**DeWeb开发指南**

(2020-06-21)

# 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.web0000.com

# 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Button
* 支持Label
* 支持Edit
* 支持CheckBox
* 支持Panel
* 支持RadioButton
* 支持SpinEdit
* 支持ComboBox
* 支持Memo
* 支持DateTimePicker
* 支持TPageControl
* 支持StringGrid
* 支持Image
* 支持TreeView
* 支持ListBox
* 支持MainMenu
* 支持PopupMenu
* 支持Timer
* 暂不支持DBGrid/DBEdit/DBLabel/DBMemo...等

# DeWeb入门

* 运行自带DEMO
* 目前仅支持10.1/10.2/10.3/10.4
* 复制Source目录内子目录相应版本(10.1/10.2/10.3/10.4)内的文件到当前Source目录
* 打开DelphiWeb.dproj, 编译运行
* 在浏览器中打开 <http://127.0.0.1>:8080/
* 创建hello,world!页面
* 在上述工程中新建Form, 存为unit9.pas, Form的Name = THello
* 放个Label, Caption = 'Hello,World'
* 在Main.pas中uses中添加unit9,
* 在Main.pas中最底部initialization中添加RegisterClass(THello);

在Main.pas中finalization中添加UnRegisterClass(THello);

* 编译运行
* 浏览器中打开http://127.0.0.1:8080/Hello.dw

# 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;

# 特殊函数

* 打开一个新窗体（页面）请使用类似dwCreateForm('TForm4',Self.Handle);

使用时请将'TForm4'替换为对应的窗体类名

* **ShowMessage**采用dwShowMessage代替

procedure dwShowMessage(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dwMessage代替

procedure dwMessageDlg(AMsg,ACaption,confirmButtonCaption,cancelButtonCaption,AMethedName:String;AForm:TForm);

其中:

* AMsg：消息内容
* ACaption：消息框标题
* confirmButtonCaption：确定按钮标题
* cancelButtonCaption：取消按钮标题
* AMethedName：返回结果后执行的过程函数名
* 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，需要等结果返回后在过程函数AMethedName（如：‘ProcessMessageDlg’）中处理

过程函数AMethedName应类似定义，且放在窗体类中（不要放在private/public等中）

procedure ProcessMessageDlg(AResult:String);

该过程函数只有一个string型参数，为1时表示为“确定”，否则为“取消”。

如果需要用到多个MessageDlg，则需要定义多个AMethedName函数

* **InputQuery**采用dwInputQuery代替

procedure dwInputQuery(AMsg,ACaption,ADefault,confirmButtonCaption,cancelButtonCaption,AMethedName: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
* AMethedName：返回结果后执行的过程函数名
* 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，需要等结果返回后在过程函数AMethedName（如：‘ProcessInputQuery’）中处理

过程函数AMethedName应类似定义，且放在窗体类中（不要放在private/public等中）

procedure ProcessInputQuery(AResult:String);

该过程函数只有一个string型参数，为输入的内容。

如果需要用到多个InputQuery，则需要定义多个AMethedName函数

* **dwRealignChildren**根据参数，对控件重新排版

ACtrl:TWinControl， 当前控件，一般建议为TPanel

AHorz:Boolean; 子控件排列方式，为True时水平排列，否则垂直排列

ASize:Integer 子控件参数，如果是水平排列,则为子控件宽度;如果是垂直排列，则为子控件高度

* **dwNavigate**打开网页

procedure dwNavigate(AUrl:String;ANew:Boolean;AForm:TForm);

AUrl: String， 网址

ANew:Boolean; 是否在新网页打开

AForm:TForm 当前窗体, 一般用self

# TForm

* 如果TransparentColor=True, 则为TransparentColorValue为网页底色，主体部分颜色为Color；否则网页颜色均为Color
* OnCreate代码有效
* OnShow代码有效
* 如果网页高度比屏幕高度，可以采用类似dwSetHeight(self,2000)来设置高度
* 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
* 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；

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

# Button

* 响应OnClick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"}

* 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>

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

# 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

# Edit

* dwSetPlaceHodler设置PlaceHolder输入提示
* 支持OnChange事件
* 前置图标{"prefixicon":"el-icon-s-home"}
* 后置图标{"suffixicon":"el-icon-s-home"}

# CheckBox

# Panel

* 如果BevelInner、BevelOuter不为bvNone，则显示边框，且为圆角；否则无框直角
* 设计过程中可以设置BevelKind=bkSoft进行设计(有时无边框看不见)
* Caption属性无效
* **注意** : Panel的Enabled不能导致子控件的Enabled

# RadioButton

# SpinEdit

# ComboBox

# Memo

# DateTimePicker

# TPageControl

* ParentBiDiMode为是否显示边框

# StringGrid

* Height应为RowCount \* 48，以保持最佳显示效果
* BorderStyle=bsNone时为无边框
* 支持OnClick事件, 在点击不同行时激活

# Image

* 可以采用设置Hint为类似{"radius":"50%"}或{"radius":"5px"}来设置圆角

# TreeView

* 树形控件
* 不要使用节点的StateIndex

# 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

# MainMenu

* 采用dwSetCompLTWH函数设置Left/Top/Width/Height
* 默认Left:0;Top:0;Width:Form.Width;Height:40
* AutoHotKeys 必须设置为 maManual
* 事件必须写在菜单项的OnClick中
* 不要使用属性Tag/DesignInfo

# PopupMenu

* 可以用于显示垂直树形菜单
* AutoHotKeys 必须设置为 maManual
* 采用dwSetCompLTWH函数设置Left/Top/Width/Height
* 默认Left:0;Top:0;Width:200;Height:Form.Height
* 不要使用属性Tag/DesignInfo

# Timer

# 附件1: ElementUI 图标

