```
**********************************************************
   14268 Fri Jun 29 20:27:05 2018
new/usr/src/lib/fm/topo/modules/common/pcibus/pcibus_labels.c
9637 Memleak in pcibus.so`pci_slot_label_lookup.
**********************************************************
______unchanged_portion_omitted_

 216 /*
 217  * Do an overall slot label lookup for the device node.
 218  */
 219 char *
 220 pci_slot_label_lookup(topo_mod_t *mod, tnode_t *node, did_t *dp, did_t *pdp)
 221 {
 222         tnode_t *anode, *apnode;
 223         did_t *adp, *apdp;
 224         char *plat, *pp, *l = NULL, *ancestor_l = NULL, *new_l = NULL;
 224         char *plat, *pp, *l, *ancestor_l = NULL, *new_l = NULL;
 225         int err, b, d, f, done = 0;
 226         size_t len;

 228         did_BDF(dp, &b, &d, &f);

 230         topo_mod_dprintf(mod, "%s: entry: node=%p, node_name=%s, "
 231             "node_inst=%d, dp=%p, dp_bdf=%d/%d/%d, pdp=%p\n",
 232             __func__, node, topo_node_name(node), topo_node_instance(node),
 233             dp, b, d, f, pdp);

 235         /*
 236          * If this device has a physical slot number then check if
 237          * an ancestor also has a slot label.
 238          *
 239          * If an ancestor has a slot label, then this node's label
 240          * is generated by concatenating a default label onto the
 241          * ancestor's label.
 242          *
 243          * We grab pairs of ancestors (parent and child) as we go up
 244          * the tree because the parent is checked for the presence
 245          * of a slot while the child contains the label.
 246          *
 247          * Note that this algorithm only applies to nodes which have
 248          * a physcal slot number. (i.e. PCIE devices or PCI/PCIX
 249          * devices off of a PCIE to PCIX switch)
 250          */
 251         if (did_physlot(pdp) >= 0) {

 253                 topo_mod_dprintf(mod, "%s: node=%p: node has a physical "
 254                     "slot=%d, checking ancestors for slots\n",
 255                     __func__, node, did_physlot(pdp));

 257                 /*
 258                  * Get this device's physical slot name.
 259                  */
 260                 l = (char *)did_physlot_name(pdp, d);

 262                 anode = topo_node_parent(node);

 264                 /*
 265                  * Check ancestors for a slot label until we
 266                  * either find one or hit a non-pci device.
 267                  */
 268                 while (!done) {

 270                         /*
 271                          * Get next ancestor node and data pointers.
 272                          */
 273                         anode = topo_node_parent(anode);
```



```
 274                         if (anode != NULL) {
 275                                 adp = did_find(mod,
 276                                     topo_node_getspecific(anode));
 277                                 apnode = topo_node_parent(anode);
 278                                 if (apnode != NULL)
 279                                         apdp = did_find(mod,
 280                                             topo_node_getspecific(apnode));
 281                                 else
 282                                         apdp = NULL;
 283                         } else {
 284                                 apnode = NULL;
 285                                 apdp = adp = NULL;
 286                         }

 288                         topo_mod_dprintf(mod, "%s: node=%p: checking next "
 289                             "two ancestors: anode=%p, adp=%p "
 290                             "apnode=%p, apdp=%p\n",
 291                             __func__, node, anode, adp, apnode, apdp);
 292                         if ((anode != NULL) && (adp != NULL)) {
 293                                 did_BDF(adp, &b, &d, &f);
 294                                 topo_mod_dprintf(mod, "%s: node=%p: "
 295                                     "anode_name=%s[%d], anode_bdf=%d/%d/%d\n",
 296                                     __func__, node, topo_node_name(anode),
 297                                     topo_node_instance(anode), b, d, f);
 298                         }
 299                         if ((apnode != NULL) && (apdp != NULL)) {
 300                                 did_BDF(apdp, &b, &d, &f);
 301                                 topo_mod_dprintf(mod, "%s: node=%p: "
 302                                     "apnode_name=%s[%d], "
 303                                     "apnode_bdf=%d/%d/%d\n",
 304                                     __func__, node, topo_node_name(apnode),
 305                                     topo_node_instance(apnode), b, d, f);
 306                         }

 308                         /*
 309                          * If the ancestors do not exist or are not pci
 310                          * devices then we're done searching.
 311                          *
 312                          * Otherwise, if the ancestor has a physical slot,
 313                          * and it is a different slot than the one we
 314                          * started with then lookup the ancestor label,
 315                          * and we're done.
 316                          */
 317                         if ((anode == NULL) || (adp == NULL) ||
 318                             (apnode == NULL) || (apdp == NULL)) {
 319                                 done++;
 320                         } else if (did_physlot_exists(apdp) &&
 321                             (apdp != pdp)) {
 322                                 if (topo_node_label(anode, &ancestor_l,
 323                                     &err) != 0) {
 324                                         topo_mod_dprintf(mod,
 325                                             "%s: node=%p: topo_node_label() "
 326                                             "FAILED!", __func__, node);
 327                                         (void) topo_mod_seterrno(mod, err);
 328                                         return (NULL);
 329                                 }
 330                                 done++;
 331                                 topo_mod_dprintf(mod, "%s: node=%p: found "
 332                                     "ancestor with a slot, label=%s ",
 333                                     __func__, node, ancestor_l);
 334                         }
 335                 }
 336                 if (ancestor_l == NULL) {
 337                         topo_mod_dprintf(mod, "%s: node=%p: no ancestor "
 338                             "slot found\n", __func__, node);
 339                 }
```

```
340         }

342         /*
343          * If we found an ancestor with a slot label, and this node has
344          * a physical slot number label then concatenate the two to form
345          * this node's label. Otherwise, do a full slot label lookup.
346          */
347         if (ancestor_l && l) {
348                 topo_mod_dprintf(mod, "%s: node=%p: concatenating "
349                     "ancestor_l=%s and l=%s\n",
350                     __func__, node, ancestor_l, l);
351                 len = strlen(ancestor_l) + strlen(l) + 2;
352                 new_l = alloca(len);
353                 (void) snprintf(new_l, len, "%s/%s", ancestor_l, l);
354                 l = new_l;
355         } else if (topo_prop_get_string(node, FM_FMRI_AUTHORITY,
356             FM_FMRI_AUTH_PRODUCT, &plat, &err) == 0) {
355         } else {
357                 /*
357                  * Get platform name used for lookups.
358                  */
359                 if (topo_prop_get_string(node, FM_FMRI_AUTHORITY,
360                     FM_FMRI_AUTH_PRODUCT, &plat, &err) < 0) {
361                         (void) topo_mod_seterrno(mod, err);
362                         return (NULL);
363                 }
364                 /*
358                  * Trim SUNW, from the platform name
359                  */
360                 pp = strchr(plat, ',');
361                 if (pp == NULL)
362                         pp = plat;
363                 else
364                         ++pp;
365                 /*
366                  * Get device number used for lookup.
367                  */
368                 did_BDF(dp, NULL, &d, NULL);

370                 /*
371                  * The slot label is determined in the following order:
372                  * - Platform specific lookup based on physical slot #.
373                  * - Platform specific lookup based on default label string.
374                  * - Platform specific lookup based on device number.
375                  * - Default label.
376                  *   The default label is based on the slot names property
377                  *   if it exists, else it is a generic name derived from
378                  *   the slot #.
379                  */
380                 if ((l = (char *)pci_label_physlot_lookup(mod, pp, pdp))
381                     == NULL) {
382                         if ((l = (char *)did_physlot_name(pdp, d)) != NULL) {
383                                 l = (char *)
384                                     pci_label_slotname_lookup(mod, pp, l, dp);
385                         }
386                         if (l == NULL) {
387                                 l = (char *)
388                                     pci_label_missing_lookup(mod, pp, dp);
389                         }
390                 }
391                 topo_mod_strfree(mod, plat);
392         } else {
393                 (void) topo_mod_seterrno(mod, err);
394                 l = NULL;
395         }
```



```
397         topo_mod_strfree(mod, ancestor_l);

399         /*
400          * If we calculated a slot label,  then save it in the
401          * node's data structure so we can free it later.
402          */
403         if (l) {
404                 if (did_slot_label_get(dp) != NULL)
405                         topo_mod_strfree(mod, did_slot_label_get(dp));
406                 l = topo_mod_strdup(mod, l);
407                 did_slot_label_set(dp, l);
408         }

410         topo_mod_dprintf(mod, "%s: exit: node=%p: label=%s\n",
411             __func__, node, (l ? l : "NULL"));

413         return (l);
414 }
```

*_____unchanged_portion_omitted_*