Existing Changes to libresolv2

Analysis of differences between libresolv2 to ISC BIND

For **Sun Microsystems**

By **Ed Posnak, Venev Inc.**
415.254.0086
www.venev.com

version 0.3
April 10, 2009

Introduction

The goal of this project is to upgrade the OpenSolaris resolver library, libresolv2, to the latest version the ISC resolver library, which we'll refer to as ISC libbind (9)6.0[1]. This task is complicated by the facts that libresolv2 is based on a very old (circa 2003) version of ISC bind and Sun has made extensive changes to the old ISC code, many of which have not been incorporated into ISC libbind (9)6.0.

To perform the port without losing any of the original changes, each of the changes not in ISC's latest code must be identified and analyzed to determine whether it has been incorporated into ISC's latest code. If not, then we must evaluate, for each change, whether and how it should be added back.

This document provides a comprehensive enumeration of the changes that were made to ISC's libbind source code. Further, it identifies which changes are in ISC libbind (9)6.0, and which are not, and categorizes the latter as either *Solaris-specific* or *ISC*. *Solaris-specific* indicates changes that are inherent to peculiarities in Solaris and *ISC* indicates general changes that Sun will try to get incorporated into future releases of ISC bind.

The methodology for enumerating the changes was to perform a "smart" diff of the libresolv2 source code with the source code from ISC libbind 8.4.1[2]. We used ISC bind 8.4.1 because libresolv2 contains a mixture of files from ISC bind 4 up to bind 8.4.2, and it most closely resembled 8.4.1. Many of these diffs were previously documented in DNS_TOI.txt by Ulf Ekberg and that data has been incorporated here. However, the current code base contains some changes not previously documented there and some of the documented changes have been removed from the code.

We then examined the latest ISC source code to see if each change had been incorporated, and partitioned our list into those that were and those that are still outstanding.

For changes that were still outstanding, we categorized each as to whether it was inherent to Solaris or needed to be submitted to ISC for incorporation into future versions of bind. We also determined, based on our goal of making minimal changes to the ISC source code, whether or not the change was worth making.

The remainder of this document is organized as follows: an overview of the structural differences between libresolv2 and libbind is given. Next the major interface incompatibilities are discussed. Finally, an enumeration and analysis of all the changes is provided.

---

[1] The latest version of ISC libbind, based on b

ind 9, is 6.0. Since we are dealing with versions of bind prior to 9, we add the (9) to indicate that this version corresponds to bind 9 and is more recent than libbind 8.4.1, for example.

[2] ISC libbind 8.4.1 refers to the include and lib directories of bind 8.4.1

Structural Differences

This section describes structural relationships among libresolv2, libbind 8.4.1 and libbind (9)6.0, enumerates the differences in file sets, and provides some general mapping rules.

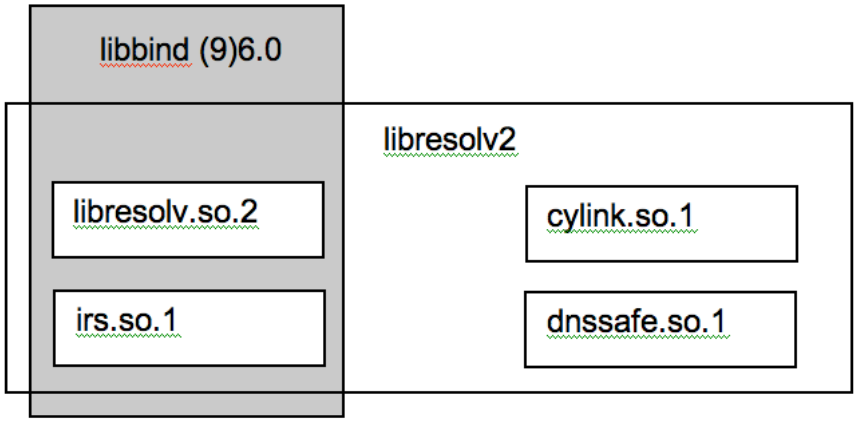
Library Packaging

Currently, libresolv2 consists of four dynamic libraries:

1. libresolv2.so.2: contains the client side functionality of the old libresolv library. Does not include IRS and crypto functions (described below).
2. irs.so.1 - - Information Retrieval Service - provides access to multiple information services
3. cylink.so.1 - CYLINK crypto library for DNSSEC/TSIG.
4. dnssafe.so.1- DNSSAFE crypto library for DNSSEC/TSIG

The splitting of libresolv2 into separate dynamic libraries, was apparently done in order to reduce the size increase of libresolv.so.2 for the on81 BIND 8.2.2-p5 port.

ISC libbind, has no such split; ISC packages all the functionality in a single library. Whereas the code for all four libraries were included in libbind 8.4.1, only the first two are in libbind (9)6.0. The figure below illustrates this relationship.



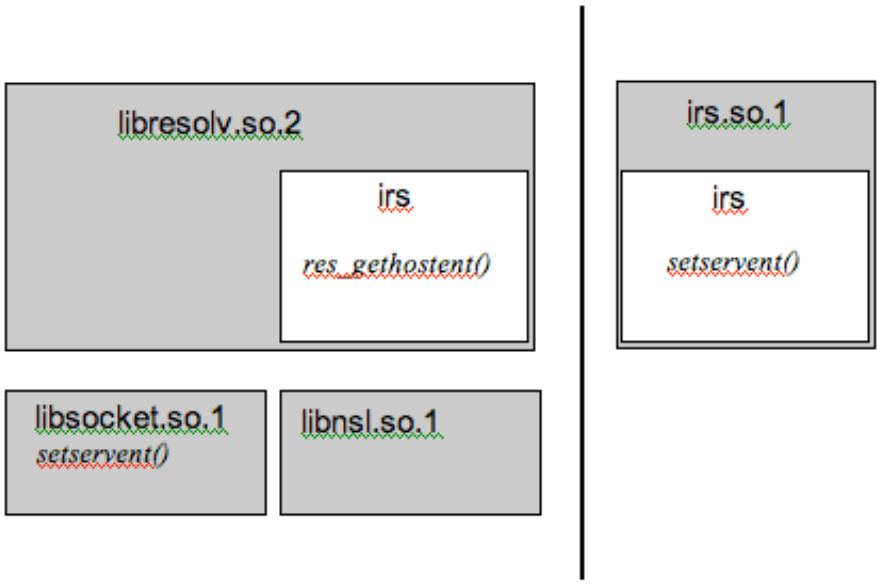
Splitting of irs

The irs functionality in ISC libbind was not completely extracted into irs.so.1. Instead, only about half of the irs functions were actually split out into irs.so.1; the other half remained in libresolv2.so.2. The criteria for determining which functions were split out is not documented, but the rationale appears to have something to do with overlap with libnsl and libsocket.

Many of the irs functions that overlap with libnsl and libsocket were moved into irs.so.1 (e.g. setservent), but some remained in libresolv2.so.2 and of those some were renamed (e.g. gethostent became res_gethostent). The result is the ISC code ends up calling the

libsocket version of some functions (e.g. setservent) but the ISC version of others (e.g. res_gethostent).

The situation with irs splitting is illustrated in the figure below.



Dependencies between the two pieces of irs cause a porting problem. The way this was dealt with in the last port was to write stub functions that reside in libresolv2. When called, the stubs load irs.so.1 and call the missing function in that library.

Consolidating the irs functions into libresolv2 might simplify future porting efforts.

Source Code Structure

The source tree of the latest version of ISC libbind is roughly a subset of older libbind, which is, in turn, a subset of libresolv2. At a high level, we have the following:

ISC libbind (9)6.0 contains the following components:

bsd - Berkeley Standard Distribution library functions
dst - Domain Signature Tools Library - a common interface to crypto libraries
inet - Internet IP address library - provides translations, etc.
irs - Information Retrieval Service - provides access to multiple information services
isc - Internet Software Consortium - general code for event handling, logging etc.
nameser - nameserver functions
resolv - high level resolver routines, e.g. res_query, etc.

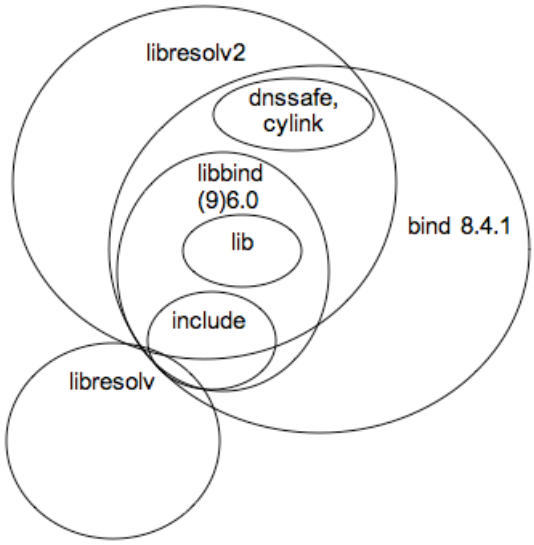
The older libbind 8.4.1 contains all of the above plus:

cylink - CYLINK crypto library (no longer included with ISC bind)
dnssafe - DNSSAFE crypto library (no longer included with ISC bind)

Sun's libresolv2 contains all of the above plus:

sunw - additional Sun-specific functions

Unfortunately, the subset/superset relationship is not exact; some files in ISC libbind are not in libresolv2 (and may reside elsewhere, such as libresolv, head/arpa, etc.). The following diagram shows a more accurate representation of the relationships.



Appendix A provides a more specific breakdown of the structural differences at the file level. The differences between libresolv2 and bind 8.4.1 are given, followed by differences between libresolv2 and libbind (9)6.0.

Interface Differences

Defining the libresolv2 Interface

The official definition of the libresolv2 public interface might be considered to reside in the file llib-lresolv (or mapfile-vers, which is fairly consistent with llib-lresolv).

Practically speaking, however, the interface is defined by the public header files <resolv.h>, <netdb.h>, <arpa/inet.h>, <arpa/nameser.h>, and <arpa/nameser_compat.h>. These public headers collectively make up a much larger interface, which is essentially a superset of the functions defined in llib-lresolv. The public headers would be a strict superset of llib-lresolv except for three things:

1. The functions res_nmkupdate and res_update are not found in the public headers
2. <resolv.h> has getlong and getshort instead of _getlong and _getshort.
3. <resolv.h> defines the argument to several functions as uchar_t*, not u_char*

Whereas most of the functions defined in the public headers are in libresolv2, some (e.g. rcmd, rexec, etc.) are not, so the public interface to libresolv2 is not just the contents of the public headers. For the purposes of this port, we take the intersection of the public headers with functions in libresolv2 as the existing public interface.

Interface Incompatibilities

ISC libbind has evolved over the years since the libresolv2 interface was defined. As a result there are many differences in the interfaces. The following is a condensed list highlighting the major incompatibilities between the interface provided by ISC libbind (9)6.0 and the existing libresolv2.

1. The ns_updrec structure has been updated. This structure is passed in many public functions.
2. res_state structure has been updated. This structure is passed in many public functions.
3. There are many data type inconsistencies that have, in previous ports, resulted in several workarounds applied in the ISC code base. A goal of this project is to get all of those fixes outside the ISC source code and into wrappers.
4. Information Retrieval Service (irs) is built into libbind, but in Solaris is a separate shared library from libresolv2.
5. In order to avoid run-time symbol conflicts between libnsl and libresolv, the libresolv2 library prepends "res_" to the names of some functions, which are public in libbind (9)6.0.

Enumeration of Changes

This section provides a comprehensive enumeration of changes made to ISC code by Sun. It is broken into four lists:

1. Documented changes not in the current libresolv2
2. Changes to ISC code that have been incorporated into libbind (9)6.0
3. Changes to ISC code that have not been incorporated into libbind (9)6.0

The first list defines changes that were previously documented but do not appear to be in the libresolv2 source tree. This list is provided for completeness and may be useful as it relates to other changes.

The second list represents changes that have apparently been accepted to ISC and incorporated into the latest version of ISC libbind. In theory, these should just work and will not require any further effort.

The third list represents the list of changes that may still need to be added for this port and possibly future upgrades. This list is further categorized by whether the change is Solaris-specific or should be incorporated into future ISC bind releases. For each of these changes a suggested action is given.

1. Documented Changes Not in libresolv2

These changes, documented in DNS_TOI.txt, are not found in the libresolv2 source tree.

1. Add new functions to the private and public interfaces,remove obsolete ones, and do some cleanup of cruft from the BIND 8.1.2 port. [4.3.2, 4.3.7, and 4.4.1.1 for the latter]

   NOTE: The spec directory does not exist in libresolv2

   Files:
   `spec/resolv2.spec`

2. Add new SUNW_2.2 public interface version for the MT-safe resolver. [4.3.1]

   NOTE: The spec directory does not exist in libresolv2

   Files:
   `spec/versions`

3. Silence compiler warning by changing "u_long" to "in_addr_t". ISC uses "u_int8_t" where Solaris defines "uint8_t".

   NOTE: The file inet/inet_addr.c does not exist in libresolv2. It does exist in ISC bind, but is completely compatible with <arpa/inet.h>

   Files:
   `inet/inet_addr.c`

4. SUNW_CANNOT_TRUST_POLL: We used to have problems in BIND 8.1.2 with poll(2) returning input indication for an fd, but read(2) or similar would find nothing and block. (see bug 4241665).

   NOTE: The SUNW_CANNOT_TRUST_POLL has been removed from ev_streams.c.

   Files:
   `isc/ev_streams.c`

5. Silence compiler warning regarding the 'strptr' argument of precsize_aton(), 'latlonstrptr' of latlon2ul(), and 'secs' of p_secstodate().

   NOTE: no change found

   Files:
   `resolv/res_debug.c`

6. SUNW_IMPLICIT_SEARCH: Restore implicit domain search even when the name to be looked up contains one or more (non-trailing) dots. SUNW_IMPLICIT_SEARCH - The Solaris port historically relies on the old resolver behavior where domain search paths are derived implicitly from the domain name (for example, if looking for "xyzzy.corp", the domain is "eng.sun.com", and ndots=1 (see resolver(3RESOLV)), we'd look for "xyzzy.corp", "xyzzy.corp.eng.sun.com", and "xyzzy.eng.sun.com", in that order). BIND 8.2 removed this type of implicit searching, but Solaris needs it back for backward compatibility.

   Files:
   `resolv/res_query.c`

7. SUNW_TRY_CONNECT_ALL: When there is more than one name server, the vanilla BIND code only tries a connect() to the first one. If we go on to the second (or higher) server, we can get a long timeout from send()/sendto(). A simple change allows connect() (and hence quicker detection of certain errors) to all name servers; see bug 4221898.

   NOTE: SUNW_TRY_CONNECT_ALL is no longer in libresolv2

   Files:
   `resolv/res_send.c`

8. Include synonyms.h, which makes us call (say) _open() instead of open(). The latter is subject to interposition by the application, and could have bad effects on the resolver. Include a few more system headers to avoid compiler warnings.

   Files:
   `include/port_before.h`
   `include/port_after.h`

9. Add #defines to prevent libresolv from picking up interposed versions of system/library routines. New for this project are thr_setspecific and thr_getspecific.

   Files:
   `usr/src/lib/libresolv2/include/synonyms.h`

10. Use switch_resolver_setup() and switch_resolver_reset()(see below) to set up and reset the resolver for nss_dns.so.1.

    Files:
    `usr/src/lib/nsswitch/dns/common/gethostent.c`
    `usr/src/lib/nsswitch/dns/common/gethostent6.c`

11. Add new 'unset_no_hosts_fallback' (the inverse of 'set_no_hosts_fallback; see bug 4198434) and 'override_retry' (rewrite of the DNS portion of the fix for 4260677 (originally 4181371)) function pointers (dynamically loaded from libresolv.so.2).

    Add new switch_resolver_setup() and switch_resolver_reset() routines to centralize setting up and resetting the resolver for use by nss_dns.so.1.

    Files:
    `usr/src/lib/nsswitch/dns/common/dns_common.h`
    `usr/src/lib/nsswitch/dns/common/dns_mt.c`
    `usr/src/cmd/cmd-`

12. Build and install the new dnskeygen(1M) utility. [2.3.1, 5.1.2]

    Files:
    `inet/usr.sbin/in.named/tools/dnskeygen/Makefile`

13. Build and install the new ndc(1M) utility. [2.3.2, 5.1.1]

    Files:
    `usr/src/cmd/cmd-net/usr.sbin/in.named/tools/ndc/Makefile`

2. Changes Already Implemented in ISC libbind 6.0

1. Undocumented changes to dst/hmac_link.c appear to have all been implemented

   Files:
   `dst/hmac_link.c`

2. Undocumented changes to getaddrinfo.c appear to have all been implemented

   Files:
   `irs/getaddrinfo.c`

3. Need to zero-fill the whole 'ctx' structure, not just the LHS and RHS fields. Otherwise, we may core dump when de-referencing garbage in (say) the 'res' field.

   Files:
   `irs/hesiod.c`

4. SUNW_POLL: Replace select(3C) with poll(2) in order to handle effectively unlimited number of file descriptors.

   Files:
   `isc/ev_files.c`

5. SUNW_POLL and SUNW_CANNOT_TRUST_POLL. In Solaris, FD_SETSIZE isn't necessarily a meaningful limit on the number of fd:s, so we need to use sysconf(_SC_OPEN_MAX).

   NOTE: SUNW_CANNOT_TRUST_POLL is no longer in libresolv2

   Files:
   `resolv/res_send.c`
   `isc/eventlib.c`
   `isc/eventlib_p.h`

6. Silence compiler warning.

   Files:
   `nameser/ns_parse.c`
   `resolv/herror.c`

7. Changes reflect replacing the private MT-safe interface with the BIND 8.2.x one (see sunw_mtctxres.c above).

   Files:
   `usr/src/lib/libresolv2/include/resolv_mt.h`

8. Returning pointer to a static buffer isn't MT-safe, so we replace with a pointer to TSD.

   Files:
   `inet/nsap_addr.c`

9. SUNW_MT_RESOLVER: Code needed to make libresolv completely (we hope) MT-safe. Static buffers become TSD for MT-safety.

   Files:
   `include/resolv_mt.h`

```
inet/nsap_addr.c
isc/memcluster.h
resolv/res_debug.c
```

10. SUNW_DOMAINFROMNIS: For backward compatibility, derive the domain name from sysinfo(SI_SRPC_DOMAIN) if not set in /etc/resolv.conf.

    Files:
    ```
    resolv/res_init.c
    ```

11. SUNW_RECV_ECONNREFUSED: For some reason, the vanilla BIND code doesn't consider ECONNREFUSED a serious problem (for UDP), and stupidly keeps on trying; see bug 4030292.

    Files:
    ```
    resolv/res_send.c
    ```

12. Need to change the sign of a return code in order to get a meaningful interpretation in p_rcode().

    Files:
    ```
    resolv/res_sendsigned.c
    ```

13. <sys/types.h> defines int8_t etc., so we must avoid doing that twice by checking if <sys/types.h> already has been included. The in64_t and u_int64_t typedef:s come from a previous version of BIND. They aren't used, and in fact are inside #if 0. Could be removed, but doesn't cause any trouble where they are, so I left them in place.

    Files:
    ```
    include/sys/bitypes.h
    ```

14. Undocumented changes to gai_strerror.c.

    NOTE: It appears that the libresolv2 version is just older.

    Category: ISC
    Files:
    ```
    irs/gai_strerror.c
    ```

15. Retry and retrans options. In previous releases, the ISC code did not offer a way to override the '_res' 'retry' and 'retrans' fields from/etc/resolv.conf.
    Files:
    ```
    resolv/res_init.c
    ```

16. Silence a compiler warning by correctly defining the NGR_R_END_RESULT macro.

    Files:
    ```
    include/port_before.h
    include/port_after.h
    ```

17. Undefine WANT_IRS_NIS. IRS largely parallels the NS switch, so we don't want it enabled by default in the resolver.

    Files:
    ```
    include/port_before.h
    include/port_after.h
    ```

3. Changes not in ISC libbind 6.0

The following list enumerates all the changes that have not been incorporated into ISC code.

Interface Changes

1. Some function names conflict with libnsl. Makefile.com renames these as:

   gethostbyname=res_gethostbyname
   gethostbyaddr=res_gethostbyaddr
   getnetbyname=res_getnetbyname
   gethostbyname2=res_gethostbyname2
   getnetbyaddr=res_getnetbyaddr
   sethostent=res_sethostent
   endhostent=res_endhostent
   gethostent=res_gethostent
   setnetent=res_setnetent
   endnetent=res_endnetent
   getnetent=res_getnetent
   socket=_socket
   getipnodebyname=res_getipnodebyname
   getipnodebyaddr=res_getipnodebyaddr
   freehostent=res_freehostent
   getaddrinfo=res_getaddrinfo
   freeaddrinfo=res_freeaddrinfo

   NOTE: many of these also have different signatures from OpenSolaris and have workarounds in the ISC code. We can use the unified adapter strategy (see GeneralPortingStrategy.doc) to get rid of all of those changes.

   Category: Solaris-Specific
   Action: Keep, but use unified strategy.
   Files:
   `Makefile.com`

2. Modified llib-lresolv to more closely reflect the actual resolver interface.

   NOTE: llib-resolv reflects only a subset of the functions defined in the public headers.

   Category: Solaris-Specific
   Action: Keep changes. Add new libbind (9)6.0 changes.
   Files:
   `llib-lresolv`

3. For BIND 8.2, ISC changed the ns_updrec struct, which, in Solaris, is defined in <arpa/nameser.h> . To retain the original interface, the ISC functions (res_update, res_mkupdate, etc.) were renamed in res_update.h and wrapped by functions defined in sunw_updrec.c.

   Category: Solaris-Specific

Action: Update the public headers to the new ns_updrec structure. This will break the interface. If not possible, update the wrapper to handle new struct.
Files:
```
sunw/sunw_updrec.c
include/res_update.h
```

4. Dummy function (never called) to avoid changing the mapfile depending on whether or not WANT_IRS_NIS is set.

   Category: Solaris-Specific
   Action: Drop this change (remove the file). The library is always built with WANT_IRS_NIS turned off.
   Files:
   ```
   sunw/sunw_irs_nis_acc.c
   irs/Makefile.com
   ```

5. Some function return 'int' on Solaris, but ‘void’ in ISC libbind.
   setnetgrent(3C)/endnetgrent(3C)
   setnetent(3SOCKET)/endnetent(3SOCKET)
   setprotoent(3SOCKET)/endprotoent(3SOCKET)
   setservent(3C)/endservent(3C)
   sethostent(3NSL) and endhosetent(3NSL)

   NOTE: Most of these are already being renamed to res_* versions to avoid conflict with libnsl.

   Category: Solaris-Specific
   Action: Keep, but use unified strategy to pull changes out of ISC code.
   Files:
   ```
   irs/getnetgrent.c
   irs/getnetent.c
   irs/getprotoent.c
   irs/getservent.c
   irs/gethostent.c
   include/port_after.h
   ```

6. SUNW_LIBNSL: Used to remove some routines that are implemented by the Solaris libnsl.so.1; see 4197284.

   Category: Solaris-Specific
   Action: Keep but remove by changing INET_OBJS in Makefile.com
   Files:
   ```
   inet/inet_ntoa.c
   inet/inet_ntop.c
   inet/inet_pton.c
   ```

7. Some ‘pragma weak’ directives were added to the ISC source.

   Category: Solaris-Specific
   Action: Keep changes for binary compatibility.
   Files:

```
resolv/res_send.c
resolv/res_data.c
resolv/res_debug.c
resolv/res_comp.c
```

8. Add ‘pragma weak’ to retain __b64_ntop and __b64_pton functions.

   Category: Solaris-Specific
   Action: Drop this change. (remove these private functions from mapfile-vers)
   Files:
   ```
   isc/base64.c
   ```

9. Some function prototypes in <resolv.h> use ‘uint_t’ (i.e. ‘unsigned int’), while ISC’s use ‘u_long’ (i.e. ‘unsigned long’).

   Category: Solaris-Specific
   Action: Update public headers (will break the interface) or keep this change
   Files:
   ```
   resolv/res_debug.c
   ```

10. Some function prototypes in <netdb.h> and <arpa/inet.h> use ‘in_addr_t’ while ISC’s use ‘u_long’ or ‘unsigned long.’

    Category: Solaris-Specific
    Action: Update public headers (will break the interface) or keep this change
    Files:
    ```
    irs/getnetent.c
    inet/inet_lnaof.c
    inet/inet_makeaddr.c
    inet/inet_netof.c
    inet/inet_network.c
    ```

11. Some function prototypes in <arpa/inet.h> and <netdb.h> use ‘socklen_t’ while ISC’s use ‘size_t.’

    Category: Solaris-Specific
    Action: Keep, but rename in port_after.h and provide a wrapper.
    Files:
    ```
    inet/inet_ntop.c
    irs/gethostent.c
    irs/getnameinfo.c
    ```

12. Get rid of “__func” #defines

    Category: Solaris-Specific
    Action: Drop changes, but ensure the public interface is supported.
    Files:
    ```
    isc/dst.h
    isc/tree.h
    ```

Structural Changes (and related implementation changes)

13. Add cylink, dnssafe, and irs subdirectories. Add EXPORT_SRC and CRYPT_SRC targets that delete all the cylink and dnssafe code.

    Category: Solaris-Specific
    Action: remove cylink and dnssafe, remove EXPORT_SRC and CRYPT_SRC targets
    Files:
    `Makefile`

14. Added copyright for the RSA Inc. files in the BIND distribution. Superceedes the copyright of the files in the dnssafe directory.

    Category: Solaris-Specific
    Action: remove dnssafe.
    Files:
    `dnssafe/README.copyright`

15. Build the /usr/lib/dns/cylink.so.1 and /usr/lib/dns/dnssafe.so.1 libraries. Add code to dynamically load the libraries (from /usr/lib/dns), and to execute the initialization functions for the respective encryption algorithms.

    Category: Solaris-Specific
    Action: remove cylink and dnssafe and associated changes.
    Files:
    `cylink/*`
    `dnssafe/*`
    `include/sunw_dst_init.h`
    `sunw/sunw_dst_init.c`

16. SC5.0 wants these constants to be 0x...U rather than 0x...L. (cylink is gone from ISC libbind)

    Category: Solaris-Specific
    Action: remove cylink.
    Files:
    `cylink/sha.c`
    `cylink/sha.h`

17. Some additional code in dsnsafe md5.h (dnssafe is gone from ISC libbind)

    Category: Solaris-Specific
    Action: remove dnssafe
    Files:
    `dnssafe/md5.h`

18. Build the /usr/lib/dns/irs.so.1 library. Add code to dynamically load the /usr/lib/dns/irs.so.1 library, and to execute the initialization functions.

    NOTE: The workaround in sunw_irs_init.c is to allow the functions irs_irp_acc and irs_nis_acc, which are not in libresolv.so.2 to be called from code within libresolv2.so2. The __SUNW_IRS_INIT_NODEFINE macro tells the compiler not to perform this mapping when it is making irs.so.1.

Category: Solaris-Specific
Action: Fold irs into libresolv2 and rename any functions that overlap libsocket and libnsl.
Files:

```
usr/src/lib/libresolv2/irs/*
usr/src/lib/libresolv2/include/sunw_irs_init.h
usr/src/lib/libresolv2/common/sunw/sunw_irs_init.c
```

Implementation Changes

19. Much of the private MT-safe resolver interface (see 1243174) used to live here. The BIND 8.2 resolver adds its own MT-safe interface; however, there are a few missing pieces to that MT-safety, so we need to retain parts of the thread-specific data (TSD) from the 1243174 fix, and to support Solaris binaries that used the old interface (DHCP dynamic DNS updates). Implement the 'set_no_hosts_fallback' and 'override_retry' fixes (see above as well as 4198434 and 4181371).

    Category: ISC
    Action: Keep only the parts not implemented yet by ISC. Some of this functionality appears to have been implemented in ISC (using<pthread.h> not <thread.h>) but some is not.
    Files:

    ```
    sunw/sunw_mtctxres.c
    irs/gethostent.c
    irs/gen.c
    ```

20. SUNW_INITCHKIF: Detect if there are no non-loopback interfaces so that we can fail quickly during boot when "dns" comes before "files" in nsswitch.conf. This fix was introduced a long time ago, and we carry on in the name of backward compatibility.

    Category: ISC
    Action: Keep changes.
    Files:

    ```
    resolv/res_init.c
    ```

21. SUNW_CONFCHECK and SUNW_AREWEINNAMED: Another hack to detect a broken configuration and avoid a timeout. SUNW_AREWEINNAMED is used to avoid triggering the SUNW_CONFCHECK code from inside the in.named.

    Category: ISC
    Action: Keep changes. Remove SUNW_AREWEINNAMED, which does nothing.
    Files:

    ```
    resolv/res_mkquery.c
    ```

22. SUNW_HOSTS_FALLBACK: When going through the NS switch, we must prevent libresolv from looking in /etc/hosts; 4198434.

    Category: Solaris-Specific
    Action: Keep changes.
    Files:

    ```
    irs/gen.c
    include/resolv_mt.h
    ```

```
sunw/sunw_mtctxres.c
```

23. SUNW_HNOK_UNDERSCORE: ISC's BIND disallows "_" in host names. While this is correct (they're illegal according to the relevant RFCs), Solaris has historically allowed underscores in host names (and so did various old BIND versions), so we do too, in the name of consistency and backward compatibility.

    Category: Solaris-Specific
    Action: Keep changes.
    Files:
    ```
    resolv/res_comp.c
    ```

24. SUNW_SETHERRNO: NS switch relies on h_errno always being set when there's an eror. On Solaris, we must use the extended ioctl's (SIOCGLIFCONF etc.) to get information for both IPv4 and IPv6 interfaces. Note that this code is used for the libresolv versions of getipnodebyname/getipnodebyaddr. While we don't expect customers to use them, they're useful during testing.

    NOTE: although the functions where this fix is applied have been renamed because they overlap with libnsl/libsocket we'll still keep them for testing.

    Category: ISC
    Action: Keep changes.
    Files:
    ```
    irs/gethostent.c
    ```

25. SUNW_OVERRIDE_RETRY: Surrounds code added to allow nss_dns.so.1 to override _res->retry (the switch wants to set it to one and do its own NSL retries; see 4181371).

    Category: ISC
    Action: Keep.
    Files:
    ```
    irs/gethostent.c
    include/resolv_mt.h
    sunw/sunw_mtctxres.c
    ```

26. SUNW_AVOIDSTDIO_FDLIMIT: Avoid 256 file descriptor limit in stdio

    Category: ISC
    Action: Drop. Not worth modifying ISC code.
    Files:
    ```
    isc/logging.c
    irs/lcl_ho.c
    resolv/res_init.c
    resolv/res_query.c
    ```

27. SUNW_LIBMD5: Use md5(3EXT) instead of internal implementation

    NOTE: ISC implemented this as HAVE_MD5, but they no longer provide dnssafe, so that file would have to be changed.

    Category: Solaris-specific (because dnssafe isn't part of ISC)
    Action: Keep, but use HAVE_MD5 not SUNW_LIBMD5.

Files:
`dst/hmac_link.c`
`dnssafe/md5.h`

28. We need RFC 1535 disabled in order to retain implicit search paths (for backward compatibility, of course).

    NOTE: no code changes to affect the behavior of "search" in /etc/resolv.conf

    Category: Solaris-Specific
    Action: No changes.
    Files:
    `resolv/res_init.c`

29. Undocumented changes: posix_getgrnam_r and posix_getpwnam_r have size_t instead of int for buflen:

    NOTE: this looks like a cut and paste bug.

    Category: N/A
    Action: Drop changes.
    Files:
    `irs/getgrent_r.c`
    `irs/getpwent_r.c`

30. Undocumented changes to irs_data.c. It appears that the diffs are because libresolv2 has an older version of this file.

    Category: N/A
    Action: Drop changes.
    Files:
    `irs/irs_data.c`

31. Add _IRS_DATA_H define so that it's safe to include this file multiple times.

    Category: ISC
    Action: Drop changes. Not worth modifying the source. Maybe send to ISC.
    Files:
    `irs/irs_data.h`

32. Silence compiler warning.

    Category: N/A
    Action: Drop changes. Not worth modifying the source.
    Files:
    `isc/tree.c`

33. Undocumented Change: add a function prototype.

    Category: N/A
    Action: Drop changes. Not necessary and not worth modifying the source.
    Files:
    `nameser/ns_print.c`

34. Undocumented change: functions irs_destroy and net_data_minimize removed from irs.h

    Category: N/A
    Action: Drop changes. Not necessary and not worth modifying the source.
    Files:

    `include/irs.h`

35. Make sure libresolv itself doesn't refer to __putlong and __putshort. [4.4.1.1].

    Category: Solaris-Specific
    Action: Drop changes, unless there is some rationale for thbis.
    Files:

    `include/port_before.h`
    `include/port_after.h`

36. SUNW_OPTIONS: Include the include file that contains all those other SUNW_-defines.

    Category: Solaris-Specific
    Action: Keep changes. Eliminate options no longer needed.
    Files:

    `include/port_before.h`

Appendix A: File Differences

File differences between libresolv2 and bind 8.4.1

The source code for the resolver library in ISC bind 8.4.1 was incorporated into libresolv2 under the following general mapping rules:

bind-8.4.1/src/lib -> libresolv2/common
bind-8.4.1/src/include -> libresolv2/include
bind-8.4.1/src/port/solaris/include -> libresolv2/include

However, there are some files that were not incorporated and libresolv2 contains some additional files not in bind 8.4.1. These changes are enumerated below:

Files in libresolv2 but not libbind-8.4.1

```
llib-lresolv
mapfile-vers
sunw/*
dnssafe/README.copyright
include/conf/sunoptions.h
include/err.h
include/inet_private.h
include/netdb_private.h
include/resolv_mt.h
include/sunw_dst_init.h
include/sunw_irs_init.h
```

Files in libbind-8.4.1 but not libresolv2

```
dnssafe/md5.c
dst/md5.h
dst/md5_dgst.c
dst/md5_locl.h
inet/inet_addr.c
isc/movefile.c
include/arpa/inet.h
include/arpa/nameser.h
include/arpa/nameser_compat.h
include/netdb.h
include/resolv.h
```

File differences between libresolv and bind (9)6.0

The source code for the new resolver library (9)6.0 maps to libresolv2 under the following general mapping rules:

libbind-6.0-rc1/[bsd|dst|inet|irs|isc|resolv] -> libresolv2/common
libbind-6.0-rc1/*.h -> libresolv2/include
libbind-6.0-rc1/include -> libresolv2/include
libbind-6.0-rc1/port/solaris/include -> libresolv2/include

The following shows the file-level differences between libresolv2 and libbind (9)6.0.

Files in libresolv2 but not libbind (9)6.0

```
llib-lresolv (for building a lint library)
mapfile-vers (library interface spec)
sunw/*
dnssafe/* (deleted crypto lib)
cylink/*  (deleted crypto lib)
dst/bsafe_link.c
dst/cylink_link.c
dst/eay_dss_link.c
dst/prandom.c
dst/rsaref_link.c
include/conf/sunoptions.h (Sun-specific #defines)
include/err.h
include/inet_private.h (ISC stuff not in head/inet.h)
include/netdb_private.h (ISC stuff not in head/netdb.h
include/prand_conf.h (used by prandom.c)
include/sunw_dst_init.h
include/sunw_irs_init.h
```

Files in libbind-9.6.0 but not in libresolv2

```
dst/md5.h
dst/md5_dgst.c
dst/md5_locl.h
inet/inet_addr.c
isc/movefile.c
isc/tree.mdoc
nameser/ns_newmsg.c
nameser/ns_rdata.c
resolv/mtctxres.c
include/arpa/inet.h
include/arpa/nameser.h
include/arpa/nameser_compat.h
include/netdb.h
include/resolv.h
```

Files generated by libbind-9.6.0 ‘configure’

```
include/port_after.h
include/port_before.h
include/config.h
include/isc/platform.h (not in libresolv2)
```

Appendix B: Source Code Differences

This section contains “smart” (non-comment, non-SCCS) diffs corresponding to the changes enumerated above. Note, the line numbers to do not correspond to source files because comments and SCCS lines were stripped out before diff was run.

```
------------------------------------------------------------------------
./cylink/sha.c
4,5d3
<
<
165,170c163,167
<
<    hash_context->state[0] = 0xEFCDAB89U;
<    hash_context->state[1] = 0x98BADCFEU;
<    hash_context->state[2] = 0x10325476U;
<    hash_context->state[3] = 0xC3D2E1F0U;
<    hash_context->state[4] = 0x67452301U;
---
>    hash_context->state[0] = 0xEFCDAB89L;
>    hash_context->state[1] = 0x98BADCFEL;
>    hash_context->state[2] = 0x10325476L;
>    hash_context->state[3] = 0xC3D2E1F0L;
>    hash_context->state[4] = 0x67452301L;
------------------------------------------------------------------------
./cylink/sha.h
16a15,18
> #define k1SHA  0x5a827999L
> #define k2SHA  0x6ed9eba1L
> #define k3SHA  0x8f1bbcdcL
> #define k4SHA  0xca62c1d6L
19,29c21,25
< #define k1SHA  0x5a827999U
< #define k2SHA  0x6ed9eba1U
< #define k3SHA  0x8f1bbcdcU
< #define k4SHA  0xca62c1d6U
<
<
< #define h0SHA  0x67452301U
< #define h1SHA  0xefcdab89U
< #define h2SHA  0x98badcfeU
< #define h3SHA  0x10325476U
< #define h4SHA  0xc3d2e1f0U
---
> #define h0SHA  0x67452301L
> #define h1SHA  0xefcdab89L
> #define h2SHA  0x98badcfeL
> #define h3SHA  0x10325476L
> #define h4SHA  0xc3d2e1f0L
------------------------------------------------------------------------
./dnssafe/ammd5.c
15,17c12,14
< static int amMD5Query PROTO_LIST ((unsigned int *, POINTER));
< static int amMD5Init PROTO_LIST ((POINTER, POINTER, A_SURRENDER_CTX*));
< static int amMD5Update PROTO_LIST
```

```
---
> static int MD5Query PROTO_LIST ((unsigned int *, POINTER));
> static int MD5Init PROTO_LIST ((POINTER, POINTER, A_SURRENDER_CTX*));
> static int MD5Update PROTO_LIST
19c16
< static int amMD5Final PROTO_LIST
---
> static int MD5Final PROTO_LIST
21c18
< static int amMD5GetMaxOutputLen PROTO_LIST ((POINTER, unsigned int *));
---
> static int MD5GetMaxOutputLen PROTO_LIST ((POINTER, unsigned int *));
24c21
<   amMD5Query, amMD5Init, amMD5Update, amMD5Final, amMD5GetMaxOutputLen
---
>   MD5Query, MD5Init, MD5Update, MD5Final, MD5GetMaxOutputLen
33c30
< static int amMD5Query (contextLen, params)
---
> static int MD5Query (contextLen, params)
44c41
< static int amMD5Init (context, params, surrenderContext)
---
> static int MD5Init (context, params, surrenderContext)
57c54
< static int amMD5Update (context, input, inputLen, surrenderContext)
---
> static int MD5Update (context, input, inputLen, surrenderContext)
70c67
< static int amMD5Final
---
> static int MD5Final
87c84
< static int amMD5GetMaxOutputLen (context, outputLen)
---
> static int MD5GetMaxOutputLen (context, outputLen)
------------------------------------------------------------------------
./dnssafe/md5.h
15,16d11
< #ifndef   SUNW_LIBMD5
<
27,36d21
< #else
<
< #include <sys/md5.h>
< #define   A_MD5_CTX         MD5_CTX
< #define   A_MD5Init(c)            MD5Init((c))
< #define   A_MD5Update(c, d, l)    MD5Update((c), (d), (l))
< #define   A_MD5Final(c, d)  MD5Final((d), (c))
<
< #endif
<
------------------------------------------------------------------------
./dst/hmac_link.c
3c7
< static const char rcsid[] = "$Header:
/proj/cvs/isc/bind8/src/lib/dst/hmac_link.c,v 1.10 2002/12/03 05:26:49
marka Exp $";
```

```
---
> static const char rcsid[] = "$Header:
/proj/cvs/isc/bind8/src/lib/dst/hmac_link.c,v 1.9 2001/05/29 05:48:10
marka Exp $";
22a27
> #ifndef   SUNW_LIBMD5
23a29,31
> #else
> #include <sys/md5.h>
> #endif
349a358
> #ifdef    ORIGINAL_ISC_CODE
350a360,362
> #else
> dst_md5_hmac_init()
> #endif
371,372d382
< #define   dst_hmac_md5_init __dst_hmac_md5_init
<



------------------------------------------------------------------------
./inet/inet_lnaof.c
18d15
< #ifdef    ORIGINAL_ISC_CODE
20,22d16
< #else
< in_addr_t
< #endif
26,30c20
< #ifdef    ORIGINAL_ISC_CODE
<     u_long i = ntohl(in.s_addr);
< #else
<     in_addr_t i = ntohl(in.s_addr);
< #endif
---
>     register u_long i = ntohl(in.s_addr);
------------------------------------------------------------------------
./inet/inet_makeaddr.c
21d17
< #ifdef    ORIGINAL_ISC_CODE
23,25d18
< #else
<     in_addr_t net, host;
< #endif
27d19
< #ifdef    ORIGINAL_ISC_CODE
29,32d20
< #else
<     in_addr_t addr;
<     struct in_addr inaddr;
< #endif
42d29
< #ifdef    ORIGINAL_ISC_CODE
45,48d31
< #else
<     inaddr.s_addr = htonl(addr);
```

```
<     return (inaddr);
< #endif
-------------------------------------------------------------------------
./inet/inet_netof.c
19d15
< #ifdef    ORIGINAL_ISC_CODE
21,23d16
< #else
< in_addr_t
< #endif
27d19
< #ifdef    ORIGINAL_ISC_CODE
29,31d20
< #else
<     register in_addr_t i = ntohl(in.s_addr);
< #endif
-------------------------------------------------------------------------
./inet/inet_network.c
20d16
< #ifdef    ORIGINAL_ISC_CODE
22,24d17
< #else
< in_addr_t
< #endif
28d20
< #ifdef    ORIGINAL_ISC_CODE
30,32d21
< #else
<     register in_addr_t val, base, n, i;
< #endif
34d22
< #ifdef    ORIGINAL_ISC_CODE
36,38d23
< #else
<     in_addr_t parts[4], *pp = parts;
< #endif
67d51
< #ifdef ORIGINAL_ISC_CODE
71,75d54
< #else
<     if (pp >= parts + 4 || val > 0xff)
<           return (INADDR_NONE);
<     if (*cp == '.') {
< #endif
-------------------------------------------------------------------------
./inet/inet_ntoa.c
24d20
< #ifndef SUNW_LIBNSL
33d28
< #endif
-------------------------------------------------------------------------
./inet/inet_ntop.c
32,33d28
< #ifndef   SUNW_LIBNSL
<
150d144
< #endif
-------------------------------------------------------------------------
```

```
./inet/inet_pton.c
26,27d22
< #ifndef   SUNW_LIBNSL
<
176d170
< #endif
-------------------------------------------------------------------------
./inet/nsap_addr.c
70,72d66
< #ifdef    SUNW_MT_RESOLVER
<     char  *tmpbuf = inet_nsap_ntoa_tmpbuf;
< #else
74d67
< #endif

-------------------------------------------------------------------------
./irs/gai_strerror.c
43c40,42
<     static pthread_key_t key = PTHREAD_ONCE_KEY_NP;
---
>     static pthread_mutex_t lock = PTHREAD_MUTEX_INITIALIZER;
>     static pthread_key_t key;
>     static int once = 0;
51,52c50,56
<     if (pthread_key_create_once_np(&key, free) != 0)
<           goto unknown;
---
>         if (!once) {
>                 pthread_mutex_lock(&lock);
>                 if (!once++)
>                         pthread_key_create(&key, free);
>                 pthread_mutex_unlock(&lock);
>         }
>
57,61c61,62
<                 goto unknown;
<           if (pthread_setspecific(key, buf) != 0) {
<                 free(buf);
<                 goto unknown;
<           }
---
>                         return ("unknown error");
>                 pthread_setspecific(key, buf);
67,71d67
<
< #ifdef    DO_PTHREADS
< unknown:
<     return ("unknown error");
< #endif
-------------------------------------------------------------------------
./irs/gen.c
36,39d32
< #ifdef SUNW_HOSTS_FALLBACK
< extern int __res_no_hosts_fallback(void);
< #endif
<
356,359d348
< #ifdef SUNW_HOSTS_FALLBACK
```

```
<     if (__res_no_hosts_fallback())
<           add_rule(irs, irs_ho, irs_dns, "");
<     else {
362,366d350
<     }
< #else
<     add_rule(irs, irs_ho, irs_dns, "continue");
<     add_rule(irs, irs_ho, irs_lcl, "");
< #endif
377,383d360
< #ifdef SUNW_HOSTS_FALLBACK
<     if (__res_no_hosts_fallback()) {
<           default_map_rules(irs);
<           return;
<     }
< #endif
<

--------------------------------------------------------------------------------
./irs/getaddrinfo.c
178,179d174
<
< #ifdef ORIGINAL_ISC_CODE
187,194d181
< #else
< #define ERR(err) \
< do { \
<      \
<     error = (err); \
<      \
< } while (0)
< #endif
265,270d251
< #ifndef ORIGINAL_ISC_CODE
< #ifdef __sparcv9
<
<     pai->_ai_pad = 0;
< #endif
< #endif
281c262
<               hints->ai_addr || hints->ai_next) {
---
>               hints->ai_addr || hints->ai_next)
283,289c264
< #if !defined(ORIGINAL_ISC_CODE)
<                 goto bad;
< #endif
<           }
<
< #ifdef ORIGINAL_ISC_CODE
<           if (hints->ai_flags & ~AI_MASK) {
---
>           if (hints->ai_flags & ~AI_MASK)
291,292d265
<           }
< #endif
300,302d272
< #if !defined(ORIGINAL_ISC_CODE)
```

```
<                 goto bad;
< #endif
306,311d275
< #ifndef ORIGINAL_ISC_CODE
< #ifdef __sparcv9
<
<           pai->_ai_pad = 0;
< #endif
< #endif
324,326d287
< #if !defined(ORIGINAL_ISC_CODE)
<                               goto bad;
< #endif
343,345d303
< #if !defined(ORIGINAL_ISC_CODE)
<           goto bad;
< #endif
368c326
<           if (error) {
---
>           if (error)
370,373d327
< #if !defined(ORIGINAL_ISC_CODE)
<                 goto bad;
< #endif
<           }
422c376
<     if (pai->ai_flags & AI_NUMERICHOST) {
---
>     if (pai->ai_flags & AI_NUMERICHOST)
424,428c378
< #if !defined(ORIGINAL_ISC_CODE)
<           goto bad;
< #endif
<     }
<     if (hostname == NULL) {
---
>     if (hostname == NULL)
430,433d379
< #if !defined(ORIGINAL_ISC_CODE)
<           goto bad;
< #endif
<     }
722c668
<                 } else {
---
>                 } else
724,727d669
< #if !defined(ORIGINAL_ISC_CODE)
<                       goto bad;
< #endif
<                 }
1025d966
<                 goto trynumeric;

--------------------------------------------------------------------------------
./irs/getgrent_r.c
7,8c4
```

```
< static const char rcsid[] =
<     "$Id: getgrent_r.c,v 8.7 2001/11/01 08:02:08 marka Exp $";
---
> static const char rcsid[] = "$Id: getgrent_r.c,v 8.7 2001/11/01
08:02:08 marka Exp $";
41c37
<           char *buf, size_t buflen, struct group **result) {
---
>           char *buf, int buflen, struct group **result) {
78c74
<           char *buf, size_t buflen, struct group **result) {
---
>           char *buf, int buflen, struct group **result) {

------------------------------------------------------------------------
./irs/gethostent.c
58,61d54
< #ifdef    SUNW_OVERRIDE_RETRY
< extern int      __res_retry(int);
< extern int      __res_retry_reset(void);
< #endif
80d72
< #ifdef ORIGINAL_ISC_CODE
82,84d73
< #else
< gethostbyaddr(const void *addr, socklen_t len, int af) {
< #endif
97d85
< #ifdef    ORIGINAL_ISC_CODE
99,101d86
< #else
< int
< #endif
105,108d89
< #ifdef    ORIGINAL_ISC_CODE
< #else
<     return (0);
< #endif
112d92
< #ifdef    ORIGINAL_ISC_CODE
114,116d93
< #else
< int
< #endif
120,123d96
< #ifdef    ORIGINAL_ISC_CODE
< #else
<     return (0);
< #endif
133,136d105
< #ifdef    SUNW_SETHERRNO
<     {
<
<           h_errno = NETDB_INTERNAL;
138,141d106
<     }
< #else
<           return (NULL);
```

```
< #endif
160,162d124
< #ifdef    SUNW_SETHERRNO
<     {
<           h_errno = NETDB_INTERNAL;
164,167d125
<     }
< #else
<           return (NULL);
< #endif
181,183d138
< #ifdef    SUNW_OVERRIDE_RETRY
<     net_data->res->retry = __res_retry(net_data->res->retry);
< #endif
185,187d139
< #ifdef    SUNW_OVERRIDE_RETRY
<     net_data->res->retry = __res_retry_reset();
< #endif
199,201d150
< #ifdef    SUNW_SETHERRNO
<     {
<           h_errno = NETDB_INTERNAL;
203,206d151
<     }
< #else
<           return (NULL);
< #endif
227,232d171
< #ifdef    SUNW_SETHERRNO
<     {
<           h_errno = NETDB_INTERNAL;
<           return (NULL);
<     }
< #else
234d172
< #endif
483a422,427
> #ifdef __hpux
> #define lifc_len iflc_len
> #define lifc_buf iflc_buf
> #define lifc_req iflc_req
> #define LIFCONF if_laddrconf
> #else
485a430,439
> #endif
>
> #ifdef __hpux
> #define lifr_addr iflr_addr
> #define lifr_name iflr_name
> #define lifr_dstaddr iflr_dstaddr
> #define lifr_flags iflr_flags
> #define ss_family sa_family
> #define LIFREQ if_laddrreq
> #else
486a441
> #endif
603a559,585
> #ifdef __linux
```

```
> #ifndef IF_NAMESIZE
> # ifdef IFNAMSIZ
> #  define IF_NAMESIZE  IFNAMSIZ
> # else
> #  define IF_NAMESIZE 16
> # endif
> #endif
> static void
> scan_linux6(int *have_v6) {
>     FILE *proc = NULL;
>     char address[33];
>     char name[IF_NAMESIZE+1];
>     int ifindex, prefix, flag3, flag4;
>
>     proc = fopen("/proc/net/if_inet6", "r");
>     if (proc == NULL)
>           return;
>
>     if (fscanf(proc, "%32[a-f0-9] %x %x %x %x %16s\n",
>              address, &ifindex, &prefix, &flag3, &flag4, name) == 6)
>           *have_v6 = 1;
>     fclose(proc);
>     return;
> }
> #endif
>
627a610,612
> #ifdef __linux
>     scan_linux6(have_v6);
> #endif
------------------------------------------------------------------------
./irs/getnameinfo.c
57d53
< #ifdef    ORIGINAL_ISC_CODE
59,61d54
< #else
<     socklen_t salen;
< #endif
63d55
< #ifdef    ORIGINAL_ISC_CODE
65,67d56
< #else
<     socklen_t hostlen;
< #endif
69d57
< #ifdef    ORIGINAL_ISC_CODE
71,73d58
< #else
<     socklen_t servlen;
< #endif
------------------------------------------------------------------------
./irs/getnetent.c
75d71
< #ifdef    ORIGINAL_ISC_CODE
77,79d72
< #else
< getnetbyaddr(in_addr_t net, int type) {
< #endif
```

```
85d77
< #ifdef    ORIGINAL_ISC_CODE
87,89d78
< #else
< int
< #endif
94,97d82
< #ifdef    ORIGINAL_ISC_CODE
< #else
<     return (0);
< #endif
101d85
< #ifdef    ORIGINAL_ISC_CODE
103,105d86
< #else
< int
< #endif
110,113d90
< #ifdef    ORIGINAL_ISC_CODE
< #else
<     return (0);
< #endif

------------------------------------------------------------------------
./irs/getnetgrent.c
41,42d37
<
< #ifdef    ORIGINAL_ISC_CODE
44,46d38
< #else
< int
< #endif
51,54d42
< #ifdef    ORIGINAL_ISC_CODE
< #else
<     return (0);
< #endif
57d44
< #ifdef    ORIGINAL_ISC_CODE
59,61d45
< #else
< int
< #endif
66,69d49
< #ifdef    ORIGINAL_ISC_CODE
< #else
<     return (0);
< #endif
83d62
< #ifdef    ORIGINAL_ISC_CODE
85c64
< getnetgrent(const char **host, const char **user, const char **domain)
{
---
> getnetgrent(char **host, char **user, char **domain) {
99,108d77
< #else
< int
```

```
< getnetgrent(char **host, char **user, char **domain) {
<     struct net_data *net_data = init();
<
<     return (getnetgrent_p((const char **)host, (const char **)user,
<           (const char **)domain, net_data));
<
< }
< #endif

------------------------------------------------------------------------
./irs/getprotoent.c
59d55
< #ifdef    ORIGINAL_ISC_CODE
61,63d56
< #else
< int
< #endif
68,71d60
< #ifdef    ORIGINAL_ISC_CODE
< #else
<     return (0);
< #endif
74d62
< #ifdef    ORIGINAL_ISC_CODE
76,78d63
< #else
< int
< #endif
83,86d67
< #ifdef    ORIGINAL_ISC_CODE
< #else
<     return (0);
< #endif

------------------------------------------------------------------------
./irs/getpwent_r.c
7,8c4
< static const char rcsid[] =
<     "$Id: getpwent_r.c,v 8.6 2001/11/01 08:02:15 marka Exp $";
---
> static const char rcsid[] = "$Id: getpwent_r.c,v 8.6 2001/11/01
08:02:15 marka Exp $";
80c76
<           char *buf, size_t buflen, struct passwd **result) {
---
>           char *buf, int buflen, struct passwd **result) {

------------------------------------------------------------------------
./irs/getservent.c
59d55
< #ifdef    ORIGINAL_ISC_CODE
61,63d56
< #else
< int
< #endif
68,71d60
< #ifdef    ORIGINAL_ISC_CODE
< #else
```

```
<     return (0);
< #endif
74d62
< #ifdef    ORIGINAL_ISC_CODE
76,78d63
< #else
< int
< #endif
83,86d67
< #ifdef    ORIGINAL_ISC_CODE
< #else
<     return (0);
< #endif
------------------------------------------------------------------------
./irs/hesiod.c
64d60
< #ifdef    ORIGINAL_ISC_CODE
68,70d63
< #else
<     memset(ctx, 0, sizeof (*ctx));
< #endif
------------------------------------------------------------------------
./irs/irs_data.c
23d19
< #include <stdlib.h>
29a26
> #include <stdlib.h>
41c38,39
< static pthread_key_t  key = PTHREAD_ONCE_KEY_NP;
---
> static pthread_key_t  key;
> static int            once = 0;
43c41
< static struct net_data      *net_data = NULL;
---
> static struct net_data      *net_data;
105a104
>     static pthread_mutex_t keylock = PTHREAD_MUTEX_INITIALIZER;
108,109c107,112
<     if (pthread_key_create_once_np(&key, net_data_destroy) != 0)
<           return (NULL);
---
>     if (!once) {
>           pthread_mutex_lock(&keylock);
>           if (!once++)
>                 pthread_key_create(&key, net_data_destroy);
>           pthread_mutex_unlock(&keylock);
>     }
118,121c121
<           if (pthread_setspecific(key, net_data) != 0) {
<                 net_data_destroy(net_data);
<                 return (NULL);
<           }
---
>           pthread_setspecific(key, net_data);
178a179,185
> #else
> #ifdef __linux
```

```
> struct __res_state *
> __res_state(void) {
>     return (&_res);
> }
> #endif

------------------------------------------------------------------------
./irs/irs_data.h
10,12d6
< #ifndef   _IRS_DATA_H
< #define   _IRS_DATA_H
<
52,53d45
<
< #endif
------------------------------------------------------------------------

./irs/lcl_ho.c
386,388d382
< #ifdef SUNW_AVOIDSTDIO_FDLIMIT
<     if (!(pvt->fp = fopen(_PATH_HOSTS, "rF")))
< #else
390d383
< #endif
------------------------------------------------------------------------
./isc/base64.c
30,35d26
< #ifdef    ORIGINAL_ISC_CODE
< #else
< #pragma weak    __b64_ntop  =     b64_ntop
< #pragma weak    __b64_pton  =     b64_pton
< #endif
<
------------------------------------------------------------------------
./isc/ev_files.c
47,48d43
< #ifdef    SUNW_POLL
< #else
51d45
< #endif
55,59d48
< #ifdef    SUNW_POLL
<
<     if (fd >= ctx->maxnfds)
<         evPollfdRealloc(ctx, 1, fd);
< #endif
116,117d104
< #ifdef    SUNW_POLL
< #else
124d110
< #endif
222,223d207
< #ifdef    SUNW_POLL
< #else
230d213
< #endif
------------------------------------------------------------------------
./isc/eventlib.c
```

```
18,20d14
< #ifdef    SUNW_POLL
< #include <limits.h>
< #endif
34,40d27
< #ifdef    SUNW_POLL
< #if defined(pselect)
< #undef pselect
< #endif
< #define   pselect     Pselect
< #endif
<
74,85d60
< #ifdef    SUNW_POLL
<     ctx->pollfds = 0;
<     ctx->maxnfds = 0;
<     ctx->firstfd = 0;
<     emulMaskInit(ctx, rdLast, EV_READ, 1);
<     emulMaskInit(ctx, rdNext, EV_READ, 0);
<     emulMaskInit(ctx, wrLast, EV_WRITE, 1);
<     emulMaskInit(ctx, wrNext, EV_WRITE, 0);
<     emulMaskInit(ctx, exLast, EV_EXCEPT, 1);
<     emulMaskInit(ctx, exNext, EV_EXCEPT, 0);
<     emulMaskInit(ctx, nonblockBefore, EV_WASNONBLOCKING, 0);
< #endif
93,95d67
< #ifdef    SUNW_POLL
<     ctx->highestFD = INT_MAX;
< #else
97d68
< #endif
101,103d71
< #ifdef    SUNW_POLL
<     ctx->fdTable = 0;
< #else
105d72
< #endif
299,301d265
< #ifdef    SUNW_POLL
<
< #else
306d269
< #endif
313,314d275
< #ifdef SUNW_POLL
< #else
323d283
< #endif
331,334d290
< #ifdef SUNW_POLL
<                 evPrintf(ctx, 4, "poll() returns %d (err: %s)\n",
<                       x, (x == -1) ? strerror(errno) : "none");
< #else
337d292
< #endif
663,668d617
< #ifdef SUNW_POLL
<     int         polltimeout = INFTIM;
```

```
<     evContext_p *ctx;
<     struct pollfd     *fds;
<     nfds_t            pnfds;
< #endif
673,675d621
< #ifdef SUNW_POLL
<           polltimeout = 1000*tv.tv_sec + tv.tv_usec/1000;
< #endif
680,713d625
< #ifdef SUNW_POLL
<
<     if (rfds != 0)
<           ctx = ((__evEmulMask *)rfds)->ctx;
<     else if (wfds != 0)
<           ctx = ((__evEmulMask *)wfds)->ctx;
<     else if (efds != 0)
<           ctx = ((__evEmulMask *)efds)->ctx;
<     else
<           ctx = 0;
<     if (ctx != 0) {
<           fds = &(ctx->pollfds[ctx->firstfd]);
<           pnfds = ctx->fdMax - ctx->firstfd + 1;
<     } else {
<           fds = 0;
<           pnfds = 0;
<     }
<     n = poll(fds, pnfds, polltimeout);
<
<     if (n > 0) {
<           int   i, e;
<           for (e = 0, i = ctx->firstfd; i <= ctx->fdMax; i++) {
<                 if (ctx->pollfds[i].fd < 0)
<                       continue;
<                 if (FD_ISSET(i, &ctx->rdLast))
<                       e++;
<                 if (FD_ISSET(i, &ctx->wrLast))
<                       e++;
<                 if (FD_ISSET(i, &ctx->exLast))
<                       e++;
<           }
<           n = e;
<     }
< #else
715d626
< #endif
723,831d633
<
< #ifdef SUNW_POLL
< void
< evPollfdRealloc(evContext_p *ctx, int pollfd_chunk_size, int fd) {
<
<     int   old_maxnfds = ctx->maxnfds;
<     int   i;
<
<     if (fd < old_maxnfds)
<           return;
<
<
```

```
<     if (pollfd_chunk_size < 20)
<           pollfd_chunk_size = 20;
<
<     ctx->maxnfds = (1 + (fd/pollfd_chunk_size)) * pollfd_chunk_size;
<
<     ctx->pollfds = realloc(ctx->pollfds,
<                         ctx->maxnfds * sizeof(*ctx->pollfds));
<     ctx->fdTable = realloc(ctx->fdTable,
<                         ctx->maxnfds * sizeof(*ctx->fdTable));
<
<     if (ctx->pollfds == 0 || ctx->fdTable == 0) {
<           evPrintf(ctx, 2, "pollfd() realloc (%lu) failed\n",
<                 ctx->maxnfds*sizeof(struct pollfd));
<           exit(1);
<     }
<
<     for (i = old_maxnfds; i < ctx->maxnfds; i++) {
<           ctx->pollfds[i].fd = -1;
<           ctx->pollfds[i].events = 0;
<           ctx->fdTable[i] = 0;
<     }
< }
<
<
< void
< evPollfdAdd(evContext opaqueCtx, int pollfd_chunk_size, int fd, short
events)
< {
< }
<
< void
< evPollfdDel(evContext opaqueCtx, int fd)
< {
< }
<
<
< short *
< __fd_eventfield(int fd, __evEmulMask *maskp) {
<
<     evContext_p *ctx = (evContext_p *)maskp->ctx;
<
<     if (!maskp->result || maskp->type == EV_WASNONBLOCKING)
<           return (&(ctx->pollfds[fd].events));
<     else
<           return (&(ctx->pollfds[fd].revents));
< }
<
<
< short
< __poll_event(__evEmulMask *maskp) {
<
<     switch ((maskp)->type) {
<     case EV_READ:
<           return (POLLRDNORM);
<     case EV_WRITE:
<           return (POLLWRNORM);
<     case EV_EXCEPT:
<           return (POLLRDBAND | POLLPRI | POLLWRBAND);
```

```
<     case EV_WASNONBLOCKING:
<           return (POLLHUP);
<     default:
<           return (0);
<     }
< }
<
<
< void
< __fd_clr(int fd, __evEmulMask *maskp) {
<
<     evContext_p *ctx = maskp->ctx;
<
<     *__fd_eventfield(fd, maskp) &= ~__poll_event(maskp);
<     if ((ctx->pollfds[fd].events & ~POLLHUP) == 0) {
<           ctx->pollfds[fd].fd = -1;
<           for ( ; ctx->fdMax > 0 && ctx->pollfds[ctx->fdMax].fd < 0;
<                 ctx->fdMax--);
<           for ( ; ctx->firstfd <= ctx->fdMax &&
<                 ctx->pollfds[ctx->firstfd].fd < 0;
<                 ctx->firstfd++);
<     }
< }
<
<
< void
< __fd_set(int fd, __evEmulMask *maskp) {
<
<     evContext_p *ctx = maskp->ctx;
<
<     *__fd_eventfield(fd, maskp) |= __poll_event(maskp);
<     if ((ctx->pollfds[fd].events & ~POLLHUP) != 0) {
<           ctx->pollfds[fd].fd = fd;
<           if (fd < ctx->firstfd || ctx->firstfd == 0)
<                 ctx->firstfd = fd;
<           if (fd > ctx->fdMax)
<                 ctx->fdMax = fd;
<     }
< }
< #endif
---------------------------------------------------------------------------
./isc/eventlib_p.h
50,54d46
< #ifdef SUNW_POLL
< #include <stropts.h>
< #include <poll.h>
< #endif
<
156,192d147
< #ifdef    SUNW_POLL
< typedef struct {
<     void        *ctx;
<     uint32_t    type;
<     uint32_t    result;
< } __evEmulMask;
<
< #define   emulMaskInit(ctx, field, ev, lastnext) \
<     ctx->field.ctx = ctx; \
```

```
<     ctx->field.type = ev; \
<     ctx->field.result = lastnext;
<
<
< #define   EV_WASNONBLOCKING 4000000001U
<
< extern short    *__fd_eventfield(int fd, __evEmulMask *maskp);
< extern short    __poll_event(__evEmulMask *maskp);
< extern void     __fd_clr(int fd, __evEmulMask *maskp);
< extern void     __fd_set(int fd, __evEmulMask *maskp);
<
< #undef    FD_ZERO
< #define   FD_ZERO(maskp)
<
< #undef    FD_SET
< #define   FD_SET(fd, maskp) \
<     __fd_set(fd, maskp)
<
< #undef    FD_CLR
< #define   FD_CLR(fd, maskp) \
<     __fd_clr(fd, maskp)
<
< #undef    FD_ISSET
< #define   FD_ISSET(fd, maskp) \
<     ((*__fd_eventfield(fd, maskp) & __poll_event(maskp)) != 0)
<
< #endif
<
204,216d158
< #ifdef SUNW_POLL
<     struct pollfd     *pollfds;
<     evFile            **fdTable;
<     int         maxnfds;
<     int         firstfd;
<     int         fdMax;
<     int         fdCount;
<     int         highestFD;
<     __evEmulMask      rdLast, rdNext;
<     __evEmulMask      wrLast, wrNext;
<     __evEmulMask      exLast, exNext;
<     __evEmulMask      nonblockBefore;
< #else
223d164
< #endif
244,247d184
< #ifdef    SUNW_POLL
< extern void evPollfdRealloc(evContext_p *ctx, int pollfd_chunk_size,
int fd);
< #endif
<
-------------------------------------------------------------------------
./isc/logging.c
41c38
<           (void)rename(old_name, new_name);
---
>           (void)isc_movefile(old_name, new_name);
44c41
<     (void)rename(chan->out.file.name, new_name);
```

```
---
>     (void)isc_movefile(chan->out.file.name, new_name);
73c70
<           regular = S_ISREG(sb.st_mode);
---
>           regular = (sb.st_mode & S_IFREG);
110,112d106
< #ifdef SUNW_AVOIDSTDIO_FDLIMIT
<     stream = fdopen(fd, "aF");
< #else
114d107
< #endif
------------------------------------------------------------------------
./isc/memcluster.c
80,94d77
< #ifdef    SUNW_MT_RESOLVER
< #include <thread.h>
< #include <synch.h>
< static mutex_t              memlock = DEFAULTMUTEX;
< #define   SUNW_MEMLOCK            (void)mutex_lock(&memlock)
< #define   SUNW_MEMUNLOCK          (void)mutex_unlock(&memlock)
< #define   SUNW_MEMLOCKBLOCK_BEGIN {
< #define   SUNW_MEMLOCKBLOCK_END   }
< #else
< #define   SUNW_MEMLOCK
< #define   SUNW_MEMUNLOCK
< #define   SUNW_MEMLOCKBLOCK_BEGIN
< #define   SUNW_MEMLOCKBLOCK_END
< #endif
<
176,177d158
<     SUNW_MEMLOCK;
<
184,185d164
< SUNW_MEMLOCKBLOCK_BEGIN
<                 SUNW_MEMUNLOCK;
187d165
< SUNW_MEMLOCKBLOCK_END
189d166
<           SUNW_MEMUNLOCK;
200d176
<                 SUNW_MEMUNLOCK;
212d187
<           SUNW_MEMUNLOCK;
218d192
<           SUNW_MEMUNLOCK;
234d207
<                       SUNW_MEMUNLOCK;
259d231
<                 SUNW_MEMUNLOCK;
316d287
<     SUNW_MEMUNLOCK;
343,344d313
<     SUNW_MEMLOCK;
<
353d321
<           SUNW_MEMUNLOCK;
397d364
```

```
<           SUNW_MEMUNLOCK;
431d397
<     SUNW_MEMUNLOCK;
458d423
<     SUNW_MEMLOCK;
460,461d424
< SUNW_MEMLOCKBLOCK_BEGIN
<           SUNW_MEMUNLOCK;
463d425
< SUNW_MEMLOCKBLOCK_END
490d451
<     SUNW_MEMUNLOCK;
---------------------------------------------------------------------------
./isc/tree.c
174d170
< #ifdef    ORIGINAL_ISC_CODE
176,178d171
< #else
<           RET(*ppr)
< #endif
---------------------------------------------------------------------------
./nameser/ns_parse.c
31d27
< #ifdef    ORIGINAL_ISC_CODE
33,35d28
< #else
< #define RETERR(err) { errno = (err); return (-1); }
< #endif
---------------------------------------------------------------------------
./nameser/ns_print.c
45,51d41
< #ifdef    ORIGINAL_ISC_CODE
< #else
<
<
< u_int16_t       dst_s_dns_key_id(const u_char *, const int);
< #endif
<
---------------------------------------------------------------------------
./resolv/res_comp.c
27,34d23
< #ifdef    ORIGINAL_ISC_CODE
< #else
< #pragma weak    __dn_skipname     =     dn_skipname
< #pragma weak    __res_dnok  =     res_dnok
< #pragma weak    __res_hnok  =     res_hnok
< #pragma weak    __res_mailok      =     res_mailok
< #pragma weak    __res_ownok =     res_ownok
< #endif
73,75d61
< #ifdef    SUNW_HNOK_UNDERSCORE
< #define   underscorechar(c) ((c) == 0x5f)
< #endif
83,85d68
< #ifdef    SUNW_HNOK_UNDERSCORE
< #define middlechar(c) (borderchar(c) || hyphenchar(c) ||
underscorechar(c))
< #else
```

```
87d69
< #endif
164,174d145
< #ifdef    ORIGINAL_ISC_CODE
< #else
< #ifdef    __putlong
< #undef    __putlong
< #endif
< #ifdef    __putshort
< #undef    __putshort
< #endif
< #pragma weak    putlong           =     __putlong
< #pragma weak    putshort    =     __putshort
< #endif
-------------------------------------------------------------------------

./resolv/res_data.c
33,40d29
< #ifdef    ORIGINAL_ISC_CODE
< #else
< #pragma weak    __fp_nquery =     fp_nquery
< #pragma weak    __fp_query  =     fp_query
< #pragma weak    __p_query   =     p_query
< #pragma weak    __hostalias =     hostalias
< #endif
<
-------------------------------------------------------------------------
./resolv/res_debug.c
<#ifdef     ORIGINAL_ISC_CODE
>#else
>#pragma weak     __dn_count_labels =     dn_count_labels
>#pragma weak     __fp_resstat            =     fp_resstat
>#pragma weak     __loc_aton        =     loc_aton
>#pragma weak     __loc_ntoa        =     loc_ntoa
>#pragma weak     __p_cdname        =     p_cdname
>#pragma weak     __p_class         =     p_class
>#pragma weak     __p_secstodate          =     p_secstodate
>#pragma weak     __p_section       =     p_section
>#pragma weak     __p_time          =     p_time
>#pragma weak     __p_type          =     p_type
>#pragma weak     __sym_ntop        =     sym_ntop
>#pragma weak     __sym_ntos        =     sym_ntos
>#pragma weak     __sym_ston        =     sym_ston
>#endif     /* ORIGINAL_ISC_CODE */
392,394d386
< #ifdef    SUNW_MT_RESOLVER
<     char  *unname = sym_ntos_unname;
< #else
396d387
< #endif
414,416d404
< #ifdef    SUNW_MT_RESOLVER
<     char  *unname = sym_ntop_unname;
< #else
418d405
< #endif
483d469
< #ifdef    ORIGINAL_ISC_CODE
```

```
485,490d470
< #else
< p_option(u_int option) {
< #endif
< #ifdef    SUNW_MT_RESOLVER
<     char  *nbuf = p_option_nbuf;
< #else
492d471
< #endif
521c500
< #ifdef    ORIGINAL_ISC_CODE
---
>
524,526d502
< #else
<     default:          sprintf(nbuf, "?0x%x?", (u_int)option);
< #endif
534,536d509
< #ifdef    SUNW_MT_RESOLVER
<     char  *nbuf = p_time_nbuf;
< #else
538d510
< #endif
586,588d557
< #ifdef    SUNW_MT_RESOLVER
<     char  *retbuf = precsize_ntoa_retbuf;
< #else
590d558
< #endif
856,858d823
< #ifdef    SUNW_MT_RESOLVER
<     char  *tmpbuf = loc_ntoa_tmpbuf;
< #else
861d825
< #endif
991d954
< #ifdef    ORIGINAL_ISC_CODE
993,998d955
< #else
< p_secstodate (u_int secs) {
< #endif
< #ifdef    SUNW_MT_RESOLVER
<     char  *output = p_secstodate_output;
< #else
1001d957
< #endif
------------------------------------------------------------------------
./resolv/res_init.c
44,60d40
< #ifdef  SUNW_INITCHKIF
< #include <net/if.h>
< #include <netinet/if_ether.h>
< #include <sys/sockio.h>
< #define MAXIFS  8192
< #endif
<
< #ifdef  SUNW_DOMAINFROMNIS
< #include <sys/systeminfo.h>
```

```
< #include <string.h>
< #endif
<
< #ifdef    ORIGINAL_ISC_CODE
< #else
< #pragma weak    __res_randomid    =     res_randomid
< #endif
<
100,102d79
<     if (statp->_u._ext.ext != NULL)
<           res_ndestroy(statp);
<
109a87,89
>     if ((statp->options & RES_INIT) != 0)
>           res_ndestroy(statp);
>
149,150c129
<     } else
<           return (-1);
---
>     }
156,249d134
< #ifdef  SUNW_INITCHKIF
<
<     {
<           int s;
<           struct lifnum lifn;
<
<           if ((s = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
<                 perror("res_init: socket");
<                 goto freedata;
<           }
<           lifn.lifn_family = AF_UNSPEC;
<           lifn.lifn_flags = LIFC_EXTERNAL_SOURCE;
<           if (ioctl(s, SIOCGLIFNUM, (char *)&lifn) < 0) {
<                 close(s);
<                 goto freedata;
<           }
<           if (lifn.lifn_count == 0) {
<
<                 struct lifconf lifc;
<                 uchar_t *buf;
<                 int buflen, i, int_up = 0;
<
<                 lifn.lifn_flags = 0;
<                 if ((ioctl(s, SIOCGLIFNUM, (char *)&lifn) < 0) ||
<                             (lifn.lifn_count < 1)) {
<                       close(s);
<                       goto freedata;
<                 }
<
<                 buflen = lifn.lifn_count * sizeof (struct lifreq);
<                 buf = (uchar_t *)malloc(buflen);
<                 if (buf == NULL) {
<                       close(s);
<                       goto freedata;
<                 }
<
```

```
<                 lifc.lifc_family = AF_UNSPEC;
<                 lifc.lifc_flags = 0;
<                 lifc.lifc_len = buflen;
<                 lifc.lifc_lifcu.lifcu_buf = (caddr_t)buf;
<                 if (ioctl(s, SIOCGLIFCONF, (char *)&lifc) < 0) {
<                       close(s);
<                       free(buf);
<                       goto freedata;
<                 }
<
<                 for (i = 0; i < lifn.lifn_count; ++i) {
<                       struct lifreq *lreqp, lreq;
<
<                       lreqp = (struct lifreq *)&lifc.lifc_req[i];
<                       strlcpy(lreq.lifr_name, lreqp->lifr_name,
<                                   sizeof (lreq.lifr_name));
<                       if (ioctl(s, SIOCGLIFFLAGS, &lreq) < 0) {
<                             close(s);
<                             free(buf);
<                             goto freedata;
<                       }
<                       if ((lreq.lifr_flags & IFF_UP) &&
<                             !(lreq.lifr_flags & IFF_NOLOCAL) &&
<                             !(lreq.lifr_flags & IFF_NOXMIT) &&
<                             !(lreq.lifr_flags & IFF_LOOPBACK)) {
<                             int_up = 1;
<                             break;
<                       }
<                 }
<                 free(buf);
<
<                 if (!int_up) {
<                       close(s);
<                       goto freedata;
<                 }
<           }
<           close(s);
<     }
< #endif
<
< #ifdef  SUNW_DOMAINFROMNIS
<
<         {
<                 char buf[sizeof(statp->defdname)], *cp;
<                 int  ret;
<                 if ((ret = sysinfo(SI_SRPC_DOMAIN, buf, sizeof(buf))) >
0 &&
<                     ret <= sizeof(buf)) {
<                         if (buf[0] == '+')
<                                 buf[0] = '.';
<                         cp = strchr(buf, '.');
<                         if (cp == 0)
<                                 strcpy(statp->defdname, buf);
<                         else
<                                 strcpy(statp->defdname, cp+1);
<                 }
<         }
< #endif
```

```
<
285,288d169
<
< #ifdef SUNW_AVOIDSTDIO_FDLIMIT
<     if ((fp = fopen(_PATH_RESCONF, "rF")) != NULL) {
< #else
290d170
< #endif
430c310
<         if (nserv > 0)
---
>         if (nserv > 1)
481,488d360
< #ifdef  SUNW_INITCHKIF
< freedata:
<     if (statp->_u._ext.ext != NULL) {
<           free(statp->_u._ext.ext);
<           statp->_u._ext.ext = NULL;
<     }
<     return (-1);
< #endif
530,538d401
< #ifdef    ORIGINAL_ISC_CODE
< #else
<           } else if (!strncmp(cp, "retrans:", sizeof("retrans:") - 1))
{
<
<                 statp->retrans = atoi(cp + sizeof("retrans:") - 1);
<           } else if (!strncmp(cp, "retry:", sizeof("retry:") - 1)){
<
<                 statp->retry = atoi(cp + sizeof("retry:") - 1);
< #endif
------------------------------------------------------------------------
./resolv/res_mkquery.c
20,26d16
<
< #ifdef SUNW_CONFCHECK
< #include <sys/socket.h>
< #include <errno.h>
< #include <sys/stat.h>
< #endif
<
38,41d27
< #ifdef SUNW_CONFCHECK
< static int      _confcheck(res_state statp);
< #endif
<
67,75d52
< #ifdef SUNW_CONFCHECK
<
<         if (_confcheck(statp) == -1) {
<                 RES_SET_H_ERRNO(statp, NO_RECOVERY);
<                 return(-1);
<         }
< #endif
<
<
161d137
```

```
< #ifdef    ORIGINAL_ISC_CODE
163,165d138
< #else
<     uchar_t *buf,
< #endif
209,258d181
<
< #ifdef SUNW_CONFCHECK
<
<
< #ifdef SUNW_AREWEINNAMED
< int __areweinnamed()
< {
<     return (0);
< }
< #endif
<
< static int _confcheck(res_state statp)
< {
<         int ns;
<         struct stat rc_stat;
<         struct sockaddr_in ns_sin;
<
<
<     if (__areweinnamed())
<           return (0);
<
<
<         if (stat(_PATH_RESCONF, &rc_stat) == -1 && errno == ENOENT) {
<
<
<                 if (statp->nsaddr.sin_addr.S_un.S_addr ==
<                       htonl(INADDR_LOOPBACK)) {
<
<
<                         ns = socket(AF_INET, SOCK_STREAM, 0);
<                         IN_SET_LOOPBACK_ADDR(&ns_sin);
<                         ns_sin.sin_port = htons(NAMESERVER_PORT);
<                         if (connect(ns, (struct sockaddr *) &ns_sin,
<                                     sizeof ns_sin) == -1) {
<                                 close(ns);
<                                 return(-1);
<                         }
<                         else {
<                                 close(ns);
<
<                                 return(0);
<                         }
<                 }
<
<                 return(0);
<         }
<
<         return (0);
< }
< #endif
-------------------------------------------------------------------------
./resolv/res_query.c
```

```
166c163
<           saved_herrno = statp->res_h_errno;
---
>           saved_herrno = h_errno;
292,294d288
< #ifdef SUNW_AVOIDSTDIO_FDLIMIT
<       if (file == NULL || (fp = fopen(file, "rF")) == NULL)
< #else
296d289
< #endif
------------------------------------------------------------------------
./resolv/res_send.c
43,51d39
< #ifdef SUNW_POLL
< #include <stropts.h>
< #include <poll.h>
< #if defined(pselect)
< #undef pselect
< #endif
< #define   pselect     Pselect
< #endif
<
59d46
< #ifdef    ORIGINAL_ISC_CODE
61,63d47
< #else
< static int highestFD = 0;
< #endif
85,91d68
< #ifdef    ORIGINAL_ISC_CODE
< #else
< #pragma weak    __res_nameinquery =     res_nameinquery
< #pragma weak    __res_queriesmatch      =     res_queriesmatch
< #pragma weak    res_nisourserver  =     res_ourserver_p
< #endif
<
124,125c101
<                       (srv6->sin6_scope_id == 0 ||
<                     srv6->sin6_scope_id == in6p->sin6_scope_id) &&
---
>                     srv6->sin6_scope_id == in6p->sin6_scope_id &&
208,213c184
< #ifdef    SUNW_POLL
<     highestFD = sysconf(_SC_OPEN_MAX) - 1;
< #endif
<
<
<     if (statp->nscount == 0 || EXT(statp).ext == NULL) {
---
>     if (statp->nscount == 0) {
600,602d570
<
<           resplen = anssiz;
<
641,644d608
< #ifdef    SUNW_POLL
<     int   polltimeout;
<     struct pollfd   pollfd;
```

```
< #endif
700,709d663
< #ifdef    SUNW_POLL
<     timeout = evSubTime(finish, now);
<     if (timeout.tv_sec < 0)
<           timeout = evConsTime(0, 0);
<     polltimeout = 1000*timeout.tv_sec +
<           timeout.tv_nsec/1000000;
<     pollfd.fd = s;
<     pollfd.events = POLLRDNORM;
<     n = poll(&pollfd, 1, polltimeout);
< #else
717d670
< #endif
726,728d678
< #ifdef    SUNW_POLL
<           Perror(statp, stderr, "poll", errno);
< #else
730d679
< #endif
-------------------------------------------------------------------------
./resolv/res_sendsigned.c
113d109
< #ifdef    ORIGINAL_ISC_CODE
116,119d111
< #else
<           Dprint(nstatp->pfcode & RES_PRF_REPLY,
<                  (stdout, ";; TSIG invalid (%s)\n", p_rcode(-ret)));
< #endif




-------------------------------------------------------------------------
./include/irs.h
4a5,8
>
>
>
157a162
> #ifdef    ORIGINAL_ISC_CODE
158a164,165
> #else
> #endif
197a205
> #ifdef    ORIGINAL_ISC_CODE
198a207,208
> #else
> #endif
207c218
< extern void           irs_destroy __P((void));
---
> extern void           irs_destroy(void);
232,272c249,305

-------------------------------------------------------------------------
./include/isc/dst.h
>
```

```
14a19
> #ifdef    ORIGINAL_ISC_CODE
> #else
> #endif
-------------------------------------------------------------------------
./include/isc/eventlib.h
>
> #ifdef    SUNW_POLL
> extern void evPollfdAdd(evContext ctx, int pollfd_chunk_size, int fd,
>                         short events);
> extern void evPollfdDel(evContext ctx, int fd);
> #endif
-------------------------------------------------------------------------
./include/isc/tree.h
3a4,6
>
>
22a26
> #ifdef    ORIGINAL_ISC_CODE
30c34,35
<
---
> #else
> #endif
-------------------------------------------------------------------------
./include/res_update.h
2a3,6
>
>
>
8a13
> #include <sys/bitypes.h>
12a18,22
> #ifdef    ORIGINAL_ISC_CODE
> #else
>
> #define   ns_updrec   __ISC_ns_updrec
> #endif
31a43
> #ifdef    ORIGINAL_ISC_CODE
37a50,58
> #else
>
> #define   res_mkupdate      __ISC_res_mkupdate
> #define   res_update  __ISC_res_update
> #define   res_mkupdrec      __ISC_res_mkupdrec
> #define   res_freeupdrec    __ISC_res_freeupdrec
> #define   res_nmkupdate     __ISC_res_nmkupdate
> #define   res_nupdate __ISC_res_nupdate
> #endif




-------------------------------------------------------------------------
port_before.h
1,8d0
```

```
<
<
< #pragma ident   "%Z%%M%     %I%   %E% SMI"
<
< #ifdef SUNW_OPTIONS
< #include "conf/sunoptions.h"
< #endif
<
10,13c2,3
<
< #ifdef    WANT_IRS_NIS
< #undef    WANT_IRS_NIS
< #endif
---
> #define SVR4
> #define WANT_IRS_NIS
70d59
< #ifdef    ORIGINAL_ISC_CODE
72,74d60
< #else
< #define NGR_R_END_RESULT(x)
< #endif
129,140d114
<
<
< #include <string.h>
< #include <strings.h>
< #include <unistd.h>
<
< #include <sys/types.h>
< #include "sys/bitypes.h"
< #include "sys/cdefs.h"
<
< #define   HAS_INET6_STRUCTS
< #define   H_ERRNO_IS_FUNCTION
------------------------------------------------------------------------------
port_after.h
1,4d0
<
<
< #pragma ident   "%Z%%M%     %I%   %E% SMI"
<
27d22
< #define   HAVE_DEV_RANDOM
33d27
< #ifndef   SUNW_LIBNSL
35d28
< #endif
227,232d219
< #include <resolv_mt.h>
< #include <netdb_private.h>
< #include <inet_private.h>
< #include <ctype.h>
< #include <sunw_dst_init.h>
< #include <sunw_irs_init.h>
234c221
< #ifdef    ORIGINAL_ISC_CODE
---
```

```
> #define NEED_STRERROR
237d223
< #endif
243,248d228
<
< #define   __putlong   putlong
< #define   __putshort  putshort
<
<
< #ifdef    ORIGINAL_ISC_CODE
251,256d230
< #endif
<
<
< #if (OS_MAJOR > 5 || (OS_MAJOR == 5 && OS_MINOR >= 8))
< #define USE_IFNAMELINKID
< #endif
265,273d238
<
< #ifndef ALIGN
< #if (OS_MAJOR == 5 && OS_MINOR > 8)
< #define ALIGN(x) (((uintptr_t)(x) + (sizeof(char*) - 1UL)) &
~(sizeof(char*) - 1UL))
< #else
< #define ALIGN(x) (((unsigned long)(x) + (sizeof(char*) - 1UL)) &
~(sizeof(char*) - 1UL))
< #endif
< #endif
<
```