

# EPG、刻录时移功能

## 目录

一 简要说明.....	3
二 EPG 功能.....	4
2.1 EPG 数据结构.....	4
2.2 获取 EPG 数据.....	5
2.3 预约事件.....	6
2.3.1 预约刻录.....	6
2.3.2 预约提醒.....	7
2.3.3 处理预约事件.....	8
三 一键刻录、时移.....	12
3.1 一键刻录、时移启动流程.....	12
3.2 选择 U 盘.....	15
3.3 SN 上报数据给 Android.....	16
四 EPG 相关问题.....	18

# 一 简要说明

- 1、**EPG**：EPG（Electronic Program Guide）菜单里面会从码流里面抓取当前的时间的样机未来一段时间的节目预告和时间安排，主要是方便用户快捷地找到自己关心的节目和查看节目的附加信息。
- 2、**一键刻录**：在 DTV 通道下且当前节目有信号，操作遥控器上面的 record 键就会自动进入刻录，将当前的播放内容刻录到 U 盘；处于刻录状态时我们可以对刻录的文件进行快进、快退、播放、暂停、停止等操作，当停止刻录之后 U 盘会保留刚才录制的文件内容。
- 3、**一键时移**：这个功能和一键刻录差不多，主要区别在于①进入一键时移时对画面做了 freeze 所以画面是静止的，操作播放之后和刻录的现象就变成一样的了。②当停止时移之后刚才录制下来的文件会被删除。
- 4、**预约刻录**：这个功能是用用户预先设置好想要刻录的频道以及时间段，时间到了之后，会弹出提示是否需要跳转刻录，如果选择否就取消本次预约刻录，选择是就马上跳转到对应的频道，不做选择的话，时间到了，系统会自动跳转到预先设置频道进行刻录，并且刻录完用户设置的时间段之后自动退出刻录。
- 5、**待机刻录**：待机刻录就是设置好预约刻录之后关机，到达预约刻录时间了会自动开机（屏幕是关的）进行刻录，刻录完了之后会自动关机。在关屏期间的待机刻录除了 POWER 键，其他按键都不能响应，而且 POWER 键只是把屏幕打开，屏幕打开之后，就和正常的一键刻录状态一致，并且刻录完了不会自动关机。
- 6、**预约观看**：预约观看就是用户预先设置好要预约观看的节目以及时间段之后。当预约时间到来弹出提示框提示是否需要自动跳转到对应的通道进行，选择是则马上跳转到对应频道，选择否则取消当次的预约观看，不做选择则倒计时时间到了之后自动跳转。
- 7、**待机预约观看**：待机预约观看就是用户预先设置好要预约观看的节目以及时间段之后，将电视关机，当预约时间到来的时候会自动开机，然后会弹出提示框提示是否需要自动跳转到对应的通道进行，选择是则马上跳转到对应频道，选择否则取消当次的预约观看，不做选择则倒计时时间到了之后自动跳转。

## 二 EPG 功能

EPG 技术在数字电视领域中的应用，体现出新型数字电视特有的交互和多媒体功能，是现代数字电视优于传统模拟电视的一个重要体现。

### 2.1 EPG 数据结构

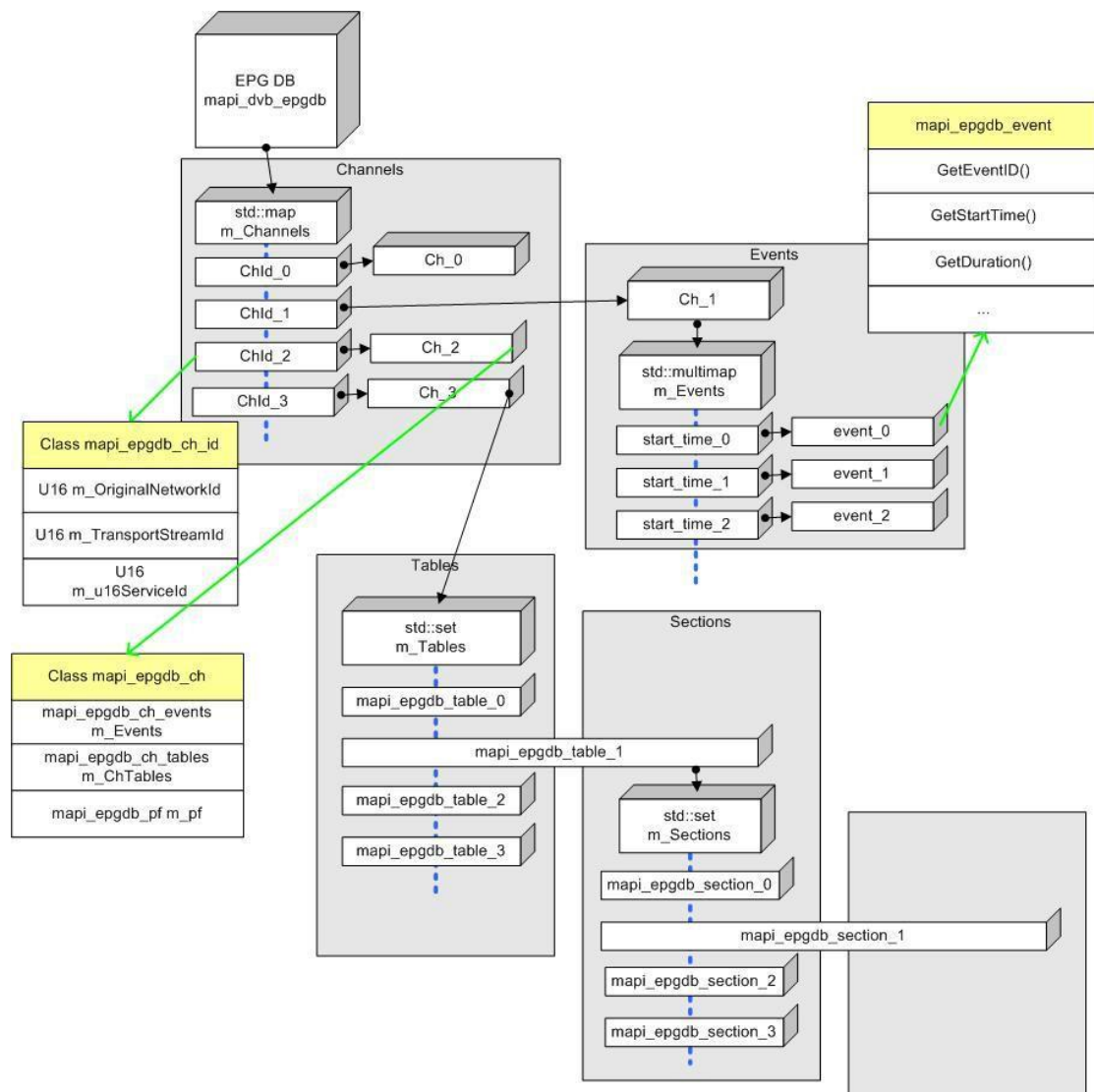
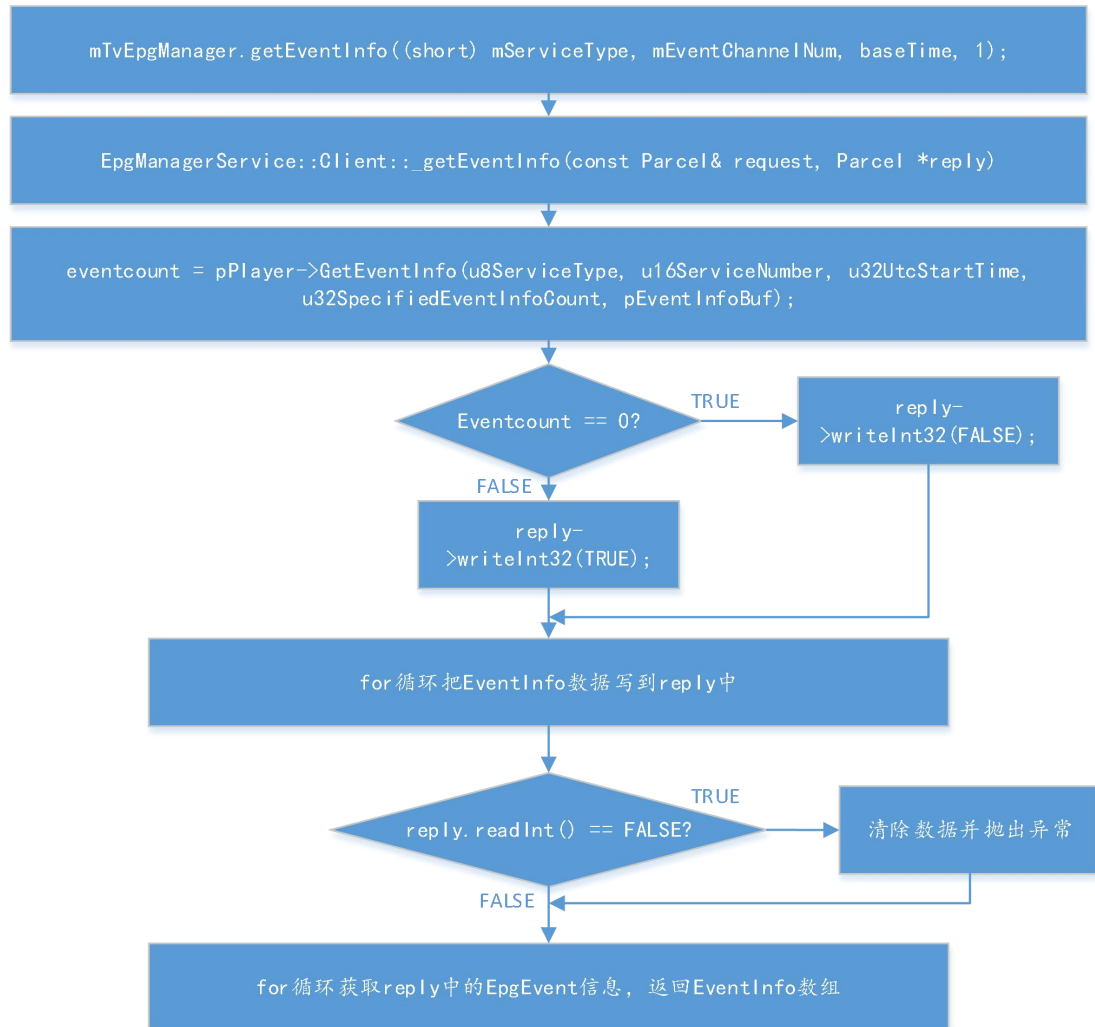


图 1-1

## 2.2 获取 EPG 数据

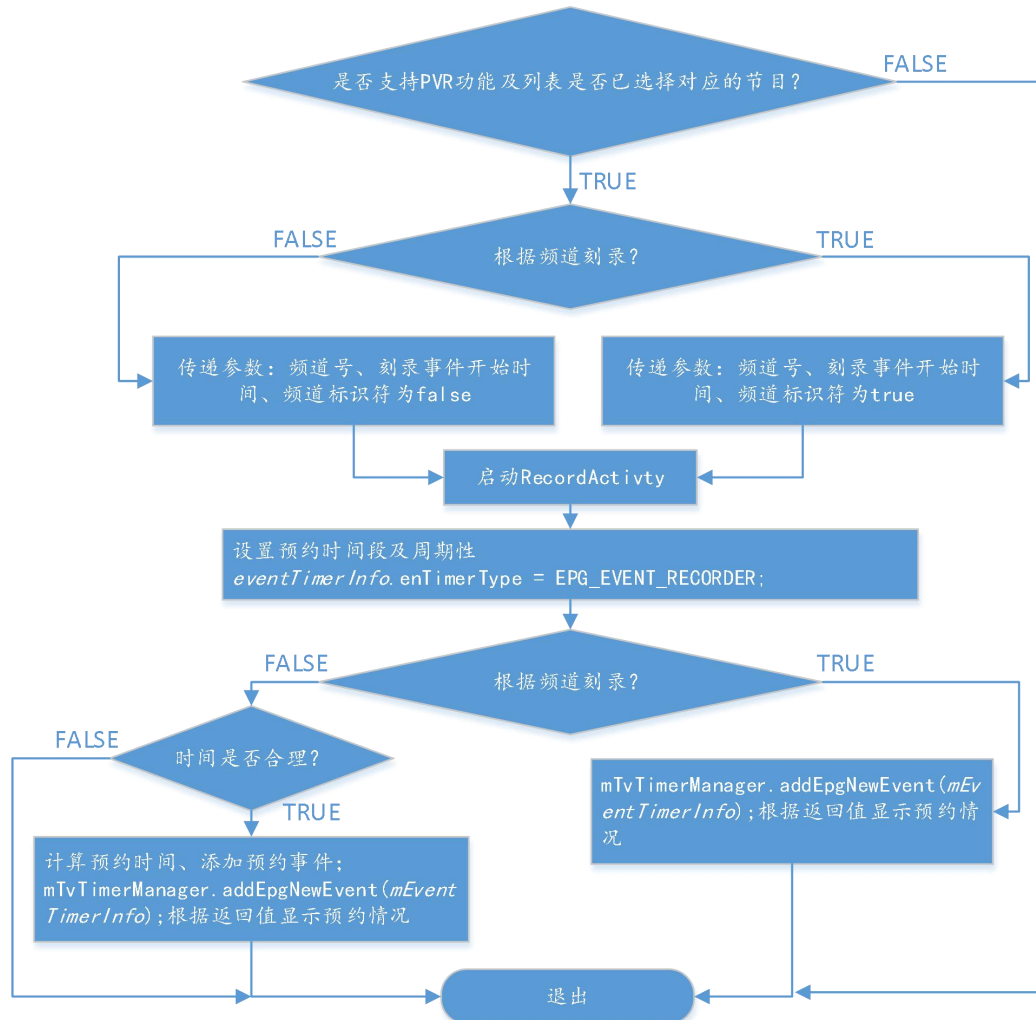
Android 层获取 EPG 信息主要流程



## 2.3 预约事件

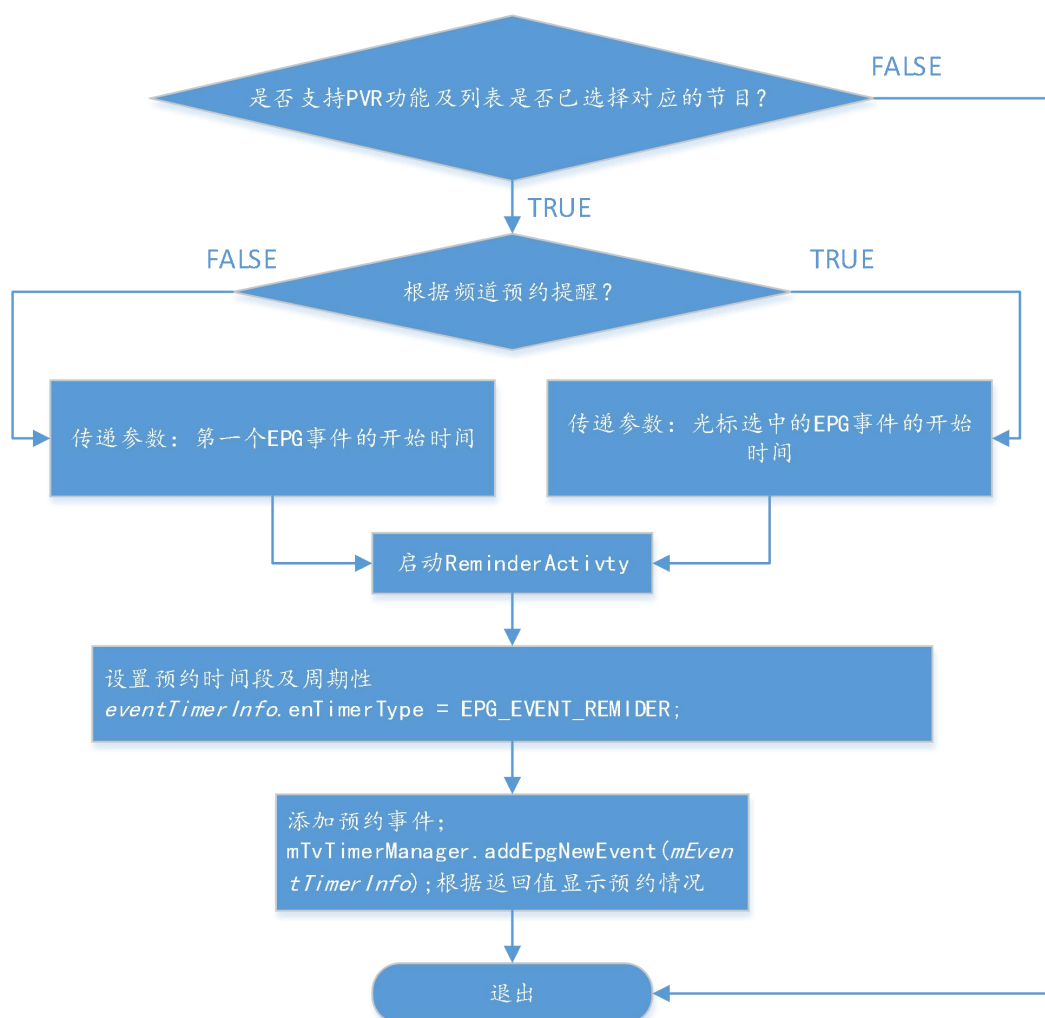
### 2.3.1 预约刻录

预约刻录整体流程如下，根据是否为预约某个节目的具体事件来传递不同的数据



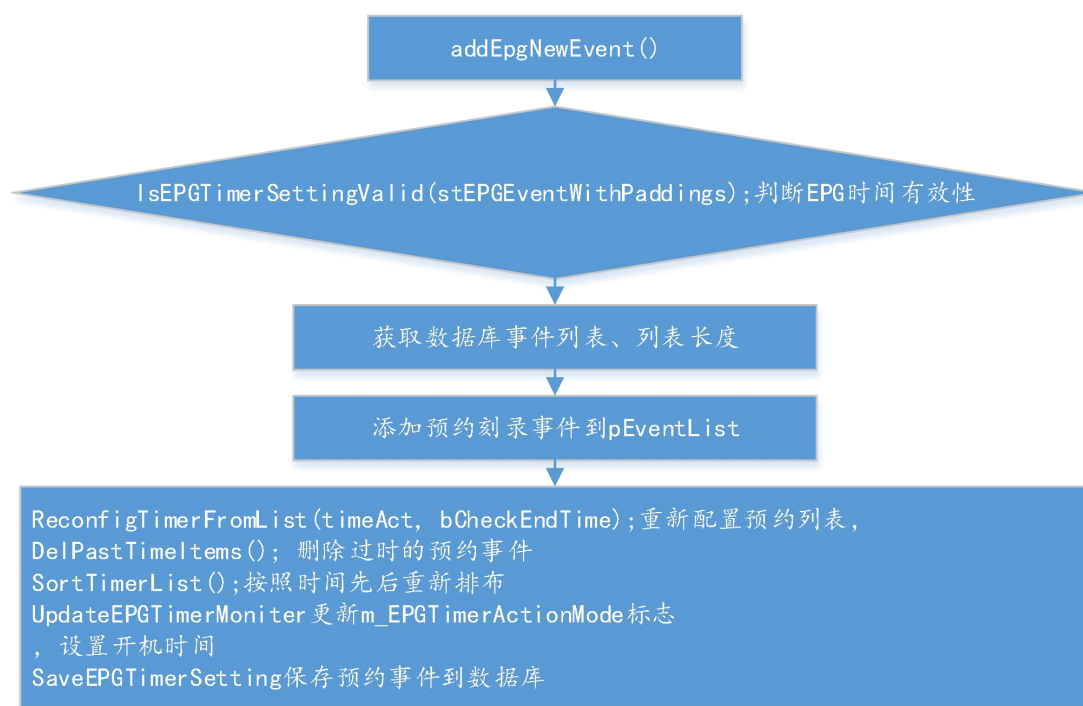
### 2.3.2 预约提醒

预约提醒流程如下：如果是预约某个节目的事件，传递选中该事件的开始时间及光标位置；如果根据时间预约，传递的数据是当前节目的第一个 EPG 事件的开始事件。



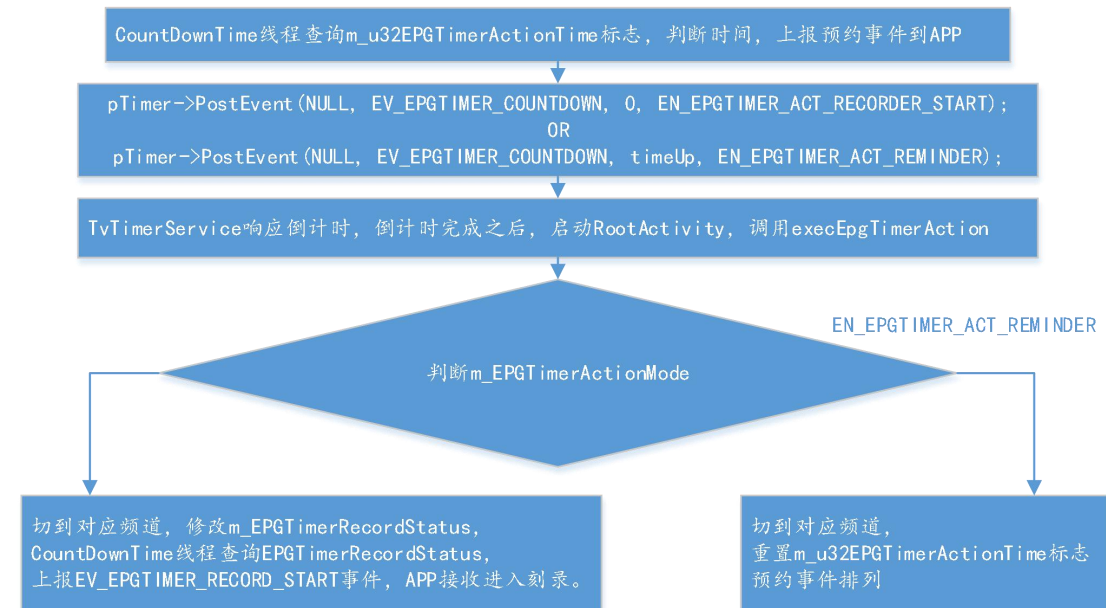
### 2.3.3 处理预约事件

添加刻录和提醒事件的方法都是调用 TvTimerManager 的 addEpgNewEvent 方法，流程如下





预约事件添加完后，由 CountdownTime()线程查询 m\_u32EPGTimerActionTime 标志，CountDownTime()线程由 MSrv\_Timer 调用 Resume()方法时初始化（PollingCurrTime：轮询当前时间的线程、MonitorOffModeTimer：监视模式变化的定时器、CountDownTime：倒计时线程），CountDownTime()流程如下



CountDownTime()上报预约事件的代码如下

```

if(pTimer->m_u32EPGTimerActionTime != INVALID_EPG_TIME)
{
    if(((pTimer->m_u32EPGTimerActionTime <= (current - EPGTIMER_CI_OP_REFRESH_LEADING_TIME_S)) && (pTimer->m_EPGTimerActionMode == EN_EPGTIMER_ACT_CI_OP_REFRESH)) || \
        ((pTimer->m_u32EPGTimerActionTime <= (current - EPGTIMER_REMINDER_LEADING_TIME_S)) && (pTimer->m_EPGTimerActionMode == EN_EPGTIMER_ACT_REMINDER)) || \
        ((pTimer->m_u32EPGTimerActionTime <= (current + EPGTIMER_RECORDER_LEADING_TIME_S)) && (pTimer->m_EPGTimerActionMode == EN_EPGTIMER_ACT_RECORDER_START)))
    {
        if(pTimer->m_EPGTimerActionMode == EN_EPGTIMER_ACT_RECORDER_START)
        {
            if(m_IsSendCountDownEvent == FALSE)
            {
                pTimer->PostEvent(NULL, EV_EPGTIMER_COUNTDOWN, 0, EN_EPGTIMER_ACT_RECORDER_START);
                m_IsSendCountDownEvent = TRUE;
            }
        }
        else
        {
            pTimer->PostEvent(NULL, EV_EPGTIMER_COUNTDOWN, 0, EN_EPGTIMER_ACT_RECORDER_START);
        }
    }
    else if(pTimer->m_EPGTimerActionMode == EN_EPGTIMER_ACT_CI_OP_REFRESH)
    {
        U32 timeUp = pTimer->m_u32EPGTimerActionTime - current;
        if((0 < timeUp) && (timeUp <= EPGTIMER_COUNTDOWN_LEADING_TIME_S))
        {
            pTimer->PostEvent(NULL, EV_EPGTIMER_COUNTDOWN, timeUp, EN_EPGTIMER_ACT_CI_OP_REFRESH);
        }
    }
    else if(pTimer->m_EPGTimerActionMode == EN_EPGTIMER_ACT_REMINDER)
    {
        U32 timeUp = pTimer->m_u32EPGTimerActionTime - current;
        if((0 < timeUp) && (timeUp <= EPGTIMER_COUNTDOWN_LEADING_TIME_S))
        {
            //If Current source is dtv and current channel is the same with channel which set up in reminder, the count down UI will be ignored
            MSrv_System_Database *pSysDB = MSrv_Control::GetInstance()->GetMSrvSystemDatabase();
            NAPI_INPUT_SOURCE_TYPE eInputSrc;
            eInputSrc = MSrv_Control::GetInstance()->GetCurrentInputSource();
            ASSERT(pSysDB);
            ST_EPG_EVENT_TIMER_INFO *pEventList = NULL;
            U32 listSize = 0;
            pSysDB->UseEpgTimerList(&pEventList, &listSize);
            //UI6 actionServiceNumber = pEventList[0].ui6ServiceNumber;
        }
    }
}

```

TvTimerService 响应倒计时弹窗如下：

```

} else {
    Log.i(TAG, "leftTimeValue > 0 show count Down page");
    if (CountDownTimerDialog.isCountDownTimerDialogShowing == false) {
        if (CountDownTimerDialog.isEPGTimerActionInvoked == true) {
            Log.i(TAG, "Bypass TvTimerManager.TVTIMER_EPG_TIMER_COUNTDOWN since EPG action be invoked directly..");
        } else {
            CountDownTimerDialog countDownTimerDialog = new CountDownTimerDialog(
                mContext, android.R.style.Theme_Panel);
            countDownTimerDialog.getWindow().setType(
                WindowManager.LayoutParams.TYPE_SYSTEM_ALERT);
            countDownTimerDialog.show();
            //add by wxy
            if(msg.arg2 == 1){
                countDownTimerDialog.setDialogTitle(countDownTimerDialog.REMINDEREVENT);
            }else {
                countDownTimerDialog.setDialogTitle(countDownTimerDialog.RECORDEREVENT);
            }
            //add end
        }
    }
}
} ? end else ?

```

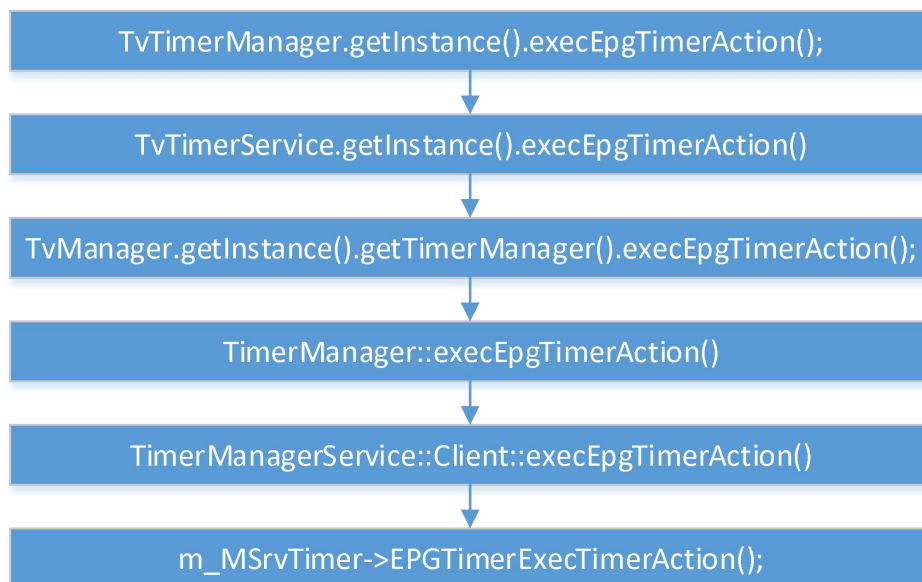
倒计时结束后，TvTimerService 启动 RootActivity 应用，并且调用 execEpgTimerAction()方法处理预约事件

```
case TvTimerManager.TVTIMER_EPG_TIMER_COUNTDOWN: {
    int leftTimeValue = msg.arg1;
    if (leftTimeValue == 0) {
        // reset countdown dialog status
        CountdownTimerDialog.isEPGTimerActionInvoked = false;

        targetIntent = new Intent("source.switch.from.storage");
        mContext.sendBroadcast(targetIntent);
        new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                TvCommonManager tvCommonManager = TvCommonManager.getInstance();
                final int currentTvInputSource = tvCommonManager.getCurrentTvInputSource();
                TvS3DManager.getInstance().setDisplayFormatForUI(
                    TvS3DManager.THREE_DIMENSIONS_DISPLAY_FORMAT_NONE);
                targetIntent = new Intent(
                    "com.mstar.tv.tvplayer.ui.intent.action.RootActivity");
                targetIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                mContext.startActivity(targetIntent);
                if (currentTvInputSource != TvCommonManager.INPUT_SOURCE_DTV) {
                    tvCommonManager.setInputSource(TvCommonManager.INPUT_SOURCE_DTV);
                }
                Log.i(TAG,
                    "leftTimeValue==0 and in current CH -> execEpgTimerAction");
                TvTimerManager.getInstance().execEpgTimerAction();
            }
        }).start();
    } ? end if leftTimeValue==0 ? else {
```

execEpgTimerAction()方法调用流程如下，最后在 MSrv\_Timer::EPGTimerExecTimerAction()这个方法中处理，remind、record 事件具体处理方法：

remind：切到对应频道，重置 m\_u32EPGTimerActionTime 标志和预约事件排列  
record: 切到对应频道，修改 m\_EPGTimerRecordStatus 值，CountDownTime 线程会一直查询 m\_EPGTimerRecordStatus 状态做相应处理。



MSrv\_Timer::EPGTimerExecTimerAction(): 以下是处理 record 事件的代码

```
        case EN_EPGTIMER_ACT_RECORDER_START:
        {
#ifdef TARGET_BUILD
        #if (MSTAR_IPC == 1)
            char appName[16];
            int ret=0;
            ret=APM_GetFocusApp(appName, sizeof(appName));
            if( (ret==0) && (strcmp(appName, "SN", strlen("SN"))) && (strcmp(appName, "CommonUI", strlen("CommonUI"))) )
            {
                APM_RequestToExit((char *)appName); //EPG_TIME_UP event exit focused AP
            }
        #endif //end of MSTAR_IPC
        #endif //end of TARGET_BUILD

        #if (ISDB_SYSTEM_ENABLE == 1)
            ST_MEDIUM_SETTING MediumSetting;
            MSrv_Control::GetMSrvSystemDatabase()->GetMediumSetting(&MediumSetting);
            if (MediumSetting.AntennaType != E_ANTENNA_TYPE_AIR)
            {
                MediumSetting.AntennaType = E_ANTENNA_TYPE_AIR;
                MSrv_Control::GetMSrvSystemDatabase()->SetMediumSetting(&MediumSetting);
            }
        #if (STB_ENABLE == 0)
            MSrv_Control::GetMSrvAtv()->SetNTSCAntenna(MSrv_ATV_Database::MEDIUM_AIR);
            MSrv_ATV_Database *pAtvDb = dynamic_cast<MSrv_ATV_Database *>(MSrv_Control::GetMSrvAtvDatabase());
            ASSERT(pAtvDb);
            pAtvDb->ConnectDatabase(MSrv_ATV_Database::DVB_T_DB);
        #endif

        #if (PVR_ENABLE == 1 && ATSC_SYSTEM_ENABLE == 0)
            // Scheduled recording has higher priority than normal record (One-Touch-Recording)
            pPlayer->PVRStopAll();
        #endif

            MSrv_Control_common::GetMSrvChannelManager()->ChangeToFirstProgram(E_FIRST_PROG_DTV, E_ON_TIME_BOOT_TYPE);
        }
        else
        {
            MSrv_Control::GetInstance()->SetInputSource(MAPI_INPUT_SOURCE_DTV);
        }
        #if (PVR_ENABLE == 1 && ATSC_SYSTEM_ENABLE == 0)
            // Scheduled recording has higher priority than normal record (One-Touch-Recording)
            pPlayer->PVRStopAll();
        #endif

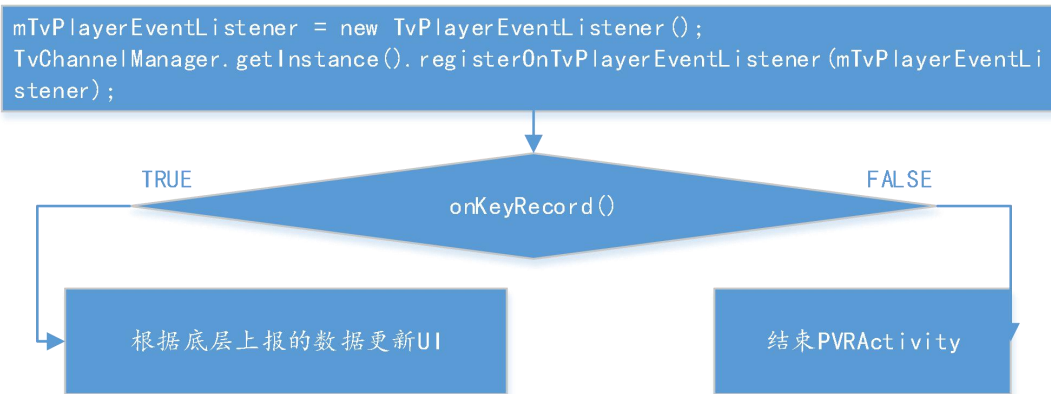
            MSrv_Control_common::GetMSrvChannelManager()->ProgramSel((U32)actionServiceNumber, actionServiceType);
        }

        U32 current = mapi_interface::Get_mapi_system()->RTCGetCLK();
        if(!((m_u32EPGTimerActionTime != INVALID_EPG_TIME) &&
            ((current+EPGTIMER_RECORDER_LEADING_TIME_S) < m_u32EPGTimerActionTime)))
        {
            m_EPGTimerRecordStatus = EN_EPGTIMER_RECORDER_TIMEUP;
        }
    }
```

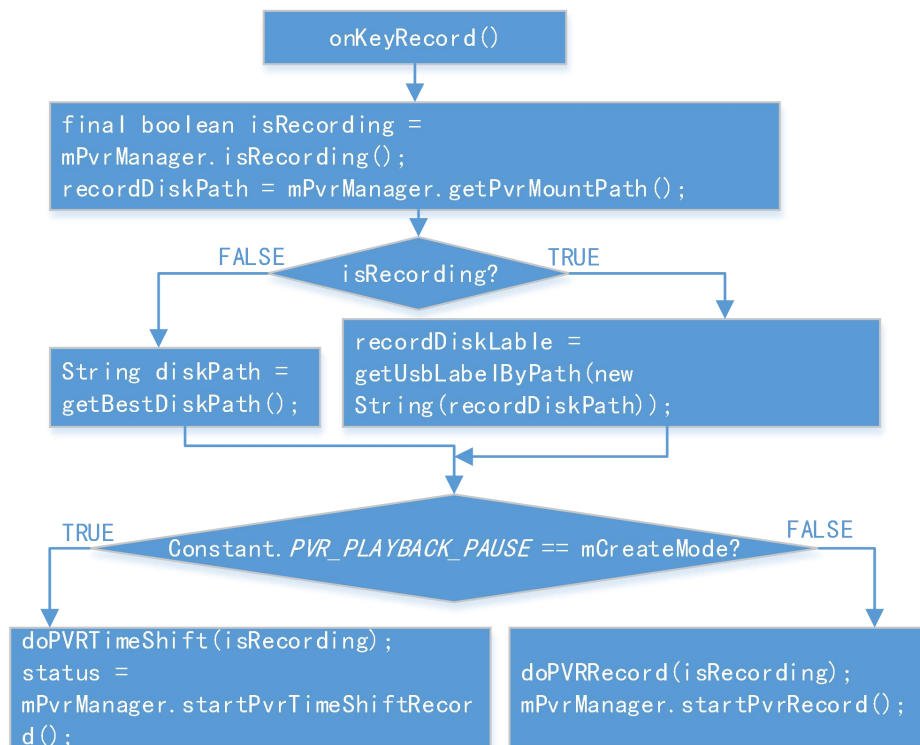
## 三 一键刻录、时移

### 3.1 一键刻录、时移启动流程

启动流程如下



执行 onKeyRecord() 流程





startPvrRecord();



startPvrTimeShiftRecord();



刻录和时移的差别：刻录结束后系统会把刻录的内容以.ts 文件保存起来；时移功能不会把刻录的文件保留在硬盘或者闪存盘等外挂设备里。也就是说，用户在完成时移操作后，是找不到任何文件的。PVR 将在时移结束后把它删除。

执行结束刻录、时移的方法最终在 PvrAdaptation::RecordStopInternal(BOOL bIsTimeshift, BOOL bIsAlwaysTimeshift, U8 u8RecordingIndex), 判断为时移功能时会执行删除文件的命令。

```
if(TRUE == bIsTimeshift)
{
    if(!bIsPreRecording)
    {
        m_PlayerDvb->m_pcAVMonitor->StartVideoFilter();
    }

    ASSERT(GetTimeShiftMwPvr() != NULL);
    if(TRUE == bIsAlwaysTimeshift)
    {
        bRet = GetTimeShiftMwPvr()->AlwaysTimeshiftRecordStop();
    }
    else
    {
        bRet = GetTimeShiftMwPvr()->TimeshiftRecordStop();
        // TimeshiftRecordStop unfreeze DTV and need reset DTV.
        if(m_PlayerDvb->m_pcdvb_t_si_Parser)
        {
            m_PlayerDvb->PlayCurrentProgram();
        }
    }

    if(GetTimeShiftMwPvr()->IsPlaybacking() != TRUE)
    {
        m_u8TimeShiftIndex = INVALID_MW_PVR_INDEX;
    }
} ? end if TRUE==bIsTimeshift ?
else
{
    GetMwPvrByIndex(u8RecordingIndex)->RecordStop();
    MSrv_Control::GetMSrvTimer()->StopEPGRecord();
}
```

## 3.2 选择 U 盘

U 盘选择目前有两种方法选择，具体如下：

- 1、主菜单里面有一个 PVR File System 的选项；点击进去会启动 PVROptionActivity，在这个 Activity 里面可以选择刻录的 U 盘，选择 AlwaysTimeShift 开关、选择时移长度、格式化 U 盘，U 盘测速。
  - 选择 U 盘：弹出选择 U 盘的 Dialog，用户选择了之后调用 saveChooseDiskSettings 保存所选择的 U 盘的信息到 SharedPreferences 中去，下次刻录或者时移的时候就会先获取这个 SharedPreferences 的值是否存在，存在就不需要选择 U 盘，直接进入下一步。
  - 时移长度选择：可以选择时移长度所占的空间。
  - AlwaysTimeShift 开关：因为我们时移的长度是固定的（512M、1G、2G、4G），所以正常情况下时移长度达到我们设置的值之后就会自动停止时移，但是如果这个开关打开的，系统就会一直时移下去，如果时移长度达到限制值自动删除前面的保留后面的。
  - 格式化 U 盘：将 U 盘格式化
  - U 盘测速：向 U 盘里面写数据测试写的速度。
- 2、刻录或者时移开始的时候选择 U 盘；在进行刻录或者时移时先判断是否在 PVROptionActivity 里面选择过、且选择的 U 盘是否还在；如果不存在先判断当前有多少个 U 盘，如果只有一个就自动选择，超过一个就弹出 Dialog 用户选择。

### 3.3 SN 上报数据给 Android

在 PVRActivity 里面 TvPlayerEventListener 实现 OnTvPlayerEventListener 然后注册。以下是一些使用到的方法：

**public boolean onPvrNotifyPlaybackTime(int what, int arg1)**

——时移的时间，arg1 表示当前时移的时间长度

```
@Override
@Override
public boolean onPvrNotifyPlaybackTime(int what, int arg1) {
    mCurrentTime = arg1;
    updateProgress();
    return true;
}
```

**public boolean onPvrNotifyPlaybackSpeedChange(int what)**

——时移/刻录的播放速度

```
@Override
public boolean onPvrNotifyPlaybackSpeedChange(int what) {
    Log.d(TAG, "onPvrNotifyPlaybackSpeedChange");
    activity.runOnUiThread(new Runnable() {

        @Override
        public void run() {
            updateSpeedIcon(mPvrManager.getPvrPlaybackSpeed());
        }
    });
    return false;
}
```

**public boolean onPvrNotifyRecordTime(int what, int arg1)**

——刻录的时间，arg1 表示当前刻录的时间长度

```
@Override
public boolean onPvrNotifyRecordTime(int what, int arg1) {
    /* When play not recording file don't update total time base on record time */
    if (false == mUpdateTotalTime) {
        return true;
    }
    mTotalTime = arg1;
    updateProgress();
    return true;
}
```

**public boolean onPvrNotifyPlaybackStop(int what)**

——时移回放结束

**public boolean onPvrNotifyParentalControl(int what, int arg1)**

——上报当前节目是否上锁



```
    @Override
public boolean onPvrNotifyParentalControl(int what, int arg1) {
    final boolean isLock = (arg1 == 1);
    mIsToPromptPassword = isLock;
    activity.runOnUiThread( new Runnable() {
        public void run() {
            showPasswordLock(isLock);
        }
    });
    return true;
}
```

## 四 EPG 相关问题

commit 6bb68488557803841812663e60568787f127ed64  
Author: zoipuus <chenwenlong@top-tech.com.cn>  
Date: Tue Feb 14 19:37:36 2017 +0800

修复ISDB的DTV、ATV都有信号时，EPG界面显示DTV、ATV节目，且Program Guide Channel左右切换时有问题

问题：ISDB 制式，当 ATV、DTV 都有节目时，虽然 EPG 菜单显示的是 DTV 节目，单点击某个 DTV 节目时播放的是 ATV 节目。

解决：当前为 ISDB 时，需要对节目类型判断是否为 ATV 类型

```
@Override
public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
    Log.e(TAG, "setOnItemClickListener -->>>> ");
    timeoutHelper.reset();

    if (isEpgChannel == false) {
        if (!isEpgChannel) {
            // go to the selected channel
            final int selectedItemPosition = epgListView.getSelectedItemPosition();
            ProgramInfo curProgInfo = new ProgramInfo();
            if (ListView.INVALID_POSITION == selectedItemPosition) {
                return;
            }
            curProgInfo = mTvChannelManager.getProgramInfoByIndex(selectedItemPosition);
            mTvChannelManager.selectProgram(curProgInfo.number, curProgInfo.serviceType);
            resetAllNextBaseTime(true);
            if (mTvSystem == TvCommonManager.TV_SYSTEM_ISDB) {
                Log.e(TAG, "TVCommonManager.TV_SYSTEM_ISDB -->>>> ");
                EpgViewHolder epgViewHolder = (EpgViewHolder) epgProgramViewHolderList.get(selectedItemPosition);
                int count = mTvChannelManager.getProgramCount(TvChannelManager.PROGRAM_COUNT_DTV) + atvProgramCount;
                for (int index = 0; index < count; index++) {
                    curProgInfo = mTvIsdbChannelManager.getProgramInfo(index);
                    Log.e(TAG, "epgViewHolder.getChannelNum() -->>>> " + epgViewHolder.getChannelNum());
                    Log.e(TAG, "curProgInfo.number -->>>> " + curProgInfo.number);
                    if (!curProgInfo.isDelete || curProgInfo.isHidden || curProgInfo.isSkip || !curProgInfo.isVisible) {
                        if (epgViewHolder.getChannelNum() == curProgInfo.number) {
                            mTvChannelManager.selectProgram(curProgInfo.number, curProgInfo.serviceType);
                            resetAllNextBaseTime(true);
                            break;
                        }
                    }
                }
            } else {
                curProgInfo = mTvChannelManager.getProgramInfoByIndex(selectedItemPosition);
                mTvChannelManager.selectProgram(curProgInfo.number, curProgInfo.serviceType);
                resetAllNextBaseTime(true);
            }
        }
    }
}

});
```

```
@Override
public boolean onKeyDown(View view, int keyCode, KeyEvent event) {
    Log.e(TAG, "setOnKeyListener -->>>> ");
    timeoutHelper.reset();
    int curFocusIdx = epgListView.getSelectedItemPosition();

    //wxl add for switch channel 2016/3/14
    if (keyCode == KeyEvent.KEYCODE_ENTER) {
        if (ListView.INVALID_POSITION != curFocusIdx) {
            ProgramInfo curProgInfo = mTvChannelManager.getProgramInfoByIndex(curFocusIdx);
            if (mTvSystem == TvCommonManager.TV_SYSTEM_ISDB) {
                EpgViewHolder epgViewHolder = (EpgViewHolder) epgProgramViewHolderList.get(curFocusIdx);
                int count = mTvChannelManager.getProgramCount(TvChannelManager.PROGRAM_COUNT_DTV) + atvProgramCount;
                for (int index = 0; index < count; index++) {
                    curProgInfo = mTvIsdbChannelManager.getProgramInfo(index);
                    Log.e(TAG, "epgViewHolder.getChannelNum() -->>>> " + epgViewHolder.getChannelNum());
                    Log.e(TAG, "curProgInfo.number -->>>> " + curProgInfo.number);
                    if (!curProgInfo.isDelete || curProgInfo.isHidden || curProgInfo.isSkip || !curProgInfo.isVisible) {
                        if (epgViewHolder.getChannelNum() == curProgInfo.number) {
                            mTvChannelManager.selectProgram(curProgInfo.number, curProgInfo.serviceType);
                            resetAllNextBaseTime(true);
                            break;
                        }
                    }
                }
            } else {
                curProgInfo = mTvChannelManager.getProgramInfoByIndex(curFocusIdx);
            }
            /*
            int
            tpIdx = curFocusIdx + mTvChannelManager
                .getProgramCount(TvChannelManager.PROGRAM_COUNT_ATV);
            Log.d(TAG, "~~~~~ tpIdx = " + tpIdx);
            ProgramInfo curProgInfo = mTvChannelManager.getProgramInfoByIndex(tpIdx);
            */
            mTvChannelManager.selectProgram(curProgInfo.number, curProgInfo.serviceType);
            resetAllNextBaseTime(true);
            mTvChannelManager.selectProgram(curProgInfo.number, curProgInfo.serviceType);
            resetAllNextBaseTime(true);
        }
    }
}

//begin wangxielin 20160326
//int lastFocusIdx = programAdapter.getLastFocusItem();
```

问题：当非 ISDB 制式时，点击 Channel 界面时也会跳转

```
commit 3224077312987e7bde2c9a4863475cbfb884a8/d
Author: zoipuus <chenwenlong@top-tech.com.cn>
Date: Thu Nov 9 15:25:36 2017 +0800

0037245: 【公版】屏蔽EPG菜单的Channel界面点击事件

diff --git a/device/mstar/common/apps/MTVPlayer/src/com/mstar/tv/tvplayer/ui/dtv/epg/EPGActivity.java b/device/mstar
index c97df98..807e874 100755
--- a/device/mstar/common/apps/MTVPlayer/src/com/mstar/tv/tvplayer/ui/dtv/epg/EPGActivity.java
+++ b/device/mstar/common/apps/MTVPlayer/src/com/mstar/tv/tvplayer/ui/dtv/epg/EPGActivity.java
@@ -731,7 +731,7 @@ public class EPGActivity extends MstarBaseActivity {
     int curFocusIdx = epglstView.getSelectedPosition();

    //wxl add for switch channel 2016/3/14
    if(keyCode == KeyEvent.KEYCODE_ENTER){
+
        if(!EPGActivity.this.isEpgChannel && keyCode == KeyEvent.KEYCODE_ENTER){
            if (mTvSystem == TvCommonManager.TV_SYSTEM_ISDB //add by chenwenlong 20170217
                && isEpgChannel)
                return true;
        }
    }
}

chenwenlong@toptech-S1200RP-SE:/media/workspace/chenwenlong/Projects/MSD338_M/msd338$ |
```

```
commit 601e85e2e6fda65fe26976f7f6f71dc935bbd743
Author: zoipuu <chenwenlong@top-tech.com.cn>
Date: Sat Oct 28 16:37:01 2017 +0800
```

K069: 0036892: 修改EPG中的刻录、提醒的时间与列表的时间不一致

问题: EPG 功能的预约刻录、提醒和当前节目预约事件的时间段显示的是当地时间, 而预约事件列表显示的是 UTC 时间; 当前节目的预约事件也少了  
解决: 参照 638 改法, 获取时间的方法全部整改, 用 UTC 时间转换为当地时间, 再去获取 EPG 事件

```
calculateEventBaseTime();
nextEventBaseTime = calculateEventBaseTime(mTvTimerManager.getRtcclock());
+// calculateEventBaseTime();
totalEventCount = mTvEpgManager.getDvbEventCount(curProgInfo.serviceType,
curProgInfo.number, nextEventBaseTime.toMillis(false));
prevConstructEventCount = 0;
@@ -915,7 +918,7 @@ public class EPGActivity extends MstarBaseActivity {
+ String.format("%02d", startTime.minute) + "-"
+ String.format("%02d", endTime.hour) + ":"
+ String.format("%02d", endTime.minute);
-
+ Log.e(TAG, "timeInfoStr --->>> " + timeInfoStr);
+ EPGViewHlder vh = new EPGViewHlder();
+ vh.setChannelInfo(timerInfoStr);
+ vh.setChannelName(eventNameStr);
@@ -1062,7 +1065,7 @@ public class EPGActivity extends MstarBaseActivity {
Time stTime = new Time();
if (timeInSec == 0) {
// use default time
- stTime = mTvTimerManager.getCurrentTVTime();
+ stTime = calculateEventBaseTime(mTvTimerManager.getRtcclock());
} else {
// use specific time
stTime.set(timeInSec * 1000);
@@ -1173,7 +1176,7 @@ public class EPGActivity extends MstarBaseActivity {
+ epgEventInfo = eventInfoList.get(0);
+ int startTime = new Time();
+ startTime.set((long) epgEventInfo.originalStartTime * 1000);
+ time curTime = mTvTimerManager.getCurrentTVTime();
+ Time curTime = calculateEventBaseTime(mTvTimerManager.getRtcclock());
+ if (curTime.before(startTime)) {
+ initState = true;
}
@@ -1382,4 +1385,10 @@ public class EPGActivity extends MstarBaseActivity {
+ int streamOffsetMilliseconds = mTvTimerManager.getClockOffset() * 1000;
+ nextEventBaseTime.set(nextEventBaseTime.toMillis(true) + timezoneOffsetMilliseconds - streamOffsetMilliseconds);
}
+
+ private Time calculateEventBaseTime(long utc) {
+ Time time = new Time();
+ time.set(utc * 1000);
+ return time;
+ }
}
```

```
Log.e("ScheduleListActivity", "timezoneOffsetSeconds --->>> " + timezoneOffsetSeconds);
Time epgTime = new Time();
+ epgTime.set(epgEventTimerInfo.startTime * 1000);
+ int epgOffsetSeconds = mTvEpgManager.getEpgEventOffsetTime(epgTime, true);
+ Log.e("ScheduleListActivity", "epgOffsetSeconds --->>> " + epgOffsetSeconds);
+
+ time.set((long) ((long) epgEventTimerInfo.startTime - timezoneOffsetSeconds + epgOffsetSeconds) * 1000);
+ end_time.set((long) ((long) epgEventTimerInfo.startTime - timezoneOffsetSeconds + epgOffsetSeconds) * 1000 + (long) epgEvent
+
+ */
+ time.set((long) epgEventTimerInfo.startTime * 1000);
+ end_time.set((long) epgEventTimerInfo.startTime * 1000 + (long) epgEventTimerInfo.durationTime * 1000); // s->ms
+
+ ScheduleListViewHlder vh = new ScheduleListViewHlder();
+ if (epgEventTimerInfo.timerType == EPG_EVENT_RECORDER) {
+ vh.setVideoFlag(true);
}
@@ -523,7 +535,7 @@ public class ScheduleListActivity extends MstarBaseActivity {
+ EpgEventTimerInfo epgEventTimerInfo = new EpgEventTimerInfo();
+ //add by jerry 20160525 start
+ epgEventTimerInfo = mTvTimerManager.getEpgTimerEventByIndex(listView.getSelecteditemPosition());
+
+ Time baseTime = mTvTimerManager.getCurrentTVTime();
+ Time baseTime = calculateEventBaseTime(mTvTimerManager.getRtcclock());
+ int startTime = epgEventTimerInfo.startTime + mTvEpgManager.getEpgEventOffsetTime(baseTime, true);
+ // TODO: use string constant
+ intent.putExtra("EventBaseTime", startTime);
@@ -537,7 +549,7 @@ public class ScheduleListActivity extends MstarBaseActivity {
+ intent.setClass(ScheduleListActivity.this, ReminderActivity.class);
+ EpgEventTimerInfo epgEventTimerInfo = new EpgEventTimerInfo();
+ epgEventTimerInfo = mTvTimerManager.getEpgTimerEventByIndex(listView.getSelecteditemPosition());
+
+ Time baseTime = mTvTimerManager.getCurrentTVTime();
+ Time baseTime = calculateEventBaseTime(mTvTimerManager.getRtcclock());
+ int startTime = epgEventTimerInfo.startTime + mTvEpgManager.getEpgEventOffsetTime(baseTime, true);
+ // TODO: use string constant
+ intent.putExtra("EventBaseTime", startTime);
@@ -553,4 +565,9 @@ public class ScheduleListActivity extends MstarBaseActivity {
+ }
+ return super.onKeyDown(keycode, event);
}
+
+ private Time calculateEventBaseTime(long utc) {
+ Time time = new Time();
+ time.set(utc * 1000);
+ return time;
+ }
}
```