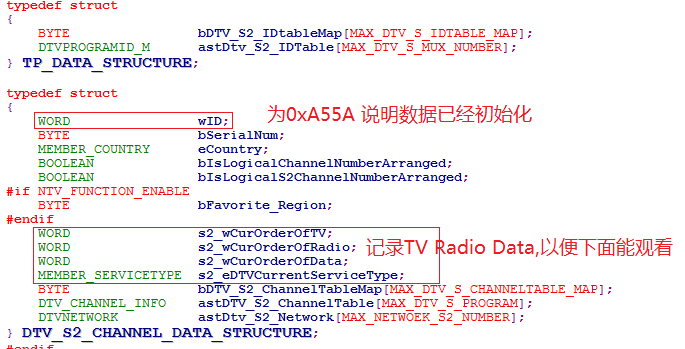
1. **有两块很重要的数据模块。**

A.**DTVPROGRAMID\_M**

B. **DTV\_CHANNEL\_INFO**



查看软件中的msAPI\_CM\_InitDTVDataManager函数中的CreatDTVProgramIndexTableAndProgramIDTable我们就知道，通过上面数据会创建下面重要数据TABLE.

1. **从数据表中初始化下面Table。**

A. 通过下面函数创建（**m\_acDVB\_S\_ProgramTableMap**）。

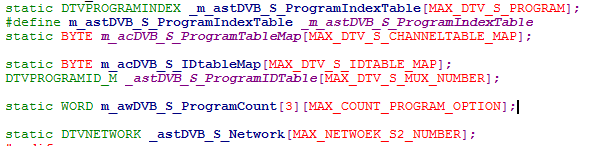


B.通过下面函数创建（**\_m\_astDVB\_S\_ProgramIndexTable**）。



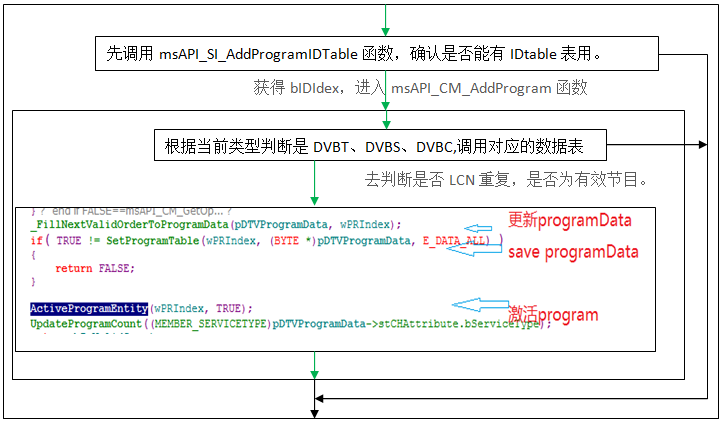
C.通过下面函数创建（**m\_acDVB\_S\_IdtableMap \_astDVB\_S\_ProgramIDTable**）。

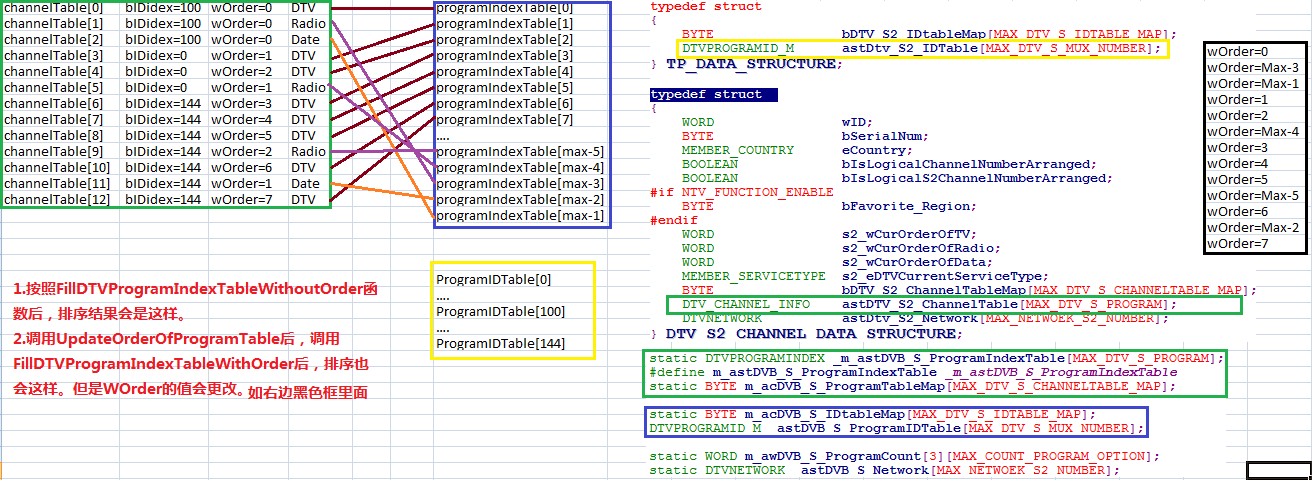




1. **我们先来看下添加节目函数msAPI\_SI\_AddProgram()。**

****

****

****

1. **我们再一起看下删除节目函数**msAPI\_CM\_DeleteProgram()**。**



注意：如果删除当前节目，如何处理。

1. **MOVE节目时，那些数据需要处理。**

主要是channelTable数据交换，还有就是channelTable里面的WOrder 内容交换。

1. **再看初始化节目函数msAPI\_CM\_InitDTVDataManager()。**



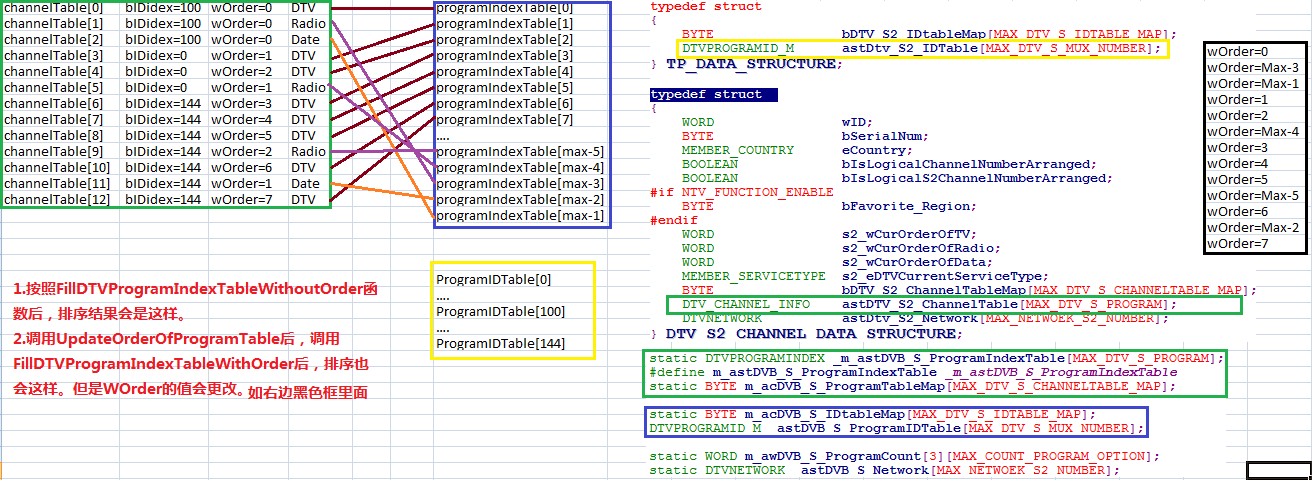
注意：初始化节目时，函数FillDTVProgramIndexTableWithOrder与FillDTVProgramIndexTableWithoutOrder的区别。

1. **下面我们就来看节目排序。**

可以把DTV\_CHANNEL\_INFO中的wOrder按照FillDTVProgramIndexTableWithoutOrder函数排列赋值。

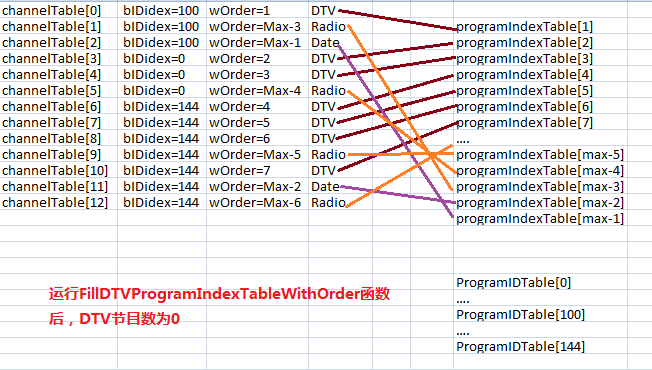
FillDTVProgramIndexTableWithOrder FillDTVProgramIndexTableWithoutOrder

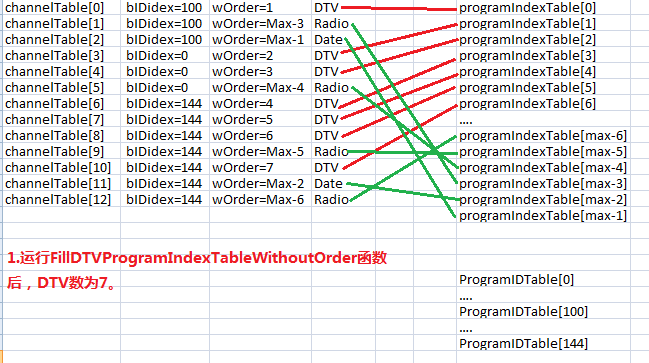
如果DTV\_CHANNEL\_INFO有下面这些数据，channelist分别调用上面函数后，显示会如何。



1. 看下下面函数结果会如何。我们分析下DTV CountProgram函数.为什么会出现这样的情况。可以通过下面的debug1(void)函数打印。

WORD CountProgram(MEMBER\_SERVICETYPE bServiceType, COUNT\_PROGRAM\_OPTION eCountOption)





B．下面我们再来看下显示的台号，通过下面的函数，我们知道其实就是显示DTV的LCN号。可以通过下面的void debug2(void)函数打印

msAPI\_CM\_GetLogicalChannelNumber((MEMBER\_SERVICETYPE)SERVICE\_TYPE\_RADIO,g\_wProgramPosition)

C.我们再来看下节目交换功能。可以通过下面的void debug3(void)函数打印

D．打印节目函数.可以通过下面的void debug4(void)函数打印

void debug1(void)

{

DTVPROGRAMINDEX \*pProgramIndexTable=m\_astDVB\_T\_ProgramIndexTable;

#if DVB\_T\_C\_DIFF\_DB

if (msApi\_GetDVBType() ==EN\_DVB\_C\_TYPE )

{

pProgramIndexTable=m\_astDVB\_C\_ProgramIndexTable;

}

#endif

#if ENABLE\_S2

if (msApi\_GetDVBType() ==EN\_DVB\_S2\_TYPE )

{

pProgramIndexTable=m\_astDVB\_S\_ProgramIndexTable;

}

#endif

FillProgramIndexWithDefault(&pProgramIndexTable[0]);

printf("ProgramCount(%d)\n",GetProgramCount(E\_SERVICETYPE\_DTV,INCLUDE\_ALL));

UpdateProgramCount(E\_SERVICETYPE\_DTV);

printf("ProgramCount(%d)\n",GetProgramCount(E\_SERVICETYPE\_DTV,INCLUDE\_ALL));

}

void debug2(void)

{

WORD wPosition;

WORD wOrder;

WORD wSimu\_LCN=0;

WORD wMaxDtvProgram= MAX\_DTVPROGRAM;

DTVPROGRAMINDEX \*pProgramIndexTable=m\_astDVB\_T\_ProgramIndexTable;

#if DVB\_T\_C\_DIFF\_DB

if (msApi\_GetDVBType() ==EN\_DVB\_C\_TYPE )

{

wMaxDtvProgram= MAX\_DTV\_C\_PROGRAM;

pProgramIndexTable=m\_astDVB\_C\_ProgramIndexTable;

}

#endif

#if ENABLE\_S2

if (msApi\_GetDVBType() ==EN\_DVB\_S2\_TYPE )

{

wMaxDtvProgram= MAX\_DTV\_S\_PROGRAM;

pProgramIndexTable=m\_astDVB\_S\_ProgramIndexTable;

}

#endif

for(wPosition=0; wPosition < wMaxDtvProgram; wPosition++)

{

wOrder = ConvertPositionToOrder(E\_SERVICETYPE\_DTV, wPosition);

if( E\_SERVICETYPE\_DTV == pProgramIndexTable[wOrder].bServiceType )

{

wSimu\_LCN=pProgramIndexTable[wOrder].wSimu\_LCN\*2;

pProgramIndexTable[wOrder].wSimu\_LCN=wSimu\_LCN;

pProgramIndexTable[wOrder].wLCN=wSimu\_LCN;

if(IsProgramEntityActive(pProgramIndexTable[wOrder].wPRIndex))

{

SetProgramTable(pProgramIndexTable[wOrder].wPRIndex, (BYTE \*)&wSimu\_LCN, E\_DATA\_LCN);

SetProgramTable(pProgramIndexTable[wOrder].wPRIndex, (BYTE \*)&wSimu\_LCN, E\_DATA\_SIMU\_LCN);

}

}

else

{

break;

}

}

}

void debug3(void)

{

DTVPROGRAMINDEX \*pProgramIndexTable=m\_astDVB\_T\_ProgramIndexTable;

#if DVB\_T\_C\_DIFF\_DB

if (msApi\_GetDVBType() ==EN\_DVB\_C\_TYPE )

{

pProgramIndexTable=m\_astDVB\_C\_ProgramIndexTable;

}

#endif

#if ENABLE\_S2

if (msApi\_GetDVBType() ==EN\_DVB\_S2\_TYPE )

{

pProgramIndexTable=m\_astDVB\_S\_ProgramIndexTable;

}

#endif

DTVPROGRAMINDEX ProgramIndexTableBack;

memcpy(&ProgramIndexTableBack,&pProgramIndexTable[2],sizeof(DTVPROGRAMINDEX));

memcpy(&pProgramIndexTable[2],&pProgramIndexTable[3],sizeof(DTVPROGRAMINDEX));

memcpy(&pProgramIndexTable[3],&ProgramIndexTableBack,sizeof(DTVPROGRAMINDEX));

}

void debug4(void)

{

BYTE bChannelName[MAX\_SERVICE\_NAME];

WORD wOrder,wOrder2;

WORD wSimu\_LCN;

WORD wMaxDtvProgram= MAX\_DTVPROGRAM;

CHANNEL\_ATTRIBUTE stCHAttribute;

DTVPROGRAMINDEX \*pProgramIndexTable=m\_astDVB\_T\_ProgramIndexTable;

#if DVB\_T\_C\_DIFF\_DB

if (msApi\_GetDVBType() ==EN\_DVB\_C\_TYPE )

{

wMaxDtvProgram= MAX\_DTV\_C\_PROGRAM;

pProgramIndexTable=m\_astDVB\_C\_ProgramIndexTable;

}

#endif

#if ENABLE\_S2

if (msApi\_GetDVBType() ==EN\_DVB\_S2\_TYPE )

{

wMaxDtvProgram= MAX\_DTV\_S\_PROGRAM;

pProgramIndexTable=m\_astDVB\_S\_ProgramIndexTable;

}

#endif

printf("ProgramIndexTable\n");

for(wOrder=0; wOrder < wMaxDtvProgram; wOrder++)

{

if(IsProgramEntityActive(pProgramIndexTable[wOrder].wPRIndex))

{

memset(bChannelName, 0x00, MAX\_SERVICE\_NAME);

GetProgramTable(pProgramIndexTable[wOrder].wPRIndex, (BYTE \*)bChannelName, E\_DATA\_SERVICE\_NAME);

GetProgramTable(pProgramIndexTable[wOrder].wPRIndex, (BYTE \*)&wOrder2, E\_DATA\_ORDER);

GetProgramTable(pProgramIndexTable[wOrder].wPRIndex, (BYTE \*)&wSimu\_LCN, E\_DATA\_SIMU\_LCN);

GetProgramTable(pProgramIndexTable[wOrder].wPRIndex, (BYTE \*)&stCHAttribute, E\_DATA\_MISC);

if(E\_SERVICETYPE\_DTV==stCHAttribute.bServiceType)

{

printf("DTV wOrder =%4d.%4d,wPRIndex=%4d,Lcn=%4d.%4d [%s]\n",wOrder,wOrder2,pProgramIndexTable[wOrder].wPRIndex,pProgramIndexTable[wOrder].wSimu\_LCN,wSimu\_LCN,bChannelName);

}

if(E\_SERVICETYPE\_RADIO==stCHAttribute.bServiceType)

{

printf("RADIO wOrder =%4d.%4d,wPRIndex=%4d,Lcn=%4d.%4d [%s]\n",wOrder,wOrder2,pProgramIndexTable[wOrder].wPRIndex,pProgramIndexTable[wOrder].wSimu\_LCN,wSimu\_LCN,bChannelName);

}

if(E\_SERVICETYPE\_DATA==stCHAttribute.bServiceType)

{

printf("DATA wOrder =%4d.%4d,wPRIndex=%4d,Lcn=%4d.%4d [%s]\n",wOrder,wOrder2,pProgramIndexTable[wOrder].wPRIndex,pProgramIndexTable[wOrder].wSimu\_LCN,wSimu\_LCN,bChannelName);

}

}

}

}