


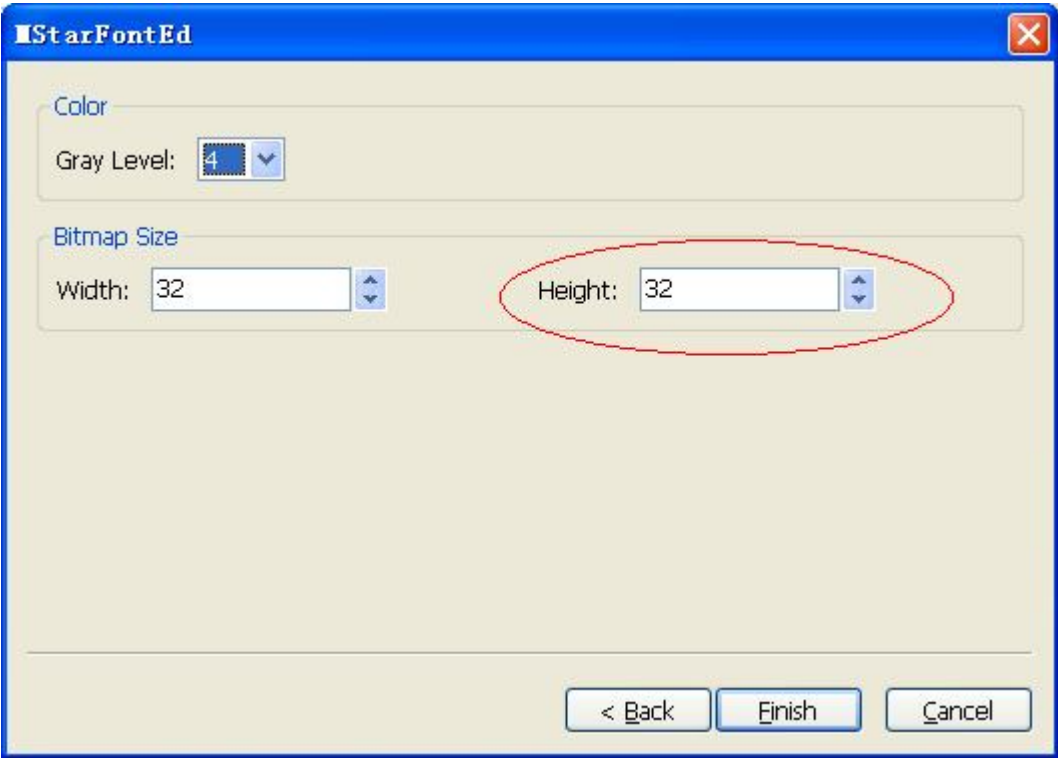
<div></div> <div>鼎科实业</div> <div>程 序 文 件</div>		文件编号	RD-TBC	
		版 本	A/0	
		制定日期	2012-7-30	
文件名称	调试报告		页 码	共 页, 第 1 页
产品名称:	MSD6306V6.0 公版平台		调试类型:	<input checked="" type="radio"/> 软件 <input type="radio"/> 硬件
调试人员:	万文国	调试时间:	2013-8-2	编制日期:

一、 字体美化

目前 Mstar 6306 方案使用为 2bits 4 色 Font，而字库本身有分为点阵和矢量两种格式。
字体美观度很大程度决定于字体库以及字体大小选择，不同字库和大小设定制作出来的字体显示会有差异。

点阵式字体制作请参考公司 Mantis 调试报告《mstar 点阵 font 字库添加调试报告.pdf》
路径: <http://172.168.0.240/view.php?id=1121>

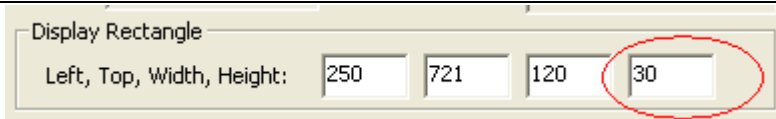
- 重点主要事项:
1. 在建立工程文件初始化设置时需要选择好 Bitmap Size，通常情况下 Height 的大小应该小于等于 UI 中文本显示框的高度



如主 UI 中文本框 MENU_TOP_BALL_NAVIGATION_TEXT

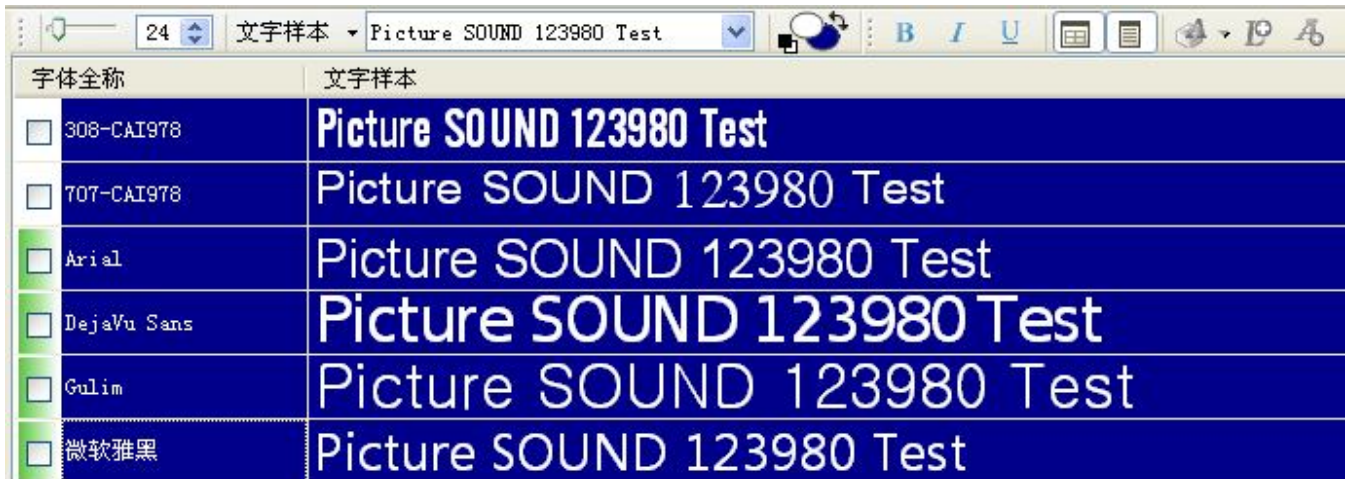


Height 大小设置为 30，则用于显示的字体的高度不能超过 30，否则无法正常显示

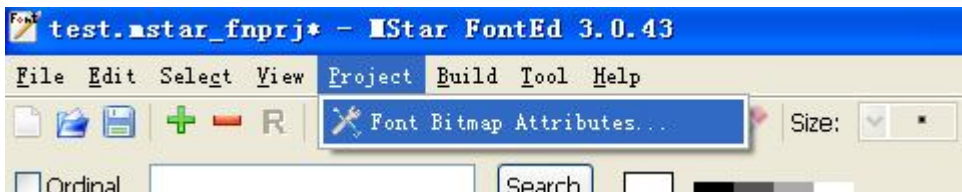


2. 字体库文件选择，客户有需求按客户需求，没有需求的请选择字体美观度佳的字体库文件。

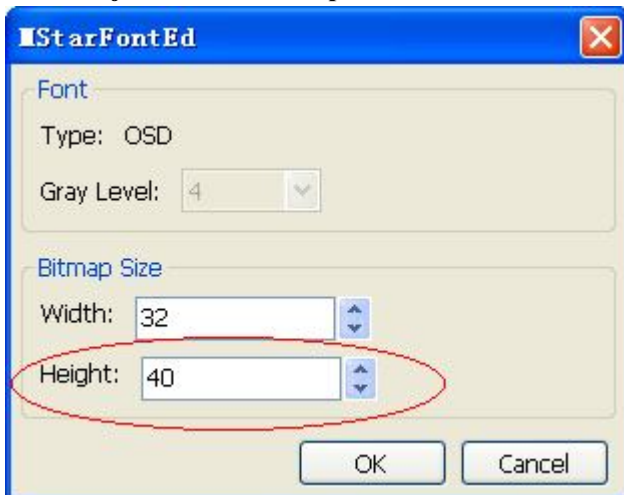
通常在 Windows 自带的 Font 文件夹中有安装一些常用字体库，也可以自行下载字体库文件进行安装。字体效果预览可以使用 word, Excel 等常用编辑类软件查看，也可以使用专业字库查看软件来查看效果（如字体试衣间）。下图为使用软件字体试衣间对比六种字库效果的截图。当然在这些软件中看到的效果都是使用的 256 色，而实际我们用到的只有 4 色，还是有一些差异，在这里查看效果的目的仅仅是为了选择好字体样式。



3. 当选择设置 bitmap size 无法满足制作的字体显示时（即导入到字体超出 bitmap 显示范围时，这时可以将 bitmap 的大小进行拉大处理。



选择 Project ->Font Bitmap Attributes...进行设置



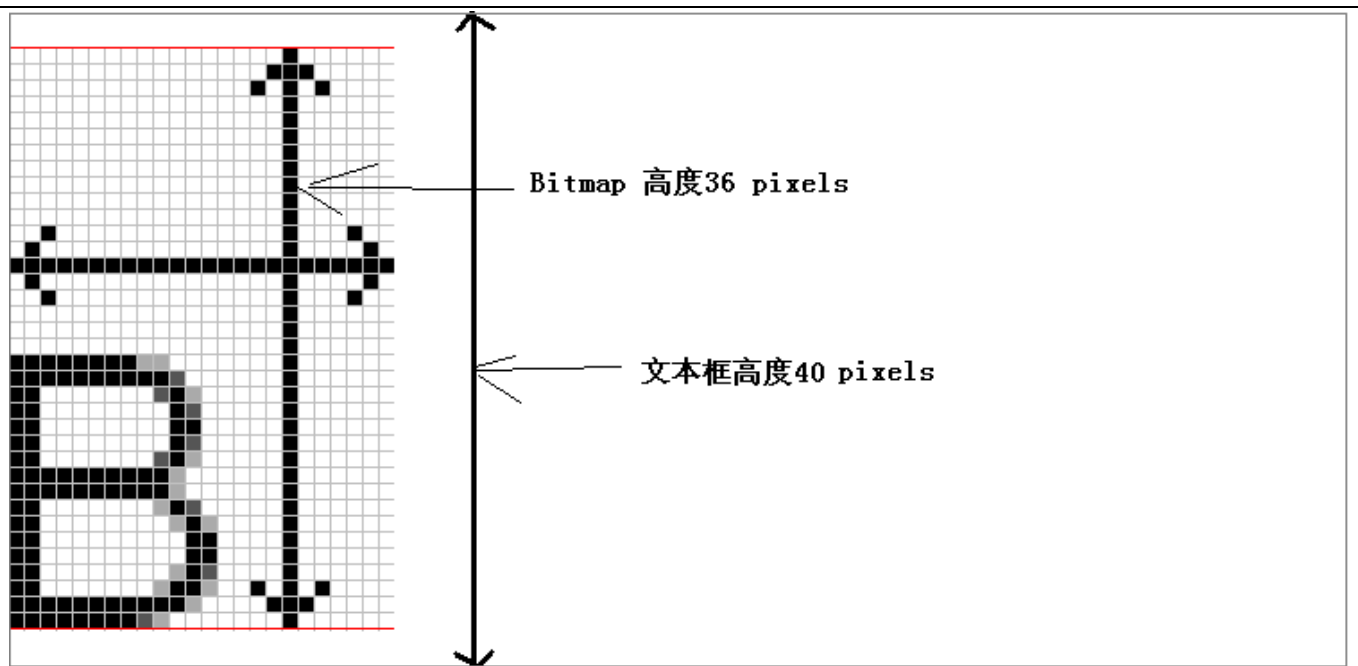
4. 字体显示位置和字体间隔大小设置

字体在文本框中上下显示位置取决于字体 Bitmap size 的设置和字体本身在 bitmap 中的上下偏移大小；

左右显示位置取决于字体本身在 Bounding Box 中的左右偏移大小；

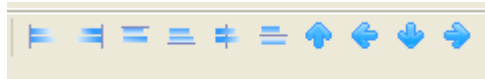
Bitmap Size 小于文本框高度时（如 bitmap Height 设置为 36，而文本框高度为 40 时），bitmap 在文本框上下方向居中显示。

Bitmap Size 等于文本框高度时，字体垂直显示位置仅与字体本身在 bitmap 中的上下偏移大小相关。



此图实际显示效果为，垂直方向字体底边离文本框底边 2 个 pixels，水平方向与文本框左侧边框对齐

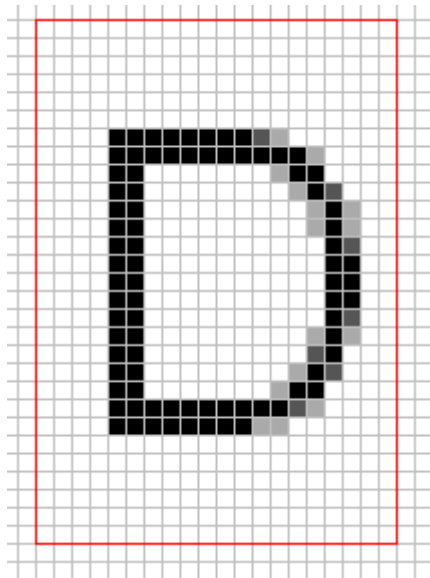
在 Mstar 字体工具中可以使用如下子功能来调整字体显示位置，包括左对齐，右对齐，上对齐，下对齐，上下左右单步移动(注意当 Bitmap 高度小于文本框高度时，需要综合考虑 bitmap 在文本框的上下偏离来调整字体)



文本框中显示字体间间隔大小取决于字体左右空白边宽度和文本框字符间距属性设置

字体左右空白边宽度指字体最左最右有效 pixel (非背景色) 到 Bounding Box 左右红线边的宽度 (单位 pixel)

如下图：左空白边：4pixels，右空白边：2 pixels



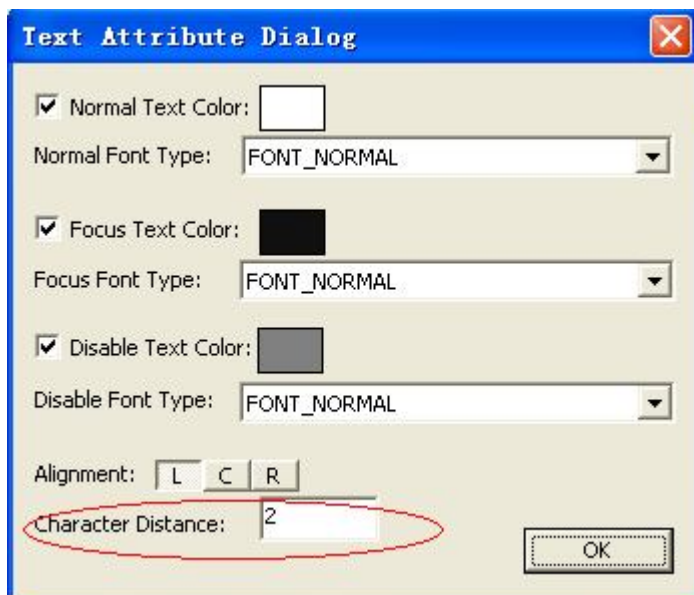
在 Mstar 字体工具中可以使用如下子功能来调整字体空白宽度



注意某些语言字符串要求连体显示，即在制作这个字符的点阵字体时需要考量是否要留左右空白边

如阿拉伯文中就有部分字符不需要左右空白边

文本框字符间距属性设置可以在 Skin Tool 中进行设置，通常默认为 2 pixels



少量字符串间距需要调整时，可以修改文本框属性

批量修改，可直接在代码中进行修改，先找到字间距赋值的地方，加上特定条件进行设定即可。

在代码查找关键词“Fontfmt.ifont_gap”，绘制前的取值动作在文件 MApp_ZUI_APIdraw.c 的绘制函数 _MApp_ZUI_API_DrawDynamicComponent 中进行，且初始值均来自于 skin tool 生成的文件 ZUI_components.c.inl 中的数据组 Zui_TextOut_List 的成员 u8dis。

如显示阿拉伯文时，在程序中就有添加如下程序段来批量修改字符间距

```
#if (ENABLE_ARABIC_OSD)
    if ( (bArabicDisplay))
    {
        pclrBtn->Fontfmt.ifont_gap = 0;
    }
#endif
```

5. 在矢量字库制作过程中，需要特别注意字体大小的设置，通常使用 TTF_CVT.exe 工具生成 MVF 文件后，可以在预览页面进行预览，并确定矢量字大中小三号字体分别需要设置到多少。

Depth：字体颜色数（2，4，16，256）常用 I2（4 色）

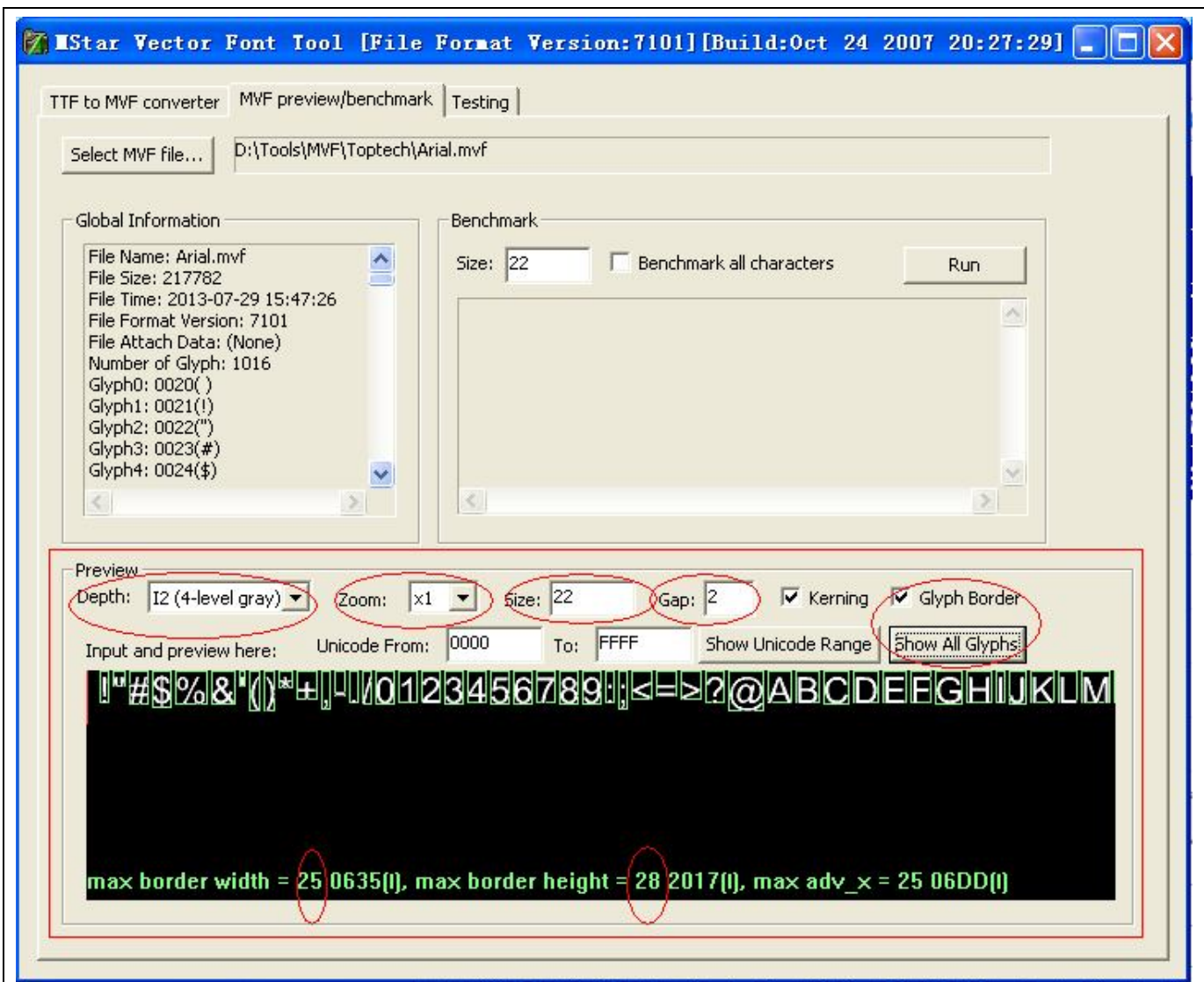
Zoom：可以放大查看字体边缘是否毛刺较多，或者字体边缘是否颜色清晰

Size：设置不同的大小来获取 MVF 文件所有字符中最宽和最高值，此值将用于程序的设定，需要配合 zoom 来设置最佳大小，前提是字符的高度和宽度最高不能超过文本框的高度和宽度

Gap：仅仅用于预览效果，相当于文本框中设置的字间距值

Glyph border：勾选后可以显示相应 size 字体的高宽，相当于点阵的 Bounding Box

Show All Glyph：可以统计 MVF 文件中所有字体中的最高和最宽值，两个值相当于点阵中的 bitmap size 设定的高宽



针对矢量字体在代码会有特别的函数来初始化一些变量后，再有 IC 自动生成相应大小的字体供调用。

```
U8 u8num = sizeof(font_info)/sizeof(LOADFONT_STRUCT) - 1;
```

```
MApp_LoadFont_SetVecFontInfo(font_info, 0, u8num);
```

初始化表 font_info 中我们可以设置相应的字体显示参数。可以自行添加显示参数定义

```
LOADFONT_STRUCT font_info[] =
```

```
{
    {
        .FontID = 0,
        .output_bpp = 2,
        .output_size = 35,
        .output_width = 30,
        .output_height = 40,
        .unicode_from = { 0x0020, 0},
        .unicode_to = { 0xFFFF, 0},
    },
    {
        .FontID = 1,
        .output_bpp = 2,
        .output_size = 28,
        .output_width = 24,
        .output_height = 32,
```



```

        .unicode_from = { 0x0020, 0},
        .unicode_to = { 0xFFFF, 0},
    },
    {
        .FontID = 2,
        .output_bpp = 2,
        .output_size = 22,
        .output_width = 19,
        .output_height = 25,
        .unicode_from = { 0x0020, 0},
        .unicode_to = { 0xFFFF, 0},
    },
};

```

FontID = 0, 1, 2 分别对应字体大小设置的 FONT_BIG, FONT_NORMAL, FONT_SMALL

output_bpp 对应为矢量字体的颜色位数, 1bit, 两色, 2bits, 四色, 依次类推

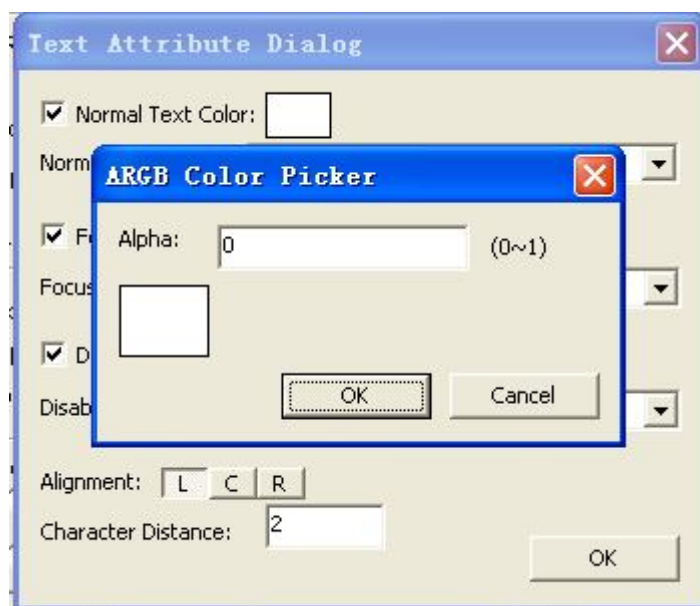
output_size 不同字体大小定义对应的实际值

output_width 和 output_height 为字体最宽和最高

unicode_from 和 unicode_to 为 unicode 范围设定

6. 字体透明度设置

默认是设置为不透。在 skin tool 中文本框属性设置 ARGB color Picker 的 Alpha 值



批量设置可以在程序中修改透明设置参数

在字符串显示前均会设置字体的透明度, 蓝色标记的参数为 skin tool 生成, 红色参数是设置不同的透明模式, 可根据不同需要去设置。

```

Mapi_GFX_SetAlpha(true, COEF_ASRC, ABL_FROM_CONST, text_info.color.a);
Mapi_GFX_TextOut(font_handle, (MS_U8*)pul6Str, pclrBtn->bStringIndexWidth+1,
&text_info);

```

7. 点阵字库和矢量字库与主程序编译整合

为了节省有效的 Flash 存储资源, 建议使用压缩格式, 软件在加载点阵字库或生成矢量字体时会自动解压后再导入内存区。路径结尾的“-7”就是编译整合使用压缩的标志

```
#define BIN_ID_FONT_MSTAR_MVF
```

0x01A0

//~271KB

//@..\..\..\tv-ap\dvb\ui2\font\MStar_Unicode5.mvf -7

#define BIN_ID_FONT_BIG5_CH

0x0105

//~410KB

//@..\..\..\tv-ap\dvb\ui2\font\Font_Big5_24x24.bin -7

压缩后会生成*.mvfc 和 *.binc 文件

对压缩感兴趣的可以研究下代码路径/ scripts/*.py 文件

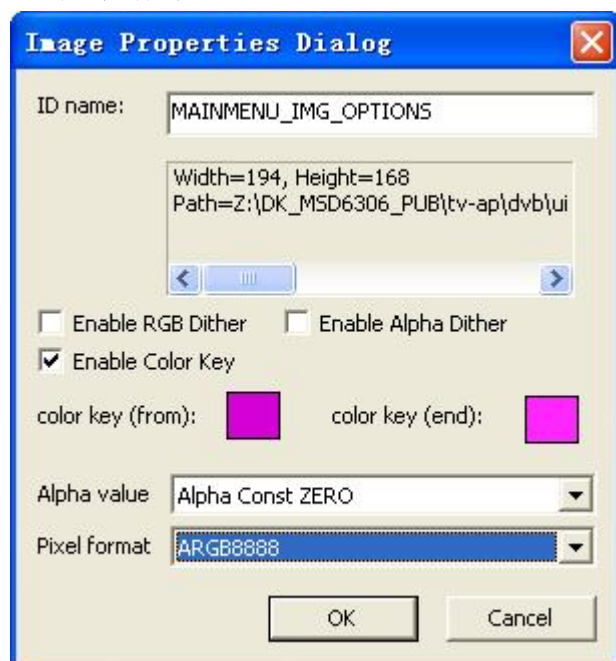
二、 图片美化

此部分工作首要条件是所用的图标边缘清晰，色彩鲜明。其次才是通过一些显示设定来达到想要的效果
主菜单图标部分：

1. 美工修改图标边缘和背景色；

需要注意如果图片在使用时需要将背景设置为透明，那么就确保图片中其他部分没有使用到背景色，否则图片会有透色现象，另外图片使用到的黑色不能低于 R=8, G=8, B=8，否则也同样会透色

2. 图片属性设置



设置相应的 color key 范围使得背景色可以完成透明

关闭整个图片的透明设置，将 Alpha 值设置为 const ZERO

把图片格式设置为 ARGB8888，使得图标可以保留更多的颜色信息，当然也会消耗更多的资源。

3. 代码中设置

因为默认代码使用的位图格式设定为 ARGB1555，所以在透明设置上需要做一些修改，使得图片显示可以满足效果需求

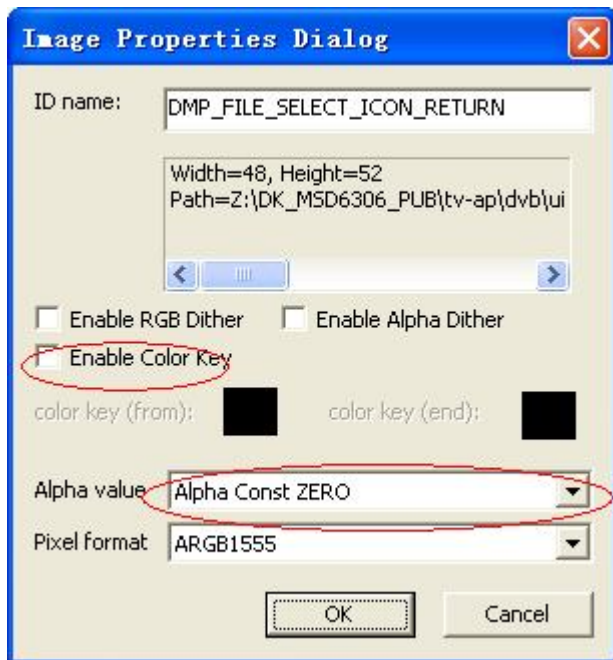
针对主 UI 图片显示在函数_MApp_ZUI_API_DrawDynamicComponent 增加条件设置

```
if((paramBitmap->u16BitmapIndex ==E_BMP_MAINMENU_IMG_CHANNEL)
|| (paramBitmap->u16BitmapIndex ==E_BMP_MAINMENU_IMG_PICTURE)
|| (paramBitmap->u16BitmapIndex ==E_BMP_MAINMENU_IMG_AUDIO)
|| (paramBitmap->u16BitmapIndex ==E_BMP_MAINMENU_IMG_TIME)
|| (paramBitmap->u16BitmapIndex ==E_BMP_MAINMENU_IMG_OPTIONS)
|| (paramBitmap->u16BitmapIndex ==E_BMP_MAINMENU_IMG_LOCK)
)
{
MApi_GFX_SetAlpha(TRUE, COEF_CONST, ABL_FROM_ROP8_DSTOUT, 255);
}
```

DMP 菜单图标部分：

主图标直接替换美工修改后的图片后，效果已很大改善。

针对其余图标显示边缘有残缺问题，考虑 DMP 菜单背景色为黑色，而大部分图标的背景色也同样为黑色，所以将原来开启的 color key 功能关闭，并将 Alpha 值设置不透明后，效果同样改善明显。



审核及见意：

