

个人资料



chenxiaohua

访问：104889次
积分：2107分
排名：第2273名

原创：101篇 转载：17篇
译文：0篇 评论：35条

文章搜索

文章分类

- [apache\(2\)](#)
- [C++/C\(21\)](#)
- [java开发\(4\)](#)
- [lex yacc专栏系列\(windows,vc.net环境\)\(1\)](#)
- [linux/unix\(29\)](#)
- [tomcat调试\(5\)](#)
- [VC/MFC/Windows/COM\(2\)](#)
- [web开发\(0\)](#)
- [web开发,管理\(0\)](#)
- [windows优化\(2\)](#)
- [中间件,应用服务器\(0\)](#)
- [数据库开发\(37\)](#)
- [管理\(0\)](#)
- [软件工程\(1\)](#)

文章存档

- [2010年04月\(1\)](#)
- [2010年02月\(1\)](#)
- [2010年01月\(2\)](#)
- [2009年12月\(2\)](#)
- [2009年08月\(2\)](#)

展开

阅读排行

- [linux下的tuxedo开发实例\(7281\)](#)
- [linux开机启动服务和chkconfi... \(6507\)](#)
- [linux下tinyxml开发入门\(4388\)](#)

博客频道4月技术图书有奖试读火爆开启

移动业界领袖会议·上海 6.20

第四届云计算大会门票抢购：史上最低价，每日限5张！

【分享季1】：网友推荐130个经典资源，分享再赠分！

linux下tinyxml开发入门

分类：[linux/unix C++/C](#)

2009-06-16 09:17

4389人阅读

[评论\(5\)](#)

[收藏](#)

[举报](#)

从http://ncu.dl.sourceforge.net/sourceforge/tinyxml/tinyxml_2_4_0.tar.gz下载tinyxml，可以根据自己的需要，选择不同的版本。

将tinyxml_2_4_0.tar.gz上传到主机，然后解压执行如下命令：

```
tar -xzf tinyxml_2_4_0.tar.gz
```

成功之后，会在当前目录出现一个tinyxml目录,进入该目录cd tinyxml,然后进行编译，顺序执行如下命令：

```
cd tinyxml
```

```
make
```

在屏幕上会打印如下输出：

```
[java]
01. [dev@localhost tinyxml]$ make
02. g++ -c -Wall -Wno-format -g -DDEBUG      tinyxml.cpp -o tinyxml.o
03. g++ -c -Wall -Wno-format -g -DDEBUG      tinyxmlparser.cpp -o tinyxmlparser.o
04. g++ -c -Wall -Wno-format -g -DDEBUG      xmltest.cpp -o xmltest.o
05. g++ -c -Wall -Wno-format -g -DDEBUG      tinyxmlerror.cpp -o tinyxmlerror.o
06. g++ -c -Wall -Wno-format -g -DDEBUG      tinystr.cpp -o tinystr.o
07. tinystr.cpp:38: warning: aggregate has a partly bracketed initializer
08. g++ -o xmltest -g tinyxml.o tinyxmlparser.o xmltest.o tinyxmlerror.o tinystr.o
```

没有出现错误，表示编译完成，这时可以执行tinyxml自带的测试程序xmltest。

我直行xmltest之后，打印出一堆乱码，后来就没有管。自己写程序测试了。

为了使用tinyxml开发，使用方便，做了一些配置。

添加环境变量TINYXML_ROOT,编辑.bash_profile，添加如下内容：

```
[java]
01. #####
02. #### for tinyxml
03. #####
04. export TINYXML_ROOT=$HOME/tinyxml
```

把tinyxml包编译打包成一个连接库，方便开发，这就要修改tinyxml目录下的Makefile。

```
[c-sharp]
01. #####
02. #
03. # Makefile for TinyXml test.
04. # Lee Thomason
05. # www.grinninglizard.com
06. #
```

[linux安装xml开发包xerces-... \(3268\)](#)
[oracle下载地址大全 \(3253\)](#)
[解决xml解析中报文格式和 encodin... \(2979\)](#)
[linux下Oracle自动启动与停止 \(... \(2601\)](#)
[tuxedo之domain应用入门 \(2528\)](#)
[编译所有子目录的makefile \(2449\)](#)
[解决虚拟机linux下鼠标不能动 \(2412\)](#)

评论排行

[linux下tinyxml开发入门 \(5\)](#)
[linux安装xml开发包xerces-... \(4\)](#)
[xerces-c和icu编译 \(2\)](#)
[windows下tuxedo开发 \(2\)](#)
[windows下yacc和lex开发环境... \(2\)](#)
[linux下Oracle自动启动与停止 \(... \(2\)](#)
[ftp自动登录 \(2\)](#)
[oracle使用to_char和next... \(2\)](#)
[tuxedo之用WTC和weblogic... \(2\)](#)
[解决虚拟机linux下鼠标不能动 \(2\)](#)

推荐文章

[*](#)
[PL/SQL](#)
[集](#)
[合](#)
[的](#)
[初](#)
[始](#)
[化](#)
[与](#)
[赋](#)

最新评论

[linux下tinyxml开发入门](#)
tianfei2007: 太感谢老母鸡了，过程都写得这么详细，如获至宝，节省了好多时间啊~~~~~

[linux开机启动服务和chkconfig使用方法](#)
huyangg: ln -s /etc/rc.d/init.d/auto_run /etc/rc.d/rc5.d/S9...

[tuxedo之domain应用入门](#)
lyach: 在googlecode上居然发现一款号称和Tuxedo兼容的交易中间件。叫 tuxone ,很山寨的...

[tuxedo之UBBCONFIG介绍](#)
lyach: 在googlecode上居然发现一款号称和Tuxedo兼容的交易中间件。叫 tuxone ,很山寨的...

[linux下tinyxml开发入门](#)
laomuji: 回复 qiongrier : 在makefile中不是有个include的section吗，它指定了程序...

[oracle数据库的物理迁移](#)
xcxxu: 寫的很詳細，謝謝樓主了！

[linux下Oracle自动启动与停止 \(二\)](#)
weiminyinhai: 我的出现了错误./oracle: line 44: syntax error: unexpected...

[解决虚拟机linux下鼠标不能动](#)
user201009: 我安装时 选的usb的鼠标，安装完成之后怎么修

```
07.  # This is a GNU make (gmake) makefile
08.  #*****
09.
10.  # DEBUG can be set to YES to include debugging info, or NO otherwise
11.  DEBUG             := YES
12.
13.  # PROFILE can be set to YES to include profiling info, or NO otherwise
14.  PROFILE           := NO
15.
16.  # TINYXML_USE_STL can be used to turn on STL support. NO, then STL
17.  # will not be used. YES will include the STL files.
18.  TINYXML_USE_STL  := NO
19.
20.  #*****
21.
22.  CC                := gcc
23.  CXX               := g++
24.  LD                := g++
25.  AR                := ar rc
26.  RANLIB            := ranlib
27.
28.  DEBUG_CFLAGS      := -Wall -Wno-format -g -DDEBUG
29.  RELEASE_CFLAGS    := -Wall -Wno-unknown-pragmas -Wno-format -O3
30.
31.  LIBS              :=
32.
33.  DEBUG_CXXFLAGS    := ${DEBUG_CFLAGS}
34.  RELEASE_CXXFLAGS  := ${RELEASE_CFLAGS}
35.
36.  DEBUG_LDFLAGS     := -g
37.  RELEASE_LDFLAGS   :=
38.
39.  ifeq (YES, ${DEBUG})
40.      CFLAGS        := ${DEBUG_CFLAGS}
41.      CXXFLAGS       := ${DEBUG_CXXFLAGS}
42.      LDFLAGS        := ${DEBUG_LDFLAGS}
43.  else
44.      CFLAGS        := ${RELEASE_CFLAGS}
45.      CXXFLAGS       := ${RELEASE_CXXFLAGS}
46.      LDFLAGS        := ${RELEASE_LDFLAGS}
47.  endif
48.
49.  ifeq (YES, ${PROFILE})
50.      CFLAGS        := ${CFLAGS} -pg -O3
51.      CXXFLAGS       := ${CXXFLAGS} -pg -O3
52.      LDFLAGS        := ${LDFLAGS} -pg
53.  endif
54.
55.  #*****
56.  # Preprocessor directives
57.  #*****
58.
59.  ifeq (YES, ${TINYXML_USE_STL})
60.      DEFS := -DTIXML_USE_STL
61.  else
62.      DEFS :=
63.  endif
64.
65.  #*****
66.  # Include paths
67.  #*****
68.
69.  #INCS := -I/usr/include/g++-2 -I/usr/local/include
70.  INCS :=
71.
72.  #*****
73.  # Makefile code common to all platforms
74.  #*****
75.
76.
77.  CFLAGS    := ${CFLAGS} ${DEFS}
78.  CXXFLAGS  := ${CXXFLAGS} ${DEFS}
79.
80.  #*****
81.  # Targets of the build
82.  #*****
83.
84.  OUTPUT := xmltest
85.  LIB := libtinyxml.so
```

改？

oracle使用to_char和next_day函数得到本周的第一天和最后一天日期

Annie769586757: 厉害，学会了，帮了我的大忙！

tuxedo之domain应用入门
匿名用户:

```

86.
87. all: ${OUTPUT} ${LIB}
88.
89.
90. #*****
91. # Source files
92. #*****
93.
94. SRCS := tinyxml.cpp tinyxmlparser.cpp xmltest.cpp tinyxmlerror.cpp tinystr.cpp
95.
96. # Add on the sources for libraries
97. SRCS := ${SRCS}
98.
99. OBJS := $(addsuffix .o,$(basename ${SRCS}))
100. LIBOBJS := tinyxml.o tinyxmlparser.o tinyxmlerror.o tinystr.o
101.
102. #*****
103. # Output
104. #*****
105.
106. ${OUTPUT}: ${OBJS}
107.         ${LD} -o $@ ${LDFLAGS} ${OBJS} ${LIBS} ${EXTRA_LIBS}
108.
109. ${LIB}: ${LIBOBJS}
110.         ar -r $@ ${LIBOBJS}
111.
112.
113. #*****
114. # common rules
115. #*****
116.
117. # Rules for compiling source files to object files
118. %.o : %.cpp
119.         ${CXX} -c ${CXXFLAGS} ${INCS} $< -o $@
120.
121. %.o : %.c
122.         ${CC} -c ${CFLAGS} ${INCS} $< -o $@
123.
124. dist:
125.         bash makedistlinux
126.
127. clean:
128.         -rm -f core ${OBJS} ${OUTPUT} ${LIB} ${TEST}
129.
130. depend:
131.         #makedepend ${INCS} ${SRCS}
132.
133. tinyxml.o: tinyxml.h tinystr.h
134. tinyxmlparser.o: tinyxml.h tinystr.h
135. xmltest.o: tinyxml.h tinystr.h
136. tinyxmlerror.o: tinyxml.h tinystr.h

```

在tinyxml目录下重新执行make，会看到多执行了一行命令：

```
ar -r libtinyxml.so tinyxml.o tinyxmlparser.o tinyxmlerror.o tinystr.o
```

生成了一个包libtinyxml.so，有了这个包，使用tinyxml开发的时候，在连接命令中加入这个包的连接，就可以正确地生成目标程序。

现在来写一个小程序测试一下，在tinyxml目录创建一个测试的xml文件，文件名为test.xml,内容如下：

```

[xhtml]
01. <Persons>
02.     <Person ID="1">
03.         <name>周星星</name>
04.         <age>20</age>
05.     </Person>
06.     <Person ID="2">
07.         <name>白晶晶</name>
08.         <age>18</age>
09.     </Person>
10. </Persons>

```

在tinyxml下创建，也添加了一个测试程序tinyxml_test.cpp，内容如下：

[cpp]

```

01.  #include "tinyxml.h"
02.  #include "tinystl.h"
03.
04.  #include <iostream>
05.
06.  using namespace std;
07.
08.  int main()
09.  {
10.      //创建一个XML的文档对象。
11.      TiXmlDocument *myDocument = new TiXmlDocument("test.xml");
12.      myDocument->LoadFile();
13.
14.      //获得根元素，即Persons。
15.      TiXmlElement *RootElement = myDocument->RootElement();
16.
17.      //输出根元素名称，即输出Persons。
18.      cout << RootElement->Value() << endl;
19.
20.      //获得第一个Person节点。
21.      TiXmlElement *FirstPerson = RootElement->FirstChildElement();
22.      //输出接点名Person
23.
24.      cout << FirstPerson->Value() << endl;
25.      //获得第一个Person的name节点和age节点和ID属性。
26.      TiXmlElement *NameElement = FirstPerson->FirstChildElement();
27.      TiXmlElement *AgeElement = NameElement->NextSiblingElement();
28.      TiXmlAttribute *IDAttribute = FirstPerson->FirstAttribute();
29.
30.      //输出第一个Person的name内容，即周星星；age内容，即20；ID属性，即1。
31.      cout << NameElement->FirstChild()->Value() << endl;
32.      cout << AgeElement->FirstChild()->Value() << endl;
33.      cout << IDAttribute->Value() << endl;
34.
35.      return 0;
36.  }

```

在tinyxml目录下修改Makefile，修改之后的内容如下：

[c-sharp]

```

01.  #*****
02.  #
03.  # Makefile for TinyXml test.
04.  # Lee Thomason
05.  # www.grinninglizard.com
06.  #
07.  # This is a GNU make (gmake) makefile
08.  #*****
09.
10.  # DEBUG can be set to YES to include debugging info, or NO otherwise
11.  DEBUG          := YES
12.
13.  # PROFILE can be set to YES to include profiling info, or NO otherwise
14.  PROFILE        := NO
15.
16.  # TINYXML_USE_STL can be used to turn on STL support. NO, then STL
17.  # will not be used. YES will include the STL files.
18.  TINYXML_USE_STL := NO
19.
20.  #*****
21.
22.  CC          := gcc
23.  CXX         := g++
24.  LD          := g++
25.  AR          := ar rc
26.  RANLIB      := ranlib
27.
28.  DEBUG_CFLAGS := -Wall -Wno-format -g -DDEBUG
29.  RELEASE_CFLAGS := -Wall -Wno-unknown-pragmas -Wno-format -O3
30.
31.  LIBS        :=
32.
33.  DEBUG_CXXFLAGS := ${DEBUG_CFLAGS}
34.  RELEASE_CXXFLAGS := ${RELEASE_CFLAGS}
35.
36.  DEBUG_LDFLAGS := -g

```

```

37.  RELEASE_LDFLAGS  :=
38.
39.  ifeq (YES, ${DEBUG})
40.      CFLAGS        := ${DEBUG_CFLAGS}
41.      CXXFLAGS      := ${DEBUG_CXXFLAGS}
42.      LDFLAGS       := ${DEBUG_LDFLAGS}
43.  else
44.      CFLAGS        := ${RELEASE_CFLAGS}
45.      CXXFLAGS      := ${RELEASE_CXXFLAGS}
46.      LDFLAGS       := ${RELEASE_LDFLAGS}
47.  endif
48.
49.  ifeq (YES, ${PROFILE})
50.      CFLAGS        := ${CFLAGS} -pg -O3
51.      CXXFLAGS      := ${CXXFLAGS} -pg -O3
52.      LDFLAGS       := ${LDFLAGS} -pg
53.  endif
54.
55.  #*****
56.  # Preprocessor directives
57.  #*****
58.
59.  ifeq (YES, ${TINYXML_USE_STL})
60.      DEFS := -DTIXML_USE_STL
61.  else
62.      DEFS :=
63.  endif
64.
65.  #*****
66.  # Include paths
67.  #*****
68.
69.  #INCS := -I/usr/include/g++-2 -I/usr/local/include
70.  INCS :=
71.
72.  #*****
73.  # Makefile code common to all platforms
74.  #*****
75.
76.
77.  CFLAGS    := ${CFLAGS} ${DEFS}
78.  CXXFLAGS  := ${CXXFLAGS} ${DEFS}
79.
80.  #*****
81.  # Targets of the build
82.  #*****
83.
84.  OUTPUT := xmltest
85.  LIB := libtinyxml.so
86.  TEST:= tinyxml_test
87.
88.  all: ${OUTPUT} ${LIB} ${TEST}
89.
90.
91.  #*****
92.  # Source files
93.  #*****
94.
95.  SRCS := tinyxml.cpp tinyxmlparser.cpp xmltest.cpp tinyxmlerror.cpp tinystr.cpp
96.
97.  # Add on the sources for libraries
98.  SRCS := ${SRCS}
99.
100.  OBJS := $(addsuffix .o, $(basename ${SRCS}))
101.  LIBOBJS := tinyxml.o tinyxmlparser.o tinyxmlerror.o tinystr.o
102.
103.  #*****
104.  # Output
105.  #*****
106.
107.  ${OUTPUT}: ${OBJS}
108.      ${LD} -o $@ ${LDFLAGS} ${OBJS} ${LIBS} ${EXTRA_LIBS}
109.
110.  ${LIB}: ${LIBOBJS}
111.      ar -r $@ ${LIBOBJS}
112.
113.  ${TEST}: tinyxml_test.o
114.      ${LD} -o $@ ${LDFLAGS} tinyxml_test.o -L${TINYXML_ROOT} -ltinyxml
115.

```

```
116.  #*****
117.  # common rules
118.  #*****
119.
120.  # Rules for compiling source files to object files
121.  %.o : %.cpp
122.      ${CXX} -c ${CXXFLAGS} ${INCS} $< -o $@
123.
124.  %.o : %.c
125.      ${CC} -c ${CFLAGS} ${INCS} $< -o $@
126.
127.  dist:
128.      bash makedistlinux
129.
130.  clean:
131.      -rm -f core ${OBJS} ${OUTPUT} ${LIB} ${TEST}
132.
133.  depend:
134.      #makedepend ${INCS} ${SRCS}
135.
136.  tinyxml.o: tinyxml.h tinystl.h
137.  tinyxmlparser.o: tinyxml.h tinystl.h
138.  xmltest.o: tinyxml.h tinystl.h
139.  tinyxmlerror.o: tinyxml.h tinystl.h
```

然后执行make，这时，可以看到多了一个tinyxml_test 生成，执行tinyxml_test，得到如下输出结果：

```
[c-sharp]
01.  Persons
02.  Person
03.  周星星
04.  20
05.  1
```

上一篇：[base64的java实现方法](#)

分享到：

下一篇：[解决xml解析中报文格式和encoding不一致的解析错误](#)

[Curtain Coating](#)
Optimized Process for Highest Coating Productivity.
www.multilayer-coating.ch

Google 提供的广告

查看评论

4楼 [tianfei2007](#) 2011-08-10 22:22发表



太感谢老母鸡了，过程都写得这么详细，如获至宝，节省了好多时间啊~~~~~

3楼 [chen_jianhui2008](#) 2010-06-11 20:10发表



直接在shell下export TINYXML_ROOT=\$HOME/tinyxml 。
然后用echo \$TINYXML_ROOT，这个目录要是本文中libtinyxml.so所在位置。
然后直接make就可以了。我test了，ok

2楼 [caixin99](#) 2010-03-13 15:27发表



1楼 [qiongrier](#) 2009-12-28 11:45发表



喂哥们 为什么我用你这个之后总是找不找类呢 连接不了类 能解决下吗 谢谢了

Re: [laomuji](#) 2011-03-10 15:55发表



回复 qiongrier：在makefile中不是有个include的section吗，它指定了程序寻找的路径，把so文件放在/usr/lib下，把几个头文件放在/usr/include下。就ok了。

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场


更多招聘职位

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告

北京创新乐知信息技术有限公司 版权所有，京 ICP 证 070598 号

世纪乐知(北京)网络技术有限公司 提供技术支持

江苏乐知网络技术有限公司 提供商务支持

 Email:webmaster@csdn.net

Copyright © 1999-2012, CSDN.NET, All Rights Reserved

