

# 功能扩展

## 讨论基于组件的开发

### Server/Client保存所有收到的消息到文件

- 文件格式不限
- 文件路径可配置

解决：

- 在原有的PM构件基础上，增加 `writeFile(String filename,String content)` 接口，实现向文件中写入收到的消息功能
- Server中，日志全都由 `LoggerHandler` 这个handler来处理，所以只要在收到聊天消息的条件下，将该消息写入文件即可，将server端收到的消息报存在./messageRecords/server.txt中，格式如下：

```
[yyyy-MM-dd HH:mm:ss] 用户帐号 消息内容 ``java if( messageType ==
MessageType.CHATTING){ receivedMessageNumber.record() //记录开始 Date now = new Date();
SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");//可以方便地修改
日期格式 String snow = dateFormat.format(now); Log.writeFile("./messageRecords/server.txt","["+
snow+ "]" + " "+message.getChatContent().getAccount()+": "+message.getChatContent().toString()); //
记录结束 if (messageStatus == MessageStatus.NEEDHANDLED) forwardMessageNumber.record();
else ignoredMessageNumber.record(); }
```

- Client中，日志全都由`ClientLoggerHandler`这个handler来处理，所以只要在确认消息类型为收到其他用户消息的条件下，将该消息记录下来即可。记录在messageRecords目录下以用户账户号为文件名的`.txt`文件中，格式如下：

`[yyyy-MM-dd HH:mm:ss] 用户帐号 消息内容`

```
````java
if(type == ACKType.OTHERSMESSAGE){
    receiveMsgNumber.record();
    //记录开始
    Date now = new Date();
    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    //可以方便地修改日期格式
    String snow = dateFormat.format(now);
    Log.writeFile("./messageRecords/"+account+".txt","["+ snow+ "]" + " "
+ ack.getChatContent().getAccount()+": " + ack.getChatContent().toString());
    //记录结束
}
```

## 每天所有的输出文件(Server/Client/PM)归档成一个压缩包

- 压缩格式不限
- 压缩包路径可配置

在Server和Client端收到消息这个部分，我们扩展了自己的Log模块部分，使它能够定时地完成压缩任务。

```
Log.setCompressPath("message_records")
//每天凌晨压缩一次
Log.startCompressSchedule()
//可以随时结束压缩计划
Log.endCompressSchedule()
```

在PM部分，我们则依然使用了第五组的PM模块，很方便地完成了pm记录文件的压缩。

```
LogReporter realtimeReporter = new RollingFileReporter("present/server-log-%d{yyyy-MM-dd_HH-mm}.log");
LogReporter archiveReporter = new RollingFileReporter("archive/server-log-%d{yyyy-MM-dd}.zip");

PerformanceMonitor monitor = new PerformanceMonitor(realtimeReporter, archiveReporter);
```

## 同组成员在线维护

用户能够时，能够查看到所有与他同组的在线的成员 其他同组成员上下线时，别的用户能够得到提示



用户登陆时，能够看到他所在的组的之前发送的消息

