void CSttTestServerBase::OnTimer()

{

//只检测具有单机测试权限的测试端心跳超时

CheckHeartbeatOverTime();

CSttTestBase \*pCurrTest = CSttServerTestCtrlCntr::GetCurrTest();

if(pCurrTest==NULL)

return;

【此处应该修改为，任何时候，都调用pCurrTest->OnTimer(bFinish)，然后根据bFinish做出处理。如果某些对象不需要做OnTimer处理，就不要在测试功能对象类中实现此函数】

if(pCurrTest->m\_strID==\_T("SttRecord"))

{

BOOL bFinish=FALSE;

pCurrTest->OnTimer(bFinish);

return;

}

【下面的逻辑，设计为：

第一步：调用pCurrTest的函数NeedReporting，判断是否需要上报

第二步：如果需要上报，调用pCurrTest的函数GetReportingParas(pCurrParas);

CSttParas \*pParas = oSysState.GetSttParas();

pCurrTest->GetReportingParas(pCurrParas);

这样做的好处：参数对象保存在SttSysState中，不需要单独销毁

第三步：CSttSysState，给所有的连接Socket

】

BOOL bSend=FALSE;

CSttSysState oSysState;

oSysState.UpdateSysStateHead(&m\_oSttTestCmd);

oSysState.SetReporting();

if(pCurrTest->m\_strID==\_T("SttIecDetectTest"))

{

CSttParas \*pCurrParas=pCurrTest->GetCurrParas();

if(pCurrParas==NULL)

{

}

else if(!pCurrParas->IsEmpty())

{

oSysState.AddNewChild((CExBaseObject\*)pCurrParas->Clone());

CSttSocketDataBase \*skt=m\_oSttTestCmd.GetRefSocketData();

if(skt!=NULL)

{

CLogPrint::LogString(XLOGLEVEL\_INFOR,"Send");

skt->SendSysState(&oSysState);

}

else

CLogPrint::LogString(XLOGLEVEL\_INFOR,"skt==NULL");

pCurrParas->DeleteAll();

}

}

else

{

CSttParas \*pParas = oSysState.GetSttParas();

pParas->m\_strID=pCurrTest->m\_strID;

CYCLE\_STRUCT\_EVENT\_BUF \*pEventBuf = pCurrTest->GetEventBuf();

long nMaxCount = pEventBuf->nMaxCount;

long nCurrCount = pEventBuf->nDataCount;

long nCurrIndex = pEventBuf->nCurrIndex;

long nIndex = pEventBuf->nCurrReadIndex;

long nBegin1,nEnd1,nBegin2,nEnd2;

xmem\_get\_cycle\_range\_index(nMaxCount, nCurrCount, nCurrIndex

, nIndex, nBegin1, nEnd1, nBegin2, nEnd2);

if(nBegin1<0||nEnd1<0)

return;

for (nIndex=nBegin1; nIndex <=nEnd1; nIndex++)

{

bSend=TRUE;

CDvmData \*pData = pParas->AddNewData("",0);

pData->m\_nIndex = pEventBuf->bufEvent[nIndex].nIndex;

pData->m\_strValue="";

pData->m\_dTime=pEventBuf->bufEvent[nIndex].dTime;

switch(pEventBuf->bufEvent[nIndex].nType)

{

case TestStart:

pData->m\_strID="TestStart";

break;

case TestStop:

pData->m\_strID="TestStop";

break;

case Bin:

pData->m\_strID.Format("Bin%03d",pEventBuf->bufEvent[nIndex].nChannel);

pData->m\_strValue.Format("%d", pEventBuf->bufEvent[nIndex].nValue);

break;

}

pEventBuf->nCurrReadIndex++;

}

}

CSttSocketDataBase \*skt=m\_oSttTestCmd.GetRefSocketData();

if(skt!=NULL&&bSend)

{

CLogPrint::LogString(XLOGLEVEL\_INFOR,"Send");

skt->SendSysState(&oSysState);

}

else if(bSend)

CLogPrint::LogString(XLOGLEVEL\_INFOR,"skt==NULL");

}