

IWU: Term up流程

//========================================================================

createIWUTask()

|🡪 xt\_start(isgCommonData\_g.iwuTaskId,T\_PREEMPT|T\_TSLICE,iwuCommonEntryPoint,NULL)

//接收到新消息时

| |🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| | |🡪 case ISG\_MODULE\_Q931: if(ISG\_SUCCESS==isgProcessMsgQ931(msg\_i))

| | | |🡪 case CONFIGURATION: if (ISG\_SUCCESS == isgCfgProcessMsg(msg\_i, &error))

| | | | |🡪 case ISG\_MODULE\_Q931: retVal = isgCfgProcessMsgQ931(msgP, ecodeP\_o);

| | | | | |🡪 retVal = isgCfgFsmHandler(msgP, &intrnlHdr, ecodeP\_o);

| | | | | | |🡪etVal = isgCfgTcbEventHandler[intrnlHdrP->tcbP->cfgState][intrnlHdrP->event]( msgDataP, intrnlHdrP, ecodeP\_o);

| |🡪 if (ISG\_FAILURE==isgMemFree(dataBufP))

isgCfgQ931InitTerminalRsp( msgP\_i, intrnlHdrP\_io, ecodeP\_o)

|🡪 retVal = sendReqToStackMgr(intrnlHdrP\_io->tcbP->isdnLineId,

intrnlHdrP\_io->transId, apiId, ecodeP\_o);

| |🡪 retVal = isgCfgSend(ISG\_MODULE\_STKMNGR, apiId, reqMsgP, reqMsgLen, ecodeP\_o);

| | |🡪 retVal = isgSendToMgmt(apiId, msgP, msgLen);

|🡪 retVal = isgCfgStartTimer(intrnlHdrP\_io->transId, MAX\_RSP\_TIME,SM\_IWU\_CONFIG\_P2P\_REQ\_TIMER\_EXP, &timerId, ecodeP\_o);

//接收到新消息================================================

| |🡪 isgProcessMsgStackMgr(msgBuf);

| | |🡪 retVal = isgCfgProcessMsg(rspMsgP, &error);

| | | |🡪 case ISG\_MODULE\_STKMNGR: retVal = isgCfgHandlePerIsdnLineReq(msgP, &buffered, ecodeP\_o);

| | | |🡪 retVal = isgCfgProcessMsgStackMgr(msgP, ecodeP\_o);

| | | | |🡪 retVal = getIptkLineidfromIsdnLineid(cfgGlobalData.isdnLineId, &cfgGlobalData.iptkLineId, ecodeP\_o);

| | | | |🡪 retVal = isgHashMapGetElement(

cfgGlobalData.smTransIdHashMapId, ISG\_HASH\_REM\_ELEMENT, smMsgP->interfaceId, (IsgVoidP \*) &transIdP);

| | | | |🡪isgMemFree(transIdP);

| | | | |🡪retVal = isgCfgFsmHandler(msgP, &intrnlHdr, ecodeP\_o);

| | | | | |🡪retVal = isgCfgGetTcb(intrnlHdrP, ecodeP\_o);

| | | | | |🡪retVal = isgCfgTcbEventHandler[intrnlHdrP->tcbP->cfgState][intrnlHdrP->event]( msgDataP, intrnlHdrP, ecodeP\_o);

| | |🡪 isgMemFree(rspMsgP);

isgCfgSmConfigP2pConfirm(msgP\_i, intrnlHdrP\_io, ecodeP\_o)

|🡪 retVal = isgStopTimer(intrnlHdrP\_io->tcbP->timerId);

| |🡪 isgMemFree(buffP);

|🡪 retVal = isgIrmGetLineInfo(intrnlHdrP\_io->tcbP->isdnLineId,&lineData, ecodeP\_o);

|🡪 retVal = sendReqToStackMgr(intrnlHdrP\_io->tcbP->isdnLineId,

intrnlHdrP\_io->transId, apiId, ecodeP\_o);

| |🡪 hashValP = (IsgVoid \*) isgMemGet(sizeof(IsgU32bit));

| |🡪 retVal = isgCfgSend(ISG\_MODULE\_STKMNGR, apiId, reqMsgP, reqMsgLen, ecodeP\_o);

| | |🡪 case ISG\_MODULE\_STKMNGR: retVal = isgSendToISDNStackMngr(apiId, msgP, msgLen);

| | | |🡪 if(0 != xq\_send(isgCommonData\_g.isgStackMgrQueueId, msgBuf))

|🡪 retVal = isgCfgStartTimer(intrnlHdrP\_io->transId, MAX\_RSP\_TIME, SM\_IWU\_CONFIG\_PERCALL\_REQ\_TIMER\_EXP, &timerId, ecodeP\_o);

| |🡪 retVal = isgStartTimer(tmrExp, duration, timerIdP);  
|🡪

//接收到新消息================================================

| |🡪 isgProcessMsgStackMgr(msgBuf);

| | |🡪 retVal = isgCfgProcessMsg(rspMsgP, &error);

| | | |🡪 case ISG\_MODULE\_STKMNGR: retVal = isgCfgHandlePerIsdnLineReq(msgP, &buffered, ecodeP\_o);

| | | |🡪 retVal = isgCfgProcessMsgStackMgr(msgP, ecodeP\_o);

| | | | |🡪 retVal = getIptkLineidfromIsdnLineid(cfgGlobalData.isdnLineId, &cfgGlobalData.iptkLineId, ecodeP\_o);

| | | | |🡪 retVal = isgHashMapGetElement(

cfgGlobalData.smTransIdHashMapId, ISG\_HASH\_REM\_ELEMENT, smMsgP->interfaceId, (IsgVoidP \*) &transIdP);

| | | | |🡪isgMemFree(transIdP);

| | | | |🡪retVal = isgCfgFsmHandler(msgP, &intrnlHdr, ecodeP\_o);

| | | | | |🡪retVal = isgCfgGetTcb(intrnlHdrP, ecodeP\_o);

| | | | | |🡪retVal = isgCfgTcbEventHandler[intrnlHdrP->tcbP->cfgState][intrnlHdrP->event]( msgDataP, intrnlHdrP, ecodeP\_o);

| | |🡪 isgMemFree(rspMsgP);

isgCfgSmConfigPermanentConfirm(msgP\_i, intrnlHdrP\_io, ecodeP\_o)

|🡪 retVal = isgStopTimer(intrnlHdrP\_io->tcbP->timerId);

| |🡪 isgMemFree(buffP);

|🡪 retVal = sendReqToStackMgr(intrnlHdrP\_io->tcbP->isdnLineId,

intrnlHdrP\_io->transId, apiId, ecodeP\_o);

| |🡪 hashValP = (IsgVoid \*) isgMemGet(sizeof(IsgU32bit));

| |🡪 retVal = isgCfgSend(ISG\_MODULE\_STKMNGR, apiId, reqMsgP, reqMsgLen, ecodeP\_o);

| | |🡪 case ISG\_MODULE\_STKMNGR: retVal = isgSendToISDNStackMngr(apiId, msgP, msgLen);

| | | |🡪 if(0 != xq\_send(isgCommonData\_g.isgStackMgrQueueId, msgBuf))

|🡪 retVal = isgCfgStartTimer(intrnlHdrP\_io->transId, MAX\_RSP\_TIME, SM\_IWU\_CONFIG\_PERCALL\_REQ\_TIMER\_EXP, &timerId, ecodeP\_o);

| |🡪 retVal = isgStartTimer(tmrExp, duration, timerIdP);  
|🡪

//接收到新消息================================================

………

isgCfgSmUnblockConfirm(msgP\_i, intrnlHdrP\_io, ecodeP\_o)

|🡪 retVal = isgStopTimer(intrnlHdrP\_io->tcbP->timerId);

| |🡪 isgMemFree(buffP);

|🡪retVal = sendEnableSubsDataRspToMgmt(ISG\_TRUE, intrnlHdrP\_io->tcbP->isdnLineId, intrnlHdrP\_io->transId, ecodeP\_o);

| |🡪respMsgP = isgMemGet(respMsgLen);

| |🡪retVal = isgCfgSend(ISG\_MODULE\_MGMT, MSG\_ENABLE\_SUBSCRIBER\_RSP, respMsgP, respMsgLen, ecodeP\_o);

| | |🡪case ISG\_MODULE\_MGMT: retVal = isgCfgGetIsdnLineIdFromMgmtMsg(msgP, &cfgGlobalData.processedIsdnLineId, ecodeP\_o);

| | |🡪retVal = isgSendToMgmt(apiId, msgP, msgLen);

|🡪retVal = cfgDelTcb(intrnlHdrP\_io->transId, ecodeP\_o);

| |🡪retVal = isgHashMapGetElement(cfgGlobalData.tcbHashMapId, ISG\_HASH\_REM\_ELEMENT, txnId,(IsgVoidP \*) &tcbP);

| |🡪retVal = isgMemFree(tcbP->msgP);

| |🡪retVal = isgMemFree(tcbP->msgP);

| |🡪retVal = isgCfgProcessBufferedMsgForIsdnLineId(cfgGlobalData.processedIsdnLineId, ecodeP\_o);

| | |🡪 retVal = isgCfgFetchElementFromList(&(cfgGlobalData.perLineBufferData[isdnLineId].msgList), &buffMsgP, ecodeP\_o);

|🡪

//接收到新消息================================================

| |🡪 isgProcessMsgStackMgr(msgBuf);

| | |🡪 retVal = isgCfgProcessMsg(rspMsgP, &error);

| | | |🡪 case ISG\_MODULE\_STKMNGR: retVal = isgCfgHandlePerIsdnLineReq(msgP, &buffered, ecodeP\_o);

| | | |🡪 retVal = isgCfgProcessMsgStackMgr(msgP, ecodeP\_o);

| | | | |🡪 retVal = getIptkLineidfromIsdnLineid(cfgGlobalData.isdnLineId, &cfgGlobalData.iptkLineId, ecodeP\_o);

| | | | |🡪 retVal = isgCfgFsmHandler(msgP, &intrnlHdr, ecodeP\_o);

| | | | | |🡪retVal = cfgCreateTcb(intrnlHdrP->transId, &(intrnlHdrP->tcbP), ecodeP\_o);

| | | | | | |🡪\*tcbP = (IsgCfgTcbSt \*) isgMemGet(sizeof(IsgCfgTcbSt));

| | | | | |🡪 retVal = isgCfgTcbEventHandler[intrnlHdrP->tcbP->cfgState][intrnlHdrP->event]( msgDataP, intrnlHdrP, ecodeP\_o);

isgCfgSmL1ActiveInd(msgP\_i, intrnlHdrP\_io, ecodeP\_o)

|🡪retVal = getIptkLineidfromIsdnLineid(intrnlHdrP\_io->tcbP->isdnLineId, &intrnlHdrP\_io->tcbP->iptkLineId, ecodeP\_o);

|🡪retVal = isgIrmGetLineInfo(intrnlHdrP\_io->tcbP->isdnLineId,&lineData, ecodeP\_o);

|🡪retVal = getIsgIwuLineConfigData(&lineconfigdataP, intrnlHdrP\_io->tcbP->isdnLineId, ISDN\_LINE\_ID, ecodeP\_o);

|🡪retVal = isgIrmSetLapdStatus(intrnlHdrP\_io->tcbP->isdnLineId, ISG\_IRM\_L1\_ACTIVE, ecodeP\_o);

|🡪retVal = isgIrmSetOprStatus(intrnlHdrP\_io->tcbP->isdnLineId, IRM\_OPR\_UP, ecodeP\_o);

|🡪retVal=isgCfgReportIsdnLineInfoIndToMgmt(intrnlHdrP\_io->tcbP->isdnLineId, IFS\_UP, ecodeP\_o);

| |🡪retVal = isgCfgSend(ISG\_MODULE\_MGMT,

MSG\_REPORT\_ISDN\_LINE\_INFO\_IND, respMsgP, respMsgLen, ecodeP\_o);

| | |🡪case ISG\_MODULE\_MGMT: retVal = isgSendToMgmt(apiId, msgP, msgLen);

|🡪retVal = isgCfgSendRegisterReqToIptk(intrnlHdrP\_io->tcbP->iptkLineId, intrnlHdrP\_io->transId, ecodeP\_o);

| |🡪retVal = isgCfgSend(ISG\_MODULE\_IPTK, ICF\_REGISTER\_REQ, reqMsgP, reqMsgLen, ecodeP\_o);

| | |🡪retVal = isgCfgSetHashMapForIptkTransId(msgP, ecodeP\_o);

| | |🡪retVal = isgSendToIPTK(apiId, msgP, msgLen);

|🡪retVal = isgCfgStartTimer(intrnlHdrP\_io->transId, MAX\_RSP\_TIME, ICF\_REGISTER\_REQ\_TIMER\_EXP, &timerId, ecodeP\_o);

| |🡪retVal = isgStartTimer(tmrExp, duration, timerIdP);

|🡪

//接收到新消息================================================

| |🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| | |🡪 case ISG\_MODULE\_IPTK: ISG\_SUCCESS==isgProcessMsgIPTK(msg\_i)

| | | |🡪 if (ISG\_FAILURE == isgCommonDecodeIPTKmsg(msg\_i, &decodedMsgP))

| | | |🡪 if (ISG\_SUCCESS == isgCfgProcessMsg(internalMsg, &error))

| | | | |🡪case ISG\_MODULE\_IPTK: retVal = isgCfgProcessMsgIPTK(msgP, ecodeP\_o);

| | | | | |🡪retVal = isgGetKeyFromIptkMsg(icfMsgP, &key, ecodeP\_o);

| | | | | |🡪retVal = isgCfgFsmHandler(msgP,&intrnlHdr,ecodeP\_o);

| | | | | | |🡪retVal = isgCfgGetTcb(intrnlHdrP, ecodeP\_o);

| | | | | | |🡪retVal = isgCfgTcbEventHandler[intrnlHdrP->tcbP->cfgState][intrnlHdrP->event] msgDataP, intrnlHdrP, ecodeP\_o);

isgCfgIptkRegisterCfm(msgP\_i, intrnlHdrP\_io, ecodeP\_o)

|🡪 retVal = isgStopTimer(intrnlHdrP\_io->tcbP->timerId);

|🡪 case MSG\_ENABLE\_SUBSCRIBER\_REQ: retVal = isgCfgSendSubsInfoIndToMgmt(

intrnlHdrP\_io->tcbP->isdnLineId,MG\_TERM\_STATE\_INSERVICE,sipcch\_termstate\_normal, ecodeP\_o);

| |🡪 retVal = isgCfgSend(ISG\_MODULE\_MGMT, MSG\_REPORT\_SUBSCRIBER\_INFO\_IND, respMsgP, respMsgLen, ecodeP\_o);

| | |🡪 case ISG\_MODULE\_MGMT: retVal = isgSendToMgmt(apiId, msgP, msgLen);

|🡪 retVal = sendAlarmReportIndToMgmt(2, 0, intrnlHdrP\_io->tcbP->isdnLineId, ecodeP\_o);

| |🡪 retVal = isgCfgSend(ISG\_MODULE\_MGMT, MSG\_REPORT\_ALARM\_IND, respMsgP, respMsgLen, ecodeP\_o);

| | |🡪 case ISG\_MODULE\_MGMT: retVal = isgSendToMgmt(apiId, msgP, msgLen);

|🡪 retVal = setIsgIwuLineRegStatus(REGISTERED, IPTK\_LINE\_ID, intrnlHdrP\_io->tcbP->iptkLineId, ecodeP\_o);

|🡪 retVal = cfgDelTcb(intrnlHdrP\_io->transId, ecodeP\_o);

|🡪