



***Terawins, Inc.***

**Quick Guide  
Version 1.00**

*Jan. 24, 2014*

## **T582 OSDTool Quick Guide**

---

Terawins Confidential

TERAWINS, Inc

<b>1</b>	<b>TW OSDTOOL .....</b>	<b>4</b>
1.1	图层与架构 .....	4
1.2	TW ART .....	5
1.3	RESOURCE MANAGER 图资管理员 .....	7
1.4	专案架构 .....	8
1.5	LAYOUT 版面设定 .....	9
1.6	MENU SPIOSD 图层设定 .....	10
1.7	SPRITE 图层设定 .....	12
1.8	OSD2 图层设定 .....	13
1.9	PROJECT 项目管理 .....	23
1.10	RESOURCE 图资输出 .....	23
1.11	输出档案 .....	24

**Revision Note**

Revisions	Content	Date
V1.00	First release	Jan. 24, 2014

Terawins Confidential

## 1 TwOSDTool

TwOSDTool 为本公司开发之屏幕视控系统(On-Screen Display)辅助开发工具包，其主接口外观如下图所示。本套件将在 OSD 开发时提供各项协助，帮助您轻松完成绚烂夺目的美丽接口。

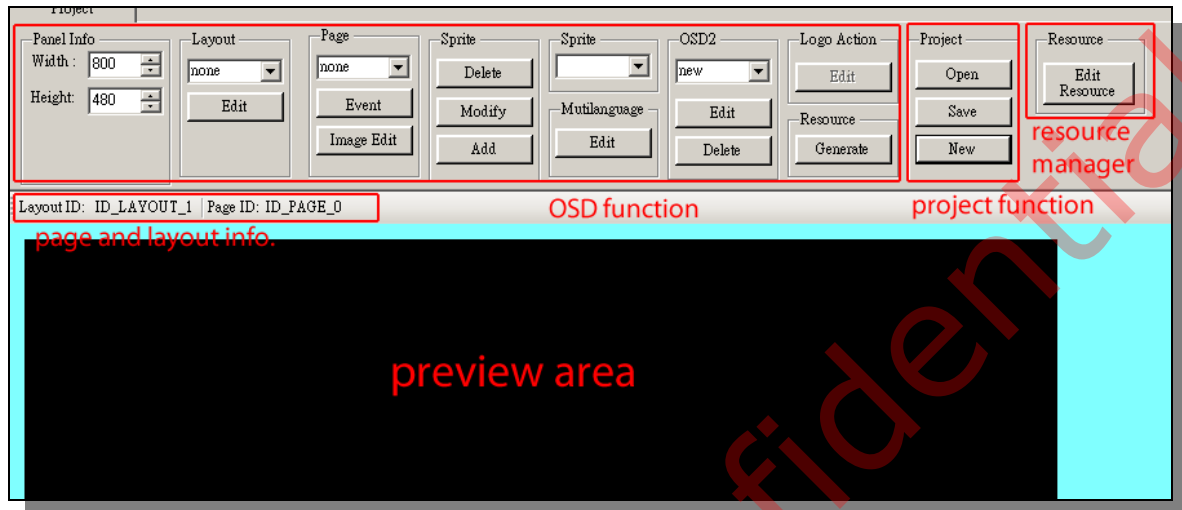


图 1-1 主接口外观

### 1.1 图层与架构

OSD 接口由四个独立图层互相迭加后显示，其顺序由前至后为 OSD2, Sprite, SPIOSED, Video，如图 1.2 所示。各图层功能如下所述：

1. OSD2: 显示以 16x24 像素(pixel) 为单位之复数静态图片，支持 PNG8 1bp, 2bp, 4bp, 8bp。
2. Sprite: 显示最大 255(8 bit)x1024(10 bit)之单一动态图片，支持 PNG8 8bp。
3. SPIOSED: 显示最大 2048(11 bit)x1024(10 bit)之单一静态图片，支持 PNG8 8bp 与 PNG32 (twbc)。
4. Video: 显示讯号影像。

于制作接口前可依据上述各图层之特性，预先根据接口功能决定各图资之位置，可收事半功倍之效。

了解 OSD 之基本特性后，接下来本文件将以创建单一页面为轴心，依照各图层之建置流程介绍 tool 之使用方式。

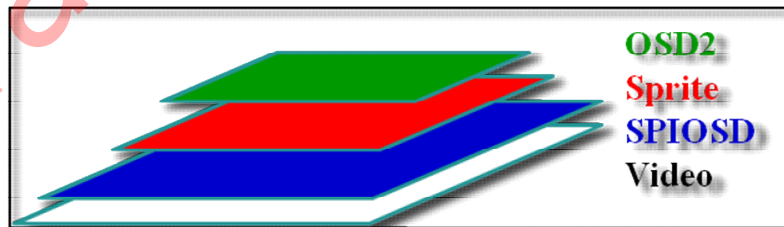


图 1-2 OSD 各图层示意图

## 1.2 TW Art

OSD 项目开发时需要美编人员与开发人员通力合作，方能完成兼具美感与实用之项目。故为了方便美工人员与开发人员进行协作，本套件特别将有关图资优化之功能独立为 TW Art 开发工具包。如此开发人员于项目建构时无需烦恼图资优化之隐忧，美编人员也无需烦恼程序架构，而可以专注于图资制作上。

如同上一节中图层特性所提到的，OSD2 图层并非以像素为单位显示画面，而是将画面以 TILE 为单位分割显示。故图片若放置于 OSD2 图层，则可藉由删除图片中重复的区块进行优化。达到降低图片容量之效果同时，也降低绘制 OSD2 图层时不必要的运作负担。

首先点击"OpenPNG"按钮并加载 PNG 图片。接下来调整 TILE 尺寸并点击 AutoCut 按钮由程序进行优化，由下图一中可看到重复的区块已被选取并删除。最后点击"OSD2 Icon"按钮进行存盘，如此存放图片之目录下将会出现"picture\_name.plt"之记录文件。而 OSDTool 将会于加载图片时尝试寻找此纪录文件，若 Tool 成功加载记录文件，则图标上将会出现 AutoCut 之标记，如下图二所示。

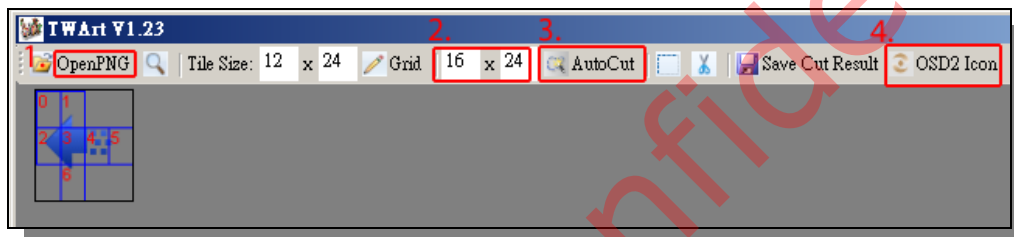


图 1-3 autocut 操作流程示意图

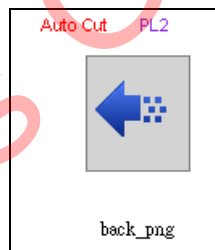


图 1-4 OSDTool 载入示意图

若项目中有同样形式之图标却颜色相异，同时加载复数图标不只浪费内存空间，也造成管理上的困扰。故本套件提供增加色盘之方式，开发人员只需切换色盘即可达成换图效果。

首先点击"Platte"选项，开启色盘编辑接口。接下来点击"Copy PL1 to PL2"按键，将第二套色盘填充为与第一套色盘相同之颜色。第三步依序点击色盘中之颜色，并由调色盘中选取颜色替换之。第四步勾选是否启用色盘，并点击"OK" 按键关闭接口，最后点击"Save Palette"储存色盘信息。如此 TW Art 将于图片目录下储存色盘信息"picture\_name.plt"，而 OSDTool 将会于加载图片时尝试寻找此色盘信息。若 Tool 成功加载色盘信息，则图标上将会出现 PL2 之标记，如下图所示。

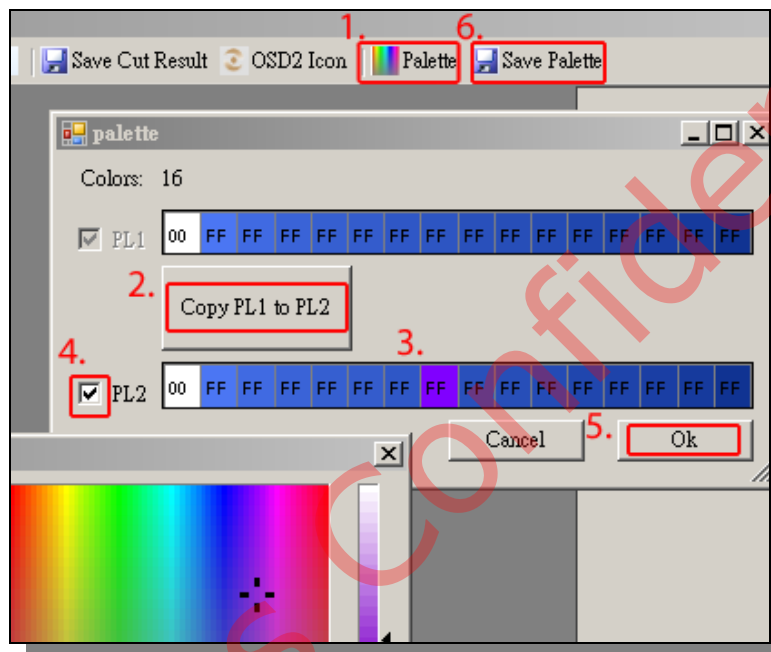


图 1-5 look up table 优化示意图

### 1.3 Resource Manager 图资管理员

此接口可将开发 OSD 所需使用之图资先行加载，并对图资统一进行加载、删除与汇入汇出等管理。但加载图资前请务必确认所有图片文件名皆为英文，避免后续流程产生错误。

如下图中所示，首先点击主接口中 **Resource** 页框之"Edit Resource"按钮，开启图资管理员。接下来，根据图片档案之格式选择"Resource type"，各选项对应之格式与图层如下表所示。第三步点击"Load Image"按钮，由文件管理器中选择图片档案(可复选)进行载入，并重复上述步骤直至所有图资载入完成。最后点击"Ok"按钮，结束图资加载之流程。

若开发中需大量替换图资档案时。首先确认旧图资之档名，并编辑新图资档名使其相同。接下来将新图资统一集中至同一目录下，并点击图资管理员之"Reload Image"按钮。最后选择新图资所在目录，确认后图资即自动进行替换。唯需注意此功能 **目前仅支持"PNG-8 with alpha"之图片格式(TWBC 没有此限制)**，使用前请务必确认。

而若需编辑单一图资档案而非批处理时，图资管理员也提供了另一种便捷的处理方式供使用者选择。若需进行替换，只需要对该图资点击右键，再点击"Change Image"选项。最后由文件管理器中选择图资档案，确认后即完成替换之动作，而项目中使用该图资之页面也将同时进行更新。而若需删除图资时，同样对该图片点击右键，并点击"Delete Resource"选项，即可将该图片由项目中移除。但若项目中有使用该图资时，图资管理员将会阻止使用者删除图资。请务必将项目中使用该图资之页面修改后，再进行删除。

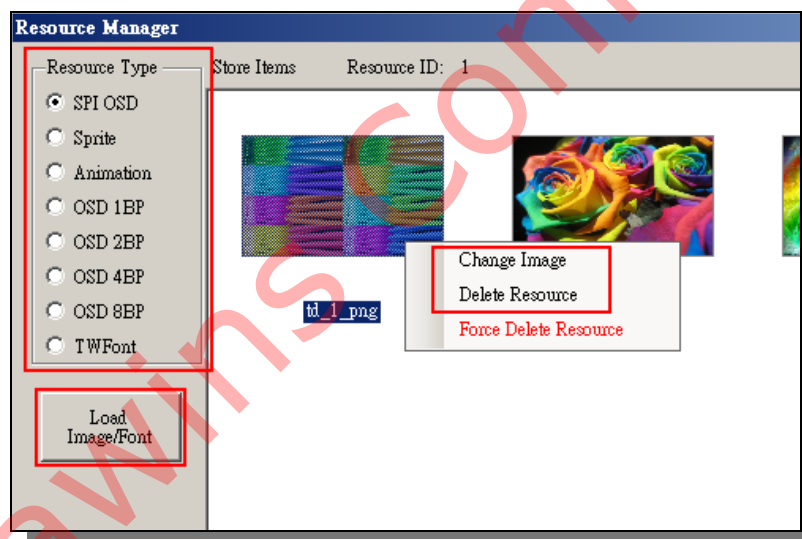


图 1-6 图资管理员界面

表 1.1 图资型态对照表

选项	说明	图层
<b>SPIOSD</b>	8 位 PNG 背景图片(image), BMP,PNG32 背景图片(twbc)	SPIOSD
<b>Sprite</b>	8 位 PNG 图片	Sprite
<b>Animation</b>	8 位 PNG 图片，使用动画效果	Sprite
<b>OSD 1BP</b>	8 位 PNG 图片，位深度 1	OSD2
<b>OSD 2BP</b>	8 位 PNG 图片，位深度 2	OSD2
<b>OSD 4BP</b>	8 位 PNG 图片，位深度 4	OSD2

## 1.4 专案架构

**Layout** 以设计 OSD 接口时之"页面"做为概念，做为项目中各种图层组合之根本。套件最后将依据此接口之设定产出程序代码，输出相同于页面数目之结构(struct)程序代码与标头(header)档。

如下图所示，项目可加入复数 **Layout**，**Layout** 则可加入复数 **Page**，且同一 **Layout** 之 **Page** 可使用 **Slide Event** 进行切换。而 **Page** 则含有单一底图(TWBC 或 PNG8)与复数之 **Sprite** 与 **OSD Menu**，并可使用 **Fade Event**。 **Sprite** 可选择使用静态之 PNG8 图片，或使用 **Animation Event** 显示动态效果。**OSD Menu** 则可加入 **ICON**、**RomFont** 与 **TwFont** 三种类型之图标，其详细说明将于后续章节介绍。

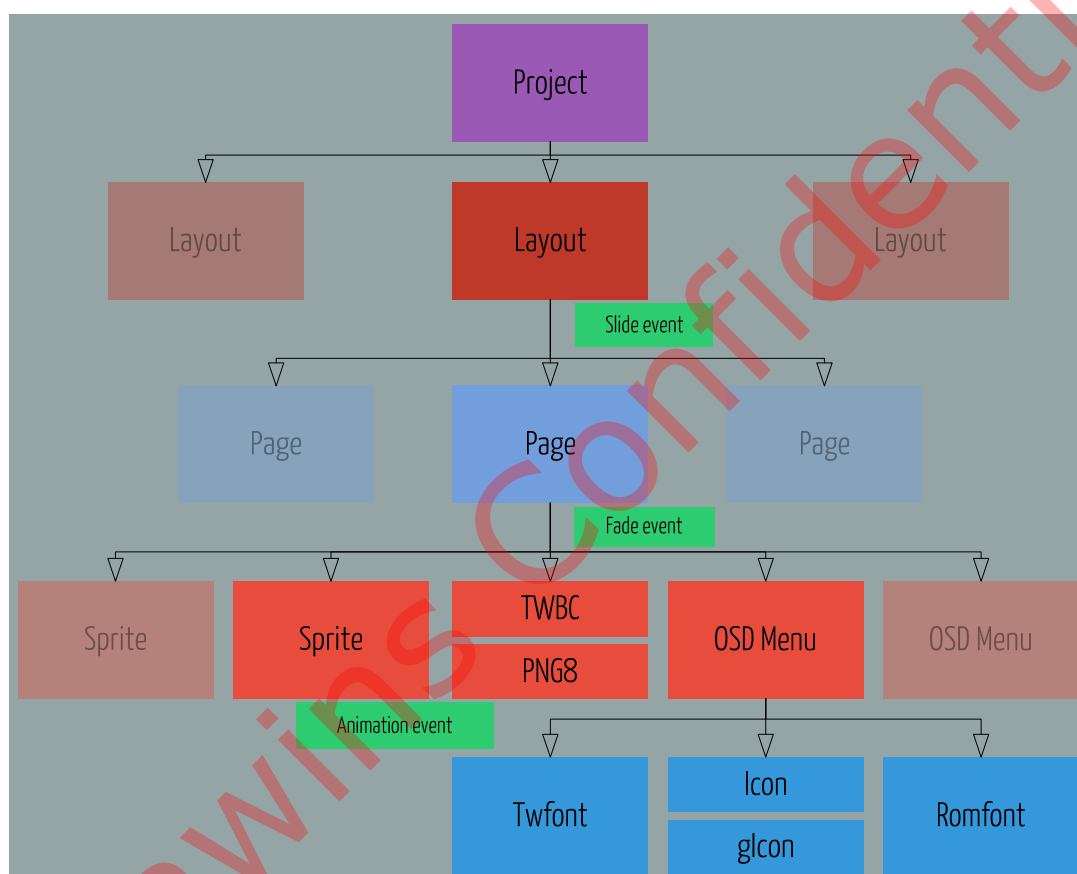


图 1-7 项目架构设定



## 1.5 Layout 版面设定

Layout 之新增方式如下图一所示，首先于主接口中编辑"Panel Info"选项，确认各图层之"画面"尺寸。第二步点击 Layout 页框中之"Edit"按钮，开启 Layout 选项，如下图二所示。第三步确认 SPIOSED 图资类型，若需使用 BMP, PNG32 格式于 SPIOSED 图层，则勾选"Compress(TWBC)"，**TWBC 为一种压缩格式，可将全彩(true color, RGB888)接近无损压缩，压缩率大约 6 (ex. 800x480x3 -> 192KB)，PNG32 (RGBA8888) 的压缩率大约 8 (ex. 800x480x4 -> 192KB)**。第四步确认页面尺寸，若须使用滑动效果时，编辑"Row"与"Column"数值以产生组合式页面(建议最大 3x3)。第五步点击"Add"按钮加入 Layout 至项目，并重复此流程至所有 Layout 皆以加入项目。最后点击"Ok"按钮完成编辑，并退出 Layout 编辑页面。

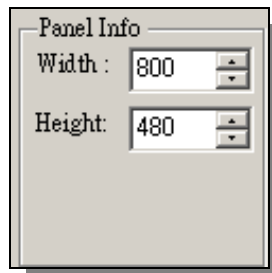


图 1-8 画面设定

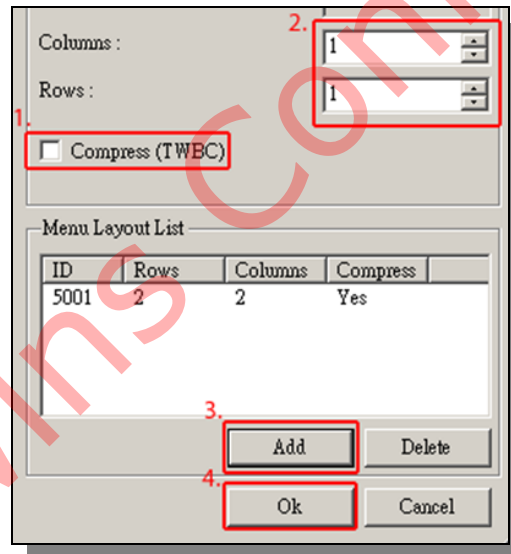


图 1-9 版面设定接口

若您需要编辑已产生之 Layout，首先于 Layout 页框中之 list 选择 Layout，并点击"Edit"按钮开启 Layout 编辑页面。接下来依据前述之流程重新设定 Layout，最后点击"Ok"确认并离开。接下来，由 Layout 页框之 list 中选择一组 Layout，开始编辑各图层之效果。

1.6 Menu SPIOSD 图层设定

此接口设定 Layout 所产生之各页面中，SPIOSD 图层将显示之图资与动画效果。

首先双击预览区域中任一页面，或由 Menu 页框中之 list 选择页面，再点击"Image Edit"进入 Menu 编辑页面。接下来点击"Load Image"按钮开启图资管理员，并点选欲使用之图资，再双击图资或点击"Select Resource"将图资载入专案。最后点击"Ok"按钮，确认并离开 Menu 编辑页面。

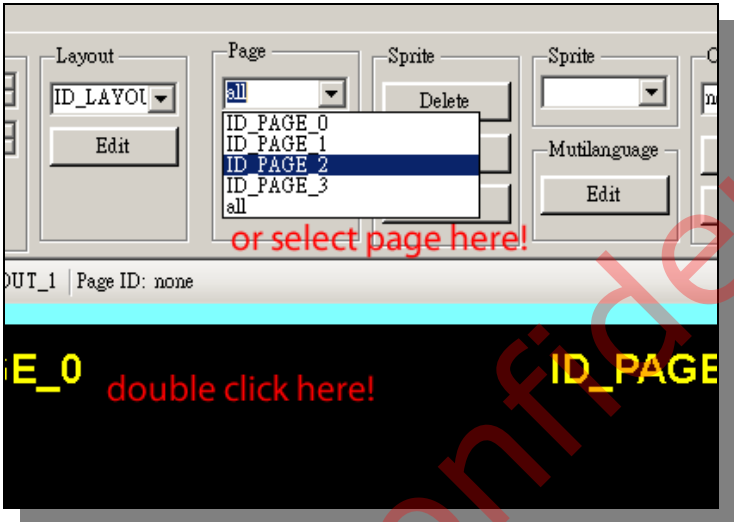


图 1-10 页面选择方式

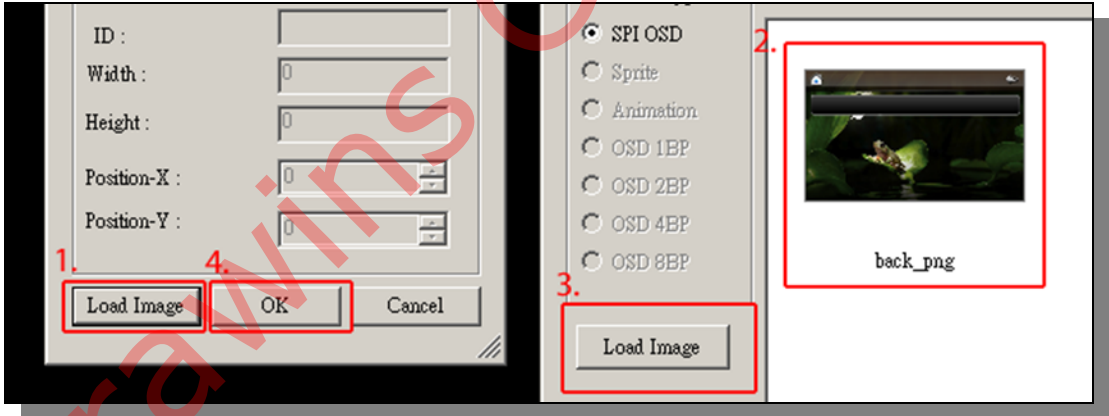


图 1-11 图资加载方式

表 1.2 事件效果对照表

事件(Event)	说明	需要组合式页面
Down	向下滑动至同一群组之其他页面	是
Left	向左滑动至同一群组之其他页面	是
Right	向右滑动至同一群组之其他页面	是
Up	向上滑动至同一群组之其他页面	是
Fade-In	画面淡入效果	否
Fade-Out	画面淡出效果	否

SPIOSD 图层之图资设定完毕后，可依设计需求设定动画效果。首先点击"Event"按钮进入 Event 编辑页面，并于"Event Select"页面中选择动画效果，详细说明如上表所示。接下来编辑"Pause Frame"之数值，设定动画之播放速度。第三步可点选"Preview"按钮，并于预览区域中确认动画效果。第四步按下"Add"加入动画效果至该页面，并重复上述步骤直至所有动画效果皆已加入页面，最后点击"OK"确认并离开 Event 编辑页面。

建议您重复上述流程直至所有页面之 SPIOSD 图层皆设定完毕，再向下进行其他图层之编辑工作。

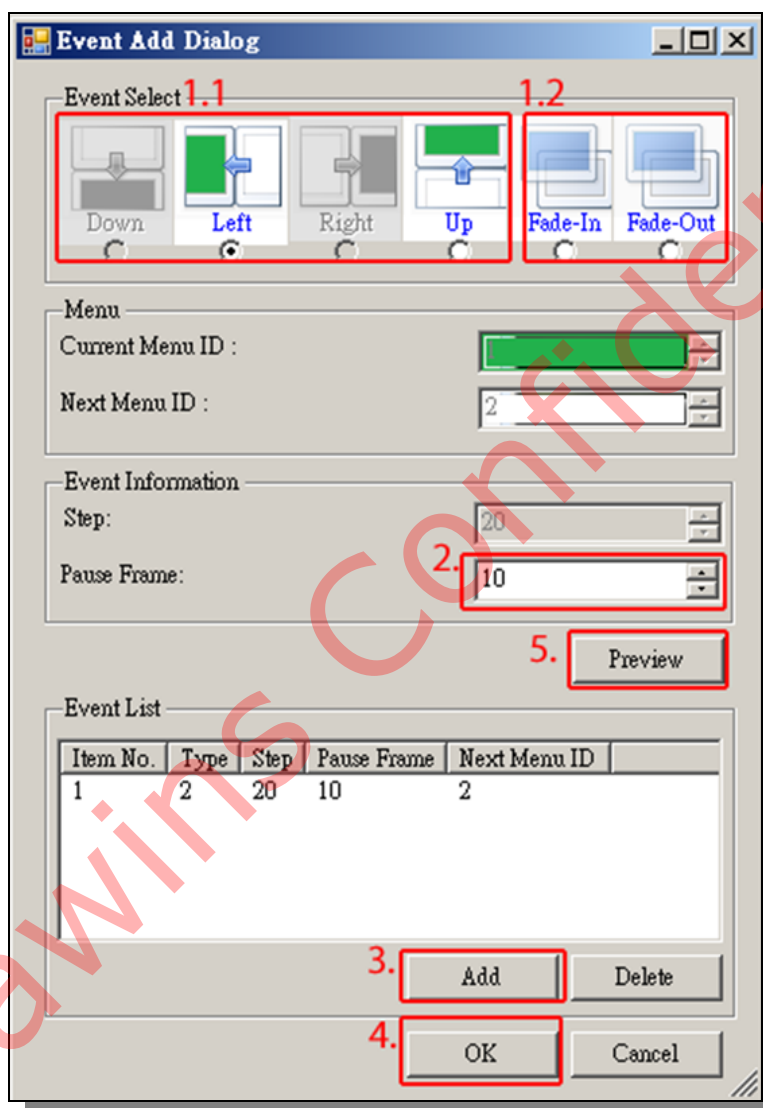


图 1-12 事件编辑接口

## 1.7 Sprite 图层设定

此接口设定 Layout 所产生之各页面中，Sprite 图层将显示之图标(Sprite Icon)与其动画效果。套件最后输出之图标结构将放置于目前选择之页面文件中，故开始编辑前请预先选择放置该图标之页面。

首先点击"Add"按钮开启 Sprite 编辑接口，再点击"Load Image"按钮，由图资管理员中选择 Sprite 图资。接下来，编辑 Location Information 页框中之"Position-X"与"Position-Y"数值，将图标(Icon)摆放至页面上之任意位置。若该图标可制作动画效果，则编辑 Sprite 页框中之"Unit Count"数值，将连环图片进行分割。接下来点击"Preview"选项则预览动画效果，点击"Stop"按钮则停止预览。若该动画效果需重复播放，则将"Sprite Repeat"选项打勾。最后点击"OK"按钮将此 Sprite 图标加入画面，并退出 Sprite 编辑接口。若需编辑图标设定，点选 Sprite 页框中之图标编号，并点击"Modify"进入 Sprite 编辑接口。反之若需删除图标，则点击"Delete"按钮将图标由画面中删除。

重复上述步骤直至所有 Sprite 图标皆已加入页面。

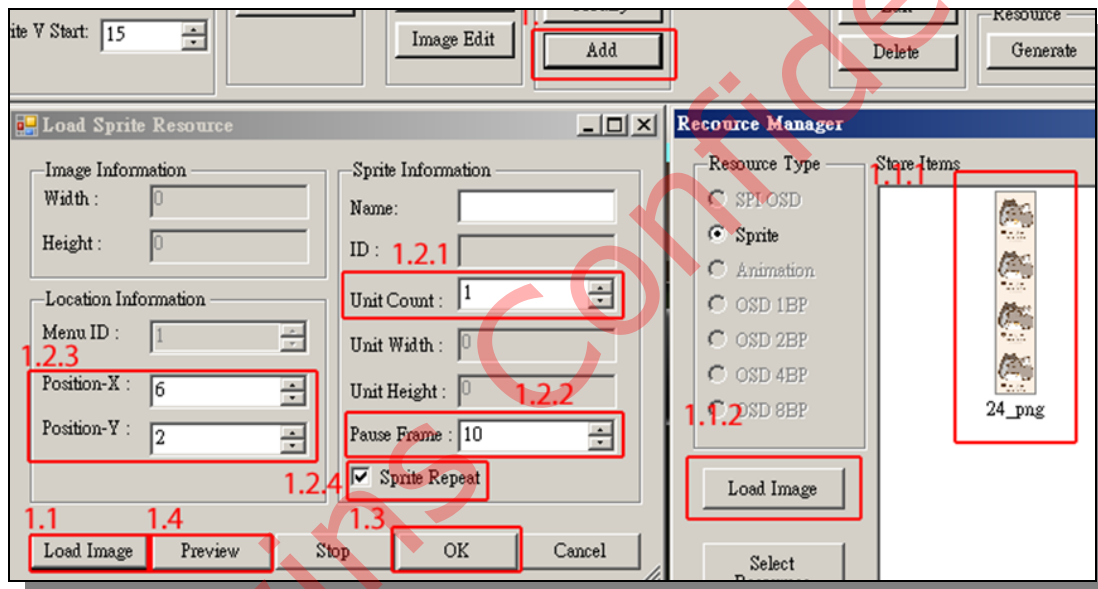


图 1-13 Sprite 编辑接口

## 1.8 OSD2 图层设定

OSD2 与其他图层最大不同之处在于“TILE”之概念与 OSD RAM 之使用，1 TILE = 16x24(默认)像素之区域，可想象为使用磁砖贴满画面。欲使用于此图层之图资，其长宽需为 TILE 之倍数，方可正常加载。加载 OSD RAM 之图资将分别构成图标(Icon)，但图标并无法单独使用。而是于加载画面(Menu)后，方可使用该画面中加载之图标。换言之，OSD2 需以画面(Menu)做为单位进行代换，故运用时需多加留意。故以下将首先介绍 OSD2 三个特殊选项 Group、Changeable 与 Optimize 之原理与使用时机，再介绍操作接口之实际使用方式。

### 1.8.1 Group

当 OSD Menu 需使用之图资量大于 OSD RAM 之容量时，若“同时”显示之图资并未超过 OSD RAM 之容量，则可使用 Group 之方式节省 OSD RAM 之使用量。若 Icon 1 与 Icon 2 为一 Group，且 Icon 1 之容量大于 Icon 2，则其占用 OSD RAM 之空间等于 Icon 2 之容量，如下图所示。

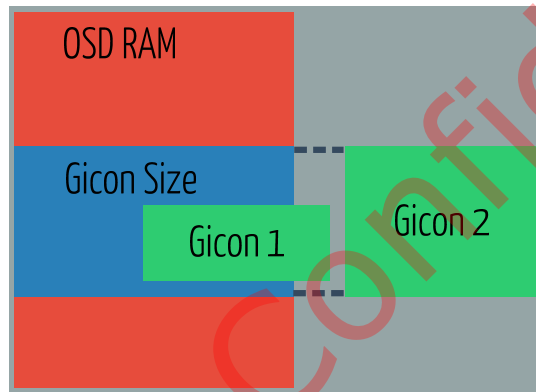


图 1-14 Group Icon 节省 OSD RAM 用量示意图

而当需显示未加载于 OSD RAM 之 Icon 时，其流程如下图所示。其与一般 Icon 相比需增加一加载 Icon 之流程，显示速度相对一般 Icon 会稍微延迟，且同样类型之 Icon 才可进行 Group。故于使用上，开发人员需谨慎考虑使用 Group 之使用时机。

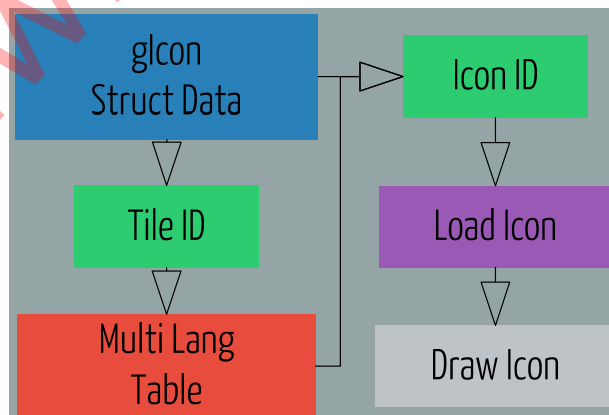


图 1-15 Group Icon 加载流程示意图

### 1.8.2 Changeable

**Changeable** 主要应用于多国语言之 **Icon** 上，当设定为不同状态时将会自动选择将加载之 **Icon**。以下图为例，当设定为 **Lang 1** 时，**changeable** 的 **Icon** 将加载 **Lang 1** 之对应 **Icon**，而设定为 **Lang 2** 时将加载 **Lang 2** 之对应 **Icon**。而其余非 **Changeable** 之 **Icon** 将加载同样之 **Icon**，而显示出对应不同语言之 **OSD** 画面。如此于语言间之切换时将可减低开发人员对于程序之额外撰写，提升开发效率。

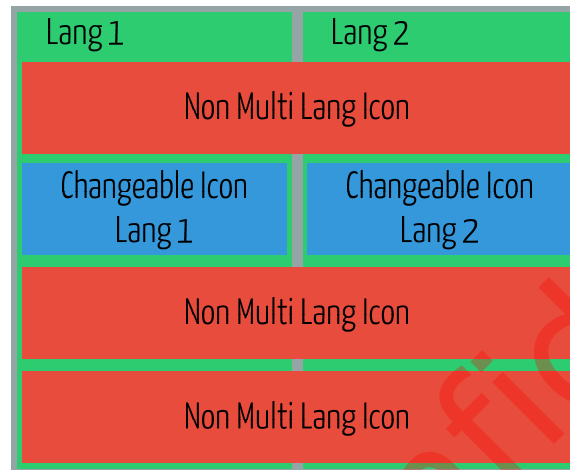


图 1-16 Changeable Icon 载入示意图

### 1.8.3 Optimize

当加载一份新 **Menu** 时，**OSD RAM** 需加载所有需要显示之 **Icon**。但 **Icon** 散落于图资各处使得 **DMA** 模块一次需搬移之数据量与 **Icon** 数量相同，当 **Icon** 数量非常多时将造成加载速度缓慢之问题。而 **Optimize** 选项将所有 **Icon** 整理后增加一备份，使 **DMA** 可发挥最大效率加快载入。但 **Optimize** 选项将会提高图资之容量，而当 **Menu** 含有 **Changeable Icon** 时，更需根据 **Set** 的数量加入备份以便载入。故开发人员在使用此选项时，请务必注意图资之容量。

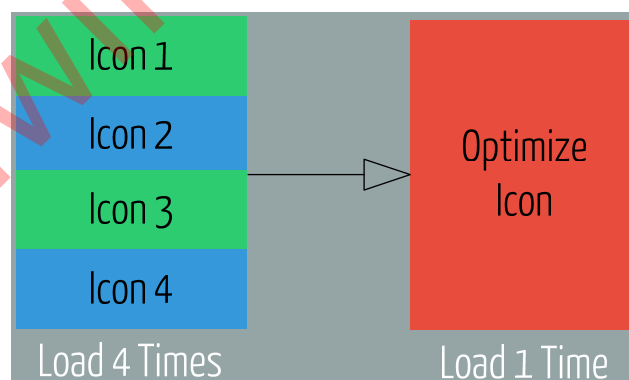


图 1-17 Optimize 搬移效率提升示意图

### 1.8.4 OSD Edit 界面

与 Sprite 相同，套件最后输出之画面与图标结构(OSD2 Menu, OSD2 Icon struct)将放置于目前选择之页面中，故开始编辑前请预先选择相应之页面。首先点击 OSD2 页框中之 list 并选择 new 选项，点击"Edit"开启 OSD2 编辑接口。开启编辑界面后，首先确认 Mode 为 icon，并根据应用编辑"Font Width"与"Font Height"。接下来点击"Load Icon"按钮开启图资管理员，并加载图资。第三步编辑 Icon 页框中之"Position-X"与"Position-Y"数值，或使用箭头按钮移动图标(icon)摆放至画面上之任意位置。若欲隐藏该图示，可取消"Visible"选项，再于程序中手动显示该图标。第四步点击"Add"按钮将图标加入该画面，并可返回第三步再次移动以重复加入同一图标。另外若于多国语言应用时需改变该图标，则将"Changeable"选项打勾，而详细流程容后再叙。

若需删除已加入之图标，于 Icon List 中点选对象后点击"Delete"按钮将其移除。而若需编辑已加入之图标，同样于 Icon List 中点选对象并编辑属性，最后点击"Modify"按钮确认编辑。欲将复数 Icon 选定为一 Group 时，直接将所有 Icon 选取并点击 Group 按钮即可；反之若需解除 Group，则点击 UnGroup 按钮。若需调整 Icon 于 ORAM 中之排序，则可使用 Icon 窗口左上角之箭头进行调整。

重复上述步骤直至所有图标加入画面，最后点击"Ok"按钮，加入此画面(Menu)至项目。而若需编辑画面设定，于 OSD2 页框之 list 中选取画面，并点击"Edit"按钮开启 OSD2 编辑接口。反之需删除该画面，则点击"Delete"按钮将其自项目中移除。

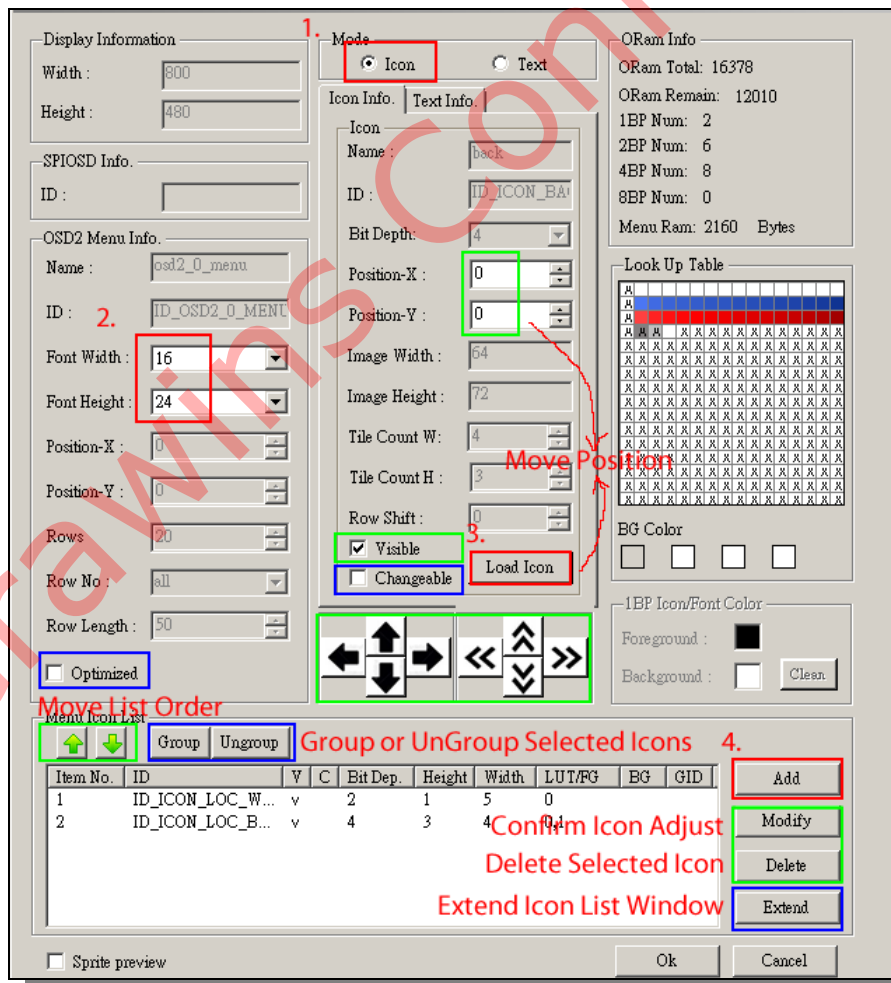


图 1-18 OSD2 编辑接口

### 1.8.5 TWFont

TWFont 为本套件为实现自选字型、可选尺寸与万国语言之需求而制作之字型库。为提供开发人员使用之弹性，故将字型库之制作功能与本套件分离。故请于 OSD 项目制作前先使用 TwFont 程序设定欲使用之字型、尺寸并进行输出，并请根据应用之特性选择不同字库选项，以节省内存空间。若对 TwFont 字库制作程序任何操作上之疑问，请参考"T582 OSD Font using TwFont.pdf"说明文件以获取更多协助。

为提升显示速度以及缩减字库容量，目前 OSD2 Font Size 仅提供 12x24、16\*31 与 16\*24 三种不同尺寸之字型。而为方便开发，TWFont 可使用图资管理员优先加载欲使用之字型，并于 OSD2 Menu 编辑时调整显示区域之尺寸并放置于画面上任意位置。但由于 TwFont 于套件中并无法进行初始化，因此画面中将以框线标示方便排版上之需求。

使用流程方面，首先调用图资管理员加载字型库，如下图所示。

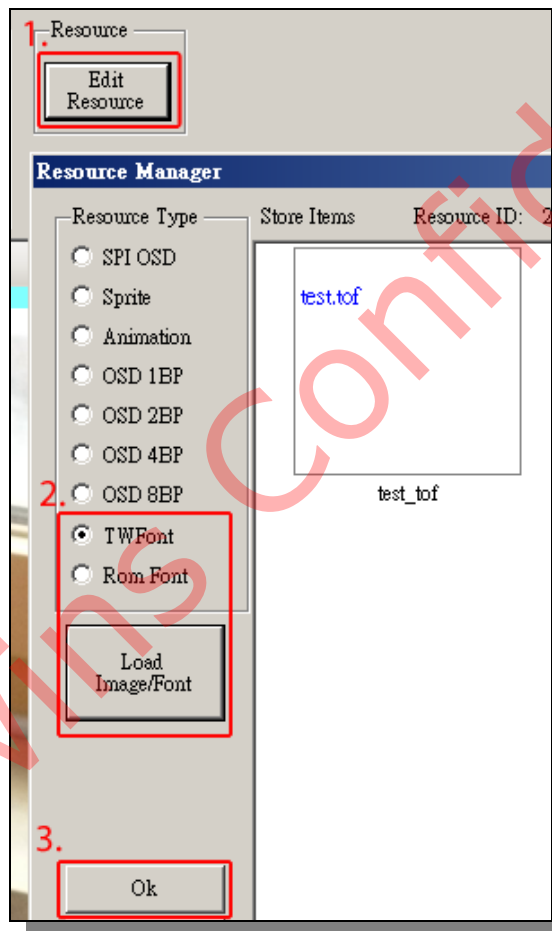


图 1-19 加载字型文件



接下来开启 OSD2 Menu 编辑接口，并于 Mode 选择 Text 模式。第三步编辑 Font Width 使其与字型文件匹配，并选择字符串的显示范围 Height 和 Width 以及选择要摆放的坐标(x, y)，另外可选择前景以及背景色(一般背景为透明色)，最后按下“Add”按钮加入即可。

套件输出文件后于 OSD2 文件中将含有 tw\_text 的结构体，在代码里透过 SDK 里的 API(tw\_text\_draw) 带入想要显示的字符串之十六位 unicode 编码，即可于 OSD2 Menu 画面中显示该字符串。

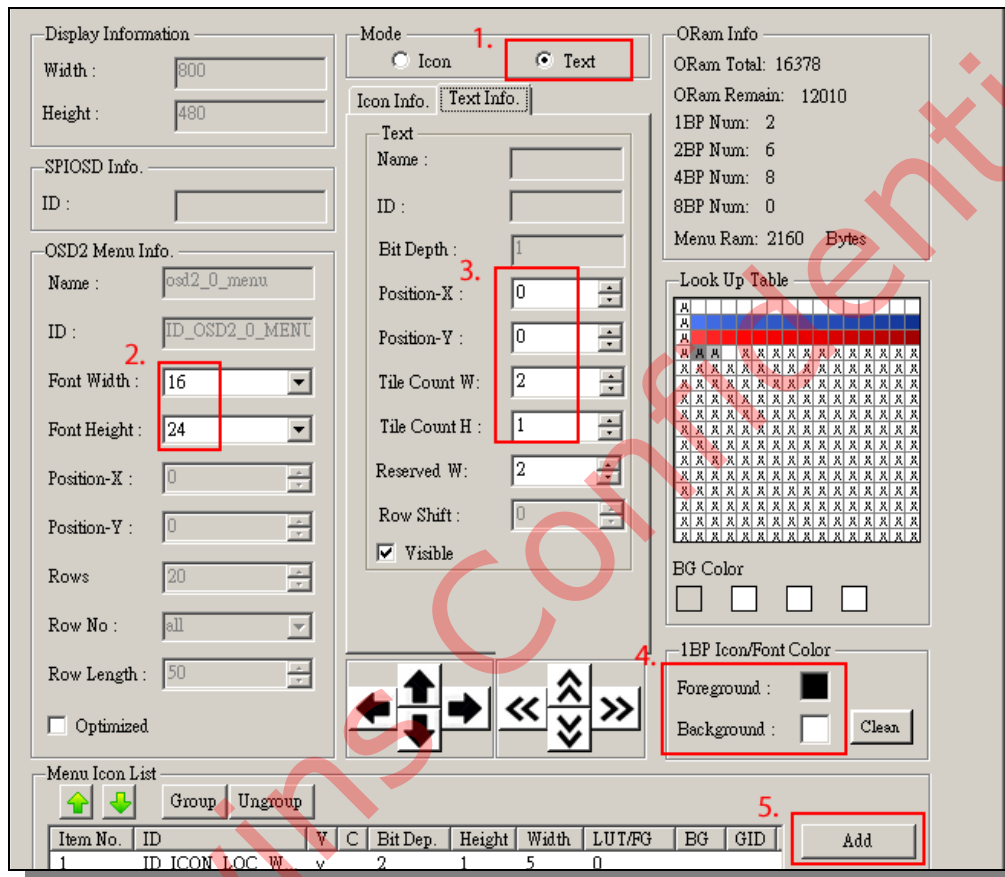


图 1-20 加入字型文件

### 1.8.6 ROM Font

ROM Font 意指 Build-in Font，也就是 T582 芯片内建的 Font。ROM Font 仅有 16x24 和 12x18 此两种大小，其中 12x18 少了最后的 6 个 Font，所以 16x24 有 120 个，而 12x18 有 114 个 Font，排列以及内容如下图三、四所示。

Tool 提供两种使用 ROM Font 的方法，字符串和 ICON。字符串的部分与 TWFont 类似，但 ROM Font 只支持 ACSII，只支持英文。ICON 的部分与 1/2/4BP ICON 的用法一样，差别只在于同样需要选择 FG 和 BG 并在 list 内挑选 Font。

图标形式的 ROM Font 在 Tool 的操作过程如下图一、二所示，首先点选"Load Icon"，并选择 Rom Font 分页，接下来点击"Load Image/Font"开启 RomFont 设定接口，选择字符串的长宽与 font 尺寸后，再于列表中选择欲显示之图标。接下来于 OSD2 编辑页面中选择 ROM Font 对象加载，并选择前景与背景颜色。最后调整位置后，点击"Add"将图标载入画面。输出后 ROM Font 与 TWFont 一样会产生 tw\_text 结构体的数据，在代码里透过 SDK 里的 API(tw\_text\_draw)带入想要显示的字符串之 ACSII，即可在 OSD2 Menu 上显示该字符串。

字符串型式的 ROM Font 与上述的操作步骤类似，但无需自列表中选择欲显示之图标，仅需设定欲显示区域并保持空白。

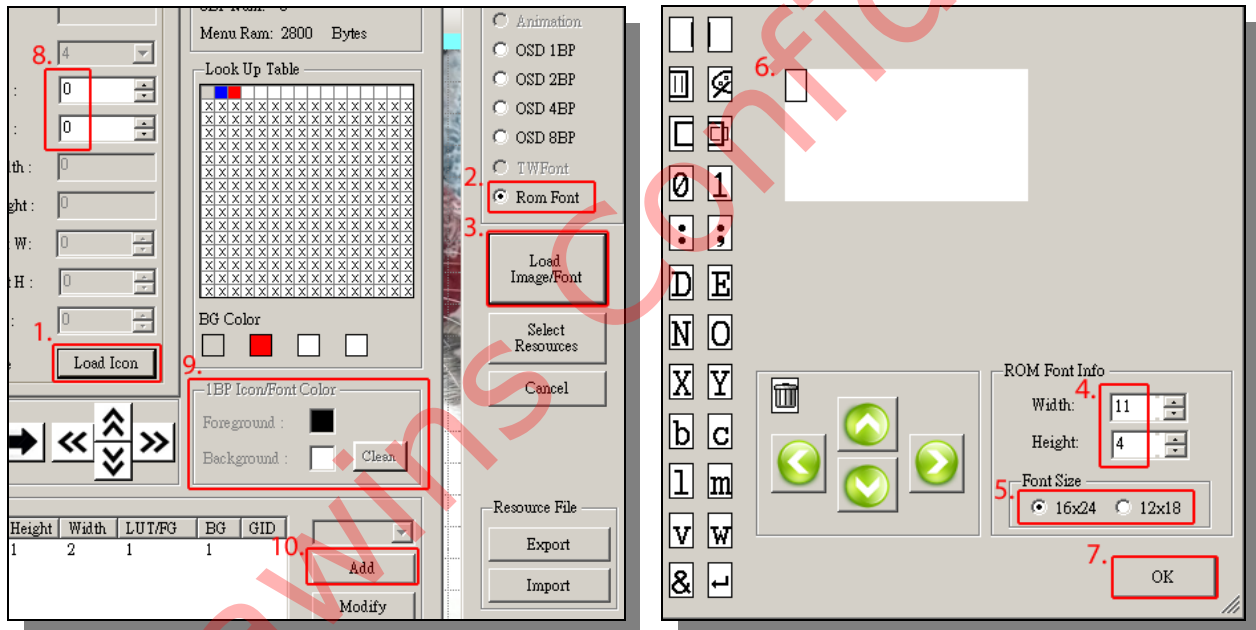


图 1-21 选择 ROM Font

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L1										...	○	✓	□	■	
L2	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
L3	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺
L4	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;
L5	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J
L6	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
L7	Z	[	\	]	^	_	'	a	b	c	d	e	f	g	h
L8	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
L9	x	y	z	←											
L10															

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L1										...	○	✓	□	■	
L2	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
L3	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺
L4	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;
L5	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J
L6	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
L7	Z	[	\	]	^	_	'	a	b	c	d	e	f	g	h
L8	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
L9	x	y	z	⓪	\$	+	%	&	←						
L10															

图 1-22 12x18(上)与 16x24(下)之 ROM Font 字型列表

### 1.8.7 Icon Reuse Example

于时钟等文字、数字应用中，常会需要将同一组数字于不同位置中重复使用。但单纯将不同位置各摆上一组图标，不仅开发人员劳心劳力、接口难以编辑，且输出文件将会尾大不掉难以辨识。故本节将以时钟做为范例进行 OSD2 制作，提供给类似应用做为参考。

首先于图资管理员中将欲使用之图标全数加载，如下图所示。接下来开启 OSD2 编辑接口，选择将显示时钟之预订位置，加载任意图标并加入。如此该组图标将带有时钟之位置信息，方便接下来于韧带专案中应用。唯需注意跳过数字 1 之原因在于该图标尺寸与其余图标相异，将造成位置错误之隐忧。第三步加载其余将显示之图标，并加入画面中任意位置。如此画面显示时，所有图标也同时载入 T582，如此方能正确显示所有图标。最后使用编辑功能，将所有动态图标自画面隐藏。如此画面加载时方能显示正常状态。



图 1-23 时钟制作范例示意图

于套件输出完毕并将文件加入韧体专案后，假设 OSD2 结构文件位于 page\_0\_struct.c，并欲于之主体函式 page\_0\_process()中显示时钟，并进行动态更新。

首先将数字图标结构根据零至九之关系排列，建立字型数组 num\_tbl。并将放置于显示位置上之图标根据前后关系排列，建立位置数组 pos\_tbl。接下来使用 SDK 提供之图标进阶绘制函数 \_icon\_draw()选择位置，并根据成是需求动态显示图标结构。详细绘制函数范例如下所示：

```

////////////////////////////////////
#include "page_0.h"
#include "tw_widget_sys.h"

struct tw_icon code *num_tbl[] = { //num font table
    &num0_png, &num1_png, &num2_png, &num3_png, &num4_png,
    &num5_png, &num6_png, &num7_png, &num8_png, &num9_png };

struct tw_icon code *pos_tbl[] = { //num position table
    &num0_png, &num2_png, &num3_png, &num4_png };

void draw_clock(unsigned char *clk_data){ //draw clock function
    unsigned char i, x;
    struct tw_icon *num;

    for(i=0;i<4;i++){ // 4 digit
        num = num_tbl[clk_data[i]]; //choose font from data

        if(clk_data[i] == 1) //icon 1 size different, need patch pos
            x = pos_tbl[i]->x + 1;
        else //set postion
            x = pos_tbl[i]->x;

        _icon_draw( num->index, num->color, x, //draw num
            pos_tbl[i]->y, num->width, num->height, 0);
    }
}
////////////////////////////////////

```

时钟绘制函式完成后，打开主体函数 page\_0\_process()并进行编辑。首先将 OSD2 画面加载 T582，再使用绘制函式将图标绘至于画面，最后将画面开启即可显示时钟画面。若需更新时钟状态，只需修改数据数组 clk\_data[]并再次调用绘制函数，即可更新画面。

```

////////////////////////////////////
#include "spiosd.h"
#include "osd2api.h"

unsigned char page_0_process (TW_EVENT* event)
{
    unsigned char clk_data[] = {2, 3, 1, 8};

    switch(event->type)
    {
        case TW_EVENT_TYPE_ENTER_SYSTEM:
            tw_menu_draw((struct tw_menu*)&osd2_1_menu); //draw osd2 menu
            draw_clock(clk_data); //draw clock
            osd_enable();

            clk_data[3] = 9; //change number
            draw_clock(clk_data); //update clock
            break;

        case TW_EVENT_TYPE_QUIT_SYSTEM:
            break;

        default: /* put to modify if no process this event */
            return TW_RETURN_NO_PROCESS;
    }
    return TW_RETURN_NONE;
}
////////////////////////////////////

```

1.8.8 Multi-Language

为应对多国语言功能应用中需制作复数 OSD2 画面对应多种语言，以致开发人员劳心劳神之窘境。故本套件以图标配置文件之方式复数替换图标，以不破坏页面架构之模式完成多国语言之开发。**目前此功能仅支持 OSD2 图层，并务必于需替换图标加入画面时勾选 changeable 属性。**

首先点击主画面中"Multi-Language"选项之"Edit"按钮，开启"MultilconEdit"管理接口。接下来，于"Select Set"选项之 list 直接点击右键，选择"Add Set"加入一组新设定。下方列表中左侧字段显示"default Set"中所设定之图标，右侧则可根据该设定需求进行替换。點選右侧图标后，再点击 Change Image 按钮。于图资管理员中选择图资进行替代，唯需注意选择图资时其尺寸需完全相同，否则系统将会阻止图资替换之动作。

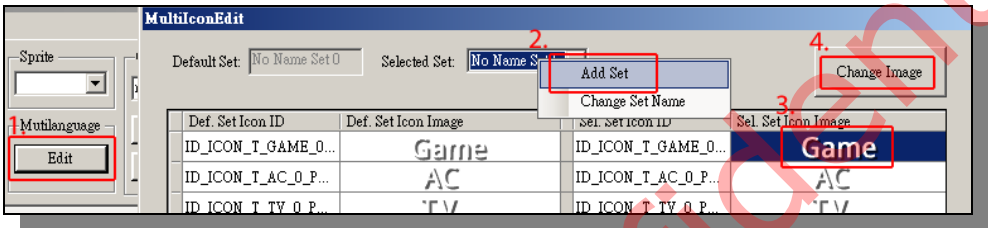


图 1-24

依序将表列图标替换完毕后，选择"Change Set Name"为设定进行更名。首先于"Select Set"选项之 list 直接点击右键，选择"Change Set Name"进行更名。并于完成后点击"Accept"按钮进行确认。按照应用中需求完成各项配置文件后，点击"OK"按钮退出编辑接口。

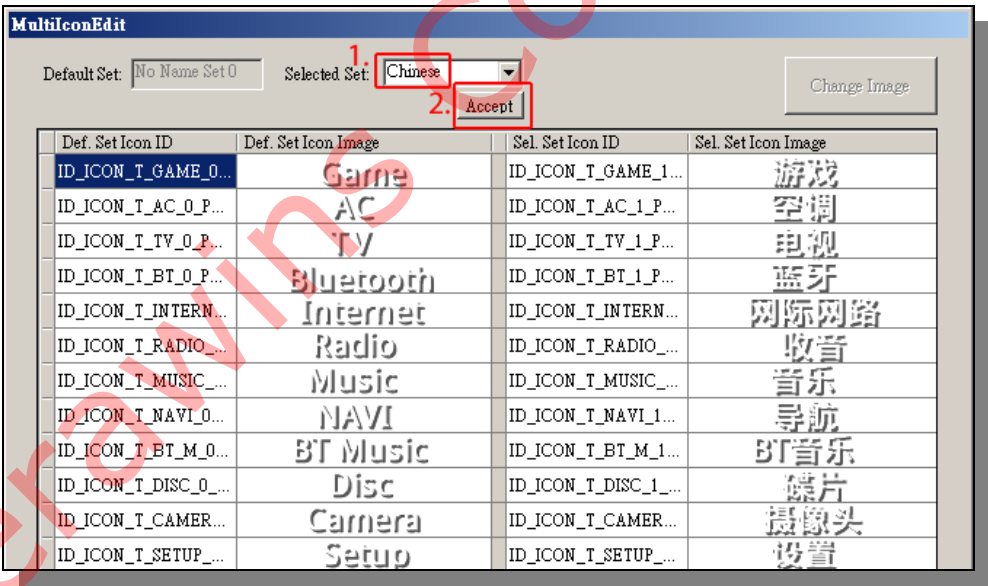


图 1-25

## 1.9 Project 项目管理

此接口提供项目之储存、加载与新增功能，加载与新增时**当前项目将会自动关闭**，使用时请多加留意。

于 Project 页框中点击"Open"按钮，开启文件管理器并选择专案档(\*.eosd)开启专案。

点击"Save"按钮，开启文件管理器并编辑档名储存项目。

点击"New"按钮，开启一份新专案。



图 1-26 项目管理接口

## 1.10 Resource 图资输出

此接口将输出项目使用图资之格式化二进制档案(thetis\_res.bin)，同时产生对应该二进制档案之 C 语言文件(C File)与头文件(Header File)。

首先点击主接口之"Generate"按钮，开启文件管理器接口。接下来选择欲储存输出数据之目录，此步骤建议创建一新目录，避免图资混淆之情形发生。最后点击"Start"按钮开始输出档案，待程序转换之进度指示条结束后，即输出完毕。

若有勾选 TWBC(建议)，由于是进行高压压缩率压缩，因此需耗费较长的时间，选用 TWBC 格式之底图越多，相应所需的时间也越久，此为正常现象毋须担忧。

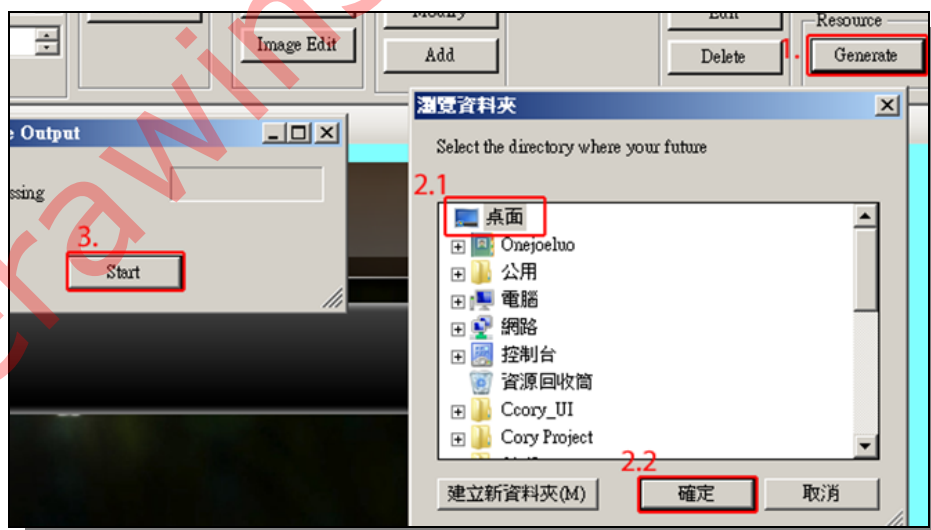


图 1-27 图资输出流程



## 1.11 输出档案

输出之档案中含有 Source, Resource, i51, part, exMCU, back\_track\_info 六个文件夹，而 T582 项目需要使用之信息储存于 Source, Resource, i51 三个文件夹中。其中 Resource 文件夹含有 res.h 与 t582\_res.bin 两个档案，分别为图资编号之定义文件与图资档案。而 i51 文件夹含有页面与图资结构之定义文件，当项目更新并输出后，上述两个文件夹中之档案需再次覆盖进韧体专案，**故强烈建议开发人员勿修改其内容，以免增加开发难度。**而 Source 文件夹中包含了欲忽略页面之定义文件，与各页面之主循环文件。与上述相异处在于只有在项目加入新的页面时，才需将此文件夹新增之页面文件加入至项目。

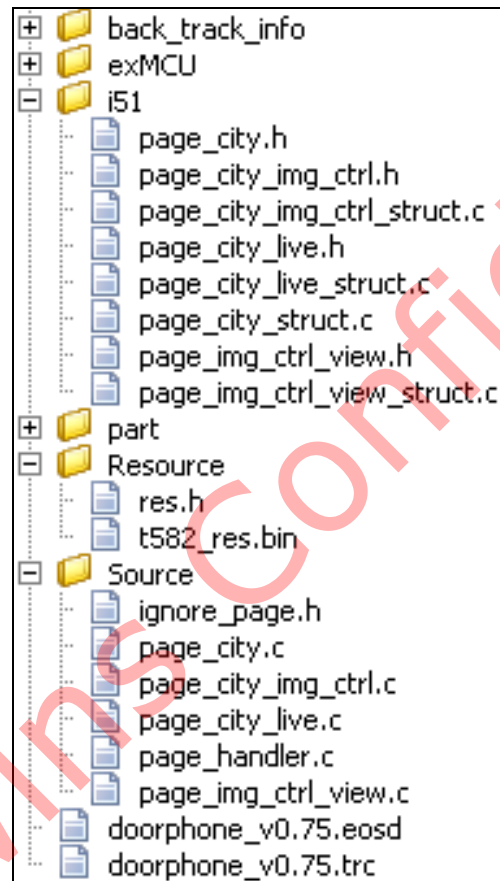


图 1-28 图资输出目录示意图