

转码配置模块说明书

版本	日期	作者	说明
1.0	2016-7-14		1、encode

1. 主要功能

模块主要功能：

- 1、编码参数设置，包括 Ittiam、Mainconcept
- 2、编码进程 Ittiam、Mainconcept 异常检测
- 3、编码进程 Ittiam、Mainconcept 异常守护线程，异常重启机制

2. 限制条件

- 1、此进程不能产生异常，否则编码进程 Ittiam、Mainconcept 不能正常工作

3. 输入

无

4. 输出

- 1、Ittiam、Mainconcept 的参数配置文件

5. 目录结构

Objs 目录：存放编译时产生的.o 等文件，是一个临时文件

.c/.h 文件：模块的源文件

Makefile 文件：编译所需文件

Build.sh 文件：一键编译文件，产生的可执行文件 encode_x64 将被放到上级目录的 build 文件中

依赖库文件： libmxml_x64.a（XML 库文件）、libmsg_x64.a（消息机制库文件）、libcjson_x64.a（json 类型文件构造库文件）

6. 流程

一、代码流程图

```

pHandle = (CD_Handle *)calloc(1,sizeof(CD_Handle));

if(pHandle)
{
    if(cd_pHandle)
        free(cd_pHandle);
    cd_pHandle = pHandle;

    //加载默认参数信息
    Cd_Core_Def_Paramter_Load(pHandle);

    //加载参数
    Cd_Xml_ChannelParamter_Load(pHandle,CD_XML_DIR,CD_CHANNEL_XML);

    //保存参数
    Cd_Xml_ChannelParamter_Save(pHandle,CD_XML_DIR,CD_CHANNEL_XML);

    //启用通信
    pHandle->mq_rcv = Msg_Rcv_Create(MQ_ENC_RECV);
    pHandle->mq_send = Msg_Send_Create(MQ_ENC_SEND);

    signal(SIGCHLD,Cd_Core_SigRcv); //接收子进程意外退出状态

    Cd_Core_Appay_Paramter(pHandle);

    /*负责重启异常转码进程 并且保存日志*/
    init_daemon(pHandle);

    pHandle->m_ready = 1;
    while(pHandle->m_ready)
    {
        memset(&rmsg, 0 ,sizeof(Msg_Param));
        lRet = Msg_Rcv(pHandle->mq_rcv,&rmsg);
    }
}

```

```

    free(cd_pHandle);
    cd_pHandle = pHandle;

    //加载默认参数信息
    Cd_Core_Def_Paramter_Load(pHandle);

    //加载参数
    Cd_Xml_ChannelParamter_Load(pHandle,CD_XML_DIR,CD_CHANNEL_XML);

    //保存参数
    Cd_Xml_ChannelParamter_Save(pHandle,CD_XML_DIR,CD_CHANNEL_XML);

    //启用通信
    pHandle->mq_rcv = Msg_Rcv_Create(MQ_TC_RECV);
    pHandle->mq_send = Msg_Send_Create(MQ_TC_SEND);
    //接收子进程意外退出状态
    signal(SIGCHLD,Cd_Core_SigRcv);
    //参数应用
    Cd_Core_Appay_Paramter(pHandle);

    /*负责重启异常转码进程 并且保存日志*/
    init_daemon(pHandle);

    pHandle->m_ready = 1;
    while(pHandle->m_ready)
    {
        memset(&rmsg, 0 ,sizeof(Msg_Param));
        lRet = Msg_Rcv(pHandle->mq_rcv,&rmsg);
        if(lRet == 0)
        {
            switch(rmsg.Msg_Type)
            {
                case MSG_TYPE_POST:
                    Cd_Msg_PostProcess(pHandle,rmsg.Msg_SubType,rmsg.Msg_Parameter,rmsg.Msg_FileName);
                    break;
                case MSG_TYPE_GET:
                    //Cd_Dprintf(" -----get:%s-----\n", rmsg.Msg_FileName);
                    Cd_Msg_GetProcess(pHandle,rmsg.Msg_SubType,rmsg.Msg_Parameter,rmsg.Msg_FileName);
                    break;
            }
        }
    }
}

```

图 1 代码流程图

二、流程框图

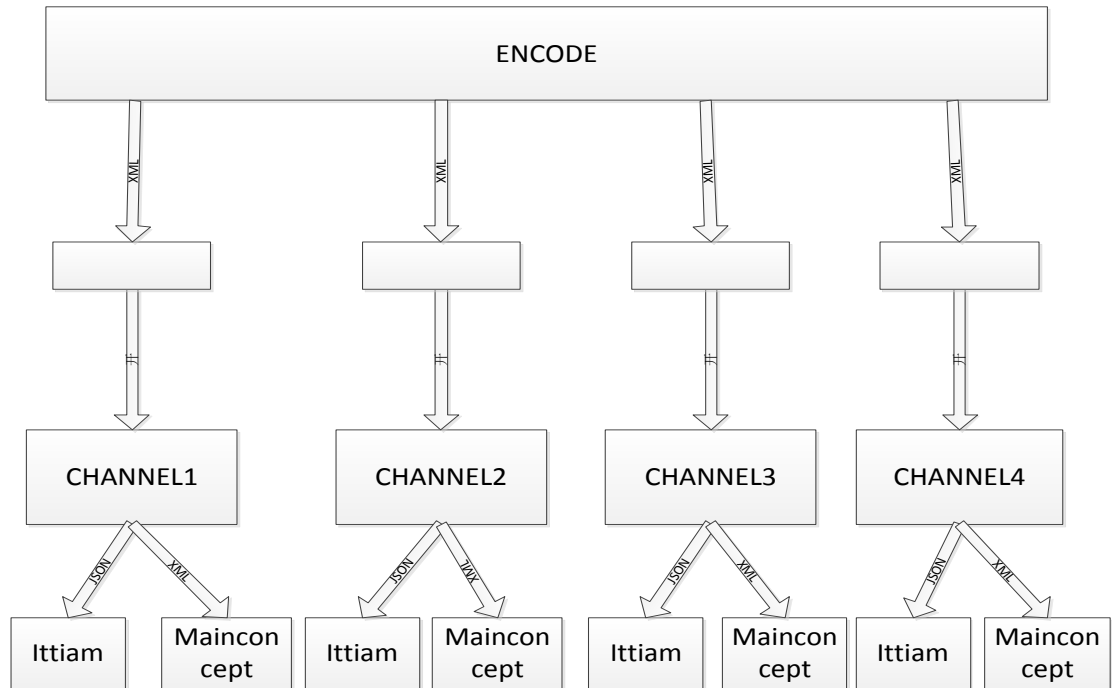


图 2 转码流程框图

三、代码详解（按执行顺序）

1、首先为进程的全局 handle 分配空间

```
pHandle = (CD_Handle *)calloc(1,sizeof(CD_Handle));
```

2、加载默认参数信息

```
Cd_Core_Def_Paramter_Load(pHandle);
```

3、加载 XML 参数

```
Cd_Xml_ChannelParamter_Load ();
```

4、保存参数到 XML 文件

```
Cd_Xml_ChannelParamter_Save ();
```

5、创建本进程消息队列

```
pHandle->mq_rcv = Msg_Rcv_Create(MQ_TC_RECV);  
pHandle->mq_send = Msg_Send_Create(MQ_TC_SEND);
```

6、接收子进程意外退出状态

```
//主要作用是检测 ittiam、mainconcept 两个转码进程是否异常  
signal(SIGCHLD,Cd_Core_SigRcv);
```

7、参数应用生效

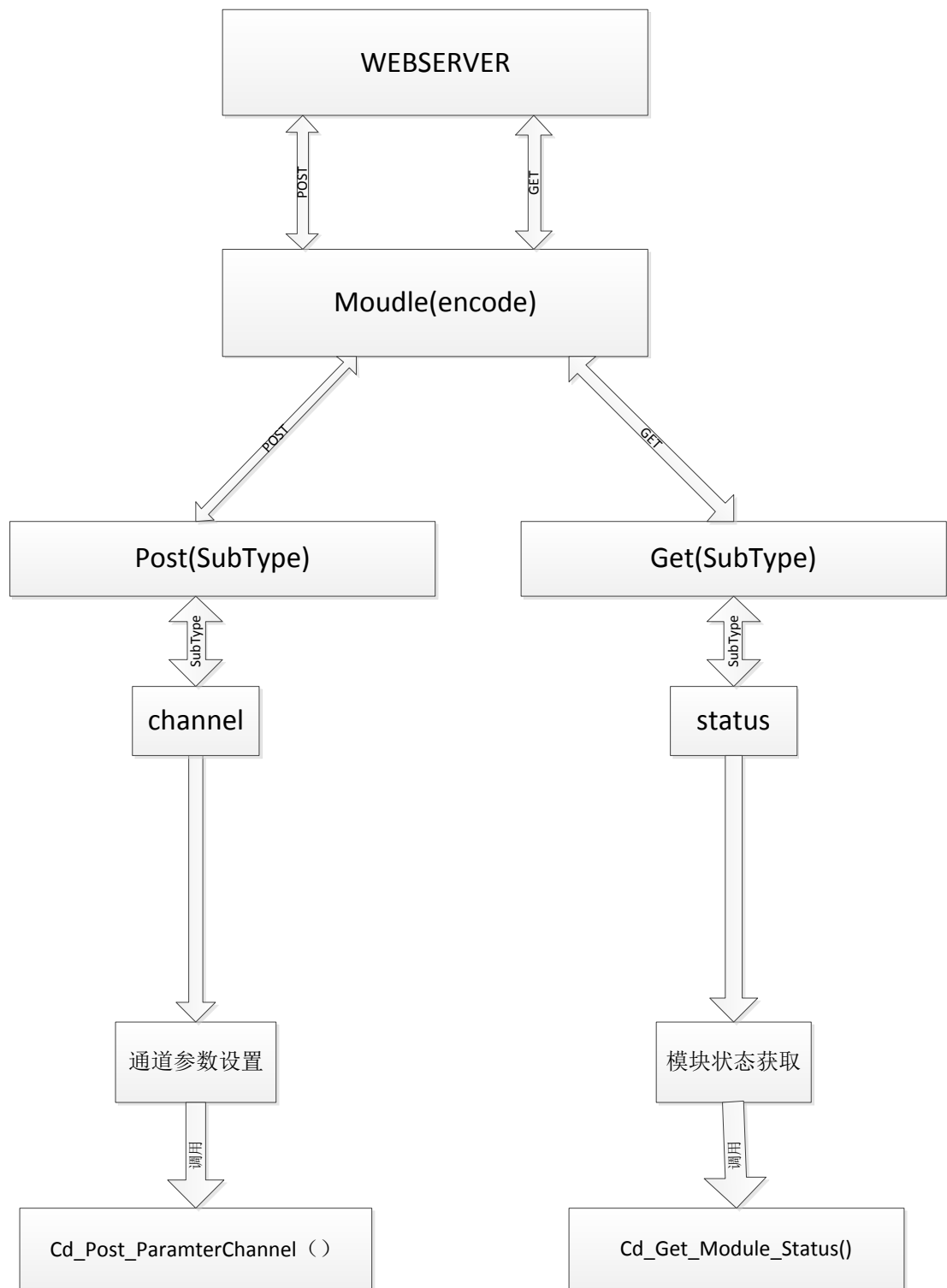
```
Cd_Core_Appay_Paramter ();
```

8、重启异常转码进程并且生成相关日志

```
init_daemon ();
```

9、阻塞等待接收消息，并且判断消息类型

```
Msg_Rcv(pHandle->mq_rcv,&rmsg);  
主要流程如下，包括 POST 和 GET 方法：
```



图三 WEB 和本进程交互过程

四、关联 WEB 界面

1、通道编码配置

通道编码配置				
	1	2	3	4
节目选择	通道 2	通道 1	无	无
编码模块	Moudle B	Moudle B	Free	Free
视频分辨率	自定义	自定义	自定义	自定义
视频宽	720	720	720	1280
视频高	576	576	576	720
视频码率	2000	4000	4000	4000
视频帧	25	25	25	50
音频类型	AAC-LC	AAC-LC	AAC-LC	AAC-LC
音频码率	96kbps	96kbps	96kbps	32kbps
音频采样率	48kHz	48kHz	48kHz	48kHz
音频声道	Stereo	Stereo	Stereo	Stereo
音频延时	0	0	0	0

提交刷新

图 4 通道转码配置

7. 鸣谢

感谢之前对这个模块有过贡献的同事们。