 #
561 #       'System' Modules (/kernel/sys):
562 #
563 SYS_KMODS       += c2audit
564 SYS_KMODS       += doorfs
565 SYS_KMODS       += exacctsys
566 SYS_KMODS       += inst_sync
567 SYS_KMODS       += kaio
568 SYS_KMODS       += msgsys
569 SYS_KMODS       += pipe
570 SYS_KMODS       += portfs
571 SYS_KMODS       += pset
572 SYS_KMODS       += semsys
573 SYS_KMODS       += shmsys
574 SYS_KMODS       += sysacct
575 SYS_KMODS       += acctctl

577 #
578 #       'Misc' Modules (/kernel/misc)
579 #       MISC_KMODS are built both 32-bit and 64-bit
580 #       MISC_KMODS_32 are built only 32-bit
581 #       MISC_KMODS_64 are built only 64-bit
582 #
583 MISC_KMODS      += ac97
584 MISC_KMODS      += acpica
585 MISC_KMODS      += agpmaster
586 MISC_KMODS      += bignum
587 MISC_KMODS      += bootdev
588 MISC_KMODS      += busra
589 MISC_KMODS      += cmlb
```



```
590 MISC_KMODS      += consconfig
591 MISC_KMODS      += ctf
592 MISC_KMODS      += dadk
593 MISC_KMODS      += dcopy
594 MISC_KMODS      += dls
595 MISC_KMODS      += drm
596 MISC_KMODS      += fssnap_if
597 MISC_KMODS      += gda
598 MISC_KMODS      += gld
599 MISC_KMODS      += hidparser
600 MISC_KMODS      += hook
601 MISC_KMODS      += hpcsvc
602 MISC_KMODS      += ibcm
603 MISC_KMODS      += ibdm
604 MISC_KMODS      += ibdma
605 MISC_KMODS      += ibmf
606 MISC_KMODS      += ibtl
607 MISC_KMODS      += idm
608 MISC_KMODS      += idmap
609 MISC_KMODS      += iommulib
610 MISC_KMODS      += ipc
611 MISC_KMODS      += kbtrans
612 MISC_KMODS      += kcf
613 MISC_KMODS      += kgssapi
614 MISC_KMODS      += kmech_dummy
615 MISC_KMODS      += kmech_krb5
616 MISC_KMODS      += ksocket
617 MISC_KMODS      += mac
618 MISC_KMODS      += mii
619 MISC_KMODS      += mwlfw
620 MISC_KMODS      += net80211
621 MISC_KMODS      += nfs_dlboot
622 MISC_KMODS      += nfssrv
623 MISC_KMODS      += neti
624 MISC_KMODS      += pci_autoconfig
625 MISC_KMODS      += pcicfg
626 MISC_KMODS      += pcihp
627 MISC_KMODS      += pcmcia
628 MISC_KMODS      += rpcsec
629 MISC_KMODS      += rpcsec_gss
630 MISC_KMODS      += rsmops
631 MISC_KMODS      += sata
632 MISC_KMODS      += scsi
633 MISC_KMODS      += sda
634 MISC_KMODS      += sol_ofs
635 MISC_KMODS      += spuni
636 MISC_KMODS      += strategy
637 MISC_KMODS      += strplumb
638 MISC_KMODS      += tem
639 MISC_KMODS      += tlimod
640 MISC_KMODS      += usba usba10 usbs49_fw
641 MISC_KMODS      += scsi_vhci_f_sym_hds
642 MISC_KMODS      += scsi_vhci_f_sym
643 MISC_KMODS      += scsi_vhci_f_tpgs
644 MISC_KMODS      += scsi_vhci_f_asym_sun
645 MISC_KMODS      += scsi_vhci_f_tape
646 MISC_KMODS      += scsi_vhci_f_tpgs_tape
647 MISC_KMODS      += fctl
648 MISC_KMODS      += emlxs_fw
649 MISC_KMODS      += qlc_fw_2200
650 MISC_KMODS      += qlc_fw_2300
651 MISC_KMODS      += qlc_fw_2400
652 MISC_KMODS      += qlc_fw_2500
653 MISC_KMODS      += qlc_fw_6322
654 MISC_KMODS      += qlc_fw_8100
655 MISC_KMODS      += hwa1480_fw
```

```
656 MISC_KMODS      += uathfw
657 MISC_KMODS      += uwba

659 MISC_KMODS      += klmmod klmops

661 $(CLOSED_BUILD)CLOSED_MISC_KMODS        += scsi_vhci_f_asym_lsi
662 $(CLOSED_BUILD)CLOSED_MISC_KMODS        += scsi_vhci_f_asym_emc
663 $(CLOSED_BUILD)CLOSED_MISC_KMODS        += scsi_vhci_f_sym_emc

665 #
666 #       Software Cryptographic Providers (/kernel/crypto):
667 #
668 CRYPTO_KMODS    += aes
669 CRYPTO_KMODS    += arcfour
670 CRYPTO_KMODS    += blowfish
671 CRYPTO_KMODS    += des
672 CRYPTO_KMODS    += ecc
673 CRYPTO_KMODS    += edonr
674 CRYPTO_KMODS    += md4
675 CRYPTO_KMODS    += md5
676 CRYPTO_KMODS    += rsa
677 CRYPTO_KMODS    += sha1
678 CRYPTO_KMODS    += sha2
679 CRYPTO_KMODS    += skein
680 CRYPTO_KMODS    += swrand

682 #
683 #       IP Policy Modules (/kernel/ipp)
684 #
685 IPP_KMODS       += dlcosmk
686 IPP_KMODS       += flowacct
687 IPP_KMODS       += ipgpc
688 IPP_KMODS       += dscpmk
689 IPP_KMODS       += tokenmt
690 IPP_KMODS       += tswtclmt

692 #
693 #       generic-unix module (/kernel/genunix):
694 #
695 GENUNIX_KMODS   += genunix

697 #
698 #       SVVS Testing Modules (/kernel/strmod):
699 #
700 #       These are streams and driver modules which are not to be
701 #       delivered with a released system. However, during development
702 #       it is convenient to build and install the SVVS kernel modules.
703 #
704 SVVS_KMODS      += lmodb lmode lmodr lmodt svvslo tidg tivc tmux

706 $(CLOSED_BUILD)SVVS             += svvs

708 #
709 #       Modules eXcluded from the product:
710 #
711 $(CLOSED_BUILD)CLOSED_XMODS =           \
712         adpu320         \
713         bnx             \
714         bnxe            \
715         lsimega         \
716         sdpib


719 #
720 #       'Dacf' Modules (/kernel/dacf):
721 #
```



```
723 #
724 #       Performance Counter BackEnd modules (/usr/kernel/pcbe)
725 #
726 PCBE_KMODS      += p123_pcbe p4_pcbe opteron_pcbe core_pcbe

728 #
729 #       MAC-Type Plugin Modules (/kernel/mac)
730 #
731 MAC_KMODS       += mac_6to4
732 MAC_KMODS       += mac_ether
733 MAC_KMODS       += mac_ipv4
734 MAC_KMODS       += mac_ipv6
735 MAC_KMODS       += mac_wifi
736 MAC_KMODS       += mac_ib

738 #
739 # socketmod (kernel/socketmod)
740 #
741 SOCKET_KMODS    += sockpfp
742 SOCKET_KMODS    += socksctp
743 SOCKET_KMODS    += socksdp
744 SOCKET_KMODS    += sockrds
745 SOCKET_KMODS    += ksslf

747 #
748 #       kiconv modules (/kernel/kiconv):
749 #
750 KICONV_KMODS    += kiconv_emea kiconv_ja kiconv_ko kiconv_sc kiconv_tc

752 #
753 #       'Dacf' Modules (/kernel/dacf):
754 #
755 DACF_KMODS      += net_dacf
```

```
**********************************************************
    1995 Mon Oct  7 19:17:23 2013
new/usr/src/uts/intel/edonr/Makefile
4185 New hash algorithm support
**********************************************************
```

```
  1 #
  2 # CDDL HEADER START
  3 #
  4 # The contents of this file are subject to the terms of the
  5 # Common Development and Distribution License (the "License").
  6 # You may not use this file except in compliance with the License.
  7 #
  8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
  9 # or http://opensource.org/licenses/CDDL-1.0.
 10 # See the License for the specific language governing permissions
 11 # and limitations under the License.
 12 #
 13 # When distributing Covered Code, include this CDDL HEADER in each
 14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 15 # If applicable, add the following below this CDDL HEADER, with the
 16 # fields enclosed by brackets "[]" replaced with your own identifying
 17 # information: Portions Copyright [yyyy] [name of copyright owner]
 18 #
 19 # CDDL HEADER END
 20 #
 21 #
 22 # Copyright 2013 Saso Kiselkov.  All rights reserved.
 23 #
 24 #
 25 #       This makefile drives the production of the edonr kernel module.
 26 #
 27 #       intel architecture dependent
 28 #


 30 #
 31 #       Path to the base of the uts directory tree (usually /usr/src/uts).
 32 #
 33 UTSBASE = ../..
 34 COMDIR  = $(COMMONBASE)/crypto

 36 #
 37 #       Define the module and object file sets.
 38 #
 39 MODULE          = edonr
 40 OBJECTS         = $(EDONR_OBJS:%=$(OBJS_DIR)/%)
 41 LINTS           = $(EDONR_OBJS:%.o=$(LINTS_DIR)/%.ln)
 42 ROOTMODULE      = $(ROOT_CRYPTO_DIR)/$(MODULE)
 43 ROOTLINK        = $(ROOT_MISC_DIR)/$(MODULE)

 45 #
 46 #       Include common rules.
 47 #
 48 include $(UTSBASE)/intel/Makefile.intel

 50 #
 51 #       Define targets
 52 #
 53 ALL_TARGET      = $(BINARY)
 54 LINT_TARGET     = $(MODULE).lint
 55 INSTALL_TARGET  = $(BINARY) $(ROOTMODULE) $(ROOTLINK)

 57 #
 58 # Linkage dependencies
 59 #
 60 LDFLAGS += -dy -Nmisc/kcf
```



```
 62 CFLAGS += -I$(COMDIR)
 63 LINTFLAGS       += -I$(COMDIR)

 65 #
 66 #       Default build targets.
 67 #
 68 .KEEP_STATE:

 70 def:            $(DEF_DEPS)

 72 all:            $(ALL_DEPS)

 74 clean:          $(CLEAN_DEPS)

 76 clobber:        $(CLOBBER_DEPS)

 78 lint:           $(LINT_DEPS)

 80 modlintlib:     $(MODLINTLIB_DEPS)

 82 clean.lint:     $(CLEAN_LINT_DEPS)

 84 install:        $(INSTALL_DEPS)

 86 $(ROOTLINK):    $(ROOT_MISC_DIR) $(ROOTMODULE)
 87         -$(RM) $@; ln $(ROOTMODULE) $@

 89 #
 90 #       Include common targets.
 91 #
 92 include $(UTSBASE)/intel/Makefile.targ
```

```
**********************************************************
    2002 Mon Oct  7 19:17:23 2013
new/usr/src/uts/intel/skein/Makefile
4185 New hash algorithm support
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://opensource.org/licenses/CDDL-1.0.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2013 Saso Kiselkov.  All rights reserved.
#
#
#       This makefile drives the production of the skein crypto kernel module.
#
#       intel architecture dependent
#


#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..
COMDIR  = $(COMMONBASE)/crypto

#
#       Define the module and object file sets.
#
MODULE          = skein
OBJECTS         = $(SKEIN_OBJS:%=$(OBJS_DIR)/%)
LINTS           = $(SKEIN_OBJS:%.o=$(LINTS_DIR)/%.ln)
ROOTMODULE      = $(ROOT_CRYPTO_DIR)/$(MODULE)
ROOTLINK        = $(ROOT_MISC_DIR)/$(MODULE)

#
#       Include common rules.
#
include $(UTSBASE)/intel/Makefile.intel

#
#       Define targets
#
ALL_TARGET      = $(BINARY)
LINT_TARGET     = $(MODULE).lint
INSTALL_TARGET  = $(BINARY) $(ROOTMODULE) $(ROOTLINK)

#
# Linkage dependencies
#
LDFLAGS += -dy -Nmisc/kcf
```



```
CFLAGS += -I$(COMDIR)
LINTFLAGS       += -I$(COMDIR)

#
#       Default build targets.
#
.KEEP_STATE:

def:            $(DEF_DEPS)

all:            $(ALL_DEPS)

clean:          $(CLEAN_DEPS)

clobber:        $(CLOBBER_DEPS)

lint:           $(LINT_DEPS)

modlintlib:     $(MODLINTLIB_DEPS)

clean.lint:     $(CLEAN_LINT_DEPS)

install:        $(INSTALL_DEPS)

$(ROOTLINK):    $(ROOT_MISC_DIR) $(ROOTMODULE)
        -$(RM) $@; ln $(ROOTMODULE) $@

#
#       Include common targets.
#
include $(UTSBASE)/intel/Makefile.targ
```

```
**********************************************************
    2962 Mon Oct  7 19:17:23 2013
new/usr/src/uts/intel/zfs/Makefile
4185 New hash algorithm support
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#       This makefile drives the production of the zfs file system
#       kernel module.
#
# Copyright 2013 Saso Kiselkov. All rights reserved.
#

#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..

ARCHDIR:sh = cd ..; basename ‘pwd‘

#
#       Define the module and object file sets.
#
MODULE          = zfs
OBJECTS         = $(ZFS_OBJS:%=$(OBJS_DIR)/%)
LINTS           = $(ZFS_OBJS:%.o=$(LINTS_DIR)/%.ln)
ROOTMODULE      = $(ROOT_DRV_DIR)/$(MODULE)
ROOTLINK        = $(ROOT_FS_DIR)/$(MODULE)
CONF_SRCDIR     = $(UTSBASE)/common/fs/zfs

#
#       Include common rules.
#
include ../Makefile.$(ARCHDIR)

#
#       Define targets
#
ALL_TARGET      = $(BINARY) $(SRC_CONFILE)
LINT_TARGET     = $(MODULE).lint
INSTALL_TARGET  = $(BINARY) $(ROOTMODULE) $(ROOTLINK) $(ROOT_CONFFILE)

#
#       Overrides and depends_on
```



```
#
MODSTUBS_DIR     = $(OBJS_DIR)
LDFLAGS         += -dy -Nfs/specfs -Ncrypto/swrand -Nmisc/idmap -Nmisc/sha2 \
        -Nmisc/skein -Nmisc/edonr
LDFLAGS         += -dy -Nfs/specfs -Ncrypto/swrand -Nmisc/idmap -Nmisc/sha2

INC_PATH        += -I$(UTSBASE)/common/fs/zfs
INC_PATH        += -I$(SRC)/common
INC_PATH        += -I$(COMMONBASE)/zfs

C99MODE=        -xc99=%all
C99LMODE=       -Xc99=%all

#
# For now, disable these lint checks; maintainers should endeavor
# to investigate and remove these for maximum lint coverage.
# Please do not carry these forward to new Makefiles.
#
LINTTAGS        += -erroff=E_SUSPICIOUS_COMPARISON
LINTTAGS        += -erroff=E_BAD_PTR_CAST_ALIGN
LINTTAGS        += -erroff=E_SUPPRESSION_DIRECTIVE_UNUSED
LINTTAGS        += -erroff=E_STATIC_UNUSED
LINTTAGS        += -erroff=E_PTRDIFF_OVERFLOW
LINTTAGS        += -erroff=E_ASSIGN_NARROW_CONV

CERRWARN        += -_gcc=-Wno-type-limits
CERRWARN        += -_gcc=-Wno-switch
CERRWARN        += -_gcc=-Wno-parentheses
CERRWARN        += -_gcc=-Wno-unused-variable
CERRWARN        += -_gcc=-Wno-unused-function
CERRWARN        += -_gcc=-Wno-unused-label

#
#       Default build targets.
#
.KEEP_STATE:

def:            $(DEF_DEPS)

all:            $(ALL_DEPS)

clean:          $(CLEAN_DEPS)

clobber:        $(CLOBBER_DEPS)

lint:           $(LINT_DEPS)

modlintlib:     $(MODLINTLIB_DEPS)

clean.lint:     $(CLEAN_LINT_DEPS)

install:        $(INSTALL_DEPS)

$(ROOTLINK):    $(ROOT_FS_DIR) $(ROOTMODULE)
        -$(RM) $@; ln $(ROOTMODULE) $@

#
#       Include common targets.
#
include ../Makefile.targ
```

```
**********************************************************
   13811 Mon Oct  7 19:17:23 2013
new/usr/src/uts/sparc/Makefile.sparc.shared
4185 New hash algorithm support
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

# Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2013 Andrew Stormont.  All rights reserved.


#
#       This makefile contains the common definitions for all sparc
#       implementation architecture independent modules.
#


#
#       Define supported builds
#
DEF_BUILDS      = $(DEF_BUILDS64)
ALL_BUILDS      = $(ALL_BUILDS64)

#
#       Everybody needs to know how to build modstubs.o and to locate unix.o.
#       Note that unix.o must currently be selected from among the possible
#       "implementation architectures". Note further, that unix.o is only
#       used as an optional error check for undefines so (theoretically)
#       any "implementation architectures" could be used. We choose sun4u
#       because it is the reference port.
#
UNIX_DIR         = $(UTSBASE)/sun4u/unix
GENLIB_DIR       = $(UTSBASE)/sun4u/genunix
IPDRV_DIR        = $(UTSBASE)/sparc/ip
MODSTUBS_DIR     = $(UNIX_DIR)
DSF_DIR          = $(UNIX_DIR)
LINTS_DIR        = $(OBJS_DIR)
LINT_LIB_DIR     = $(UTSBASE)/sparc/lint-libs/$(OBJS_DIR)

UNIX_O           = $(UNIX_DIR)/$(OBJS_DIR)/unix.o
MODSTUBS_O       = $(MODSTUBS_DIR)/$(OBJS_DIR)/modstubs.o
GENLIB           = $(UTSBASE)/sun4u/lint-libs/$(OBJS_DIR)/libgenunix.so

LINT_LIB_32      = $(UTSBASE)/sun4u/lint-libs/$(OBJS_DIR)/llib-lunix.ln
GEN_LINT_LIB_32  = $(UTSBASE)/sun4u/lint-libs/$(OBJS_DIR)/llib-lgenunix.ln

LINT_LIB_64      = $(UTSBASE)/sun4u/lint-libs/$(OBJS_DIR)/llib-lunix.ln
GEN_LINT_LIB_64  = $(UTSBASE)/sun4u/lint-libs/$(OBJS_DIR)/llib-lgenunix.ln
```



```
LINT_LIB         = $(LINT_LIB_$(CLASS))
GEN_LINT_LIB     = $(GEN_LINT_LIB_$(CLASS))

LINT32_DIRS      = $(LINT32_BUILDS:%=$(UTSBASE)/sparc/lint-libs/%)
LINT32_FILES     = $(LINT32_DIRS:%=%/llib-l$(MODULE).ln)

LINT64_DIRS      = $(LINT64_BUILDS:%=$(UTSBASE)/sparc/lint-libs/%)
LINT64_FILES     = $(LINT64_DIRS:%=%/llib-l$(MODULE).ln)

#
#       Include the makefiles which define build rule templates, the
#       collection of files per module, and a few specific flags. Note
#       that order is significant, just as with an include path. The
#       first build rule template which matches the files name will be
#       used. By including these in order from most machine dependent
#       to most machine independent, we allow a machine dependent file
#       to be used in preference over a machine independent version
#       (Such as a machine specific optimization, which preserves the
#       interfaces.)
#
include $(UTSBASE)/sparc/Makefile.files
include $(UTSBASE)/sparc/v9/Makefile.files
include $(UTSTREE)/sun/Makefile.files
include $(UTSTREE)/common/Makefile.files

#
# ----- TRANSITIONAL SECTION --------------------------------------------------
#

#
#       Not everything which *should* be a module is a module yet. The
#       following is a list of such objects which are currently part of
#       genunix but which might someday become kmods.   This must be
#       defined before we include Makefile.uts, or else genunix's build
#       won't be as parallel as we might like.
#
NOT_YET_KMODS    = $(OLDPTY_OBJS) $(PTY_OBJS) $(VCONS_CONF_OBJS) $(MOD_OBJS)

#
# ----- END OF TRANSITIONAL SECTION -------------------------------------------
#
#       Include machine independent rules. Note that this does not imply
#       that the resulting module from rules in Makefile.uts is machine
#       independent. Only that the build rules are machine independent.
#
include $(UTSBASE)/Makefile.uts

#
#       machine specific optimization, override default in Makefile.master
#
XARCH_32         = -xarch=v8
XARCH_64         = -m64
XARCH            = $(XARCH_$(CLASS))

COPTIMIZE_32     = -xO3
COPTIMIZE_64     = -xO3
COPTIMIZE        = $(COPTIMIZE_$(CLASS))

CCMODE           = -Xa

CFLAGS_32        = -xcg92
CFLAGS_64        = -xchip=ultra $(CCABS32) $(CCREGSYM)
CFLAGS           = $(CFLAGS_$(CLASS))

CFLAGS           += $(XARCH)
```

```
128 CFLAGS          += $(COPTIMIZE)
129 CFLAGS          += $(EXTRA_CFLAGS)
130 CFLAGS          += $(XAOPT)
131 CFLAGS          += $(INLINES) -D_ASM_INLINES
132 CFLAGS          += $(CCMODE)
133 CFLAGS          += $(SPACEFLAG)
134 CFLAGS          += $(CERRWARN)
135 CFLAGS          += $(CTF_FLAGS_$(CLASS))
136 CFLAGS          += $(C99MODE)
137 CFLAGS          += $(CCUNBOUND)
138 CFLAGS          += $(CCSTATICSYM)
139 CFLAGS          += $(CC32BITCALLERS)
140 CFLAGS          += $(CCNOAUTOINLINE)
141 CFLAGS          += $(IROPTFLAG)
142 CFLAGS          += $(CGLOBALSTATIC)
143 CFLAGS          += -xregs=no%float
144 CFLAGS          += -xstrconst
145 CFLAGS          += $(CSOURCEDEBUGFLAGS)
146 CFLAGS          += $(CUSERFLAGS)

148 ASFLAGS         += $(XARCH)

150 LINT_DEFS_32    =
151 LINT_DEFS_64    = -m64
152 LINT_DEFS       += $(LINT_DEFS_$(CLASS))

154 #
155 #       The following must be defined for all implementations:
156 #
157 #       MODSTUBS:       Module stubs source file.
158 #
159 MODSTUBS         = $(UTSBASE)/sparc/ml/modstubs.s

161 #
162 #       Define the actual specific platforms - obviously none.
163 #
164 MACHINE_DEFS     =

166 #
167 #       Debugging level
168 #
169 #       Special knowledge of which special debugging options effect which
170 #       file is used to optimize the build if these flags are changed.
171 #
172 #       XXX: The above could possibly be done for more flags and files, but
173 #            is left as an experiment to the interested reader. Be forewarned,
174 #            that excessive use could lead to maintenance difficulties.
175 #
176 DEBUG_DEFS_OBJ32        =
177 DEBUG_DEFS_DBG32        = -DDEBUG
178 DEBUG_DEFS_OBJ64        =
179 DEBUG_DEFS_DBG64        = -DDEBUG
180 DEBUG_DEFS              = $(DEBUG_DEFS_$(BUILD_TYPE))

182 DEBUG_COND_OBJ32        = $(POUND_SIGN)
183 DEBUG_COND_DBG32        =
184 DEBUG_COND_OBJ64        = $(POUND_SIGN)
185 DEBUG_COND_DBG64        =
186 IF_DEBUG_OBJ            = $(DEBUG_COND_$(BUILD_TYPE))$(OBJS_DIR)/

188 $(IF_DEBUG_OBJ)syscall.o        :=      DEBUG_DEFS      += -DSYSCALLTRACE
189 $(IF_DEBUG_OBJ)clock.o          :=      DEBUG_DEFS      += -DKSLICE=1

191 # Comment these out if you don't want dispatcher lock statistics.

193 # $(IF_DEBUG_OBJ)disp_lock.o    := DEBUG_DEFS   += -DDISP_LOCK_STATS
```



```
195 #
196 #       Collect the preprocessor definitions to be associated with *all*
197 #       files.
198 #
199 ALL_DEFS         = $(MACHINE_DEFS) $(DEBUG_DEFS) $(OPTION_DEFS)
200 #
201 #
202 #       The kernels modules which are "implementation architecture"
203 #       specific for this machine are enumerated below. Note that most
204 #       of these modules must exist (in one form or another) for each
205 #       architecture.
206 #
207 #       Common Drivers (usually pseudo drivers) (/kernel/drv):
208 #
209 DRV_KMODS       += aggr arp audio bl blkdev bofi clone cn conskbd consms cpuid
210 DRV_KMODS       += crypto cryptoadm devinfo dump
211 DRV_KMODS       += dtrace fasttrap fbt lockstat profile sdt systrace dcpc
212 DRV_KMODS       += fssnap icmp icmp6 ip ip6 ipnet ipsecah
213 DRV_KMODS       += ipsecesp iptun iwscn keysock kmdb kstat ksyms llc1
214 DRV_KMODS       += lofi
215 DRV_KMODS       += log logindmux kssl mm nca physmem pm poll pool
216 DRV_KMODS       += pseudo ptc ptm pts ptsl ramdisk random rsm rts sad
217 DRV_KMODS       += simnet softmac sppp sppptun sy sysevent sysmsg
218 DRV_KMODS       += spdsock
219 DRV_KMODS       += tcp tcp6 tl tnf ttymux udp udp6 wc winlock zcons
220 DRV_KMODS       += ippctl
221 DRV_KMODS       += dld
222 DRV_KMODS       += ipf
223 DRV_KMODS       += rpcib
224 DRV_KMODS       += dlpistub
225 DRV_KMODS       += vnic
226 DRV_KMODS       += xge
227 DRV_KMODS       += rds
228 DRV_KMODS       += rdsv3
229 DRV_KMODS       += chxge
230 DRV_KMODS       += smbsrv
231 DRV_KMODS       += vscan
232 DRV_KMODS       += nsmb
233 DRV_KMODS       += fm
234 DRV_KMODS       += nulldriver
235 DRV_KMODS       += bridge trill
236 DRV_KMODS       += bpf
237 DRV_KMODS       += dca

239 $(CLOSED_BUILD)CLOSED_DRV_KMODS += glm
240 $(CLOSED_BUILD)CLOSED_DRV_KMODS += isp
241 $(CLOSED_BUILD)CLOSED_DRV_KMODS += mpt
242 $(CLOSED_BUILD)CLOSED_DRV_KMODS += qus
243 $(CLOSED_BUILD)CLOSED_DRV_KMODS += se

245 #
246 #       Hardware Drivers in common space
247 #

249 DRV_KMODS       += afe
250 DRV_KMODS       += audio1575
251 DRV_KMODS       += audioens
252 DRV_KMODS       += audiols
253 DRV_KMODS       += audiop16x
254 DRV_KMODS       += audiopci
255 DRV_KMODS       += audiots
256 DRV_KMODS       += e1000g
257 DRV_KMODS       += efe
258 DRV_KMODS       += hxge
259 DRV_KMODS       += mxfe
```

```
260 DRV_KMODS       += pcan
261 DRV_KMODS       += pcwl
262 DRV_KMODS       += rge
263 DRV_KMODS       += rtls
264 DRV_KMODS       += sfe
265 DRV_KMODS       += aac
266 DRV_KMODS       += igb
267 DRV_KMODS       += ixgbe
268 DRV_KMODS       += vr
269 DRV_KMODS       += mr_sas
270 $(CLOSED_BUILD)CLOSED_DRV_KMODS += ixgb
271 DRV_KMODS       += yge

273 #
274 #       Machine Specific Driver Modules (/kernel/drv):
275 #
276 DRV_KMODS       += audiocs
277 DRV_KMODS       += bge dmfe eri fas hme qfe
278 DRV_KMODS       += openeepr options sd ses st
279 DRV_KMODS       += ssd
280 DRV_KMODS       += ecpp
281 DRV_KMODS       += hid hubd ehci ohci uhci usb_mid usb_ia scsa2usb usbprn ugen
282 DRV_KMODS       += usbser usbsacm usbsksp usbsprl
283 DRV_KMODS       += usb_as usb_ac
284 DRV_KMODS       += usbskel
285 DRV_KMODS       += usbvc
286 DRV_KMODS       += usbftdi
287 DRV_KMODS       += wusb_df hwahc hwarc wusb_ca
288 DRV_KMODS       += usbecm
289 DRV_KMODS       += hci1394 av1394 scsa1394 dcam1394
290 DRV_KMODS       += sbp2
291 DRV_KMODS       += ib ibp eibnx eoib rdsib sdp iser daplt hermon tavor sol_ucma
292 DRV_KMODS       += sol_umad
293 DRV_KMODS       += pci_pci pcieb pcieb_bcm
294 DRV_KMODS       += i8042 kb8042 mouse8042
295 DRV_KMODS       += fcode
296 DRV_KMODS       += mpt_sas
297 DRV_KMODS       += socal
298 DRV_KMODS       += sgen
299 DRV_KMODS       += myri10ge
300 DRV_KMODS       += smp
301 DRV_KMODS       += dad
302 DRV_KMODS       += scsi_vhci
303 DRV_KMODS       += fcp
304 DRV_KMODS       += fcip
305 DRV_KMODS       += fcsm
306 DRV_KMODS       += fp
307 DRV_KMODS       += qlc
308 DRV_KMODS       += qlge
309 DRV_KMODS       += stmf
310 DRV_KMODS       += stmf_sbd
311 DRV_KMODS       += fct
312 DRV_KMODS       += fcoe
313 DRV_KMODS       += fcoet
314 DRV_KMODS       += fcoei
315 DRV_KMODS       += qlt
316 DRV_KMODS       += iscsit
317 DRV_KMODS       += pppt
318 DRV_KMODS       += ncall nsctl sdbc nskern sv
319 DRV_KMODS       += ii rdc rdcsrv rdcstub
320 DRV_KMODS       += iscsi
321 DRV_KMODS       += emlxs
322 DRV_KMODS       += oce
323 DRV_KMODS       += srpt
324 DRV_KMODS       += pmcs
325 DRV_KMODS       += pmcs8001fw
```



```
327 $(CLOSED_BUILD)CLOSED_DRV_KMODS += ifp
328 $(CLOSED_BUILD)CLOSED_DRV_KMODS += uata
329 $(CLOSED_BUILD)CLOSED_DRV_KMODS += usbser_edge

331 #
332 #       I/O framework test drivers
333 #
334 DRV_KMODS       += pshot
335 DRV_KMODS       += gen_drv
336 DRV_KMODS       += tvhci tphci tclient
337 DRV_KMODS       += emul64

339 #
340 # PCMCIA specific module(s)
341 #
342 DRV_KMODS       += pcs
343 MISC_KMODS      += busra cardbus dada pcmcia
344 DRV_KMODS       += pcata
345 DRV_KMODS       += pcic

347 $(CLOSED_BUILD)CLOSED_DRV_KMODS += pcser

349 # Add lvm
350 #
351 DRV_KMODS       += md
352 MISC_KMODS      += md_mirror md_stripe md_hotspares md_raid md_trans md_notify
353 MISC_KMODS      += md_sp

355 #
356 #       Exec Class Modules (/kernel/exec):
357 #
358 EXEC_KMODS      += aoutexec elfexec intpexec shbinexec javaexec

360 #
361 #       Scheduling Class Modules (/kernel/sched):
362 #
363 SCHED_KMODS     += RT TS RT_DPTBL TS_DPTBL IA FSS FX FX_DPTBL SDC

365 #
366 #       File System Modules (/kernel/fs):
367 #
368 FS_KMODS        += dev devfs fdfs fifofs hsfs lofs namefs nfs pcfs tmpfs zfs
369 FS_KMODS        += zut specfs udfs ufs autofs cachefs procfs sockfs mntfs
370 FS_KMODS        += ctfs objfs sharefs dcfs smbfs

372 #
373 #       Streams Modules (/kernel/strmod):
374 #
375 STRMOD_KMODS    += bufmod connld dedump ldterm ms pckt pfmod
376 STRMOD_KMODS    += pipemod ptem redirmod rpcmod rlmod telmod timod
377 STRMOD_KMODS    += spppasyn spppcomp
378 STRMOD_KMODS    += tirdwr ttcompat
379 STRMOD_KMODS    += usbkbm usbms usbwcm usb_ah
380 STRMOD_KMODS    += drcompat
381 STRMOD_KMODS    += cryptmod
382 STRMOD_KMODS    += vuid3ps2

384 #
385 #       'System' Modules (/kernel/sys):
386 #
387 SYS_KMODS       += c2audit
388 SYS_KMODS       += exacctsys
389 SYS_KMODS       += inst_sync kaio msgsys semsys shmsys sysacct pipe
390 SYS_KMODS       += doorfs pset acctctl portfs
```

```
392 #
393 #       'User' Modules (/kernel/misc):
394 #
395 MISC_KMODS      += ac97
396 MISC_KMODS      += bignum
397 MISC_KMODS      += consconfig gld ipc nfs_dlboot nfssrv scsi
398 MISC_KMODS      += strplumb swapgeneric tlimod
399 MISC_KMODS      += rpcsec rpcsec_gss kgssapi kmech_dummy
400 MISC_KMODS      += kmech_krb5
401 MISC_KMODS      += fssnap_if
402 MISC_KMODS      += hidparser kbtrans usba usba10 usbs49_fw
403 MISC_KMODS      += s1394
404 MISC_KMODS      += hpcsvc pcihp
405 MISC_KMODS      += rsmops
406 MISC_KMODS      += kcf
407 MISC_KMODS      += ksocket
408 MISC_KMODS      += ibcm
409 MISC_KMODS      += ibdm
410 MISC_KMODS      += ibdma
411 MISC_KMODS      += ibmf
412 MISC_KMODS      += ibtl
413 MISC_KMODS      += sol_ofs
414 MISC_KMODS      += idm
415 MISC_KMODS      += idmap
416 MISC_KMODS      += hook
417 MISC_KMODS      += neti
418 MISC_KMODS      += ctf
419 MISC_KMODS      += mac dls
420 MISC_KMODS      += cmlb
421 MISC_KMODS      += tem
422 MISC_KMODS      += pcicfg fcodem fcpci
423 MISC_KMODS      += scsi_vhci_f_sym scsi_vhci_f_tpgs scsi_vhci_f_asym_sun
424 MISC_KMODS      += scsi_vhci_f_sym_hds
425 MISC_KMODS      += scsi_vhci_f_tape scsi_vhci_f_tpgs_tape
426 MISC_KMODS      += fctl
427 MISC_KMODS      += emlxs_fw
428 MISC_KMODS      += qlc_fw_2200
429 MISC_KMODS      += qlc_fw_2300
430 MISC_KMODS      += qlc_fw_2400
431 MISC_KMODS      += qlc_fw_2500
432 MISC_KMODS      += qlc_fw_6322
433 MISC_KMODS      += qlc_fw_8100
434 MISC_KMODS      += spuni
435 MISC_KMODS      += hwa1480_fw uwba
436 MISC_KMODS      += mii

438 MISC_KMODS      += klmmod klmops

440 $(CLOSED_BUILD)CLOSED_MISC_KMODS        += scsi_vhci_f_asym_lsi
441 $(CLOSED_BUILD)CLOSED_MISC_KMODS        += scsi_vhci_f_asym_emc
442 $(CLOSED_BUILD)CLOSED_MISC_KMODS        += scsi_vhci_f_sym_emc

444 #
445 #       Software Cryptographic Providers (/kernel/crypto):
446 #
447 CRYPTO_KMODS    += aes
448 CRYPTO_KMODS    += arcfour
449 CRYPTO_KMODS    += blowfish
450 CRYPTO_KMODS    += des
451 CRYPTO_KMODS    += ecc
452 CRYPTO_KMODS    += edonr
453 CRYPTO_KMODS    += md4
454 CRYPTO_KMODS    += md5
453 CRYPTO_KMODS    += ecc
455 CRYPTO_KMODS    += rsa
456 CRYPTO_KMODS    += sha1
```



```
457 CRYPTO_KMODS    += sha2
458 CRYPTO_KMODS    += skein
459 CRYPTO_KMODS    += swrand

461 #
462 # IP Policy Modules (/kernel/ipp):
463 #
464 IPP_KMODS       += dlcosmk
465 IPP_KMODS       += flowacct
466 IPP_KMODS       += ipgpc
467 IPP_KMODS       += dscpmk
468 IPP_KMODS       += tokenmt
469 IPP_KMODS       += tswtclmt

471 #
472 # 'Dacf' modules (/kernel/dacf)
473 DACF_KMODS      += consconfig_dacf

475 #
476 #       SVVS Testing Modules (/kernel/strmod):
477 #
478 #       These are streams and driver modules which are not to be
479 #       delivered with a released system. However, during development
480 #       it is convenient to build and install the SVVS kernel modules.
481 #
482 SVVS_KMODS      += lmodb lmode lmodr lmodt svvslo tidg tivc tmux

484 $(CLOSED_BUILD)SVVS             += svvs

486 #
487 #       Modules eXcluded from the product:
488 #
489 XMODS           +=
490 $(CLOSED_BUILD)CLOSED_XMODS =           \
491         sdpib           \
492         wsdrv

494 #
495 #       'Dacf' Modules (/kernel/dacf):
496 #
497 DACF_KMODS      += net_dacf

499 #
500 #       MAC-Type Plugin Modules (/kernel/mac)
501 #
502 MAC_KMODS       += mac_6to4
503 MAC_KMODS       += mac_ether
504 MAC_KMODS       += mac_ipv4
505 MAC_KMODS       += mac_ipv6
506 MAC_KMODS       += mac_wifi
507 MAC_KMODS       += mac_ib

509 #
510 # socketmod (kernel/socketmod)
511 #
512 SOCKET_KMODS    += sockpfp
513 SOCKET_KMODS    += socksctp
514 SOCKET_KMODS    += socksdp
515 SOCKET_KMODS    += sockrds
516 SOCKET_KMODS    += ksslf

518 #
519 #       kiconv modules (/kernel/kiconv):
520 #
521 KICONV_KMODS    += kiconv_emea kiconv_ja kiconv_ko kiconv_sc kiconv_tc
```

```
**********************************************************
    1995 Mon Oct  7 19:17:23 2013
new/usr/src/uts/sparc/edonr/Makefile
4185 New hash algorithm support
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://opensource.org/licenses/CDDL-1.0.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # Copyright 2013 Saso Kiselkov.  All rights reserved.
  23 #
  24 #
  25 #       This makefile drives the production of the edonr kernel module.
  26 #
  27 #       sparc architecture dependent
  28 #

  30 #
  31 #       Path to the base of the uts directory tree (usually /usr/src/uts).
  32 #
  33 UTSBASE = ../..
  34 COMDIR  = $(COMMONBASE)/crypto

  36 #
  37 #       Define the module and object file sets.
  38 #
  39 MODULE          = edonr
  40 OBJECTS         = $(EDONR_OBJS:%=$(OBJS_DIR)/%)
  41 LINTS           = $(EDONR_OBJS:%.o=$(LINTS_DIR)/%.ln)
  42 ROOTMODULE      = $(ROOT_CRYPTO_DIR)/$(MODULE)
  43 ROOTLINK        = $(ROOT_MISC_DIR)/$(MODULE)

  45 #
  46 #       Include common rules.
  47 #
  48 include $(UTSBASE)/sparc/Makefile.sparc

  50 #
  51 #       Define targets
  52 #
  53 ALL_TARGET      = $(BINARY)
  54 LINT_TARGET     = $(MODULE).lint
  55 INSTALL_TARGET  = $(BINARY) $(ROOTMODULE) $(ROOTLINK)

  57 #
  58 # Linkage dependencies
  59 #
  60 LDFLAGS += -dy -Nmisc/kcf
```



```
  62 CFLAGS += -I$(COMDIR)
  63 LINTFLAGS       += -I$(COMDIR)

  65 #
  66 #       Default build targets.
  67 #
  68 .KEEP_STATE:

  70 def:            $(DEF_DEPS)

  72 all:            $(ALL_DEPS)

  74 clean:          $(CLEAN_DEPS)

  76 clobber:        $(CLOBBER_DEPS)

  78 lint:           $(LINT_DEPS)

  80 modlintlib:     $(MODLINTLIB_DEPS)

  82 clean.lint:     $(CLEAN_LINT_DEPS)

  84 install:        $(INSTALL_DEPS)

  86 $(ROOTLINK):    $(ROOT_MISC_DIR) $(ROOTMODULE)
  87         -$(RM) $@; ln $(ROOTMODULE) $@

  89 #
  90 #       Include common targets.
  91 #
  92 include $(UTSBASE)/sparc/Makefile.targ
```

```
**********************************************************
    2002 Mon Oct  7 19:17:24 2013
new/usr/src/uts/sparc/skein/Makefile
4185 New hash algorithm support
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://opensource.org/licenses/CDDL-1.0.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2013 Saso Kiselkov.  All rights reserved.
#
#
#       This makefile drives the production of the skein crypto kernel module.
#
#       sparc architecture dependent
#


#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..
COMDIR  = $(COMMONBASE)/crypto

#
#       Define the module and object file sets.
#
MODULE          = skein
OBJECTS         = $(SKEIN_OBJS:%=$(OBJS_DIR)/%)
LINTS           = $(SKEIN_OBJS:%.o=$(LINTS_DIR)/%.ln)
ROOTMODULE      = $(ROOT_CRYPTO_DIR)/$(MODULE)
ROOTLINK        = $(ROOT_MISC_DIR)/$(MODULE)

#
#       Include common rules.
#
include $(UTSBASE)/sparc/Makefile.sparc

#
#       Define targets
#
ALL_TARGET      = $(BINARY)
LINT_TARGET     = $(MODULE).lint
INSTALL_TARGET  = $(BINARY) $(ROOTMODULE) $(ROOTLINK)

#
# Linkage dependencies
#
LDFLAGS += -dy -Nmisc/kcf
```



```
CFLAGS += -I$(COMDIR)
LINTFLAGS       += -I$(COMDIR)

#
#       Default build targets.
#
.KEEP_STATE:

def:            $(DEF_DEPS)

all:            $(ALL_DEPS)

clean:          $(CLEAN_DEPS)

clobber:        $(CLOBBER_DEPS)

lint:           $(LINT_DEPS)

modlintlib:     $(MODLINTLIB_DEPS)

clean.lint:     $(CLEAN_LINT_DEPS)

install:        $(INSTALL_DEPS)

$(ROOTLINK):    $(ROOT_MISC_DIR) $(ROOTMODULE)
        -$(RM) $@; ln $(ROOTMODULE) $@

#
#       Include common targets.
#
include $(UTSBASE)/sparc/Makefile.targ
```

```
**********************************************************
    3086 Mon Oct  7 19:17:24 2013
new/usr/src/uts/sparc/zfs/Makefile
4185 New hash algorithm support
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#       This makefile drives the production of the zfs file system
#       kernel module.
#
# Copyright 2013 Saso Kiselkov. All rights reserved.
#

#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..

ARCHDIR:sh = cd ..; basename ‘pwd‘

#
#       Define the module and object file sets.
#
MODULE          = zfs
OBJECTS         = $(ZFS_OBJS:%=$(OBJS_DIR)/%)
LINTS           = $(ZFS_OBJS:%.o=$(LINTS_DIR)/%.ln)
ROOTMODULE      = $(ROOT_DRV_DIR)/$(MODULE)
ROOTLINK        = $(ROOT_FS_DIR)/$(MODULE)
CONF_SRCDIR     = $(UTSBASE)/common/fs/zfs

#
#       Include common rules.
#
include ../Makefile.$(ARCHDIR)

#
#       Define targets
#
ALL_TARGET      = $(BINARY) $(SRC_CONFILE)
LINT_TARGET     = $(MODULE).lint
INSTALL_TARGET  = $(BINARY) $(ROOTMODULE) $(ROOTLINK) $(ROOT_CONFFILE)

#
#       Overrides and depends_on
```



```
#
#       We require sched/SDC because by the time vfs_mountroot() runs,
#       we can no longer load modules through OBP.
#
MODSTUBS_DIR     = $(OBJS_DIR)
LDFLAGS         += -dy -Nfs/specfs -Ncrypto/swrand -Nmisc/idmap \
        -Nsched/SDC -Nmisc/sha2 -Nmisc/skein -Nmisc/edonr
        -Nsched/SDC -Nmisc/sha2

INC_PATH        += -I$(UTSBASE)/common/fs/zfs
INC_PATH        += -I$(SRC)/common
INC_PATH        += -I$(COMMONBASE)/zfs

C99MODE=        -xc99=%all
C99LMODE=       -Xc99=%all

#
# For now, disable these lint checks; maintainers should endeavor
# to investigate and remove these for maximum lint coverage.
# Please do not carry these forward to new Makefiles.
#
LINTTAGS        += -erroff=E_SUSPICIOUS_COMPARISON
LINTTAGS        += -erroff=E_BAD_PTR_CAST_ALIGN
LINTTAGS        += -erroff=E_SUPPRESSION_DIRECTIVE_UNUSED
LINTTAGS        += -erroff=E_STATIC_UNUSED
LINTTAGS        += -erroff=E_PTRDIFF_OVERFLOW
LINTTAGS        += -erroff=E_ASSIGN_NARROW_CONV

CERRWARN        += -_gcc=-Wno-type-limits
CERRWARN        += -_gcc=-Wno-switch
CERRWARN        += -_gcc=-Wno-parentheses
CERRWARN        += -_gcc=-Wno-unused-variable
CERRWARN        += -_gcc=-Wno-unused-function
CERRWARN        += -_gcc=-Wno-unused-label

#
#       Default build targets.
#
.KEEP_STATE:

def:            $(DEF_DEPS)

all:            $(ALL_DEPS)

clean:          $(CLEAN_DEPS)

clobber:        $(CLOBBER_DEPS)

lint:           $(LINT_DEPS)

modlintlib:     $(MODLINTLIB_DEPS)

clean.lint:     $(CLEAN_LINT_DEPS)

install:        $(INSTALL_DEPS)

$(ROOTLINK):    $(ROOT_FS_DIR) $(ROOTMODULE)
        -$(RM) $@; ln $(ROOTMODULE) $@

#
#       Include common targets.
#
include ../Makefile.targ
```