

```
#include "filefindervideo.h"
```

```
FileFinderVideo::FileFinderVideo(QObject *parent) :
    DatabaseOperationsVideo(parent)
{
}
```

```
void FileFinderVideo::getDirTree()
{
```

```
    if(mPath.front()==NULL)
    {
        return;
    }
```

```
    mDir= new QDir(mPath.front());
```

```
    //Assuming the directory mentioned in the mPath exist
```

```
    if(mDir->exists()==true)
```

```
    {
        mFileQueue.push(mDir->absolutePath());
    }
```

```
    else
    {
```

```
        //      QMessageBox::StandardButton vReply;
```

```
        //      vReply=QMessageBox::warning(NULL,"Error","Directory :: "+mPath.front()+"doesn't exist",QMessageBox::Ok);
```

```
        ///      QMessageBox::information(NULL,"Error","Directory :: /*+mPath.front()+*/ doesn't exist");
```

```
        //      if(vReply == QMessageBox::Ok)
```

```
        //      {
        //          return;
```

```
        //      }
    }
```

```
    QFileInfoList vFileList;
```

```
    QStringList vNameFilter,vDefaultNameFilter=mDir->nameFilters();
```

```
    vNameFilter<<"*.mp4"<<"*.mkv"<<"*.avi"<<"*.vorbis"<<"*.wmv"<<"*.mpeg"<<"*.3gp"<<"*.flv"<<"*.FLV";
```

```
    mDir->setSorting(QDir::Name);
```

```
    while(mFileQueue.empty()==false)
```

```
    {
        mDir->setPath(mFileQueue.front());
```

```
        mDir->setNameFilters(vNameFilter);
```

```
        mDir->setFilter(QDir::NoDotAndDotDot|QDir::Readable|QDir::Files);
```

```
        //      vFileList=mDir->entryInfoList();
```

```
        //      emitting the current directory as mDir -> absolutePath()
```

```
        //      and the files in the directory as mDir-> entryInfoList()
```

```
        updateDB(mPath.front(),QFileInfo(mDir->absolutePath()),mDir->entryInfoList());
```

```
        //      emit(updateDirTreeView(QFileInfo(mDir->absolutePath()),mDir->entryInfoList()));
```

```
        mDir->setNameFilters(vDefaultNameFilter);
```

```
        mDir->setFilter(QDir::NoDotAndDotDot|QDir::Readable|QDir::Dirs);
```

```
        vFileList=mDir->entryInfoList();
```

```
        for(QFileInfoList::size_type i=0; i <vFileList.count(); ++i)
```

```
        {
            mFileQueue.push(vFileList[i].absoluteFilePath());
```

```
    }

    mFileQueue.pop();

}
delete mDir;
}

void FileFinderVideo::setPath(QString vPath)
{
    mPath.push(vPath);
}

void FileFinderVideo::initiator()
{
    if(mPath.front() != NULL)
    {
        updateSource(mPath.front());
        getDirTree();
        updateTreeView();
        mPath.pop();
    }
}
```