## Hwcfg\_Concerto\_Default.dll 介绍

**Ni kai**

1. 功能简介 ………………………………………………………………………………………………………………… 2
2. 命名方式 ………………………………………………………………………………………………………………… 3
3. 使用逻辑 ………………………………………………………………………………………………………………… 4

## 功能简介

hw\_concerto\_default.dll是用来获取flash.bin中hareware config数据的库。

工具HarewareConfig.exe通过调用hw\_concerto\_default.dll获取到需要展现到UI上的数据，并show到UI上供用户查看或修改。

为什么要使用Hwcfg\_concert\_Default.dll?

Hwcfg\_concerto\_Default.dll使得flash.bin对外部调用者透明，调用者不需要关心当前的flash.bin是属于哪个平台，flash.bin内部的数据时如何存放的，每个block的id,name及偏移量是多少等等，总而言之是使得flash.bin的内部特性与UI分离。

## 命名方式

hw\_concerto\_default.dll中，Concerto是平台名，Default是客户名，这两个部分是可以修改的，比如对应Magic平台的A客户，那么这个dll的名称就应该是Hwcfg\_Magic\_A.dll。也就是说，Hwcfg\_Magic\_A.dll就是专门用来提取Magic平台A客户所需要的hwcfg信息的。

## 使用逻辑

Hwcfg\_Concert\_Default.dll的导出函数如下：

int ProjectInitial(wchar\_t\* initial\_parameter); //初始化

int Load(wchar\_t\* file\_path, struct NativePropertyArray\* all\_propertys); //解析file\_path指定的文件，将解析出来的NativeProperty保存到NativePropertyArray中

int Save(wchar\_t\* file\_path, struct NativePropertyArray\* all\_propertys); //将修改好的NativePropertyArray反序列化，保存至指定的file\_path

int Clean();

void GetLastError(wchar\_t\* info, size\_t info\_length);

用户通过调用Load接口传入flash.bin的路径，获取到一个指向NativePropertyArray结构体指针，此结构体定义如下：

typedef struct NativePropertyArray

{

unsigned int size;

NativeProperty\* first;

} NativePropertyArray;

NativePropertyArray包含了一个NativeProperty数组，以及里面存放的NativeProperty的个数size。

NativeProperty的定义如下：

typedef struct NativeProperty

{

wchar\_t key[SHORTLENGTH]; //the property's name

wchar\_t value[MIDDLELENGTH]; //the property's value

wchar\_t group[SHORTLENGTH]; //shows what group does the property belong to

wchar\_t options[LONGLENGTH]; //combine all the option strings into one.

// And split the options with semicolon(";")

// E.g. LNB type's options is "Standard;User;Universal"

pf\_Validation m\_validation; //this a function pointer points to validation function

} NativeProperty;

它抽象出了hwcfg需要得到flash.bin中的元素的property，调用者(HarewareConfig)无需知道当前需要展现在界面上的元素是属于哪个group，也不必关心这些key的意义，也不必关心这个key在不同平台的区别。

假设现在用户需要导入Magic平台A客户的flash.bin。

用户使用HardwareConfig导入flash.bin，HardwareConfig动态加载Hwcfg\_Magic\_A.dll，然后调用里面的Load接口，获取到一个NativePropertyArray，将里面的NativeProperty展现到UI上。

用户编辑完UI上的数据，需要保存时，HardwareConfig首先先将UI上的数据序列化到NativePropertyArray中，然后调用Save接口，将序列化好的NativePropertyArray和文件需要保存的路径传入，即可。