Qt TAR

Qt TAR Site : http://qttar.sourceforge.net

Qt TAR is a GNU Tar data archiver wrapper for Qt.

Original TAR library site is http://www.gnu.org/software/tar, authors are John Gilmore and Jay Fenlason. Current version is "1.28". The author of QtTAR is Brian Lin, who is an astrophysicist.

For details, please pay a visit to each of the following languages:

English	正體中文	简体中文

Qt TAR is also a part of CIOS Compression Subsystem, which belongs to CIOS Data Manipulation Subsystem.

CIOS / Compression Subsystem			
QtGZip	QtBZip2	QtXz	QtLzo
QtLZip	QtUCL		
QtCompression			
QtTAR	QtZIP	QtRAR	Qt7z
QtArchivers			

Download

Method	Filename
7z	QtTAR-2015-11-04.7z
Tar.gz	QtTAR-2015-11-04.tar.gz
Tar.xz	QtTAR-2015-11-04.tar.xz
Mercurial	hg clone ssh://linfoxman@hg.code.sf.net/p/qttar/mercurial QtTAR
Git	git clone ssh://linfoxman@git.code.sf.net/p/qttar/git QtTAR

Contacts

Author : Brian Lin (a.k.a Vladimir Lin , Vladimir Forest , Владимир Лесной)

The author generated over 6 millions lines of codes manually in diverse fields and maintains hundreds of projects. Please be specific on which project you are referring. In case you want to report and ask questions about Qt LZO, please add a "Qt LZO" in a short and brief form when you report or ask questions.

E-mail	lin.foxman@gmail.com
E-mail	lin.vladimir@gmail.com
E-mail	wolfram_lin@yahoo.com
E-mail	wolfram_lin@sina.com
E-mail	wolfram_lin@163.com
Skype	wolfram_lin
WeChat	153-0271-7160
WhatsApp	153-0271-7160
ICQ	153-0271-7160
QQ	lin.vladimir@gmail.com

Qt TAR 說明文件

目錄

GNU Tar 檔案集成包裝工具說明

GNU Tar 於1985 發表第一版，其後持續改進，目前版本為1.28。

GNU Tar 一種將檔案包裝集結的工具，由於經常與壓縮工具同時使用，經常被誤認為是一種壓縮方法，實際上這是錯誤的認知。

GNU Tar 封包目前有幾種方式的封裝方法，如下：

- POSIX
- S-TAR
- Old GNU
- Sparse
- S-TAR In
- S-TAR Extension

Qt TAR 的使用方式

Qt TAR 有兩種使用方式，編譯成Qt 模組或是直接將原始碼包含入您的開發計畫當中。

編譯成Qt 模組的方法，請見「QtTAR 編譯說明」。

直接將原始碼包含入您的開發計畫當中的方法，如下：

1. 下載「Qt TAR」（QtTAR-2015-11-04.7z），並且解開檔案。
2. 進入「QtTAR/Src/Embedded」目錄當中。
3. 將「QtTAR/Src/Embedded/LZO」目錄複製到您的開發計畫當中。
4. 在您的Qt PRO 計畫檔當中加入「TAR/TAR.pri」。

Qt TAR 編譯說明

網站：http://qttar.sourceforge.net
原始碼：https://sourceforge.net/projects/qttar/files/
版本：2015-11-04

目錄

Qt LZO 目錄結構

- QtTAR
  - 3rdparty
    - scripts
      - Update-TAR.js
    - sources
      - tar-1.28.tar.gz
  - doc
    - HTML
      - index.html
    - ODT
      - QtTAR.odt
      - QtTAR-TW.odt
      - QtTAR-CN.odt

- PDF
  - QtTAR.pdf
  - QtTAR-TW.pdf
  - QtTAR-CN.pdf
- Qt
  - cn
    - examples.html
    - index.html
    - lzo.html
    - others.html
    - qtlzo.html
    - Replacements.txt
  - en
    - examples.html
    - index.html
    - lzo.html
    - others.html
    - qtlzo.html
  - tw
    - examples.html
    - index.html
    - lzo.html
    - others.html
    - qtlzo.html
  - classic.css
  - index.html
  - Qt.pri
  - QtTAR.qhp
- TeX
  - QtTAR.tex

- examples
  - tartool
    - tartool.cpp
    - tartool.ico
    - tartool.pro
    - tartool.rc
  - examples.pro
- include
  - QtTAR
    - headers.pri
    - QtTAR
    - qttar.hpp
- src
  - Embedded
    - TAR
      - TAR.pri
      - qttar.cpp
      - qttar.hpp
  - QtTAR
    - qttar.cpp
    - QtTAR.pro
  - src.pro
- tests
  - auto
    - cmake
      - cmake.pro
      - CMakeLists
    - auto.pro

•tests.pro

編譯QtTAR

您需要使用Qt 的原始碼來編譯QtTAR 模組, QtTAR 模組的擺設位置如下:

- •QtTargetDirectory
- •QtSourceDirectory
  - •QtTAR
  - •qtbase
  - •gnuwin32
  - •...

編譯Qt 一般使用原始碼與目標目錄分開的方式, QtTargetDirectory 是Qt 編譯完成後的最終目錄, QtSourceDirectory 是Qt 的原始碼目錄。

當您編譯完成Qt 以後, 不要進行任何清除的動作, 將QtTAR 解壓縮以後, 擺到QtSourceDirectory 目錄當中, 並且更名為QtTAR。

Unix

```
cd QtTAR
qmake
make
make install
```

Windows

```
cd QtTAR
qmake
nmake
nmake install
```

或是

```
cd QtTAR
qmake
jom
nmake install
```

如此即可完成。

製作QtTAR 文件

切換到「QtTAR\doc\Qt」目錄當中:

cd QtTAR\doc\Qt
qhelpgenerator.exe QtTAR.qhp -o QtTAR.qch

新增文件到Qt Creator

打開Qt Creator，「工具」→「選項」→「說明」→「文件」→「新增」，選取QtTAR.qch，按下完成即可新增。

新增文件到Qt Assistant

打開Qt Assistant，「編輯」→「喜好設定」→「文件」→「新增」，選取QtTAR.qch，按下完成即可新增。

Qt TAR 類別使用說明

目錄

QtTAR 類別

功能

處理TAR 的封裝數據。

宣告

class QtTAR

```
{

  public:

    explicit        QtTAR         (void) ;

    virtual        ~QtTAR         (void) ;

    virtual int     BlockSize     (void) const ;
```

```
    virtual bool    isBlock       (QByteArray & block) ;

    virtual bool    isPadding     (QByteArray & block) ;

    virtual qint64 FileBlocks    (qint64 size) ;

    virtual int     Checksum      (QByteArray & block,char replace = ' ') ;

    virtual bool    Extract       (QByteArray & block          ,void       *
hiddenFileInfo) ; // block => File

    virtual bool    Bale          (void       * hiddenFileInfo,QByteArray & data
) ; // File  <= Data

  protected:

    QString         toOct         (int checksum) ;

    void            PackOct       (char * buf,qint64 size,int length) ;

    qint64          FromOct       (char * buf,int length) ;

  private:

    void            Copy          (char * buf,QString string) ;

} ;
```

說明

```
int BlockSize(void) const
```

TAR 的封包大小，一般是512 位元組。

```
bool isBlock(QByteArray & block)
```

檢查是否為TAR 封包。

```
bool isPadding(QByteArray & block)
```

檢查是否為填空封包。

```
qint64 FileBlocks(qint64 size)
```

將size 轉換成512 位元組對齊的數量。TAR 檔案格式均必須是512 位元組的倍數，這個函式把檔案大小轉換成512 位元組對齊的數值。

```
int Checksum(QByteArray & block,char replace = ' ')
```

計算TAR 封包512 位元組的Checksum 值。

```
bool Extract(QByteArray & block,void * hiddenFileInfo)
```

將TAR 封包數據解譯為檔案記錄資訊。

```
bool Bale(void * hiddenFileInfo,QByteArray & data)
```

將檔案記錄資訊封裝為TAR 封包數據。

```
QString toOct(int checksum)
```

將Checksum 轉換成八進位字串。

```
void PackOct(char * buf,qint64 size,int length)
```

將size 數據封裝成八進位數值，並且複製到buf 當中。

```
qint64 FromOct(char * buf,int length)
```

將buf 以八進位格式轉換成qint64。

```
void Copy(char * buf,QString string)
```

將string 字串複製到buf 當中。

QtTarBall 類別

功能

處理TarBall 檔案內容，一般的TAR 檔案處理需要繼承這個類別。

宣告

```
class QtTarBall : public QtTAR

{

  public:

    explicit       QtTarBall        (void) ;

    virtual       ~QtTarBall        (void) ;

    virtual bool   List             (QDir root,QIODevice & IO) ;

    virtual bool   List             (QDir root,QString filename) ;

    virtual bool   Extract          (QDir root,QIODevice & IO) ;

    virtual bool   Extract          (QDir root,QString filename) ;

    virtual bool   TarBall          (QIODevice & IO,QDir root,QDir source,bool
recursive = true) ;

    virtual bool   TarBall          (QString filename,QDir root,QDir source,bool
recursive = true) ;

  protected:

    virtual bool   Interval         (void) ;

    virtual void   Report           (void * hiddenFileInfo) ;

    virtual bool   ListFile         (QDir root,QIODevice & IO,void *
hiddenFileInfo) ;

    virtual bool   Extract          (QDir root,QIODevice & IO,void *
hiddenFileInfo) ;

    virtual bool   Read             (QIODevice & IO,QByteArray & data,qint64
size) ;

    virtual bool   Skip             (QIODevice & IO,qint64 size) ;

    virtual bool   Write            (QIODevice & IO,QByteArray & data) ;

    virtual bool   WriteClose       (QIODevice & IO) ;

    virtual bool   WriteFile        (QIODevice & IO,QFileInfo & file) ;

    virtual bool   ExtractFile      (QDir root,QIODevice & IO,void *
hiddenFileInfo) ;

    virtual bool   ExtractDir       (QDir root,QIODevice & IO,void *
hiddenFileInfo) ;
```

```
    virtual bool   ExtractLink      (QDir root,QIODevice & IO,void *
hiddenFileInfo) ;

    virtual bool   ExtractDEVs      (QDir root,QIODevice & IO,void *
hiddenFileInfo) ;

    virtual bool   ExtractNext      (QDir root,QIODevice & IO,void *
hiddenFileInfo) ;

    virtual bool   ExtractEXT       (QDir root,QIODevice & IO,void *
hiddenFileInfo) ;

    virtual bool   setFileTime      (QDir root,QIODevice & IO,void *
hiddenFileInfo) ;

    virtual bool   setFileMode      (QDir root,QIODevice & IO,void *
hiddenFileInfo) ;

    virtual QFileInfoList Listing   (QDir & root,QDir source) ;

    virtual bool   WriteTAR         (QIODevice & IO,QDir & root,QFileInfo &
file) ;

    virtual bool   ToHiddenFileInfo (QDir & root,QFileInfo & file,void *
hiddenFileInfo) ;

    virtual void * NewHiddenFile    (void) ;

    virtual void   CleanHiddenFile  (void * hiddenFileInfo) ;

  private:

} ;
```

說明

```
bool List(QDir root,QIODevice & IO)
```

對TAR 檔IO 檔案的內部包含檔案進行列表。

```
bool List(QDir root,QString filename)
```

對TAR 檔file 檔案的內部包含檔案進行列表。

```
bool Extract(QDir root,QIODevice & IO)
```

對TAR 檔IO 檔案的內部包含檔案進行解開。

```
bool Extract(QDir root,QString filename)
```

對TAR 檔file 檔案的內部包含檔案進行解開。

```
bool TarBall(QIODevice & IO,QDir root,QDir source,bool recursive = true)
```

將目錄source 當中的所有檔案進行TAR 檔IO 檔案執行封裝。

```
bool TarBall(QString filename,QDir root,QDir source,bool recursive = true)
```

將目錄source 當中的所有檔案進行TAR 檔filename 檔案執行封裝。

```
bool Interval(void)
```

大部份較長的執行程序當中都會執行一次Interval()函式。如果您希望在執行解壓縮或封裝的時期，插入自己的中點處理功能，您應該繼承這個函式。一般在Qt 當中，這個函式應該再去呼叫qApp->processEvents()。

```
void Report(void * hiddenFileInfo)
```

報告檔案記錄資訊。這個函式僅在對TAR 進行ListFile 檔案列表時被呼叫。內定沒有做任何事，您需要繼承這個函式來列出檔案資訊。

```
bool ListFile(QDir root,QIODevice & IO,void * hiddenFileInfo)
```

由IO 檔案中，抽取hiddenFileInfo 檔案，並且呼叫Report 來列出檔案資訊。

```
bool Extract(QDir root,QIODevice & IO,void * hiddenFileInfo)
```

由IO 檔案中，抽取hiddenFileInfo 檔案，並且分配相對應的處理方式。

```
bool Read(QIODevice & IO,QByteArray & data,qint64 size)
```

由IO 檔案中讀取size 位元組的數據到data 當中，IO 檔案一般是TAR 檔。

```
bool Skip(QIODevice & IO,qint64 size)
```

相等於IO . seek (IO.pos() + size) 。

一般用於TAR 檔案列表時，快速跳過不需要讀取的部分。

```
bool Write(QIODevice & IO,QByteArray & data)
```

將數據寫入IO 檔案中，一般是TAR 檔案。

```
bool WriteClose(QIODevice & IO)
```

關閉IO 檔案，一般是TAR 檔案。

```
bool WriteFile(QIODevice & IO,QFileInfo & file)
```

將檔案file 寫入IO 檔案中，一般是讀取硬碟中的檔案寫入TAR 檔案中。

```
bool ExtractFile(QDir root,QIODevice & IO,void * hiddenFileInfo)
```

創建並處理正常檔案資訊。

```
bool ExtractDir(QDir root,QIODevice & IO,void * hiddenFileInfo)
```

創建並處理目錄資訊。

```
bool ExtractLink(QDir root,QIODevice & IO,void * hiddenFileInfo)
```

創建並處理Symbolic link 檔資訊。

```
bool ExtractDEVs(QDir root,QIODevice & IO,void * hiddenFileInfo)
```

創建並處理設備檔資訊。

```
bool ExtractNext(QDir root,QIODevice & IO,void * hiddenFileInfo)
```

處理S-TAR In 封包。

```
bool ExtractEXT(QDir root,QIODevice & IO,void * hiddenFileInfo)
```

處理S-TAR Extended 封包。

```
bool setFileTime(QDir root,QIODevice & IO,void * hiddenFileInfo)
```

設定檔案的創建日期及最後修改日期。

```
bool setFileMode(QDir root,QIODevice & IO,void * hiddenFileInfo)
```

設定檔案的存取權限等相關資訊。

```
QFileInfoList Listing(QDir & root,QDir source)
```

掃描source 目錄當中的檔案列表。

```
bool WriteTAR(QIODevice & IO,QDir & root,QFileInfo & file)
```

將正常檔案file 寫入TAR 檔案中。

```
bool ToHiddenFileInfo(QDir & root,QFileInfo & file,void * hiddenFileInfo)
```

將QFileInfo 轉換成檔案記錄資訊。

```
void * NewHiddenFile(void)
```

配置檔案記錄資訊。使用者需要繼承這個函式，並自行配置用戶定義的等價資料結構。

```
void CleanHiddenFile(void * hiddenFileInfo)
```

清除檔案記錄資訊。

HiddenFileTypes 列舉

HiddenFileTypes 是檔案類型的列舉型態。

您必須定義ENABLE_HIDDEN_FILE_INFO_STRUCTURE 才能使用這個隱藏的定義，一般而言不建議您這樣做。建議的方式為自行定義一個等價的列舉及資料結構。

名稱	值	涵意

None	0	無。
Regular	1	正常檔案。
Link	2	Symbolic link 檔案。
Symbol	3	Symbol 設備檔。
Char	4	Char 設備檔。
Block	5	Block 設備檔。
Directory	6	目錄。
FIFO	7	FIFO 設備檔。
Reserved	8	保留使用。
Next	9	S-TAR 的下個封包記號。
Extended	10	S-TAR Extended 封包。

HiddenFileInfo 資料結構

HiddenFileInfo 是一個檔案相關資訊記錄。

您必須定義ENABLE_HIDDEN_FILE_INFO_STRUCTURE 才能使用這個隱藏的定義，一般而言不建議您這樣做。建議的方式為 自行定義一個等價的列舉及資料結構。

```
typedef struct HiddenFileInfo {
  bool            Archive     ; /* inside a Tar file or Zip file, and so on */
  QString         Root        ; /* Root directory or tarball name */
  QString         Filename    ; /* Filename or directory name of this entry */
  QString         System      ; /* Normally, this is operation system */
  qint64          mode        ; /* Unix only */
  qint64          uid         ; /* Unix only */
  qint64          gid         ; /* Unix only */
  qint64          size        ; /* File size, for directory, it is 0 */
  QDateTime       Time        ; /* Creation time */
  QDateTime       Lastest     ; /* Last modified */
  QString         CheckSum    ; /* CheckSum of this file */
  HiddenFileTypes Type        ; /* File Type */
```

```
  QString         LinkName    ; /* Unix only */
  QString         uname       ; /* Unix only */
  QString         gname       ; /* Unix only */
  qint64          Major       ; /* Unix only */
  qint64          Minor       ; /* Unix only */
  QString         Prefix      ; /* Normally, this is prefix directory */
  QString         Comment     ; /* This is from GZIP format, however, sometimes it is useful */
} HiddenFileInfo              ;
```

Qt TAR 使用範例

目錄

「QtTAR/examples/tartool」目錄當中有使用範例程式。一般而言，您必須自行繼承 QtTarBall 來處理TAR 檔案。如果您有使用壓縮格式，您應該使用QFile 當中的 DecoderFn 及EncoderFn 來處理壓縮格式。如果您是使用QtArchivers 的功能，QtArchivers 當中使用CIOS Data Manipulation Subsystem 的統一串流格式，您則不需要處理這類問題。

對Tar 檔案內容列表

```
void ListTarBall(QString filename)

{
  QDir     d = QDir::current ( )    ;
  TarBALL tarball                   ;
  tarball . List ( d , filename ) ;
}
```

解開Tar 檔案

```
void ExtractTarBall(QString filename,QDir root)

{
  TarBALL tarball                          ;
  tarball . Extract ( root , filename ) ;
}
```

將指定目錄的檔案列表封裝到 Tar 檔案

```
void MakeTarBall(QString filename,QDir src)

{
  TarBALL tarball                                ;
  QDir    root = QDir::current ( )               ;
  tarball . TarBall ( filename , root , src ) ;
}
```

Qt TAR 其他需要注意的事項

目錄

修改所支援的Qt 版本

目前的QtTAR 支援Qt 5.5.0，如果您使用的Qt 版本有所更新，您需要修改以下檔案：

QtTAR/.qmake.conf

```
load(qt_build_config)

MODULE_VERSION = 5.5.0

# change version into current Qt version every time your upgrade your Qt

CIOS_VERSION = 1.6.0
```

將 MODULE_VERSION 修改成例如 5.5.1 即可。