# Source Tree: #

```
qfuse
  |---trunk
     |---setenv.sh
     |---lib
          |---fuse, ublio, uuid
     |---utils
          |---n
             |---ntfs-3g, ntfsprogs
          |---e
              |---ext2fuse, e2fsprogs
          |---z
              |---zfsfuse
     |---qa
         |---pjd-fstest
  |---vendor
     |---trunk
        |---lib
             |---fuse-2.7.4, libublio-20070103
        |---utils
           |---n
              |---ntfs-3g-2009.4.4, ntfsprogs-2.0.0
           |---e
              |---ext2fuse-src-0.8.1, e2fsprogs-1.41.5
           |---z
              |---zfsfuse
         |---qa
             |---pjd-fstest-20090130-RC
```

# How to build libfuse and other fs daemon?#

1. Enter qfuse/trunk and run ". ./setenv.sh"
2. on QNX host, cd lib/fuse and run "make install" or "CPULIST=x86 make install" if you only want to build libfuse for x86 target.
   This will install libfuse to qfuse/trunk/install/cpudir/lib and fuse headers to qfuse/trunk/install/include.
   On other hosts, for example: cd lib/fuse/x86-o, sh ./run.sh, make
3. cd lib/ublio and make install./* default ntfs-3g needs ublio but you can change*/
4. cd utils/n/ntfs-3g and run "make" or "make install". This will install ntfs-3g and libntfs to qfuse/trunk/install/cpudir too.

# How to run filesystem daemon?#

1. copy libfuse to /usr/local/lib
2. copy filesystem daemon's own library to /usr/local/lib, for example: ntfs-3g needs libntfs-3g.
3. copy filesystem daemon executable file to /usr/local/bin
4. run the fs daemon with arguments. For example: ntfs-3g -o debug,no_detach /dev/hd1t7 /ntfs

# How to port other filesystem to QNX?#

1. Get filesystem source code
2. Write a run.sh to try on QNX host. If configure complains something like you have to rerun automake or autoconf, then you have to install these on QNX(pkgsrc QNX port project you could try) or you can try on Linux host with QNX development tools installed. There is run.sh example in ntfs-3g.

3. If previous step succeeded then you can take a look at ext2fuse or ntfs-3g to see what you need to modify. Most likely you have to modify getopt because QNX doesn't support "--" long option yet.
4. If the filesystem registers FUSE lowlevel API then you have to modify its *_read function to use qfuse's buffer. There is example in libfuse or ext2fuse's op_read function. Because ntfs-3g registers highlevel API it doesn't care, necessary modification is in libfuse's default lowlevel API.
5. Try to port it to QNX' recursive compiling environment with multi-platform support.

# How to test?#

1. There is a posix filesystem test suite I ported from FreeBSD in trunk/qa/pjd-fstest.
   Please read README file to see how to run it.

# Benchmark #

/* below is test no fs daemon using ublio library*/
$ uname -a
QNX localhost 6.3.2 2006/03/16-14:19:50EST x86pc x86
#ext2fuse /home/ext2_256M.fs /ext2
#ntfs-3g /home/ntfs_256M.fs /ntfs
#mount -t qnx4 /home/qnx4_256M.fs /qnx4

/* default 8k record testing*/
# rw -t 128M -k /ext2/rw.tmp
Sequential File Write/Read Benchmark
OS:  QNX 6.3.2 x86pc
Filesys: disk, fsys
Config:  128MiB file, 8KiB record, fd, malloc
Create:  10 msec
Write:  7783 msec,  475 usec/write(),  89% CPU, 16.83 MiB/sec
Read:  9538 msec,  582 usec/read(),  72% CPU, 13.74 MiB/sec
# rw -t 128M -k /ntfs/rw.tmp
Sequential File Write/Read Benchmark
OS:  QNX 6.3.2 x86pc
Filesys: disk, fsys
Config:  128MiB file, 8KiB record, fd, malloc
Create:  4 msec
Write:  8774 msec,  535 usec/write(), 100% CPU, 14.93 MiB/sec
Read:  7164 msec,  437 usec/read(),  61% CPU, 18.29 MiB/sec
# rw -t 128M -k /qnx4/rw.tmp
Sequential File Write/Read Benchmark
OS:  QNX 6.3.2 x86pc
Filesys: ST340014A, UDMA5, qnx4, 1% full
Config:  128MiB file, 8KiB record, fd, fsync, malloc
Create:  11 msec
Write:  5624 msec,  343 usec/write(),  43% CPU, 23.30 MiB/sec
Read:  3812 msec,  232 usec/read(),  58% CPU, 34.38 MiB/sec

I am disappointed with ext2fuse's performance as ntfs-3g is using FUSE's high level API, every read/write will do inode(resmgr)->pathname(fuse)->inode(ntfs). But ext2fuse has a lot which can be improved also ext2 is not a difficult filesystem.

# Wireless:#

1. ifconfig run0 list wme

   AC_BE cwmin  4 cwmax 10 aifs  3 txopLimit  0

```
        cwmin  4 cwmax 10 aifs  3 txopLimit   0
    AC_BK cwmin  4 cwmax 10 aifs  7 txopLimit   0
        cwmin  4 cwmax 10 aifs  7 txopLimit   0
    AC_VI cwmin  3 cwmax  4 aifs  2 txopLimit  94
        cwmin  3 cwmax  4 aifs  2 txopLimit  94
    AC_VO cwmin  2 cwmax  3 aifs  2 txopLimit  47
        cwmin  2 cwmax  3 aifs  2 txopLimit  47
```

2. ifconfig run0 list chan

```
Channel   1 : 2412  Mhz 11g        Channel 102 : 5510  Mhz 11a
Channel   2 : 2417  Mhz 11g        Channel 104 : 5520  Mhz 11a
Channel   3 : 2422  Mhz 11g        Channel 108 : 5540  Mhz 11a
Channel   4 : 2427  Mhz 11g        Channel 110 : 5550  Mhz 11a
Channel   5 : 2432  Mhz 11g        Channel 112 : 5560  Mhz 11a
Channel   6 : 2437  Mhz 11g        Channel 116 : 5580  Mhz 11a
Channel   7 : 2442  Mhz 11g        Channel 118 : 5590  Mhz 11a
Channel   8 : 2447  Mhz 11g        Channel 120 : 5600  Mhz 11a
Channel   9 : 2452  Mhz 11g        Channel 124 : 5620  Mhz 11a
Channel  10 : 2457  Mhz 11g        Channel 126 : 5630  Mhz 11a
Channel  11 : 2462  Mhz 11g        Channel 128 : 5640  Mhz 11a
Channel  12 : 2467  Mhz 11g        Channel 132 : 5660  Mhz 11a
Channel  13 : 2472  Mhz 11g        Channel 134 : 5670  Mhz 11a
Channel  14 : 2484  Mhz 11g        Channel 136 : 5680  Mhz 11a
Channel  36 : 5180  Mhz 11a        Channel 140 : 5700  Mhz 11a
Channel  38 : 5190  Mhz 11a        Channel 149 : 5745  Mhz 11a
Channel  40 : 5200  Mhz 11a        Channel 151 : 5755  Mhz 11a
Channel  44 : 5220  Mhz 11a        Channel 153 : 5765  Mhz 11a
Channel  46 : 5230  Mhz 11a        Channel 157 : 5785  Mhz 11a
Channel  48 : 5240  Mhz 11a        Channel 159 : 5795  Mhz 11a
Channel  52 : 5260  Mhz 11a        Channel 161 : 5805  Mhz 11a
Channel  54 : 5270  Mhz 11a        Channel 165 : 5825  Mhz 11a
Channel  56 : 5280  Mhz 11a        Channel 167 : 5835  Mhz 11a
Channel  60 : 5300  Mhz 11a        Channel 169 : 5845  Mhz 11a
Channel  62 : 5310  Mhz 11a        Channel 171 : 5855  Mhz 11a
Channel  64 : 5320  Mhz 11a        Channel 173 : 5865  Mhz 11a
Channel 100 : 5500  Mhz 11a
```

3. ifconfig run0 list caps

```
run0=0x583c50b<WEP,TKIP,AES_CCM,IBSS,HOSTAP,SHSLOT,SHPREAMBLE,MONITOR,TKIPMIC,WPA1,WPA2,WI
```

4. How do you capture wireless packets with QNX?

```
1).ifconfig run0 mediaopt monitor
2).tcpdump -L
Data link types (use option -y to set):
  EN10MB (Ethernet)
  IEEE802_11 (802.11)
  IEEE802_11_RADIO (802.11 plus BSD radio information header)
3).tcpdump -i run0 -y IEEE802_11_RADIO -s0 -vvvv
tcpdump: data link type IEEE802_11_RADIO
tcpdump: listening on run0, link-type IEEE802_11_RADIO (802.11 plus BSD radio information header), capture size 65535 b
06:59:12.559480 6.0 Mb/s 2412 MHz (0x04e0) -23dB signal antenna 1 21dB signal 0us Beacon () [6.0* 9.0 12.0* 18.0 24.0* 3
Note: mediaopt monitor is not necessary for all the cases. If you want to see all packets in this channel and closed channels ther
```

# FAQ:#

1. Can I compile them on Linux or Windows HOST with QNX development toools installed?

Answer: You can run ". ./run.sh" to make it on Linux or Windows host and there is an example in ntfs-3g or fuse. And you'd better not to mix QNX' makefile and GNU's makefile together so I create nto-cpu-endian for QNX multi arch compiling and cpu-endian for GNU multi arch compiling so put run.sh to corresponding directory. If want to switch between GNU makefile.gnu and QNX makefile.qnx, you can run "make -f Makefile.gnu distclean" to clean it with GNU makefile.

Resources:

1. FUSE project
2. ntfs-3g project