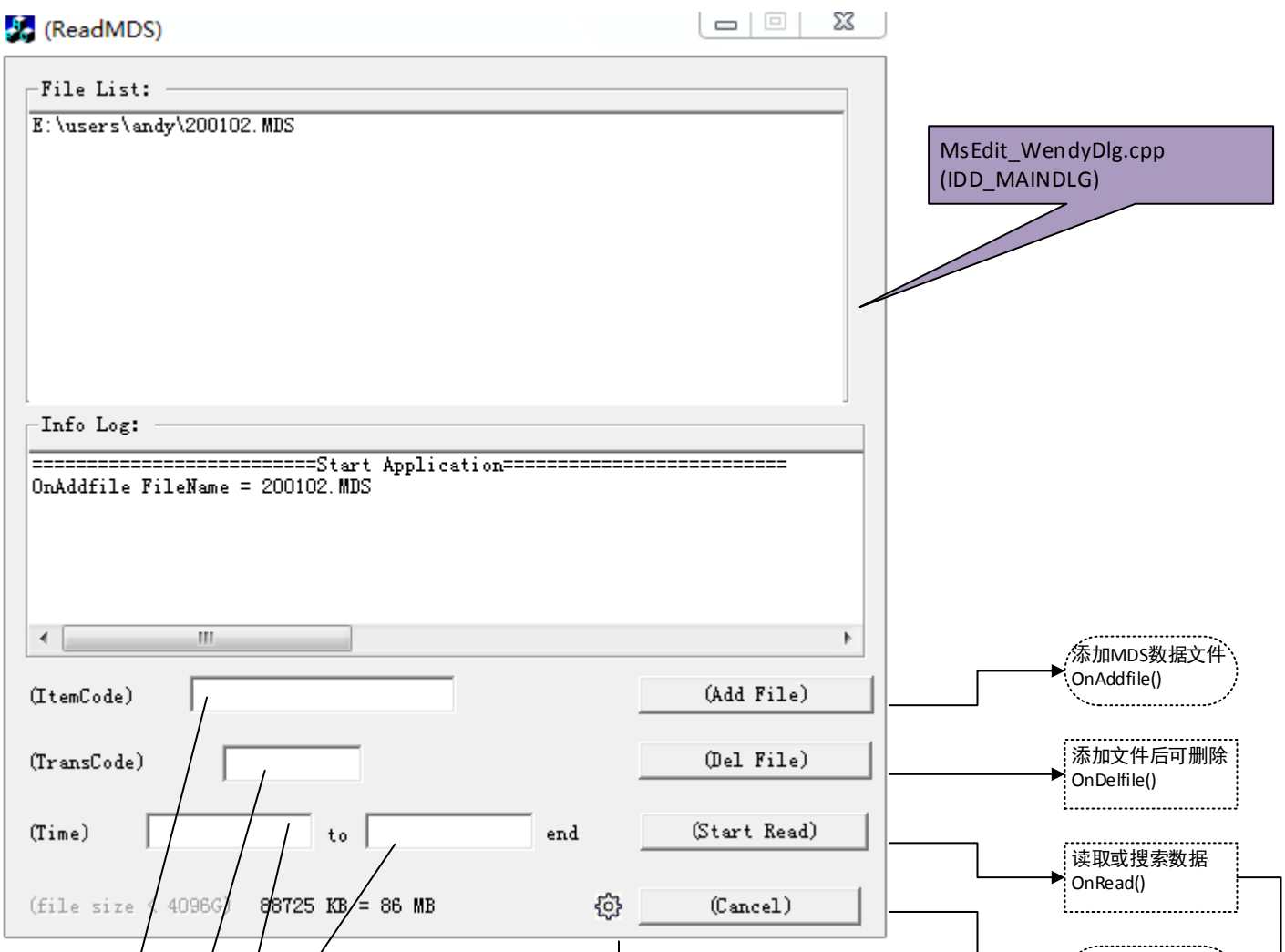
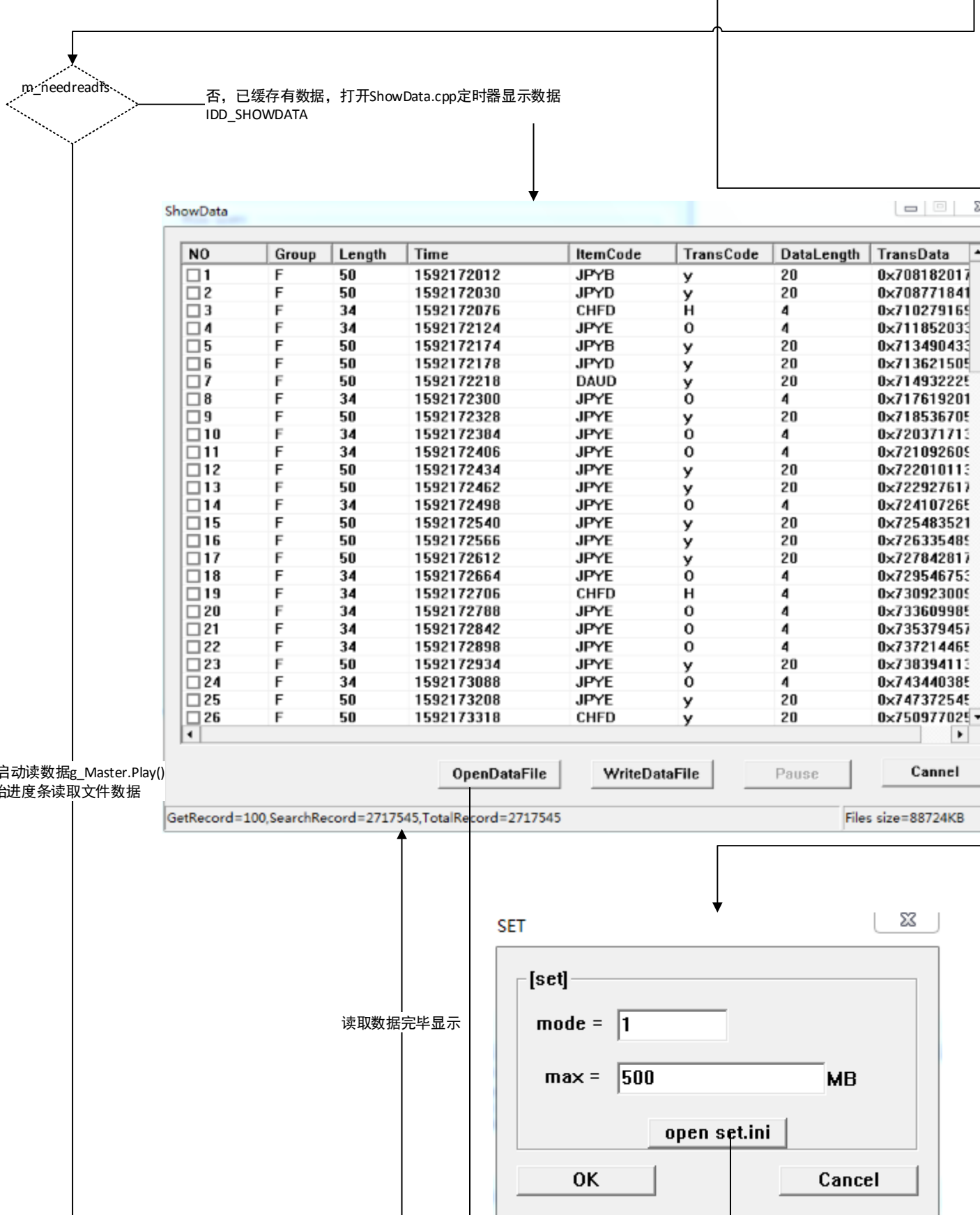
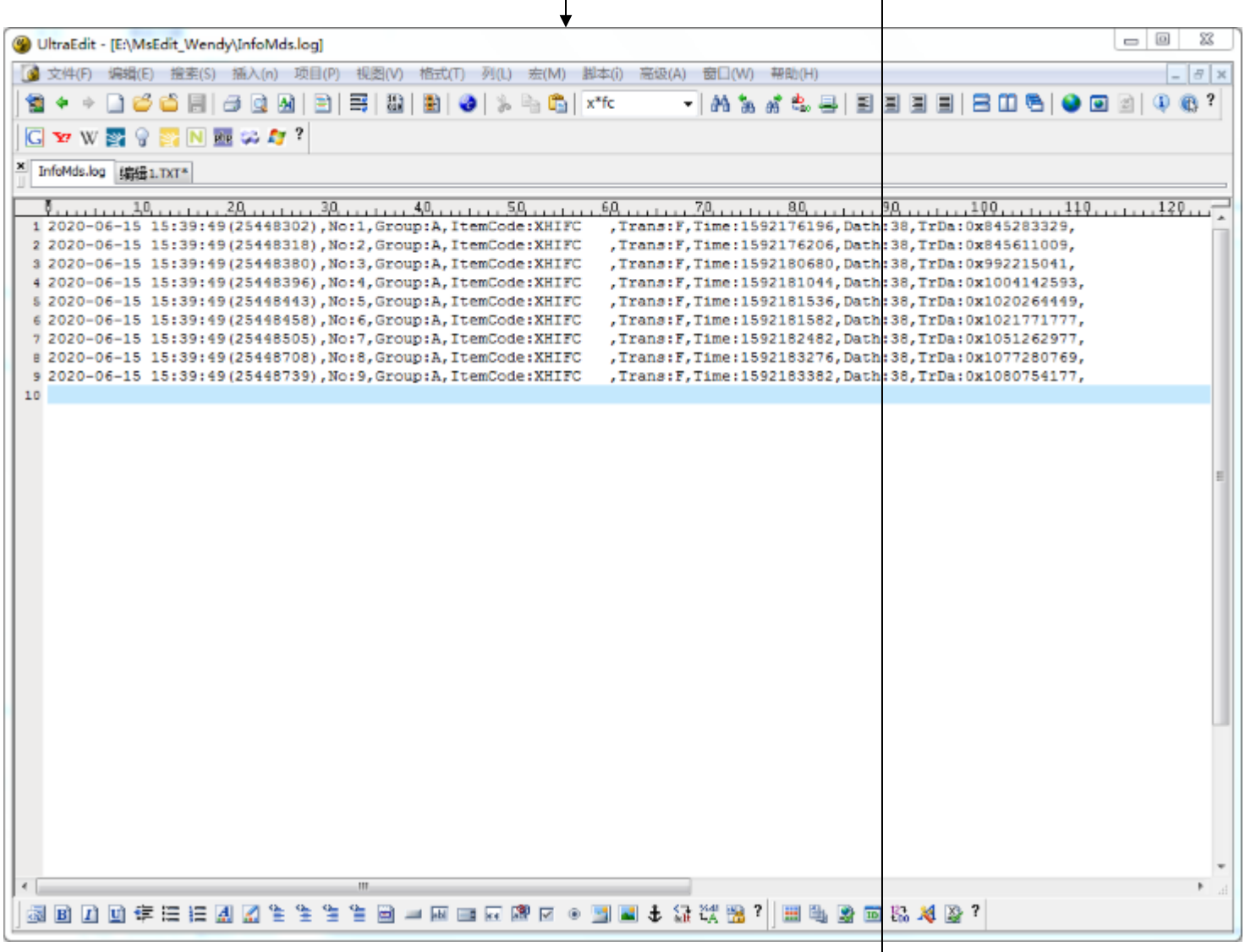
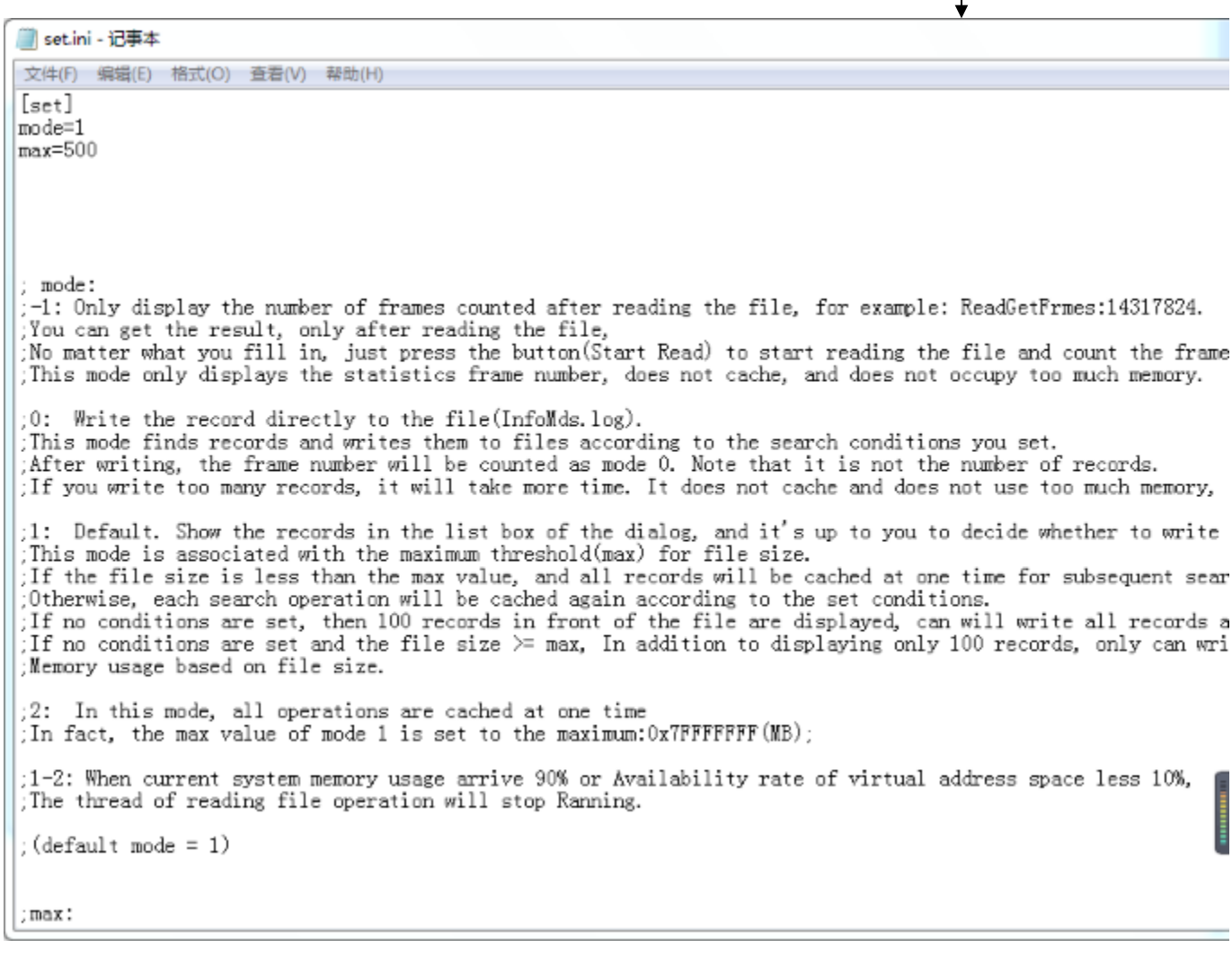
软件界面简单流程图

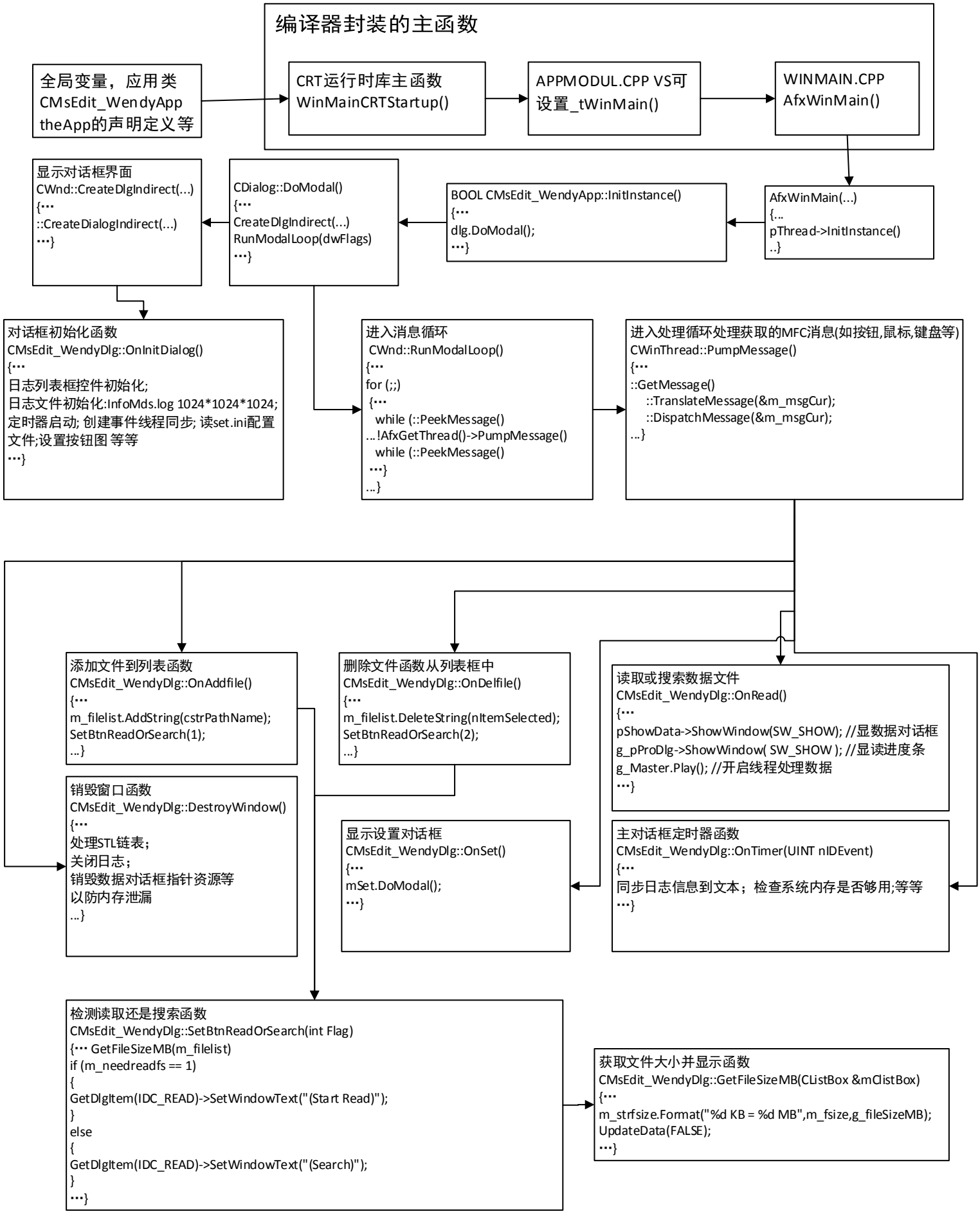

(ReadMDS)
File List:
E:\users\andy\200102.MDS
Info Log:
==========Start Application==========
OnAddfile FileName = 200102.MDS
(ItemCode)
(TransCode)
(Time)
to
end
(file size < 4096G)
88725 KB = 86 MB
(Add File)
(Del File)
(Start Read)
(Cancel)
MsEdit_WendyDlg.cpp
(IDD_MAINDLG)
添加MDS数据文件
OnAddfile()
添加文件后可删除
OnDelfile()
读取或搜索数据
OnRead()
退出程序
设置搜索条件的编辑框
m_item m_TransCode m_time
设置mode，max值
OnSet()


m_needreadfs
否，已缓存有数据，打开ShowData.cpp定时器显示数据
IDD_SHOWDATA
ShowData
NO Group Length Time ItemCode TransCode DataLength TransData
1 F 50 1592172012 JPYB y 20 0x708182017
2 F 50 1592172030 JPYD y 20 0x708771841
3 F 34 1592172076 CHFD H 4 0x710279169
4 F 34 1592172124 JPYE 0 4 0x711852033
5 F 50 1592172174 JPYB y 20 0x713490433
6 F 50 1592172178 JPYD y 20 0x713621505
7 F 50 1592172218 DAUD y 20 0x714932225
8 F 34 1592172300 JPYE 0 4 0x717619201
9 F 50 1592172328 JPYE y 20 0x718536705
10 F 34 1592172384 JPYE 0 4 0x720371713
11 F 34 1592172406 JPYE 0 4 0x721092609
12 F 50 1592172434 JPYE y 20 0x722010113
13 F 50 1592172462 JPYE y 20 0x722927617
14 F 34 1592172498 JPYE 0 4 0x724107265
15 F 50 1592172540 JPYE y 20 0x725483521
16 F 50 1592172566 JPYE y 20 0x726335485
17 F 50 1592172612 JPYE y 20 0x727842817
18 F 34 1592172664 JPYE 0 4 0x729546753
19 F 34 1592172706 CHFD H 4 0x730923009
20 F 34 1592172788 JPYE 0 4 0x733609985
21 F 34 1592172842 JPYE 0 4 0x735379457
22 F 34 1592172898 JPYE 0 4 0x737214465
23 F 50 1592172934 JPYE y 20 0x738394113
24 F 34 1592173088 JPYE 0 4 0x743440385
25 F 50 1592173208 JPYE y 20 0x747372545
26 F 50 1592173318 CHFD y 20 0x750977025
OpenDataFile
WriteDataFile
Pause
Cannel
GetRecord=100,SearchRecord=2717545,TotalRecord=2717545
Files size=88724KB
是,启动读数据g_Master.Play()
开始进度条读取文件数据
读取数据完毕显示
SET
[set]
mode = 1
max = 500 MB
open set.ini
OK
Cancel
Reading data...Do not operate other
Read: 24060 KB
27%
FilesSize: 88725 KB
OK
Cannel

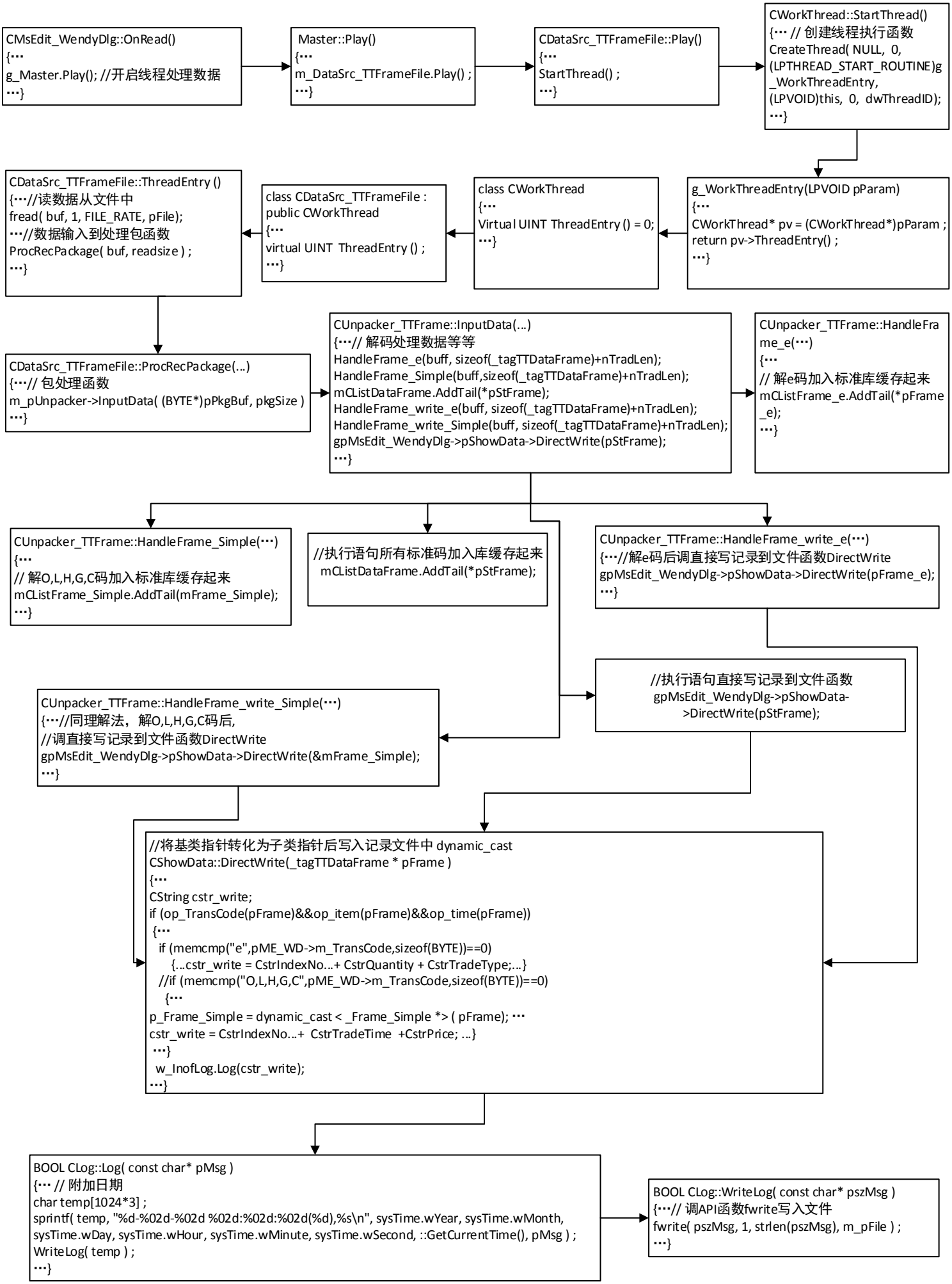

UltraEdit - [E:\MsEdit_Wendy\InfoMds.log]
InfoMds.log
编辑1.TXT*
1 2020-06-15 15:39:49(25448302),No:1,Group:A,ItemCode:XHIFC ,Trans:F,Time:1592176196,Dath:38,TrDa:0x845283329,
2 2020-06-15 15:39:49(25448318),No:2,Group:A,ItemCode:XHIFC ,Trans:F,Time:1592176206,Dath:38,TrDa:0x845611009,
3 2020-06-15 15:39:49(25448380),No:3,Group:A,ItemCode:XHIFC ,Trans:F,Time:1592180680,Dath:38,TrDa:0x992215041,
4 2020-06-15 15:39:49(25448396),No:4,Group:A,ItemCode:XHIFC ,Trans:F,Time:1592181044,Dath:38,TrDa:0x1004142593,
5 2020-06-15 15:39:49(25448443),No:5,Group:A,ItemCode:XHIFC ,Trans:F,Time:1592181536,Dath:38,TrDa:0x1020264449,
6 2020-06-15 15:39:49(25448458),No:6,Group:A,ItemCode:XHIFC ,Trans:F,Time:1592181582,Dath:38,TrDa:0x1021771777,
7 2020-06-15 15:39:49(25448505),No:7,Group:A,ItemCode:XHIFC ,Trans:F,Time:1592182482,Dath:38,TrDa:0x1051262977,
8 2020-06-15 15:39:49(25448708),No:8,Group:A,ItemCode:XHIFC ,Trans:F,Time:1592183276,Dath:38,TrDa:0x1077280769,
9 2020-06-15 15:39:49(25448739),No:9,Group:A,ItemCode:XHIFC ,Trans:F,Time:1592183382,Dath:38,TrDa:0x1080754177,
10


set.ini - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
[set]
mode=1
max=500
; mode:
;-1: Only display the number of frames counted after reading the file, for example: ReadGetFrmes:14317824.
;You can get the result, only after reading the file,
;No matter what you fill in, just press the button(Start Read) to start reading the file and count the frame
;This mode only displays the statistics frame number, does not cache, and does not occupy too much memory.
;0: Write the record directly to the file(InfoMds.log).
;This mode finds records and writes them to files according to the search conditions you set.
;After writing, the frame number will be counted as mode 0. Note that it is not the number of records.
;If you write too many records, it will take more time. It does not cache and does not use too much memory,
;1: Default. Show the records in the list box of the dialog, and it's up to you to decide whether to write
;This mode is associated with the maximum threshold(max) for file size.
;If the file size is less than the max value, and all records will be cached at one time for subsequent sear
;Otherwise, each search operation will be cached again according to the set conditions.
;If no conditions are set, then 100 records in front of the file are displayed, can will write all records a
;If no conditions are set and the file size >= max, In addition to displaying only 100 records, only can wri
;Memory usage based on file size.
;2: In this mode, all operations are cached at one time
;In fact, the max value of mode 1 is set to the maximum:0x7FFFFFFF(MB);
;1-2: When current system memory usage arrive 90% or Availability rate of virtual address space less 10%,
;The thread of reading file operation will stop Ranning.
;(default mode = 1)
;max:

程序主对话框IDD_MAINDLG流程



线程读数据文件


CMsEdit_WendyDlg::OnRead()
{…
g_Master.Play(); //开启线程处理数据
…}
Master::Play()
{…
m_DataSrc_TTFrameFile.Play() ;
…}
CDataSrc_TTFrameFile::Play()
{…
StartThread() ;
…}
CWorkThread::StartThread()
{… // 创建线程执行函数
CreateThread(NULL, 0,
(LPTHREAD_START_ROUTINE)g_WorkThreadEntry,
(LPVOID)this, 0, dwThreadID);
…}
g_WorkThreadEntry(LPVOID pParam)
{…
CWorkThread* pv = (CWorkThread*)pParam ;
return pv->ThreadEntry() ;
…}
class CWorkThread
{…
Virtual UINT ThreadEntry () = 0;
…}
class CDataSrc_TTFrameFile :
public CWorkThread
{…
virtual UINT ThreadEntry () ;
…}
CDataSrc_TTFrameFile::ThreadEntry ()
{…//读数据从文件中
fread(buf, 1, FILE_RATE, pFile);
…//数据输入到处理包函数
ProcRecPackage(buf, readsize) ;
…}
CDataSrc_TTFrameFile::ProcRecPackage(…)
{…// 包处理函数
m_pUnpacker->InputData((BYTE*)pPkgBuf, pkgSize)
…}
CUnpacker_TTFrame::InputData(…)
{…// 解码处理数据等等
HandleFrame_e(buff, sizeof(_tagTTDataFrame)+nTradLen);
HandleFrame_Simple(buff,sizeof(_tagTTDataFrame)+nTradLen);
mCListDataFrame.AddTail(*pStFrame);
HandleFrame_write_e(buff, sizeof(_tagTTDataFrame)+nTradLen);
HandleFrame_write_Simple(buff, sizeof(_tagTTDataFrame)+nTradLen);
gpMsEdit_WendyDlg->pShowData->DirectWrite(pStFrame);
…}
CUnpacker_TTFrame::HandleFrame_e(…)
{…
// 解e码加入标准库缓存起来
mCListFrame_e.AddTail(*pFrame_e);
…}
CUnpacker_TTFrame::HandleFrame_Simple(…)
{…
// 解O,L,H,G,C码加入标准库缓存起来
mCListFrame_Simple.AddTail(mFrame_Simple);
…}
//执行语句所有标准码加入库缓存起来
mCListDataFrame.AddTail(*pStFrame);
CUnpacker_TTFrame::HandleFrame_write_e(…)
{…//解e码后调直接写记录到文件函数DirectWrite
gpMsEdit_WendyDlg->pShowData->DirectWrite(pFrame_e);
…}
//执行语句直接写记录到文件函数
gpMsEdit_WendyDlg->pShowData->DirectWrite(pStFrame);
CUnpacker_TTFrame::HandleFrame_write_Simple(…)
{…//同理解法，解O,L,H,G,C码后,
//调直接写记录到文件函数DirectWrite
gpMsEdit_WendyDlg->pShowData->DirectWrite(&mFrame_Simple);
…}
//将基类指针转化为子类指针后写入记录文件中 dynamic_cast
CShowData::DirectWrite(_tagTTDataFrame * pFrame)
{…
CString cstr_write;
if (op_TransCode(pFrame)&&op_item(pFrame)&&op_time(pFrame))
{…
if (memcmp("e",pME_WD->m_TransCode,sizeof(BYTE))==0)
{…cstr_write = CstrIndexNo…+ CstrQuantity + CstrTradeType;…}
//if (memcmp("O,L,H,G,C",pME_WD->m_TransCode,sizeof(BYTE))==0)
{…
p_Frame_Simple = dynamic_cast < _Frame_Simple *> (pFrame); …
cstr_write = CstrIndexNo…+ CstrTradeTime +CstrPrice; …}
…}
w_InofLog.Log(cstr_write);
…}
BOOL CLog::Log(const char* pMsg)
{… // 附加日期
char temp[1024*3] ;
sprintf(temp, "%d-%02d-%02d %02d:%02d:%02d(%d),%s\n", sysTime.wYear, sysTime.wMonth,
sysTime.wDay, sysTime.wHour, sysTime.wMinute, sysTime.wSecond, ::GetCurrentTime(), pMsg) ;
WriteLog(temp) ;
…}
BOOL CLog::WriteLog(const char* pszMsg)
{…// 调API函数fwrite写入文件
fwrite(pszMsg, 1, strlen(pszMsg), m_pFile) ;
…}

对话框显示数据记录

```
//读取数据文件函数
CMsEdit_WendyDlg::OnRead()
{…
pShowData-
>Create(IDD_SHOWDATA);//
…}
```

```
BOOL CShowData::OnInitDialog()
{…
//状态栏初始化
//列表控件列名定义等
//记录总数量获取
//定时器启动
//计算文件总大小并显示等等
…}
```

```
//定时器函数
void CShowData::OnTimer(UINT nIDEvent)
{…KillTimer(1);
DWORD retEvent = WaitForSingleObject(g_hEvent, 0);
//循环检测有无读完数据的事件信号，有则进入函数
OnTimer_Frame_To_List();
SetTimer( 1, 130, NULL ) ;
…}
```

```
显示数据到列表控件中
int CShowData::OnTimer_Frame_To_List(void)
{…(POSITION ps;)
//获取首位置Ps,获取记录总数TotalRecord,
 for循环获取STL容器中的数据GetAt(ps)
for(int i=0; (ps && i<1000) ; )
数据显示到列表控件中
        if (op_TransCode(pFramePara)&&op_item(pFramePara)&&op_time(pFramePara))
        {
                OnTimer_Frame_To_ListPara(pFramePara,m_list_data);          // wendy add 2020.06.04
                i++;
        }
…}
SearchRecord ++;
SearchRecord = TotalRecord;
GetNext(ps); i++; 到1000次或ps为空时跳出循环.
if (SearchRecord < TotalRecord)
{…SetTimer( 1, 30, NULL ) ;当小于总数时继续启动定时器搜索…}
Else
{…enable_timer = false;…}
```

```
int CShowData::OnTimer_Frame_To_ListPara(_tagTTDataFrame *pFrame, CListCtrl &mp_list_data)  // wendy add 2020.06.04
{…
        itoa(GetRecord+1,string,10);
        mp_list_data.InsertItem(GetRecord,string,0);

        memcpy(GroupCodeString,&pFrame->btGroupCode,1);
        mp_list_data.SetItemText(GetRecord,1, GroupCodeString);

//将基类指针转化为子类指针, 然后数据显示到列表控件中. dynamic_cast
p_Frame_Simple = dynamic_cast < _Frame_Simple *> ( pFrame);

        CStrpTransData.Format("0x%ld",(long)pFrame->pTransData);
        mp_list_data.SetItemText(GetRecord,7, CStrpTransData);
        GetRecord++;

        return 0;
}
```

```
处理循环获取的MFC消息(如按钮,鼠标,键盘等)
CWinThread::PumpMessage()
{…
::GetMessage()
   ::TranslateMessage(&m_msgCur);
   ::DispatchMessage(&m_msgCur);
…}
```

```
//点击写数据到文件函数，与显示到列表控件类似，但又有所不同。
void CShowData::OnWriteSearch()
{…POSITION ps;
while(ps )
if (op_TransCode(pFramePara_wf)&&op_item(pFramePara_wf)&&op_time(pFramePara_wf))
p_Frame_Simple = dynamic_cast < _Frame_Simple *> ( pFramePara_wf);
cstr_write = CstrIndexNo + CstrGroup + CStrItemCode  + CstrTrans +  CstrTradeTime  +CstrPrice;
w_InofLog.Log(cstr_write);
GetDlgItem(IDC_WRITE_SEARCH)->EnableWindow(TRUE);
…}
```

```
//打开同目录下的数据记录文件
void CShowData::OnOpenDatafile()
{
// TODO: Add your control notification handler code here
ShellExecute(NULL,"open",".\\InfoMds.log",NULL,NULL,SW_SHOWNORMAL); // 打开当前目录下的set.ini 文件
}
```

设置模式对话框

显示设置对话框
CMsEdit_WendyDlg::OnSet()
{…
mSet.DoModal();
…}

```
//设置对话框初始化
BOOL CSet::OnInitDialog()
{…//获取配置文件的参数并显示
mode_set =GetPrivateProfileInt("set","mode",1,".\\set.ini");
maxMB_set =GetPrivateProfileInt("set","max",500,".\\set.ini");
SetDlgItemText(IDC_EDIT_MODE,str);
SetDlgItemText(IDC_EDIT_MAX,str);
…}
```

```
//Set设置对话框的处理循环获取的MFC消息(如按钮,鼠标,键盘等)
//注意：虽然代码相同，但是与主对话框的特环消息处理函数不是同一实例
CWinThread::PumpMessage()
{…
::GetMessage()
    ::TranslateMessage(&m_msgCur);
    ::DispatchMessage(&m_msgCur);
…}
```

```
BOOL CSet::PreTranslateMessage(MSG* pMsg)
{… // 一系列代码限制并保证输入的参数正确性
// 获取edit中文本
//只允许输入 数字, Backspace,Delete,-
//只允许输入一个-且放在开始位置.
// 不允许在-前输入数字
…}
```

```
//打开同目录下的设置模式的配置文件
void CSet::OnOpenSetfile()
{
// TODO: Add your control notification handler code here
ShellExecute(NULL,"open",".\\set.ini",NULL,NULL,SW_SHOWNORMAL); // 打开当前目录下的set.ini 文件
}
```

```
//点击OK确定按钮后的处理函数
void CSet::OnOK()
{…
WritePrivateProfileString("set","mode",str,".\\set.ini");  //模式参数写入文件
WritePrivateProfileString("set","max",str,".\\set.ini");   //最大文件数值参数写入文件
char wlist1[] = "\n\n\n\n\n\
; mode:\n\…
char wlist2[] = "\n\n\
;max:\n\
;Calculati
// 检查新建文件或注释不对时，重新写注释到文件
if( (stream = fopen( "set.ini", "r+t" )) != NULL )
{
        numread = fread( list, sizeof(char), 10240, stream );
        if (!strstr(list,wlist1))
        {
                wlist1[strlen(wlist1)]='\n';
                numwritten = fwrite( wlist1, sizeof( char ), sizeof(wlist1), stream );

        }
        if (!strstr(list,wlist2))
        {
                wlist2[strlen(wlist2)]='\n';
                numwritten = fwrite( wlist2, sizeof( char ), sizeof(wlist2), stream );

        }
}
fclose( stream );
…}
```

维护修改说明或日志记录

一,维护需要添加的解码代码说明

查找类于解码O，L，H，G，C 的流程代码，然后仿之添加即可。
1, 修改函数BOOL CUnpacker_TTFrame::InputData(BYTE* pBuf, int bufSize) 中加处理新解码帧流程.
2,添加处理函数void CUnpacker_TTFrame::HandleFrame_XXX(BYTE * buff,WORD Len)为上流程等所用.
3,类似于添加STL容器缓存解码后的帧数据 如:CListFrame_xxx mCListFrame_xxx 为上流程等所用.
4,类似于添加 typedef CList<_Frame_xxx, _Frame_xxx&> CListFrame_xxx; 为上流程等所用.
5,类似于添加 struct _Frame_xxx ： _tagTTDataFrame // wendy add 2020.06.04 为上流程等所用.
6,类似于添加写记录文件void CUnpacker_TTFrame::HandleFrame_write_xxx(BYTE * buff,WORD Len) .
7.修改函数int CShowData::DirectWrite(_tagTTDataFrame * pFrame) 增加处理写新解码帧数据到文件.
8.搜索类似于mCListFrame_Simple 添加 处理mCListFrame_Xxx. 其中可能涉及到修改以下文件函数
MsEdit_WendyDlg.cpp
void CMsEdit_WendyDlg::OnRead()
BOOL CMsEdit_WendyDlg::DestroyWindow()
ShowData.cpp
BOOL CShowData::OnInitDialog()
void CShowData::OnWriteSearch()　　// wendy add 2020.06.04
int CShowData::OnTimer_Frame_To_List(void)　　// wendy add 2020.06.04
Unpacker_TTFrame.cpp

二,其它维护

1, 函数BOOL CUnpacker_TTFrame::InputData(BYTE* pBuf, int bufSize) 中加处理流程 看是否独立出来，方便以后的维护修改等.