

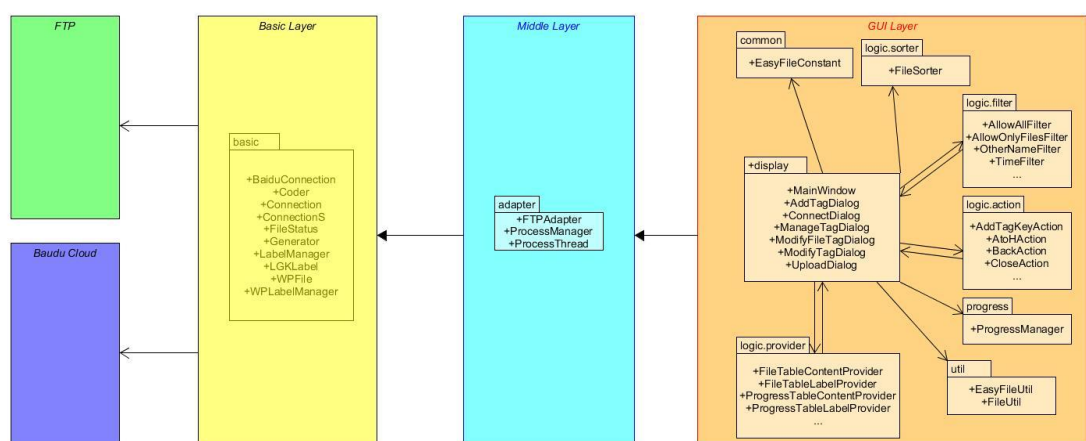
## 模块接口设计

## 模块间接口综述

对于工程中的三个模块，简述三个模块的分工如下：

1. **GUI Layer:** 为最上层，负责前端的用户界面显示，接口面向用户操作，向下通过一个中端模块的 **Singleton** 调用中端模块的接口和函数。
2. **Middle Layer:** 为中间层，负责前后端的连接，起到隔离前后端的相互影响的作用，向上层 GUI 模块提供接口，对于相应的接口，向下调用 **Basic** 层的接口和函数。
3. **Basic Layer:** 为底层，向上层提供客户端对 **FTP** 和网盘的操作接口，向下调用 **FTP** 和网盘的 **API**。

如下为模块之间的调用关系的 UML 包图:



## GUI Layer 接口设计

## 层间接口

GUI 层为最上层，所有的接口都是面向用户操作的，所以层间接口就不详细介绍了。

## 层内接口

从 UML 图中可以看到，GUI 层以 `display` 包为中心，进行交互。下面列举了 GUI 层的所有提供给同层类的接口。

## 类 MainWindow

构造函数	参数
<b>MainWindow()</b>	

返回类型	方法	说明
void	<b>addFTPHistory(String address)</b>	在 ftp 历史中加入 address 这个地址
String	<b>getAddressText()</b>	得到输入栏中的文件地址
void	<b>getFTPHistory()</b>	获取 FTP 的历史地址
String	<b>getFTPText()</b>	得到当前的 ftp 地址
int	<b>getMode()</b>	得到当前的 ftp 地址
int	<b>getSaveToWP()</b>	得到 saveToWP
IStructuredSelection	<b>getTableSelection()</b>	得到当前的表格选中项
void	<b>gotoFTP(String ftpAddress)</b>	连接到一个 ftp
void	<b>gotoPath(String path)</b>	进入到文件夹
void	<b>newDownload(List&lt;String&gt; r, List&lt;String&gt; l, int cm)</b>	新建一个下载任务
void	<b>newUpload(List&lt;String&gt; l, List&lt;String&gt; r, int wp, int cm)</b>	新建一个上传任务
void	<b>refresh()</b>	刷新主界面
void	<b>setFtpText(String ftp)</b>	设置 FTP 栏的地址
void	<b>updateProgressTable()</b>	更新进度栏

## 类 ManageTagDialog

构造函数	参数
<b>ManageTagDialog(MainWindow _window)</b>	_window - 父窗口

返回类型	方法	说明
void	<b>closeDialog()</b>	关闭对话框
IStructuredSelection	<b>getTableSelection()</b>	得到选中的标签
void	<b>refresh()</b>	刷新标签表格

## 类 ModifyFileTagDialog

构造函数	参数
<b>ModifyFileTagDialog(MainWindow _window, String _filePath)</b>	_window - 父窗口 _filePath - 文件路径

## 类 ModifyTagDialog

构造函数	参数
<b>ModifyTagDialog(ManageTagDialog _window, LGKLabel Tag)</b>	_window - 父窗口 Tag - 待修改的标签

类 UploadDialog

构造函数		参数
<b>UploadDialog(MainWindow _window)</b>		<b>_window</b> - 父窗口
返回类型	方法	说明
String	<b>getFileText()</b>	获取文件文本框中的内容
void	<b>setFileText(String fileName)</b>	设置文件输入框的内容

类 ConnectDialog

构造函数		参数
<b>ConnectDialog(MainWindow _window)</b>		<b>_window</b> - 父窗口
返回类型	方法	说明
void	<b>closeDialog()</b>	关闭窗口
String	<b>getPassword()</b>	获取输入框中的密码
String	<b>getUsername()</b>	获取输入框中的用户名
void	<b>login()</b>	通过用户名密码进行登录

类 AddTagKeyDialog

构造函数	参数
<b>public</b> <b>AddTagKeyDialog(ManageTagDialog _window)</b>	<b>_window</b> - 父窗口

类 AddTagDialog

构造函数	参数
<b>AddTagDialog(UploadDialog _window, String[] _uploadFiles)</b>	<b>_window</b> - 父窗口 <b>_uploadFiles</b> - 上传的文件列表

类 EasyFileConstant

返回类型	方法	说明
static Map<String,Integer>	<b>getSortMap()</b>	
static List<String>	<b>getSortTypeList()</b>	
static List<String>	<b>getSortTypeList(String types)</b>	
static Map<String,String>	<b>getSortTypeMap()</b>	
static Map<String,String>	<b>getSortTypeMap(String keys, String values)</b>	

类 ProgressManager

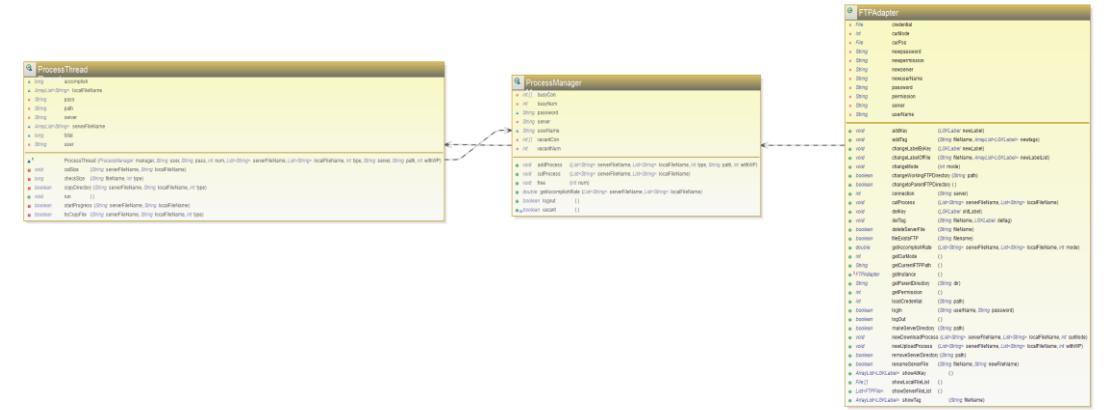
构造函数	参数
ProgressManager(MainWindow w)	w - 父窗口

返回类型	方法	说明
void	clicked(int index)	
void	deleteProgress(int index, int done)	
void	readFromFile()	
void	saveToFile()	
void	updateProgress()	

Middle Layer 接口设计

Middle Layer 层极为工程中的 adapter 包，它负责了前端和后端的连接，在整个工程中起到了至关重要的作用，它作用相当网络传输中 socket，包装了后端的各种操作，完全隔离了前端和后端的相互影响，前端不需要知道后端的实现细节，后端也不需要在意前端的实现方式，一切数据流动，都在这一层进行处理，把处理好的数据转交给前端和后端，并且对于后端会对前端卡死的进程，分配相应的线程，使得软件界面不卡死。所以，这一层的层间接口相当复杂与重要

Middle Layer UML 类图



层间接口

中间层提供给上层的接口都位于 FTPAdapter 类中，其层间接口如下：

类 FTPAdapter

返回类型	方法	说明
void	addKey(LGKLabel newLabel)	增加一种新的标签

void	<b>addTag(String</b> fileName, <b>ArrayList&lt;LGKLabel&gt;</b> newtags)	对文件增加标签
void	<b>changeLabelByKey(LGKLabel</b> newLabel)	修改一种标签的属性
void	<b>changeLabelOfFile(String</b> fileName, <b>ArrayList&lt;LGKLabel&gt;</b> newLabelList)	修改文件的各种标签
void	<b>changeMode(int</b> mode)	更改当前模式
boolean	<b>changetoParentFTPDirectory()</b>	前往 ftp 的父目录
boolean	<b>changetoParentLocalDirectory()</b>	前往本地的父目录
boolean	<b>changeWorkingFTPDirectory(String</b> path)	前往特定的 ftp 目录
boolean	<b>changeWorkingLocalDirectory(String</b> path)	前往特定的本地目录
int	<b>connection(String</b> server)	尝试连接 ftp 服务器
void	<b>cutProcess(List&lt;String&gt;</b> serverFileName, <b>List&lt;String&gt;</b> localFileName)	暂停或停止某任务——用于上传或下载任务
boolean	<b>deleteServerFile(String</b> fileName)	删除 ftp 服务器的某文件
void	<b>delKey(LGKLabel</b> oldLabel)	删除一种标签
void	<b>delTag(String</b> fileName, <b>LGKLabel</b> deltag)	删除标签
void	<b>downloadFileFromWP(String</b> serverFileName, <b>String</b> localFileName)	从网盘上下载文件
boolean	<b>fileExistsFTP(String</b> filename)	询问 ftp 服务器上某文件是否存在
boolean	<b>fileExistsLocal(String</b> filename)	询问本地某文件是否存在
double	<b>getAccomplishRate(List&lt;String&gt;</b> serverFileName, <b>List&lt;String&gt;</b> localFileName, <b>int</b> mode)	得到任务完成百分比
int	<b>getCurMode()</b>	得到当前模式
String	<b>getCurrentFTPPath()</b>	得到当前 ftp 路径
String	<b>getCurrentLocalPath()</b>	得到当前本地路径
String	<b>getCurServer()</b>	得到当前服务器名字
FTPFile	<b>getFile(String</b> file)	得到某路径的文件
List<FTPFile>	<b>getFileList(String</b> dir)	显示特定路径的文件列表
static FTPAdapter	<b>getInstance()</b>	FTPAdapter 为单件
String	<b>getParentDirectory(String</b> dir)	得到特定路径的父目录路径
int	<b>getPermission()</b>	获得用户的权限等级
FTPFile	<b>getWPFile(String</b> file)	得到网盘的文件
List<FTPFile>	<b>getWPFileList(String</b> dir)	得到特定网盘路径下

		的文件
String	<b>getWPParentDirectory(String dir)</b>	得到网盘特定目录的父目录
boolean	<b>haveLoggedIn()</b>	判断当前是否已登录
boolean	<b>isConnection()</b>	判断当前是否正在连接
int	<b>loadCredential(String path)</b>	导入证书
boolean	<b>login(String userName, String password)</b>	用户登录
boolean	<b>logout()</b>	用户登出
boolean	<b>makeServerDirectory(String path)</b>	在 ftp 所求目录中新建一个空文件夹
void	<b>newDownloadProcess(List&lt;String&gt; serverFileName, List&lt;String&gt; localFileName, int curMode)</b>	新建或继续下载任务
void	<b>newUploadProcess(List&lt;String&gt; serverFileName, List&lt;String&gt; localFileName, int withWP)</b>	新建或继续上传任务
boolean	<b>removeServerDirectory(String path)</b>	把一个空文件夹在 ftp 服务器目录中删掉
boolean	<b>renameServerFile(String fileName, String newFileName)</b>	对 ftp 服务器的文件进行重命名
ArrayList<LGKLabel>	<b>showAllKey()</b>	显示所有种类的标签
File[]	<b>showLocalFileList()</b>	显示当前本地目录下的文件列表
List< FTPFile>	<b>showServerFileList()</b>	显示当前 ftp 目录下的文件列表
ArrayList<LGKLabel>	<b>showTag(String fileName)</b>	显示文件的标签列表

## 层内接口

adapter 包中的其余两个类都是服务于 FTPAdapter 类的，因此这两个类中的接口均为层内接口。类 ProcessManager 负责处理所有来自 FTPAdapter 的上传或下载的任务请求，类 ProcessThread 负责具体执行一个任务，并且得出任务完成的百分比。

类 ProcessManager

构造函数	参数
<b>ProcessManager(String userName, String password, String server)</b>	userName – 用户名字 password – 用户密码 server – 服务器名字

返回类型	方法	说明
boolean	<b>logout()</b>	用户登出
void	<b>free(int num)</b>	释放已完成的任务
boolean	<b>vacant()</b>	检查是否有可用资源
void	<b>addProcess(List&lt;String&gt; serverFileName, List&lt;String&gt; localFileName, int type, String path, final int withWP)</b>	新增任务
void	<b>cutProcess(List&lt;String&gt; serverFileName, List&lt;String&gt; localFileName)</b>	中断任务
double	<b>getAccomplishRate(List&lt;String&gt; serverFileName, List&lt;String&gt; localFileName)</b>	得到任务完成百分比

## 类 ProcessThread

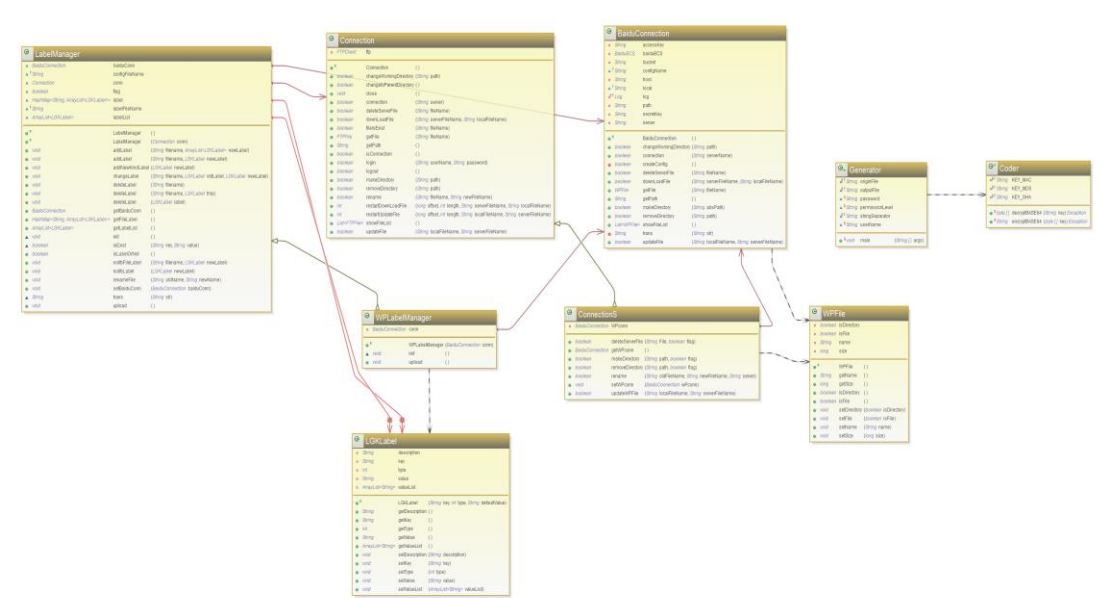
构造函数	参数
<b>ProcessThread(ProcessManager manager, String user, String pass, int num, List&lt;String&gt; serverFileName, List&lt;String&gt; localFileName, int type, String server, String path, int withWP)</b>	manager – 任务管理器的引用 user – 用户名字 pass – 用户密码 num – 所用的线程序号 serverFileName – 服务器文件路径 localFileName – 本地文件路径 type – 上传或者下载 server – 服务器名字 path – 服务器文件父目录 withWP – 是否和网盘同步

返回类型	方法	说明
void	<b>run()</b>	线程的执行
boolean	<b>startProgress(String serverFileName, String localFileName)</b>	开始复制的过程
void	<b>calSize(String serverFileName, String localFileName)</b>	计算文件列表的总大小
long	<b>checkSize(String fileName, int type)</b>	计算任务的文件总大小
boolean	<b>copyDirectory(String serverFileName, String localFileName, int type)</b>	复制文件夹
boolean	<b>tryCopyFile(String serverFileName, String localFileName, int type)</b>	复制文件

# Basic Layer 接口设计

Basic Layer 作为本次工程的最底端代码，负责了所有与 FTP 和网盘的直接操作，主要就是调用了 FTP 和网盘的 API。在该层中，接口间的调用较为复杂，其为上层提供相应的层间接口，同时这些接口也会提供给同层的一些函数的使用。

## Basic Layer UML 类图



## 接口概述：

### 类 BaiduConnection

构造函数		参数
BaiduConnection()		
返回类型	方法	说明
boolean	<code>createConfig()</code>	向百度网盘建立连接
boolean	<code>connection()</code>	
boolean	<code>downloadFile(String serverFileName, String localFileName)</code>	从百度云下载文件
boolean	<code>updateFile(String localFileName, String serverFileName)</code>	向百度云上传文件
boolean	<code>makeDirectory(String absPath)</code>	在网盘上新建一个文件夹
boolean	<code>deleteServerFile(String fileName)</code>	删除服务器上的文



		件
boolean	<b>removeDirectory(String path)</b>	删除服务器上的文件夹
String	<b>trans(String str)</b>	转换路径格式
boolean	<b>changeWorkingDirectory(String path)</b>	切换当前路径
List<WPFile>	<b>showFileList()</b>	获取当前路径下的文件列表
WPFile	<b>getFile(String fileName)</b>	获取特定文件的信息

## 类 Coder

构造函数		参数
<b>Coder()</b>		
返回类型	方法	说明
byte[]	<b>decryptBASE64(String key)</b>	解密 BASE64
String	<b>encryptBASE64(byte[] key)</b>	加密 BASE64

## 类 Connection

构造函数		参数
<b>Connection()</b>		
返回类型	方法	说明
boolean	<b>changetoParentDirectory()</b>	ftp 文件路径返回父目录
boolean	<b>changeWorkingDirectory(String path)</b>	ftp 跳转到指定文件目录
void	<b>close()</b>	关闭连接
boolean	<b>connection(String server)</b>	建立一个 ftp 链接
boolean	<b>deleteServerFile(String fileName)</b>	删除服务器端指定文件
boolean	<b>downloadFile(String serverFileName, String localFileName)</b>	下载 ftp 指定文件
boolean	<b>fileIsExist(String fileName)</b>	返回 ftp 服务器上是否有某个文件
FTPFile	<b>getFile(String fileName)</b>	返回 ftp 中的指定文件
String	<b>getPath()</b>	得到 ftp 服务器当前文件路径
boolean	<b>isConnection()</b>	返回 ftp 是否连接
boolean	<b>login(String userName, String password)</b>	用户登录
boolean	<b>logout()</b>	用户登出

boolean	<b>makeDirectory(String path)</b>	在 ftp 客户端新建一个文件夹
boolean	<b>removeDirectory(String path)</b>	删除服务器端一个文件夹
boolean	<b>rename(String fileName, String newFileName)</b>	重命名一个服务器端文件
int	<b>restartDownloadFile(long offset, int length, String serverFileName, String localFileName)</b>	下载服务器文件到本地文件, 数据位置从 offset 位置开始 length 个 bytes
int	<b>restartUpdateFile(long offset, int length, String localFileName, String serverFileName)</b>	上传服务器文件到本地文件, 数据位置从 offset 位置开始 length 个 bytes
List<FTPFile>	<b>showFileList()</b>	返回当前目录的文件列表
boolean	<b>updateFile(String localFileName, String serverFileName)</b>	上传指定文件

## 类 ConnectionS

构造函数	参数
<b>ConnectionS()</b>	

返回类型	方法	说明
boolean	<b>deleteServerFile(String File, boolean flag)</b>	ftp 文件路径返回父目录
BaiduConnection	<b>getWPconn()</b>	ftp 跳转到指定文件目录
boolean	<b>makeDirectory(String path, boolean flag)</b>	关闭连接
boolean	<b>removeDirectory(String path, boolean flag)</b>	建立一个 ftp 链接
boolean	<b>rename(String oldFileName, String newFileName, String server)</b>	删除服务器端指定文件
void	<b>setWPconn(BaiduConnection wPconn)</b>	下载 ftp 指定文件
boolean	<b>updateWPFile(String localFileName, String serverFileName)</b>	返回 ftp 服务器上是否有某个文件

## 类 LabelManager

构造函数	参数
<b>WPFile()</b>	
<b>LabelManager(Connection conn)</b>	conn — FTP 连接

返回类型	方法	说明
void	<b>addLabel</b> (String filename, ArrayList<LGKLabel> newLabel)	ftp 文件路径返回父目录
void	<b>addLabel</b> (String filename, LGKLabel newLabel)	ftp 跳转到指定文件目录
void	<b>addNewKindLabel</b> (LGKLabel newLabel)	关闭连接
void	<b>changeLabel</b> (String filename, LGKLabel oldLabel, LGKLabel newLabel)	建立一个 ftp 链接
void	<b>deleteLabel</b> (LGKLabel label)	删除服务器端指定文件
void	<b>deleteLabel</b> (String filename)	下载 ftp 指定文件
void	<b>deleteLabel</b> (String filename, LGKLabel tmp)	返回 ftp 服务器上是否有某个文件
HashMap<String,ArrayList<LGKLabel>>	<b>getFileLabel</b> ()	返回这个 ftp 系统所有文件对应的标签
ArrayList<LGKLabel>	<b>getLabelList</b> ()	得到这个 ftp 系统有什么标签, 返回一个标签列表
boolean	<b>isLabelOrNot</b> ()	返回这个 ftp 系统是否有标签
void	<b>notifyLabel</b> (LGKLabel newLabel)	更改一个标签类型的属性, 但是不能给该 key 值。
void	<b>renameFile</b> (String oldName, String newName)	重命名一个文件, 对标签系统进行相应操作
void	<b>upload</b> ()	上传标签系统文件到 ftp 上。

## 类 LGKLabel

构造函数	参数
<b>LGKLabel</b> (String key, int type, String defaultVal)	key – 标签关键字 type – 标签类型 defaultVal – 标签默认值

## 类 WPLabelManager

构造函数	参数
<b>WPLabelManager</b> (BaiduConnection conn)	

返回类型	方法	说明
void	<b>upload</b> ()	上传标签文件

