

deweb 开发指南

1. deweb 介绍

deweb 是一个可以直接将 Delphi 程序快速转换为网页应用的工具！

使用 deweb，开发者不需要学习 HTML、JavaScript、Java、PHP、ASP、C#等新知识，用 Delphi 搞定一切。

deweb 开发的网页支持所有客户端，包括手机、平板等。

网站: <http://www.delphibbs.com>

2. 支持控件

- 支持所有数据连接组件(ADOConnection/FDConnection/UniConnection)等;
- 支持所有数据表组件 (Query/Table) ;
- 支持所有三层中间件;
- 支持 BitBtn(用于选择拟上传的文件/图像/视频)
- 支持 Button
- 支持 Chart
- 支持 CheckBox
- 支持 ComboBox
- 支持 DateTimePicker
- 支持 Edit
- 支持 Image
- 支持 Label
- 支持 ListBox
- 支持 MainMenu
- 支持 Memo
- 支持 Panel
- 支持 PageControl
- 支持 RadioButton
- 支持 SpinEdit
- 支持 Shape(用于扫描二维码)
- 支持 SpeedButton
- 支持 StringGrid
- 支持 Timer
- 支持 TreeView

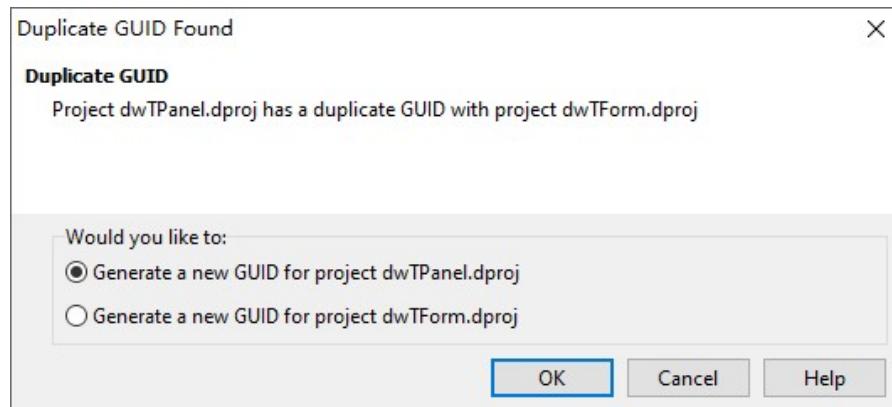
3. deweb 入门

- 开发版本

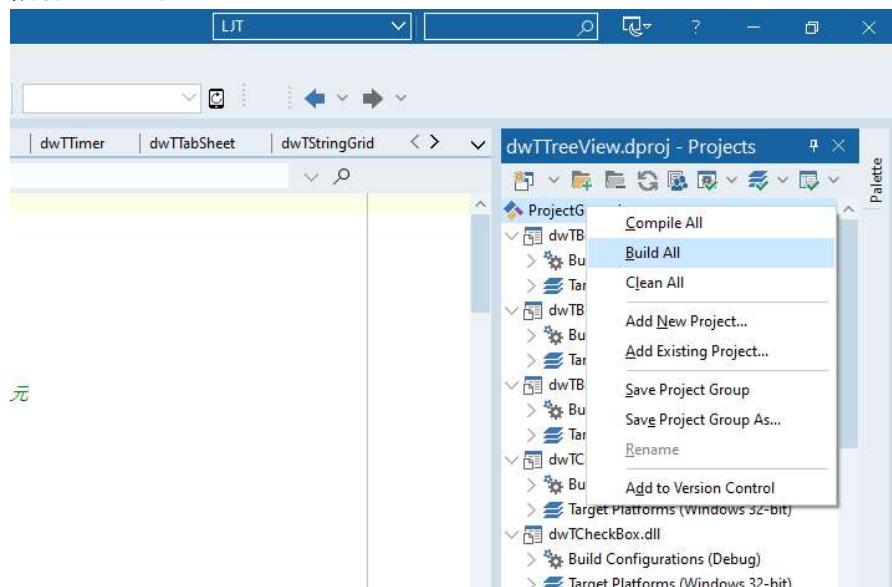
- 目前支持 Delphi 10.1/10.2/10.3/10.4。
- deweb 默认使用 Delphi 10.4.2 编译，推荐使用此版本开发！

- 第一次编译前准备

- 打开 Source\dwVcls\ AllVcls.groupproj
如果遇到以下情况，选择“OK”按钮



- 然后 Build all;



- 打开 Source\dwThirdVcls\ 中所有的 dpr, 然后 Build all
- 复制对应的服务器程序(仅 Delphi 版本非 10.4.x 的需要)。
先删除 dewebServer.exe,
如果是 D10.1.x, 则重命名 dewebServer101.exe 为 dewebServer.exe;
如果是 D10.2.x, 则重命名 dewebServer102.exe 为 dewebServer.exe;
如果是 D10.3.x, 则重命名 dewebServer103.exe 为 dewebServer.exe;

- 入门教程 hello,world!

- 新建一个 DLL, 命名为 hello1.dpr
- 新增一个 Form.(File->New->VCL Form - Delphi)。注：不要更改单元名称和 Form 名称，即分别为 unit1.pas 和 Form1
- 在 Form1 中添加一个 Label, 设置其 Caption 为"Hello,deweb"
- 将 hello1.dpr 源文件代码更改为

```
library hello1;
```

```
uses
  ShareMem,
  SysUtils,
  Forms,
  Messages,
  StdCtrls,
  Variants,
  Windows,
  Classes,
  Data.Win.ADODB,
  unit1 in 'unit1.pas' {Form1};
```

```
{$R *.res}
```

```
var
  DLLApp      : TApplication;
  DLLScreen   : TScreen;

function dwLoad(AParams:String;AConnection:TADOConnection;AApp:TApplication;AScreen:TScreen):TForm;stdcall;
begin
  //
  Application := AApp;
  Screen     := AScreen;
  //

  //
  Form1      := TForm1.Create(nil);
  //Form1.ADOQuery1.Connection := AConnection;

  Result     := Form1;
end;

procedure DLLUnloadProc(dwReason: DWORD);
begin
```

```

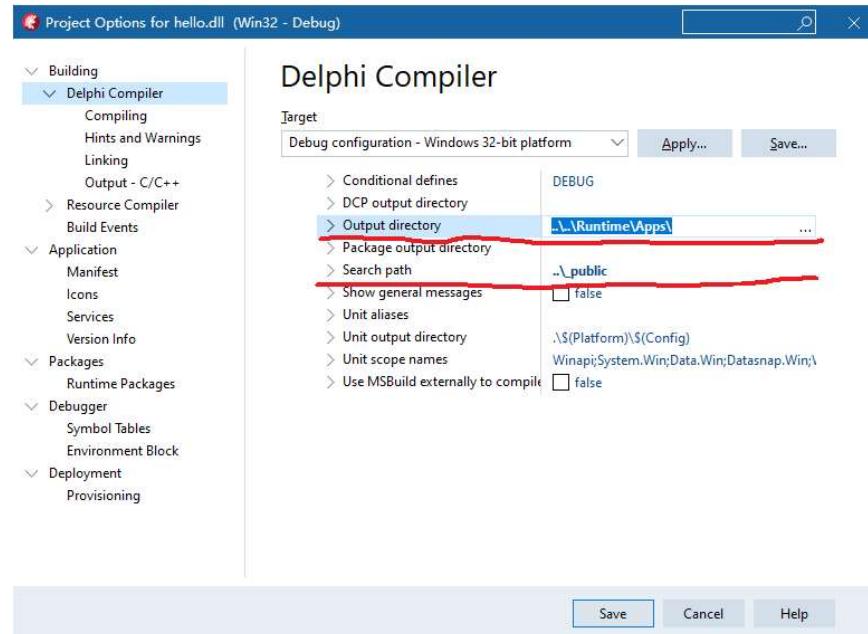
if dwReason = DLL_PROCESS_DETACH then begin
    Application := DLLApp; //恢复
    Screen := DLLScreen;
end;
end;

exports
dwLoad;

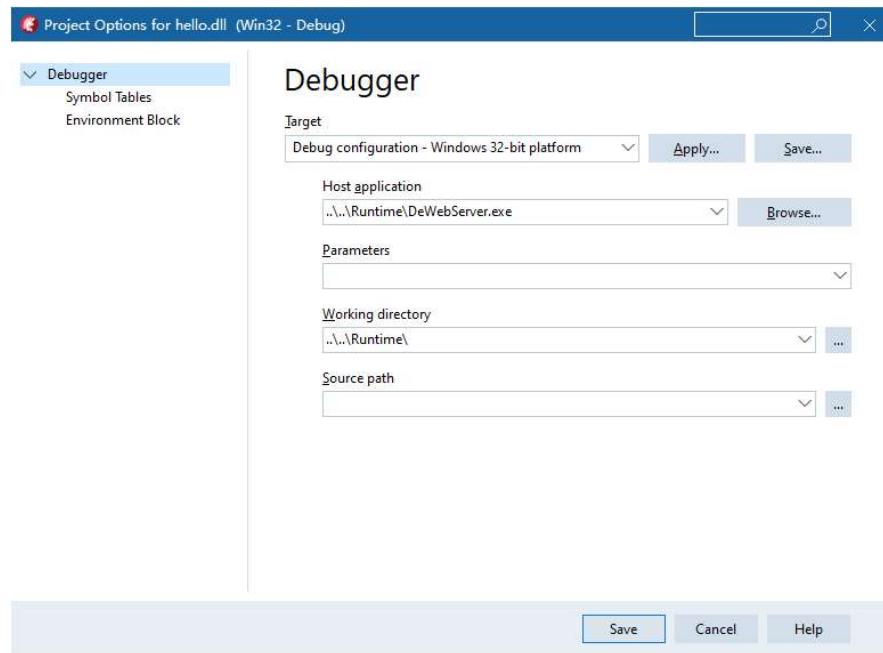
begin
    DLLApp := Application; //保存 DLL 中初始的 Application 对象
    DLLScreen := Screen;
    DLLProc := @DLLUnloadProc; //保证 DLL 卸载时恢复原来的 Application
    DLLUnloadProc(DLL_PROCESS_DETACH);
end.

```

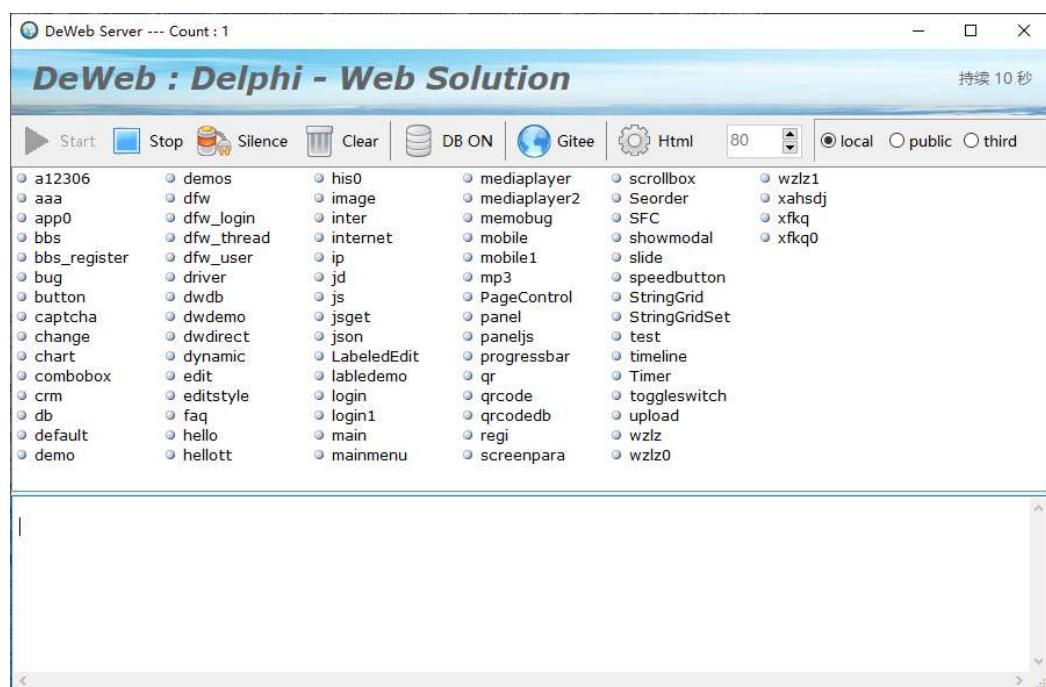
- 保存到 source\demos\hello1 目录中
- 更改 Project 设置为



- 设置运行参数
菜单:Run – Parameters



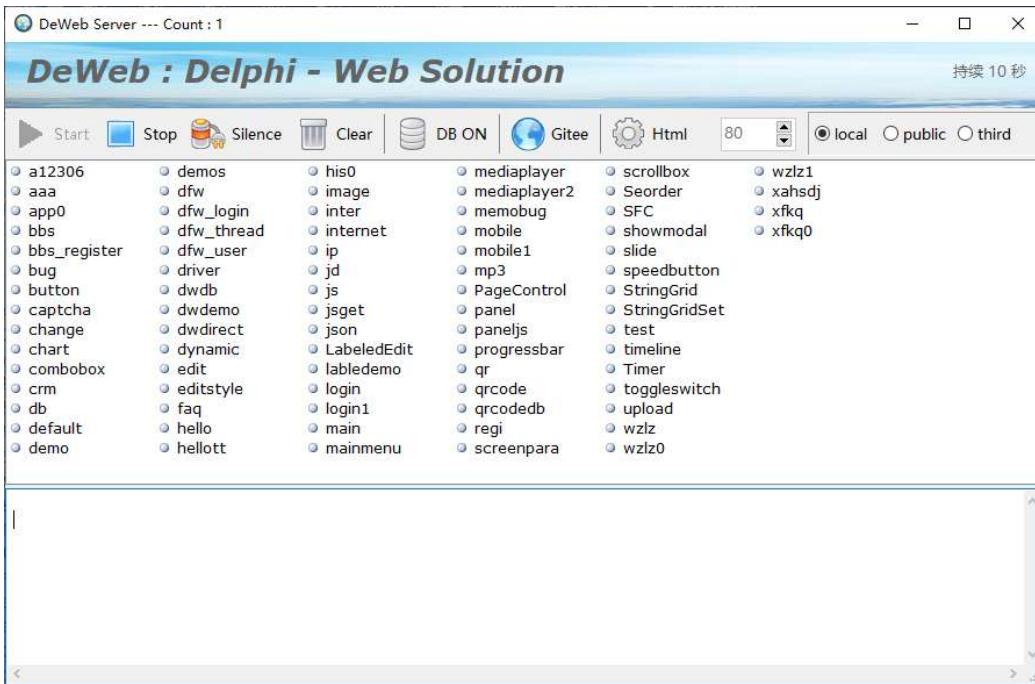
- 编译运行，即可以直接打开服务器程序 dewebsrvr.exe



- 双击 hello1,即可打开当前应用

4. 服务程序

- 服务程序为 dewanServer.exe
双击直接运行
- 带参数命令行运行
如果 1 个参数，则为运行端口号，如：dewanServer 8081
如果 3 个参数，则分别为端口号、窗体 Left、窗体 Top
如果 5 个参数，则分别为端口号、窗体 Left、窗体 Top、窗体 Width、窗体 Height
- 如果点击 start 按钮后，start 按钮仍处于可用状态，则表明当前端口已被占用，请更换一个端口号或关闭使用该端口的程序后重新 start。



其中，

- 按钮 Start 用于启动 dewan 服务；
- 按钮 Stop 用于停止 dewan 服务；
- 按钮 Silence 用于系统调试，点击切换为 Debug，可以查看客户端与服务器交互信息
- 按钮 Clear 用于清空交互信息
- 按钮 DB ON/DB OFF 用于显示当前公用数据库连接状态。公用数据库连接字符串在配置文件中
- 按钮 Gitee 用于直接打开 dewan 在码云 (<https://www.gitee.com>) 的链接
- 按钮 Html 用于生成可离线使用的同名 Html 文件（需要在工程中设置 Form1 的 AlphaBlend=True）。
- local/public/third 用于设置系统采用的 js/css 源，分别为本地、公用和第三方。一般建议为本地，如果部署在互联网，建议采用 CDN 加速；如果采用第三方，请先在配置文件中进行设置。

5. 配置文件

dewebServer 的配置文件为同目录下的 config.json 文件，基本如下

```
{  
    "port": 80,  
    "resource": 0,  
    "third": "http://cdn.delphibbs.com",  
    "redir": [  
        ["mm","hello.dw"],  
        ["gg","login.dw"]  
    ],  
    "upload_dir":"upload",  
    "max_upload_size":60,  
    "connectionstring": "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=deweb.mdb"  
}
```

其中：

port：整型，为默认启动时的端口；

resource：整型，为启动时资源文件的位置。0 表示本地源，1 表示公用源，2 表示第三方源；

third：字符串型，和 resource 为 2 时配合使用，为第三方源网址；

redir：字符串二维数组，为自动地址切换表，每组由 2 个字符串组成。如：["mm", "hello.dw"]；

当访问 http://127.0.0.1/mm 的时候，dewebServer 会自动打开 http://127.0.0.1/hello.dw；

upload_dir：为上传文件存放目录；

max_upload_size 为上传文件最大 size，单位为 MB

connectionstring：字符串型，为公用数据库 ADOConnection 的连接字符串。

6. 部署

部署时需要 runtime 中的大部分文件

需要的目录有：

- apps，仅包括需要部署的 dll 即可
- dist，必须。请勿改动
- media，非必须，一般媒体文件存在于此目录
- upload，必须，接收上传文件存放目录
- vcls，必须，包括支持的控件，可以删除未用于的控件
- zxing_files，非必须，仅当使用扫码识别时用到

需要的文件有：

- dewebServer.exe 或 dewebServer_1042.exe 等，必须！
- borlndmm.dll，必须！
- config.json，必须！配置文件，可以手动修改
- libcrypto-1_1.dll，必须！
- libeay32.dll，必须！
- libssl-1_1.dll，必须！
- sqlite3.dll，非必须！仅使用 sqlite 数据库时使用

- ssleay32.dll, 必须!

7. 应用基本设计

- 为了尽量和网络显示一致, 建议在 Form 默认字体为"微软雅黑, 11 号", BorderStyle = bsNone;
- 大部分控件的 Hint 用来存放额外的信息, 请勿使用;
- 不能使用 ShowMessage/MessageDlg/InputQuery 等所有模态窗体;
- 所有窗体不能 Show/ShowModal;

8. 数据库连接

- 数据库连接可以在 Form 中创建
- 也可以在 dewanServer 中的配置文件中设置数据库连接, 然后使用函数 dwLoad 中的参数 AConnection, 即在 dwLoad 函数中加入类似代码:
AForm.ADOQuery1.Connection := AConnection;

9. 纯数据交互 (post/get)

dewan 支持通过 post/get 进行纯数据交互

- 使用方法, 类似:
<https://delphibbs.com/dwdirect.dll?name=westwind&age=18>
- 例程 dwdirect 位于 source\demos\dwdirect
关键函数如下:

```
function dwDirectInteraction(AData:PWideChar):PWideChar;stdcall;
function _Get(AParName:String):String;
var
    iPos      : Integer;      //位置,类似:name=
begin
    Result    := AData;
    //删除以前的
    iPos      := Pos(AParName+'=',Result);
    Delete(Result,1,iPos+Length(AParName));
    //删除&后面的内容
    iPos      := Pos('&',Result);
    if iPos>0 then begin
        Result    := Copy(Result,1,iPos-1);
    end;
end;
begin
    Result :=PWideChar(Format(' Name is %s Age is %s',[_Get('name'),_Get('age')]));
end;
```

- 注: 此功能支持其他语言混合编程

10. 特殊函数

- 打开一个新页面请使用

```
dwOpenUrl(self,'/discuss.dw','_self'); //当前页面打开  
dwOpenUrl(self,'/plan.dw','_blank'); //新建页面打开
```

- **ShowMessage** 采用 dwShowMessage 代替

```
procedure dwShowMessage(AMsg:String;AForm:TForm);
```

其中:

- AMsg: 消息内容
- AForm: 当前窗体, 一般可以使用 self

例: dwShowMessage('Welecome to deweb!',Self);

如果需要自定义标题和按钮标题, 则可以用 dwShowMsg 函数

```
procedure dwShowMsg(AMsg,ACaption,AButtonCaption:String;AForm:TForm);
```

其中:

- AMsg: 消息内容
- ACaption: 消息框标题
- AButtonCaption: 按钮标题
- AForm: 当前窗体, 一般可以使用 self

例: dwShowMessage('Welecome to deweb!','deweb Demo','确定',Self);

- 确定取消的 **MessageDlg** 采用 dwMessageDlg 代替

```
procedure
```

```
dwMessageDlg(AMsg,ACaption,confirmButtonCaption,	cancelButtonCaption,AMethodName:String;AForm:TForm);
```

其中:

- AMsg: 消息内容
- ACaption: 消息框标题
- confirmButtonCaption: 确定按钮标题
- cancelButtonTitle: 取消按钮标题
- AMethodName: 返回结果后触发 Form 的 OnStartDock 中的事件标志
- AForm: 当前窗体, 一般可以使用 self

dwMessageDlg 没返回值, 所以不能在当前函数/过程中处理, 需要等结果返回后会自动激活 Form 的 OnStartDock 函数,

所以需要在 Form 的 OnStartDock 函数中添加事件, 在事件中通过以下取得前期拟触发的事件标志

```
dwGetProp(Self,'interactionmethod');
```

通过类似以下得到返回的结果, 为'1'时表示为"确定", 否则为"取消"

```
dwGetProp(Self,'interactionvalue');
```

- **InputQuery** 采用 dwInputQuery 代替

```
procedure
```

```
dwInputQuery(AMsg,ACaption,ADefault,confirmButtonCaption,	cancelButtonCaption,AMethodName:String;AForm:TForm);
```

其中:

- AMsg: 消息内容

- ACaption: 消息框标题
- ADefault: 输入框的默认值
- confirmButtonCaption: 确定按钮标题
- cancelButtonCaption: 取消按钮标题
- AMethodName: 返回结果后触发 Form 的 OnStartDock 中的事件标志
- AForm: 当前窗体, 一般可以使用 self

dwInputQuery 没返回值, 所以不能在当前函数/过程中处理, 需要等结果返回后会自动激活 Form 的 OnStartDock 函数,

所以需要在 Form 的 OnStartDock 函数中添加事件, 在事件中通过以下取得前期拟触发的事件标志

```
dwGetProp(Self,'interactionmethod');
```

通过类似以下得到返回的结果, 为输入的内容。

```
dwGetProp(Self,'interactionvalue');
```

- **dwOpenUrl** 打开网页

```
procedure dwOpenUrl(AForm:TForm;AUrl:String;AParams:String);
```

AForm:TForm 当前窗体, 一般用 self

AUrl: String, 网址

AParams: String; '_blank'是新窗口打开, '_self'是当前窗口打开

- **dwSetCookie** 设置 Cookie

```
function dwSetCookie(AForm:TForm;AName,AValue:String;AExpireHours:Double):Integer
```

AForm:TForm 当前窗体, 一般用 self

AName: String, Cookie 名称

AValue: String; Cookie 值

AExpireHours:Double 有效期 (小时)

- **dwGetCookie** 读取 Cookie 值

```
function dwGetCookie(AForm:TForm;AName:String):String;
```

AForm:TForm 当前窗体, 一般用 self

AName: String, Cookie 名称

- **dwRunJS** 运行 javascript 代码

例: alert('hello,world!');

原生的弹框函数

例: this.Button1_cap='btn1';this.Label1_lef='20px';

设置按钮 Button1 的 Caption 为“btn1”,设置 Lable1 的 Left 为 20

- **用 Javascript 向 Form 发送消息 (传递数据)**

可以在 javascript 中执行以下代码:

```
let me=this;axios.get({"m":"interaction","i":'+this.clientid+',"t":"varname","v":"varvalue"})then(resp=>{me.procResp(resp.data)});
```

以上代码中加下划线的部分可以根据需要自行设置 (其他部分尽量不要修改)

varname 一般用作拟传递变量的名称;

varvalue 一般用作拟传递变量的值。

执行以上代码后, 会自动激活 Form 的 OnStartDock 函数,

所以需要在 Form 的 OnStartDock 函数中添加事件, 在事件中通过以下取得前期拟触发变量的名称

```
dwGetProp(Self,'interactionmethod');
```

通过类似以下得到拟传递变量的值。

```
dwGetProp(Self,'interactionvalue');
```

- **dwGetMobileAvailHeight** 取得移动端实际可用高度函数

可以通过类似以下代码得到

```
Height := dwGetMobileAvailHeight(self);
```

- **dwSetMobileMode** 设置当前屏幕显示模式为移动应用模式

```
function dwSetMobileMode(AForm:TForm;ADefaultWidth,ADefaultHeight:Integer):Integer;
```

其中:

ADefaultWidth,ADefaultHeight 为电脑浏览时的默认大小。如果有一项为 0, 则为当前屏幕大小 (只设置宽度, 不设置高度)

11. 开发 dewan 控件

开发控件非常简单，只需要开发一个适合需要的 DLL 即可。

注意几点：

- 1) 命名：工程名必须以“dw”+“控件类名”，如：dwTButton.dpr, dwTEdit.dpr 等；
- 2) 源码存放目录：Source\dwVcls
- 3) 工程输出目录：Source\RunTime\Vcls\
- 4) uses 的第 1 个单元必须是 sharemem；
- 5) 包含以下函数，分别为：

```
function dwGetExtra(ACtrl:TComponent):String;stdCall;
function dwGetEvent(ACtrl:TComponent;AData:String):String;StdCall;
function dwGetHead(ACtrl:TComponent):String;StdCall;
function dwGetTail(ACtrl:TComponent):String;StdCall;
function dwGetData(ACtrl:TComponent):String;StdCall;
function dwGetMethod(ACtrl:TComponent):String;StdCall;
```

- 6) dwGetExtra 用于在 html 中增加对第三方网页控件的支持。

目前 dewan 中默认采用 element-ui 进行处理。

如果使用 element-ui 中的控件，则代码如下（返回一个空的 JSON 字符串数组）：

```
Result := '[]';
```

如果 element-ui 不存在某控件，比如 TChart，就需要引入第三方 UI 库。

在 dwGetExtra 中可以增加以下代码。

```
Result := '['<script src=\"dist/charts/echarts.min.js\"></script>','
+ "<script src=\"dist/charts/lib/index.min.js\"></script>",
+ "<link rel=\"stylesheet\" href=\"dist/charts/lib/style.min.css\">"]';
```

- 7) dwGetEvent 用于处理控件事件

有 2 个参数

ACtrl:TComponent, 控件自身

AData:String, 控件信息，为 JSON 字符串，一般应先转换为 JSON 对象，比如 joData。其中 joData.e 为事件名称，如“onchange”或“onclick”等（大小写敏感）；joData.v 为值，如 Edit 的 Value 或 CheckBox 的 Checked 等。

在这个函数中，需要根据这 2 个参数，运行相应的事件，如：

```
function dwGetEvent(ACtrl:TComponent;AData:String):String;StdCall;
```

```
var
```

```
    joData : Variant;
```

```
begin
```

```
    //
```

```
    joData := _Json(AData);
```

```
    if joData.e = 'onclick' then begin
```

```
        TButton(ACtrl).OnClick(TButton(ACtrl));
```

```
    end;
```

```
    end else if joData.e = 'onenter' then begin
```

```
        if Assigned(TButton(ACtrl).OnEnter) then begin
```

```
            TButton(ACtrl).OnEnter(TButton(ACtrl));
```

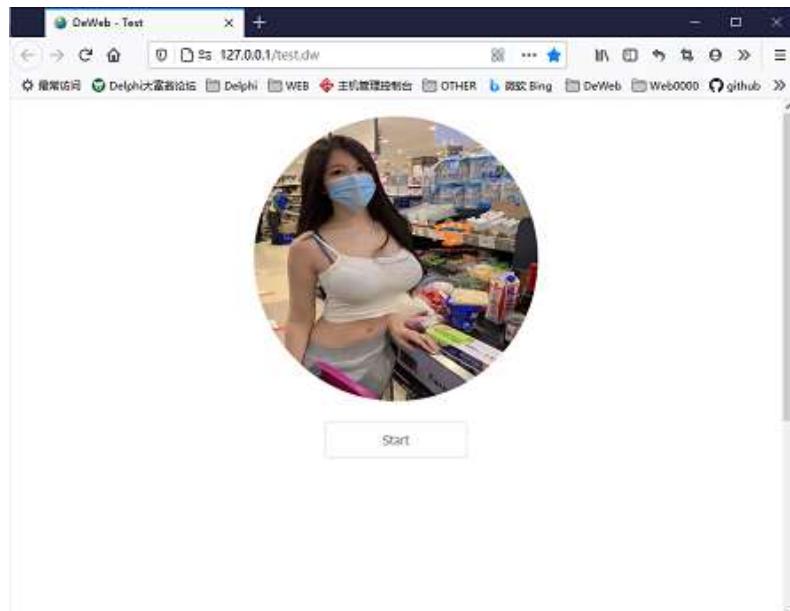
```
        end;
    end else if joData.e = 'onexit' then begin
        if Assigned(TButton(ACtrl).OnExit) then begin
            TButton(ACtrl).OnExit(TButton(ACtrl));
        end;
    end;
end;
```

- 8) dwGetHead 用于生成当前控件改造时的头部信息，需要根据控件对应的 VUE 控件来组成字符串。
- 9) dwGetTail 用于生成当前控件改造时的尾部信息，一般与 dwGetHead 中的 HTML 要素对应即可。
- 10) dwGetData 用于当前控件的各项关键控制参数值，比如控件的 Left,Top,Text 等。
- 11) dwGetMethod 用于当前控件的事件控制值，

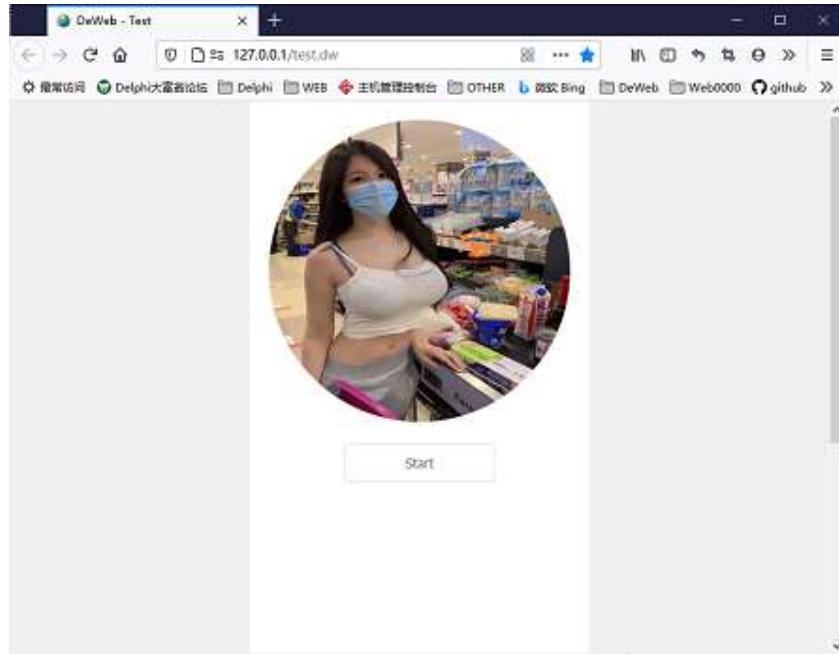
12. Form

- 有 2 种模式：

- (1) 如果 TransparentColor=False，则为全色模式，如图：



- (2) 如果 TransparentColor=True，则为双色模式，如图：



此时，TransparentColorValue 为网页底色，Color 为主体部分颜色

- 属性 VertScrollBar.Visible 决定当窗体高度不够时是否自动显示滚动条
- OnCreate 代码有效
- OnShow 代码有效

在 OnShow 中可以通过 dwGetProp(Self,'params')得到 URL 中的参数

- 如果网页高度比屏幕高度高，可以在 **OnShow** 中采用类似 dwSetHeight(self,2000)来设置高度。

注意：需要将窗体的 VertScrollBox 的 Visible 设置为 False

- 如果窗体 Name = TDefault，则可以 URL 中不输入 http://127.0.0.1:8080/default.dw, 直接通过 <http://127.0.0.1:8080> 即可访问
- ip 可以通过 dwGetProp(Self,'ip')得到（目前好像不准确）；
- URL 中的参数可以通过 dwGetProp(Self,'params')得到；
- AlphaBlend=True 时可以通过 dewebsServer.exe 的 Html 按钮生成纯 html 文件，可用于非 deweb 环境；
- 在代码中设置

self.ParentFont :=True;

可以在动态创建和删除控件时刷新整个窗体；

注：不要用 Form1. ParentFont :=True;

- 多平台适配代码在 OnMouseUp 中。

X,Y 分别表示当前设备的 Width/Height；

Button : mbLeft : 屏幕纵向, mbRight:屏幕横向；

Shift: ssShift, ssAlt, ssCtrl,ssLeft, ssRight,

分别对应 0:未知/1:PC/2:Android/3:iPhone/4:Tablet

另外，浏览窗体的

screenWidth 可以通过 dwGetProp(Self,'screenwidth')得到；

screenHeight 可以通过 dwGetProp(Self,'screenheight')得到；

innerWidth 可以通过 dwGetProp(Self,'innerwidth')得到；

innerHeight 可以通过 dwGetProp(Self,'innerheight')得到；
clientWidth 可以通过 dwGetProp(Self,'clientwidth')得到；
clientHeight 可以通过 dwGetProp(Self,'clientheight')得到；
物理分辨率宽度可以通过 dwGetProp(Self,'truewidth')得到；
物理分辨率高度可以通过 dwGetProp(Self,'trueheight')得到；
设置像素比可以通过 dwGetProp(Self, 'devicepixelratio')得到；

其中：

screenWidth/screenHeight 为屏幕分辨率。 注意：移动端为虚拟屏幕分辨率
innerWidth/innerHeight 为可视区的宽高，包括了滚动条的宽度
clientWidth/clientHeight 为可视区的宽高，不包括了滚动条的宽度

- 页面滚动事件代码在 OnHelp 中。

Data 表示当前滚屏的顶部，例如：设置

```
Label1.Top := Data;
```

可以将当前 Label1 一直在顶部

如果想一直在底部，需要取得浏览器高度，需要根据设备是移动端还是 PC 端分别进行处理。

如果是 PC 端，则需要用到 clientwidth/clientheight

如果是移动端，则需要用到 screenwidth/screenheight

- 支持通过修改 hint 属性引入第三方 js/css.下面的例子中当前 Form 对应的应用会引入 2 个第三方 js/css

例如：

```
{  
  "head": [  
    "<script src=\"http://aaa.com/a.js\" type=\"text/javascript\"> </script>",  
    "<link rel=\"stylesheet\" type=\"text/css\" href=\"dist/b.css\" />"  
  ]  
}
```

13. BitBtn(用于选择拟上传的文件/图像/视频)

- 响应 OnEnter 事件
- 响应 OnExit 事件
- 设计阶段通过更改 Hint 来改变颜色(默认为空或无 type)
分别为

```
{"type":"primary"}  
{"type":"success"}  
{"type":"info"}  
{"type":"warning"}  
{"type":"danger"}  
{"type":"text"}
```

- 设计阶段通过更改 Hint 来改变样式(默认为空)
分别为

```
{"style":"plain"}  
{"style":"round"}  
{"style":"circle"}
```

- 设计阶段通过更改 Hint 来改变图标(默认为无)
例:

```
{"icon":" el-icon-star-off"}
```

各图标名称见附件或 <https://element.eleme.cn/#/zh-CN/component/icon>

- 如果无 accept, 则默认为上传文件
{"accept":"image/*"}表示上传图片;
{"accept":"video/*"}表示上传视频;
- 如果无 capture, 则默认为使用文件管理器、相册或相机
{"capture":"camera"}表示使用相机
- 在 Hint 中设置{"auto":1}, 则可以选择后自动上传
- 注意图标和颜色可以同时配合使用, 例: {"type":"success","icon":" el-icon-star-off"}
- 开始上传自动激活 OnStartDock 事件
上传完成自动激活 OnEndDock 事件
- 可以通过 dwGetProp(self,'_upload')得到上传成功的真实文件名 (如果同名文件存在, 则会自动重命名)
如果在弹出式窗体中, 则使用类似以下的代码
dwGetProp(TForm(self.Parent),'_upload')
- 实际上传时可以通过
dwRunJS('this.dwInputSubmit('+IntToStr(Self.Handle)+',"BitBtn1");self);
如果在弹出式窗体中, 则使用类似以下的代码
dwRunJS('this.dwInputSubmit('+IntToStr(Self.Handle)+','"+Self.Name+'__BitBtn1");self);
- 注意: 除双引号内的部分, 不要使用全角符号!, 请特别注意引号和逗号

14. Button

- 响应 OnClick 事件
- 响应 OnEnter 事件
- 响应 OnExit 事件
- 设计阶段通过更改 Hint 来进行配置
- 改变按钮样式(默认为空或无 type)

分别为

```
{"type":"primary"}  
{"type":"success"}  
{"type":"info"}  
{"type":"warning"}  
{"type":"danger"}  
{"type":"text"}
```

- 自定义颜色

```
{"backgroundcolor":"rgb(51,123,208)","color":"rgb(200,200,200)"}  
其中：backgroundcolor 为按钮背景色， color 为按钮字体颜色
```

- 自定义字体大小

```
{"fontsize":"16px"}
```

其中： fontsize 为按钮字体大小

- 设计阶段通过更改 Hint 来改变样式(默认为空)

分别为

```
{"style":"plain"}  
{"style":"round"}  
{"style":"circle"}
```

- 设计阶段通过更改 Hint 来改变图标(默认为无)

例：

```
{"icon":" el-icon-star-off"}
```

各图标名称见附件或 <https://element.eleme.cn/#/zh-CN/component/icon>

- 通过更改 Hint 来改变圆角半径

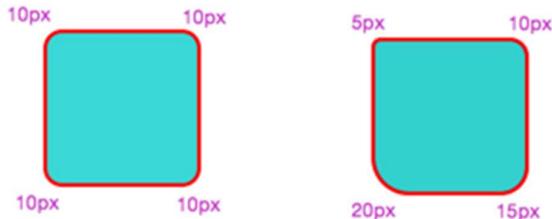
```
{"borderradius":"5px"}  
{"borderradius":"5px 5px 0px 0px"}
```

数值有三种表示方法： px、%、em，一般推荐在用 px；

如果你在 border-radius 属性中只指定一个值，那么将生成 4 个 圆角。

但是，如果你要在四个角上一一指定，可以使用以下规则：

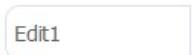
- 四个值：第一个值为左上角，第二个值为右上角，第三个值为右下角，第四个值为左下角。
- 三个值：第一个值为左上角，第二个值为右上角和左下角，第三个值为右下角
- 两个值：第一个值为左上角与右下角，第二个值为右上角与左下角
- 一个值： 四个圆角值相同



- 常用的左侧有圆角，右侧没有，则可以

```
{"borderradius":"10px 0 0 10px"}
```

如图



- 左侧无圆角，右侧有，则可以类似设置 hint

```
{"borderradius":"0 10px 10px 0"}
```

如图



- 注意图标和颜色可以同时配合使用，例：{"type":"success","icon":"el-icon-star-off"}

- 注意：**除双引号内的部分，不要使用全角符号！，请特别注意引号和逗号

- 本地执行 javascript 代码（**注：本功能需要熟练掌握 javascript 知识，请谨慎使用！**）

可以在 Hint 中添加可在本地执行的 javascript 代码，执行时机分别 o 为
OnEnter,OnExit,OnClick

类似：

```
{"onenter":"this.Button1_cap='enter';","onexit":"this.Button1_cap ='exit';"}
```

表示当鼠标进入时，置 Button1 的 caption 为 enter；

当鼠标退出时，置 Button1 的 caption 为 exit；

或者

```
{"onclick":"this.Button1_cap='click';"}
```

表示当鼠标单击时，置 Button1 的 caption 为 click；

可以控制的变量有：

Button1_top : 上边距

Button1_lef : 左边距

Button1_wid : 宽

Button1_hei : 高

Button1_vis : 可见性

Button1_cap : 标题

其中 Button1 需要根据当前控件名相应更改，支持在控件 A 中控制另一控件 B 的属性。



15. Chart

- 支持 TLineSeries/TBarSeries/THorizBarSeries/TAreaSeries/TFastLineSeries/TPieSeries
- 每个 Chart 只支持一种 Series, 即所有 Series 按第一个 Series 的类型处理;
- 同一 Chart 的所有 Series 需要按同一 X 值添加数据, 且数据个数相同;
- ShowHint 用来指示是否显示提示框;
- BackWall.Pen.Color 用来指示边框颜色;
- Legend.Visilbe 用来指示是否显示 Legend;
- Legend.Alignment 指示 Legend 位置
- Legend.Invert 用来指示是横向 (False) 和纵向排列 (True) ;
- Legend.ColorWidth 用来指示当 Legend 在左边或右边时所占的宽度;
- Legend.Pen.Color 用来指示 Legend 的文本颜色;
- Color 用来配置背景色;

16. CheckBox

17. ComboBox

- 高度可以通过设置 Hint 为{"height":35}来实现;
- 高度也可以通过改变字体来实现。但实际显示效果中字体还是默认的，仅高度改变
- 指定 ItemIndex 时, 需要手动指定 Text, 类似:

```
ComboBox1.ItemIndex := 1;  
ComboBox1.Text := ComboBox1.Items[ComboBox1.ItemIndex];
```

- 取得 ItemIndex 时, 需要手动取得, 类似:

```
procedure TForm1.Button_GetRowClick(Sender: TObject);  
var  
  I : Integer;  
  iRes : Integer;  
begin  
  //  
  iRes := -1;  
  for I:=0 to ComboBox1.Items.Count-1 do begin  
    if ComboBox1.Items[I] = ComboBox1.Text then begin  
      iRes := I;  
      break;  
    end;  
  end;  
  //  
  ComboBox1.ItemIndex := iRes;  
  
  //
```

```
dwShowMessage('ItemIndex : '+IntToStr(ComboBox1.ItemIndex),self);
end;
```

18. DateTimePicker

19. Edit

- 支持 OnChange 事件 (必须有代码, 可以是注释)
- 支持 OnEnter 事件
- 支持 OnExit 事件
- 支持 OnMouseEnter 事件
- 支持 OnMouseLeave 事件
- 注意:
 - ◆ Edit 的 Text 值中有“或<或>时会影响解析, 需要用\"或\<或\>来代替; 同时在代码中注意: 处理时多一个\, 而在前端显示时无此\
 - ◆ Edit 的 Text 值前后的空格会被忽略
- 可以通过设置 Hint 来前置图标, 如{"prefix-icon":"el-icon-s-home"}
- 可以通过设置 Hint 来后置图标, 如{"suffix-icon":"el-icon-s-home"}
- 可以通过设置 Hint 来设置输入提示, 如{"placeholder":"请输入用户名"}
- 可以组合, 如: {"placeholder":"请输入用户名","prefix-icon":"el-icon-s-home","suffix-icon":"el-icon-search"}
- 注意: 设置 Hint 时除双引号内的部分, 不要使用全角符号!, 请特别注意引号和逗号
- 注意: OnChange 事件中必须有代码, 可以是一个空的注释

20. Image

- 可以采用设置 Hint 为类似{"radius":"50%"}或{"radius":"5px"}来设置圆角
- 可以采用设置 Hint 中的 src 来设置图片源。如果不设置, 当直接显示当前显示的图片 (如果窗体较多, 请尽可能将 TImage 的名称和其他 TImage 的名称不一样)。
- 设置 Hint 中的 src 来设置图片源的源文件可能是所有网络支持类型 (*.jpg, *.png, *.gif 等), 如果不设置, 系统将自动保存当前显示的图片为 jpg 格式, 并显示。

21. Image(用作验证码)

- 注: 用作验证码时需要设置其 HelpKeyword='captcha'
- 生成的图片默认保存在目录 runtime/media/images, 请定时清理
- 通过 dwGetProp(Image1,'captcha')得到验证码的字符串值 (uses dwBase;)
- 用类似以下代码主动刷新验证码

```
dwRunJS('document.getElementById("'+Image1.Name+'").click();',self);
```

22. Label

- Delphi 界面效果和网页效果会有一些区别, 宽度和高度最好都留一些余量
- 支持 OnClick 事件
- 支持 OnMouseEnter 事件

- 支持 OnMouseLeave 事件
- 如果需要强制换行，可以在 Caption 中增加
，
如： delphi 是一个很棒的开发工具！我喜欢！
可以写成： delphi 是一个很棒的开发工具！
我喜欢！
- 如果需要在运行过程中动态改变 Label 的字体、字号、颜色、样式、对齐等，可以通过设置 HelpKeyword 为 rich 为支持该功能

23. ListBox

- 支持 Enabled/Visible/Left/Top/Width/Height
- 支持 OnClick 事件
- 支持单选/多选
- 暂不支持在 OnCreate 后动态改变 Items

24. MainMenu

- 采用 dwSetCompLTWH 函数设置 Left/Top/Width/Height （默认： 0, 0, 600, 50）
- 采用 dwSetMenuDefault 函数设置默认选中的菜单项，
如： dwSetMenuDefault(MainMenu,'1-0-2');
注： 序号从 0 开始，每层之间用-隔开
- 默认 Left:0;Top:0;Width:600;Height:38
- AutoHotKeys 必须设置为 maManual
- 事件必须写在菜单项的 OnClick 中
- ParentBiDiMode 为 True 时为水平显示,为 True 时垂直显示
- 不要使用属性 Tag/DesignInfo
- 第 1 个菜单项的 Hint 用于保存前景色，字体色等信息；
`{"background-color":"#FF0000","text-color":"#0000FF","active-text-color":"#FFFFFF"}`
- ImageIndex 见 dwVars 中的 dwIcons 常量和 ElementUI 中图标的定义，网址：
<https://element.eleme.cn/#/zh-CN/component/icon>





25. MediaPlayer

本控件有 2 种模式：

- ◆ 普通模式 (HelpKeyword <> 'm3u8') ,支持的视频格式有：
 - ✧ MP4 = MPEG 4 文件使 dao 用 H264 视频 zhuan 编解码器和 AAC 音频 shu 编解码器
 - ✧ WebM = WebM 文件使用 VP8 视频编解码器和 Vorbis 音频编解码器
 - ✧ Ogg = Ogg 文件使用 Theora 视频编解码器和 Vorbis 音频编解码器
- ◆ 视频流模式 (在控件中设置 HelpKeyword = 'm3u8')，支持 m3u8 等视频流格式，可以播放在线电视。

具体的属性有

- AutoRewind 表示是否循环播放；
- Enabled 表示是否自动播放（一些浏览器不支持），注意不是 AutoOpen；
- FileName 表示当前视频位置，注意需要采用网络位置，如./ocean.mp4
- 播放，采用 MediaPlayer1.EnabledButtons := [btPlay];
- 暂停，采用 MediaPlayer1.EnabledButtons := [btPause];
- 设置时刻，采用类似 MediaPlayer1.HelpContext := 6; //表示播放位置为 6-1 秒处

26. Memo

- 支持 OnChange 事件

- 支持 OnEnter 事件
- 支持 OnExit 事件
- 支持 OnMouseEnter 事件
- 支持 OnMouseLeave 事件

27. Memo (用于直接显示 HTML 代码)

- 注：用作验证码时需要设置其 HelpKeyword='html'
- Memo 的 Text 为拟显示的 HTML 代码。

注：请谨慎使用！ 注意 HTML 代码的正确性！

28. PageControl

- ParentBiDiMode 和 ParentShowHint 联合配置是否显示边框
 - ParentBiDiMode=True 且 ParentShowHint=True 时，则显示全边框，如



- ParentBiDiMode=True 且 ParentShowHint=False 时，则显示顶部边框，如



Label4

- ParentBiDiMode=False 时，不论 ParentShowHint 值，都不显示边框，如



ParentBiDiMode为是否显示边框

- 设置 Hint 为{"dwater": "editable"}, 可以在 TabSheet 的标题处显示关闭按钮和新增按钮，当按下按钮时，激活 OnEndDock 事件，类似：
procedure TForm1.PageControl1EndDock(Sender, Target: TObject; X, Y: Integer);
其中参数：
 - X = 1 为新增 Tab；
 - X = 0 为删除 Tab, 待删除 Tab 的序号为 Y (从 0 开始)
- 设置 Hint 为 {"dwater": "closable"}, 可以在 TabSheet 的标题处显示关闭按钮，当按下关闭按钮时，激活 OnEndDock 事件，参数和以上相同。

29. PageControl (用作时间线)

- 注：PageControl 用作时间线时需要设置其 HelpKeyword='timeline'，如下图



- 每个 TabSheet 为一个时间节点
 - Tag 为时间值
 - Caption 为时间节点标题（黑体）
 - 可以在 Hint 中通过类似{"color":"#FF0000"}来设置节点颜色
 - 每个 TabSheet 中可以有若干 Label 或 Memo, 按创建顺序显示为标题下面的文本
 - TabSheet 中不能有其他控件, 否则容易造成混乱!!!

30. PageControl (用作轮播组件)

- 注：PageControl 用作轮播组件时需要设置其 HelpKeyword='carousel';
- 每个 TabSheet 为一个页面

31. Panel

- BorderStyle 为 bsSingle, 显示边框, 且为圆角; 否则无框直角
- Caption 属性无效, 实现显示时不显示 Caption;
- HelpContext 用来设置透明度, 0: 不透明, 2~9: 半透明, 10: 完全透明;
- 通过更改 Hint 来改变圆角半径


```
{"borderradius":"5px"}  
{"borderradius":"5px 5px 0px 0px"}
```

数值有三种表示方法: px、%、em, 一般推荐用 px;
 如果你在 border-radius 属性中只指定一个值, 那么将生成 4 个 圆角。
 但是, 如果你要在四个角上一一指定, 可以使用以下规则:

 - 四个值: 第一个值为左上角, 第二个值为右上角, 第三个值为右下角, 第四个值为左下角。
 - 三个值: 第一个值为左上角, 第二个值为右上角和左下角, 第三个值为右下角
 - 两个值: 第一个值为左上角与右下角, 第二个值为右上角与左下角
 - 一个值: 四个圆角值相同- 可以通过设置其 Hint 属性 dwstyle 完成设置 Panel 单独边框, 类似


```
{"dwstyle":"border-bottom:solid 2px #aaa; border-top:solid 2px #aaa;"}
```

本例完成实现边框: 显示下边框, 样式为 solid, 宽度为 2, 颜色 为#aaa

```
{"dwstyle":"border:solid 2px #aaa;"}
```

本例完成实现边框: 显示所有边框 (上下左右), 样式为 solid, 宽度为 2, 颜色 为#aaa

```
{"dwstyle":"border-bottom:solid 2px #800; border-top:dashed 1px #008;"}
```

本例完成实现边框: 显示上边框和下边框, 其中下边框样式为 solid, 宽度为 2, 颜色 为#800,

上边框样式为 dashed 虚线, 宽度为 1, 颜色 为#008

其实, dwstyle 属性还可以实现许多其他的功能, html 中 div 元素中的 style 属性可

- 以实现的功能都可以通过 dwstyle 来设置实现
- 常用的左侧有圆角，右侧没有，则可以
`{"borderradius":"10px 0 0 10px"}`
 如图



- 左侧无圆角，右侧有，则可以类似

`{"borderradius":"0 10px 10px 0"}`

如图



- 注意：Panel 的 Enabled 不能导致子控件的 Enabled
- 当 Color 为 clNone 时，显示为透明色；
- 可以采用 dwattr 来设置渐变色，如：
`{"dwstyle":"background-image: linear-gradient(rgba(50,50,50,0.8),rgba(50,50,50,0);")}`
- 本地执行 javascript 代码（注：本功能需要熟练掌握 javascript 知识，请谨慎使用！）

可以在 Hint 中添加可在本地执行的 javascript 代码，执行时机分别 o 为 OnEnter,OnExit,OnClick

类似：

`{"onenter":"Panel1_col='#ccc';","onexit":"Panel1_col='#eee';"}`

表示当鼠标进入时，置 Panel1 的 color 为#ccc；

当鼠标退出时，置 Panel1 的 color 为#eee；

或者

`{"onclick":"Panel1_col='#bbb';"}`

表示当鼠标单击时，置 Panel1 的 color 为#bbb；

可以控制的变量有：

Panel1_top : 上边距

Panel1_lef : 左边距

Panel1_wid : 宽

Panel1_hei : 高

Panel1_vis : 可见性

Panel1_col : 颜色

其中 Panel1 需要根据当前控件名相应更改，支持在控件 A 中控制另一控件 B 的属性。

32. Panel (用作 iframe，可用来显示 pdf)

- BorderStyle 为 bsSingle 时，显示边框；否则无边框；
- Hint 中采用 src 来控制 iframe 的 src，如`{"src":"/a.html"}`
- 显示 pdf 时采用类似`{"src":"/dist/pdf/web/viewer.html?file=c.pdf"}`，其中 c.pdf 位于 runtime/dist/pdf/web 目录

33. ProgressBar

34. RadioButton

35. RzPageControl

- SoftCorners 和 ParentShowHint 联合配置是否显示边框
 - SoftCorners=True 且 ParentShowHint=True 时，则显示全边框，如



- SoftCorners=True 且 ParentShowHint=False 时，则显示顶部边框，如



Label4

- SoftCorners=False 时，不论 ParentShowHint 值，都不显示边框，如



- 设置 Hint 为{"dwattr":"editable"}, 可以在 TabSheet 的标题处显示关闭按钮和新增按钮，当按下按钮时，激活 OnEndDock 事件，类似：
procedure TForm1.PageControl1EndDock(Sender, Target: TObject; X, Y: Integer);
其中参数：
 - X = 1 为新增 Tab；
 - X = 0 为删除 Tab, 待删除 Tab 的序号为 Y (从 0 开始)
- 设置 Hint 为{"dwattr":"closable"}, 可以在 TabSheet 的标题处显示关闭按钮，当按下关闭按钮时，激活 OnEndDock 事件，参数和以上相同。
- ShowCardFrame=False 时其内部的 RzTabSheet 的 Left=0，这样显示的 RzPageControl 没有额外的边框；

36. ScrollBox

- 其内必须有且仅有一个 Panel，其他控件请放到该 Panel 中
- 滚动条事件自动激活 OnEndDock 事件，其中 X 为滚动值，Y 为方向，1：向下，-1：向上

37. Shape(用于扫描二维码)

- Pen.Width 为扫描框边框宽度
- Pen.Color 为扫描框边框颜色

- 开始扫描采用函数 dwSetZXing(AShape,AValue);
输入参数：
AShape 类型为 TShape, 为当前 TShape 对象;
AValue 类型为整型, 备用
- 当扫描识别完成后自动激活 TShape 对象的 OnEndDock 事件, 并通过类似的代码取得识别的值:
dwISO8859ToChinese(dwUnEscape(Shape1.HelpKeyword))

38. SpeedButton

用于生成图标居上的按钮

- 采用设置 Hint 中的 src 来设置图片源, 例如:
{"src":"media/images/kehу.png"}
- Flat 为 False 为有外框, 为 True 时为无外框;
- 用 Layout 来控制类型
 - blGlyphLeft 为普通类型



- blGlyphRight 右上显示三角形, 颜色为 groupindex
(常用红色为 255, 蓝色为 16711680)



- blGlyphTop 右上显示显示 Badge 标记, 字体色为白色, 数量为 Spacing, 最大限制为 Margin, 底色通过 hint 中的类似 Button 的 {"type": "primary"} 等控制。



- blGlyphLeft 为右上红点提示



39. SpinEdit

40. StaticText

TStaticText 用于链接，其 Hint 属性中有 3 项

- href 用来标志链接地址，如：http://www.163.com
- type 用于标志链接类型（颜色），选项有：primary,success,warning,danger,info。默认认为 primary
- target 用来设置是否新标签页打开，“_blank”表示在新窗口打开，“_self”表示在当前窗口打开，如：{"href":"http://www.163.com","target":"_blank"}
- ParentFont=True 时为系统配置的字体（字体字号颜色样式），否则为当前设置的的

41. StringGrid

- Height 应为 RowCount * DefaultRowHeight+1，以保持最佳显示效果
- BorderStyle=bsNone 时为无边框
- DefaultDrawing=False 时如果单元格内容超出长度，则会自动换行。
注意：如果内容字数太多，行高可能会被撑开
- 支持 OnClick 事件，在点击时激活
- 标题栏（第一行）可以通过在列标题中添加[*left*]或[*center*]或[*right*]设置当前列的对齐属性。列默认为左对齐。
- TStringGrid 的 Hint 提供一种新的简洁高效(简单粗暴)的方式，即 dwstyle
deweb 会把 dwstyle 的字符直接封装到 TStringGrid 对应的 el-table 中，快速实现一些 el-table 已有的功能
如 {"dwstyle": "border"} 表示带边框和竖线
{"dwstyle": "stripe"} 表示带斑马纹
{"dwstyle": "show-summary"} 表示带合计

还可以组合，比如：

{"dwstyle": "border stripe"} 表示带边框和竖线,带斑马纹

- 设置单元格样式

```
function dwSGSetCellStyle(
    ASG : TStringGrid;
    ARow,ACol : Integer;
    BackGroundColor : TColor;
    FontName : String;
    FontSize : Integer;
    FontBold : Boolean;
    FontItalic : Boolean;
    FontColor : TColor
    ):Integer;
```

输入值：

- ASG : TStringGrid; 为控件名称

- ARow,ACol : Integer;为行、列 ID
- BackGroundColor : TColor;为背景色
- FontName : String;为字体名称
- FontSize : Integer;为字体大小
- FontBold : Boolean;为字体是否粗体
- FontItalic : Boolean;为字体是否斜体
- FontColor : TColor 为字体颜色

参考：

```
dwSGSetCellStyle(StringGrid1,0,0,clGreen,'宋体',20,True,False,clWhite);
```

- 设置列样式

```
function dwSGSetCol(
  ASG:TStringGrid;
  ARow:Integer;
  BackGroundColor:TColor;
  FontColor:TColor
 ):Integer;
```

输入值：

- ASG:TStringGrid;为控件名称
- ACol:Integer;为列 ID
- BackGroundColor:TColor;为背景色
- FontColor:TColor 为字体色

- 设置行样式

```
function dwSGSetRow(
  ASG:TStringGrid;
  ARow:Integer;
  BackGroundColor:TColor;
  FontColor:TColor
 ):Integer;
```

输入值：

- ASG:TStringGrid;为控件名称
- ARow:Integer;为行 ID
- BackGroundColor:TColor;为背景色
- FontColor:TColor 为字体色

42. Timer

- !!! 注!!! 通过设置 DesignInfo 来控制是否运行，其中：

1 为 Enabled=True；0 为 Enabled=False;不要通过 Timer 自身的 Enabled 来控制

43. ToggleSwitch



- State 为当前状态;
- ThumbColor 为选中时的颜色
- ThumbWidth 为选择框宽度
- StateCaptions-CaptionOff/CaptionOn 为未选中/选中的标题

44. TrackBar

45. TrackBar (用作 Pagination 分页)

- 注：TrackBar 用作轮播组件时需要设置其 HelpKeyword='page';
- Max 为总数量 (页数*每页数据量);
- Min 建议为 0;
- ThumbLength 为每页显示的数据量;
- Position 为当前页序号 (1 为首页);
- 点击页码 (包括上一页, 下一页) 后会触发 OnChange 事件
- 可以通过设计 Hint 的 dwattr 为设置显示的组件

如{"dwattr":"layout=\\"prev, pager, next\\""}
表示仅显示“下一页”，“页码”和“下一页”，如图



layout 的选项有: sizes, prev, pager, next, jumper, ->, total, slot

- 可以通过设计 Hint 的 dwattr 为设置显示背景
{"dwattr":"background"}
如图



- 更多设置请参考
<https://element.eleme.cn/#/zh-CN/component/pagination>

46. TreeView

- 树形控件
- 不要使用节点的 StateIndex

附件 1: ElementUI 图标

网址: <https://element.eleme.cn/#/zh-CN/component/icon>

0					
10					
20					
30					
40					
50					
60					
70					
80					
90					

100					
110					
120					
130					
140					
150					
160					
170					
180					
190					

200					
210					
220					
230					
240					
250					
260					
270					

附件 2: ElementUI 图标编号 (ImageIndex)

- 1. el-icon-platform-eleme
- 2. el-icon-eleme
- 3. el-icon-delete-solid
- 4. el-icon-delete
- 5. el-icon-s-tools
- 6. el-icon-setting
- 7. el-icon-user-solid
- 8. el-icon-user
- 9. el-icon-phone
- 10. el-icon-phone-outline
- 11. el-icon-more
- 12. el-icon-more-outline
- 13. el-icon-star-on
- 14. el-icon-star-off
- 15. el-icon-s-goods
- 16. el-icon-goods
- 17. el-icon-warning
- 18. el-icon-warning-outline
- 19. el-icon-question
- 20. el-icon-info
- 21. el-icon-remove
- 22. el-icon-circle-plus
- 23. el-icon-success
- 24. el-icon-error
- 25. el-icon-zoom-in
- 26. el-icon-zoom-out
- 27. el-icon-remove-outline
- 28. el-icon-circle-plus-outline
- 29. el-icon-circle-check
- 30. el-icon-circle-close
- 31. el-icon-s-help
- 32. el-icon-help
- 33. el-icon-minus
- 34. el-icon-plus
- 35. el-icon-check
- 36. el-icon-close
- 37. el-icon-picture
- 38. el-icon-picture-outline
- 39. el-icon-picture-outline-round
- 40. el-icon-upload
- 41. el-icon-upload2
- 42. el-icon-download
- 43. el-icon-camera-solid
- 44. el-icon-camera
- 45. el-icon-video-camera-solid
- 46. el-icon-video-camera
- 47. el-icon-message-solid
- 48. el-icon-bell
- 49. el-icon-s-cooperation
- 50. el-icon-s-order
- 51. el-icon-s-platform
- 52. el-icon-s-fold
- 53. el-icon-s-unfold
- 54. el-icon-s-operation
- 55. el-icon-s-promotion
- 56. el-icon-s-home
- 57. el-icon-s-release
- 58. el-icon-s-ticket
- 59. el-icon-s-management
- 60. el-icon-s-open
- 61. el-icon-s-shop
- 62. el-icon-s-marketing
- 63. el-icon-s-flag
- 64. el-icon-s-comment
- 65. el-icon-s-finance
- 66. el-icon-s-claim
- 67. el-icon-s-custom
- 68. el-icon-s-opportunity
- 69. el-icon-s-data
- 70. el-icon-s-check
- 71. el-icon-s-grid
- 72. el-icon-menu
- 73. el-icon-share
- 74. el-icon-d-caret
- 75. el-icon-caret-left
- 76. el-icon-caret-right
- 77. el-icon-caret-bottom
- 78. el-icon-caret-top
- 79. el-icon-bottom-left
- 80. el-icon-bottom-right
- 81. el-icon-back
- 82. el-icon-right
- 83. el-icon-bottom
- 84. el-icon-top

- 85. el-icon-top-left
- 86. el-icon-top-right
- 87. el-icon-arrow-left
- 88. el-icon-arrow-right
- 89. el-icon-arrow-down
- 90. el-icon-arrow-up
- 91. el-icon-d-arrow-left
- 92. el-icon-d-arrow-right
- 93. el-icon-video-pause
- 94. el-icon-video-play
- 95. el-icon-refresh
- 96. el-icon-refresh-right
- 97. el-icon-refresh-left
- 98. el-icon-finished
- 99. el-icon-sort
- 100. el-icon-sort-up
- 101. el-icon-sort-down
- 102. el-icon-rank
- 103. el-icon-loading
- 104. el-icon-view
- 105. el-icon-c-scale-to-original
- 106. el-icon-date
- 107. el-icon-edit
- 108. el-icon-edit-outline
- 109. el-icon-folder
- 110. el-icon-folder-opened
- 111. el-icon-folder-add
- 112. el-icon-folder-remove
- 113. el-icon-folder-delete
- 114. el-icon-folder-checked
- 115. el-icon-tickets
- 116. el-icon-document-remove
- 117. el-icon-document-delete
- 118. el-icon-document-copy
- 119. el-icon-document-checked
- 120. el-icon-document
- 121. el-icon-document-add
- 122. el-icon-printer
- 123. el-icon-paperclip
- 124. el-icon-takeaway-box
- 125. el-icon-search
- 126. el-icon-monitor
- 127. el-icon-attract
- 128. el-icon-mobile
- 129. el-icon-scissors
- 130. el-icon-umbrella
- 131. el-icon-headset
- 132. el-icon-brush
- 133. el-icon-mouse
- 134. el-icon-coordinate
- 135. el-icon-magic-stick
- 136. el-icon-reading
- 137. el-icon-data-line
- 138. el-icon-data-board
- 139. el-icon-pie-chart
- 140. el-icon-data-analysis
- 141. el-icon-collection-tag
- 142. el-icon-film
- 143. el-icon-suitcase
- 144. el-icon-suitcase-1
- 145. el-icon-receiving
- 146. el-icon-collection
- 147. el-icon-files
- 148. el-icon-notebook-1
- 149. el-icon-notebook-2
- 150. el-icon-toilet-paper
- 151. el-icon-office-building
- 152. el-icon-school
- 153. el-icon-table-lamp
- 154. el-icon-house
- 155. el-icon-no-smoking
- 156. el-icon-smoking
- 157. el-icon-shopping-cart-full
- 158. el-icon-shopping-cart-1
- 159. el-icon-shopping-cart-2
- 160. el-icon-shopping-bag-1
- 161. el-icon-shopping-bag-2
- 162. el-icon-sold-out
- 163. el-icon-sell
- 164. el-icon-present
- 165. el-icon-box
- 166. el-icon-bank-card
- 167. el-icon-money
- 168. el-icon-coin
- 169. el-icon-wallet
- 170. el-icon-discount
- 171. el-icon-price-tag
- 172. el-icon-news

- 173. el-icon-guide
- 174. el-icon-male
- 175. el-icon-female
- 176. el-icon-thumb
- 177. el-icon-cpu
- 178. el-icon-link
- 179. el-icon-connection
- 180. el-icon-open
- 181. el-icon-turn-off
- 182. el-icon-set-up
- 183. el-icon-chat-round
- 184. el-icon-chat-line-round
- 185. el-icon-chat-square
- 186. el-icon-chat-dot-round
- 187. el-icon-chat-dot-square
- 188. el-icon-chat-line-square
- 189. el-icon-message
- 190. el-icon-postcard
- 191. el-icon-position
- 192. el-icon-turn-off-microphone
- 193. el-icon-microphone
- 194. el-icon-close-notification
- 195. el-icon-bangzhu
- 196. el-icon-time
- 197. el-icon-odometer
- 198. el-icon-crop
- 199. el-icon-aim
- 200. el-icon-switch-button
- 201. el-icon-full-screen
- 202. el-icon-copy-document
- 203. el-icon-mic
- 204. el-icon-stopwatch
- 205. el-icon-medal-1
- 206. el-icon-medal
- 207. el-icon-trophy
- 208. el-icon-trophy-1
- 209. el-icon-first-aid-kit
- 210. el-icon-discover
- 211. el-icon-place
- 212. el-icon-location
- 213. el-icon-location-outline
- 214. el-icon-location-information
- 215. el-icon-add-location
- 216. el-icon-delete-location
- 217. el-icon-map-location
- 218. el-icon-alarm-clock
- 219. el-icon-timer
- 220. el-icon-watch-1
- 221. el-icon-watch
- 222. el-icon-lock
- 223. el-icon-unlock
- 224. el-icon-key
- 225. el-icon-service
- 226. el-icon-mobile-phone
- 227. el-icon-bicycle
- 228. el-icon-truck
- 229. el-icon-ship
- 230. el-icon-basketball
- 231. el-icon-football
- 232. el-icon-soccer
- 233. el-icon-baseball
- 234. el-icon-wind-power
- 235. el-icon-light-rain
- 236. el-icon-lightning
- 237. el-icon-heavy-rain
- 238. el-icon-sunrise
- 239. el-icon-sunrise-1
- 240. el-icon-sunset
- 241. el-icon-sunny
- 242. el-icon-cloudy
- 243. el-icon-partly-cloudy
- 244. el-icon-cloudy-and-sunny
- 245. el-icon-moon
- 246. el-icon-moon-night
- 247. el-icon-dish
- 248. el-icon-dish-1
- 249. el-icon-food
- 250. el-icon-chicken
- 251. el-icon-fork-spoon
- 252. el-icon-knife-fork
- 253. el-icon-burger
- 254. el-icon-tableware
- 255. el-icon-sugar
- 256. el-icon-dessert
- 257. el-icon-ice-cream
- 258. el-icon-hot-water
- 259. el-icon-water-cup
- 260. el-icon-coffee-cup

- | | |
|---------------------------------|-------------------------------|
| 261. el-icon-cold-drink | 271. el-icon-pear |
| 262. el-icon-goblet | 272. el-icon-orange |
| 263. el-icon-goblet-full | 273. el-icon-coffee |
| 264. el-icon-goblet-square | 274. el-icon-ice-tea |
| 265. el-icon-goblet-square-full | 275. el-icon-ice-drink |
| 266. el-icon-refrigerator | 276. el-icon-milk-tea |
| 267. el-icon-grape | 277. el-icon-potato-strips |
| 268. el-icon-watermelon | 278. el-icon-lollipop |
| 269. el-icon-cherry | 279. el-icon-ice-cream-square |
| 270. el-icon-apple | 280. el-icon-ice-cream-round |