

更多

ZN'S学习笔记

我的简介

dmzn

[查看我的完整个人资料](#)

标签

uniGUI

GoLang

Delphi

Java

System

DB

随想

搜索此博客

博客归档

八月 2021 (3)

七月 2021 (2)

六月 2021 (2)

五月 2021 (2)

四月 2021 (3)

一月 2021 (1)

十月 2020 (1)

二月 2020 (1)

十二月 2019 (1)

十一月 2019 (5)

九月 2019 (7)

一月 2019 (1)

十月 2018 (1)

九月 2018 (2)

八月 2018 (2)

七月 2018 (2)

六月 2018 (1)

五月 2018 (3)

四月 2018 (1)

三月 2018 (5)

二月 2018 (2)

一月 2018 (6)

举报滥用情况

显示标签为“uniGUI”的博文。 [显示所有博文](#)

2021年8月14日星期六

uniGUI执行前端方法并返回结果

以获取 TUniHTMLMemo.OuterText 为例:

```

procedure TForm.BtnGetTextClick(Sender: TObject);
var nStr: string;
begin
  nStr := 'var t1=%s.getEditorBody().outerText;' + //调用前端方法
        'ajaxRequest(%s.form, ''getHTMLtext'', [''t1='' + t1]);'; //%.window等效
  nStr := Format(nStr, [Memo1.JSName, Self.Name]);

  UniSession.SendResponse(nStr); //UniSession.AddJS等效
  UniSession.Synchronize; //执行并等待
  ShowMessageN(FText);
end;

```

```

procedure TForm.UniFormAjaxEvent(Sender: TComponent;
  nEvent: string; nParams: TUniStrings);
begin
  if nEvent='getHTMLtext' then
    FText := nParams.Values['t1'];
  //接收结果
end;

```

需注意, MainModule.EnableSynchronousOperations=true 是 Synchronize 能否正确执行的关键参数.

时间: [八月.14..2021](#) 没有评论:

标签: uniGUI

2021年8月8日星期日

uniGUI编译时自动退出旧实例

uniGUI开发每次编译运行时,实例会在后台运行,每次重新编译时必须手工在任务栏里将应用退出,非常麻烦.可以在项目编译的参数里加上杀进程的命令,重新编译时就会自动将旧的进程杀掉.方法如下:

```

Project > Options > Build Events> Target: All configurations > Pre-build events > Commands,
填写命令: taskkill /f /im $(OUTPUTFILENAME) 2>1 || exit /B 0

```

时间: [八月.08..2021](#) 没有评论:

标签: uniGUI

2021年7月4日星期日

uniGUI前端与服务器的交互方法

以下前端指浏览器,后端指服务器.前后端交互分两种情况:

- 前端和后端拥有相同或对应的事件,如: Ext.button.click = TUniButton.OnClick

前端代码:

```

1. function click(sender, e, eOpts){ alert('a'); }
2. function ajaxRequest(sender, url, data){ alert('b');}
3. function ajaxCallback(sender, response){ alert('d');}

```

后端代码:

```

procedure BtnClick(Sender: TObject);
begin
  ShowMsg('c');
end;

```

前端点击按钮时,先执行前端click,然后发送ajaxRequest到后端,后端执行BtnClick后返回,触发ajaxCallback.执行顺序为:click>ajaxRequest>BtnClick>ajaxCallback;前端响应顺序为:a>b>**d>c**,前端先执行ajaxCallback,然后执行BtnClick的应答.

- 前端有而后端没有的事件,如: TuniDBGrid.headercontextmenu



前端代码:

```
nStr := 'headercontextmenu=function headercontextmenu(ct, column, e, t, eOpts)' +
      '{ ajaxRequest($0, ''$E'', [''xy'' + e.getXY(), ''col='' + column.dataIndex]) }';
nStr := MacroValue(nStr, [MI('$0', nGrid.JSName), MI('$E', sEvent_DBGridHeaderPopmenu)]);

if nGrid.ClientEvents.ExtEvents.IndexOf(nStr) < 0 then
  nGrid.ClientEvents.ExtEvents.Add(nStr);
//动态绑定前端事件
```

后端代码:

```
procedure DBGridAjaxEvent(Sender: TComponent;
  EventName: string; Params: TUniStrings);
begin
  if EventName = TGridHelper.sEvent_DBGridHeaderPopmenu then
  begin
    //执行前端事件, Params中为传递的参数
    end;
  end;
end;
```

前端事件触发后, 调用ajaxRequest向后端传递事件和参数。第一个参数为接收对象的名称(JSName, 如: 0A1B) 或对对象关系路径 (如: MainForm.Panel1.DBGrid1), 后端对象则触发AjaxEvent。

- 后端主动调用前端方法:

```
UniSession.SendResponse(Btn.JSName + '.setText(''NewText'')');
Btn.JSInterface.JSCall('fireEvent', ['click', BtnLoad]);
```

时间: [七月.04.2021](#) 没有评论:

标签: uniGUI

2021年6月3日星期四

uniGUI使用HTML登录页

```
const
  LoginFile = 'login.html';

procedure TUniServerModule.UniGUIServerModuleHTTPCommand(ARequestInfo:
  TIdHTTPRequestInfo; AResponseInfo: TIdHTTPResponseInfo; var Handled: Boolean);
begin
  if (SameText(ARequestInfo.URI, '/') and (SameText(ARequestInfo.Command, 'GET'))) then
  begin
    Handled := True;
    AResponseInfo.Redirect(LoginFile);
    Exit;
  end;
end;

procedure TUniMainModule.UniGUIMainModuleCreate(Sender: TObject);
var User, Pass: string;
begin
  User := Trim(uniGUIApplication.UniApplication.Parameters.Values['user']);
  Pass := Trim(uniGUIApplication.UniApplication.Parameters.Values['pass']);
  if (Length(User) = 0) or (Length(Pass) = 0) then
  begin
    UniSession.AResponse.Redirect(LoginFile);
    UniSession.Terminate;
  end;
end;
```

时间: [六月.03.2021](#) 没有评论:

标签: uniGUI

uniGUI路由(router)跳转

```
procedure TUniServerModule.UniGUIServerModuleHTTPCommand(
  ARequestInfo: TIdHTTPRequestInfo; AResponseInfo: TIdHTTPResponseInfo;
  var Handled: Boolean);
begin
  if SameText(ARequestInfo.URI, '/server') then
  begin
    Handled := True;
    AResponseInfo.Redirect('/');
  end;

  if SameText(ARequestInfo.URI, '/bing') then
  begin
    Handled := True;
```

```
AResponseInfo.Redirect('http://cn.bing.com');
end;
end;
```

时间: [六月 03, 2021](#) 没有评论:

标签: uniGUI

2021年5月21日星期五

uniGUI中组件居中的属性设定

测试例子如下: 要求图片出现在面板的中间 (水平和垂直居中)。

父容器: myPanel, TUniPanel

- AlignmentControl = uniAlignmentClient, 使用客户端布局
- Layout = hbox, 使用水平布局
- LayoutAttribs.Align = middle, 子组件垂直居中排列
- LayoutAttribs.Pack = center, 子组件水平居中排列

子组件: myImage, TUniImage

- myImage.Url := 'Images/Welcome.gif'

图片加载后,显示在myPanel的中间位置, 官方对于LayoutAttribs的描述:

Align: Controls how the child items of the container are aligned. The value is used to position items "perpendicularly". That is, for horizontal boxes (where vertical is False), then this will position items vertically. Otherwise, this will position items horizontally. The acceptable values for this property are best explained in context with the value of vertical.

If the container is an hbox this property operates as follows:

- top: Child items are aligned vertically at the top of the container.
- middle: Child items are vertically centered in the container.
- bottom: Child items are aligned vertically at the bottom of the container.
- stretch: Child items are stretched vertically to fill the height of the container.
- stretchmax: Child items are stretched vertically to the height of the largest item.

If the container is a vbox this property operates as follows:

- left: Child items are aligned horizontally at the left side of the container.
- center: Child items are horizontally centered in the container.
- right: Child items are aligned horizontally at the right of the container.
- stretch: Child items are stretched horizontally to fill the width of the container.
- stretchmax: Child items are stretched horizontally to the size of the largest item.

Columns: Number of columns in a table layout.

Pack: Controls how the child items of the container are packed together. Acceptable configuration values for this property are:

- start - child items are packed together at left (HBox) or top (VBox) side of container (default*)
- center - child items are packed together at mid-width (HBox) or mid-height (VBox) of container
- end - child items are packed together at right (HBox) or bottom (VBox) side of container

Defaults to 'start'.

Padding: The padding to use on this Component. Can be specified as a number (in which case all edges get the same padding) or a CSS string like '5 10 10 10'.

This property can be specified as a string containing space-separated, numeric padding values. The order of the sides associated with each value matches the way CSS processes padding values:

If there is only one value, it applies to all sides.

If there are two values, the top and bottom borders are set to the first value and the right and left are set to the second.

If there are three values, the top is set to the first value, the left and right are set to the second, and the bottom is set to the third.

If there are four values, they apply to the top, right, bottom, and left, respectively.

In other words:

```
'x' -> top: x, right:x, bottom:x, left:x
'x y' -> top:x, right:y, bottom:x, left: y
'x y z' -> top:x, right:y, bottom: z, left: y
'x y z t' -> top: x, right:y, bottom:z, left:t
```

时间: [五月 21, 2021](#) 没有评论:

标签: uniGUI

2020年10月27日星期二

uniGUI三方组件UniFalcon的安装方法

以我使用的UniFalcon 1.2.0.72为例:

- 解压缩安装包, 运行Demos目录下的任意演示程序。由于未安装Falcon的支持文件, 当浏览器访问演示程序地址 (<http://localhost:8077>)时, 演示程序会在日志文件中输出错误信息。如:
Demos\FSConfirm\executable\nlog\fsconfirm_exe\A2020-10-27.log

FSConfirm.exe: 0000DA4: 15:28:44 [HandleFileRequest[127.0.0.1]]:File not found:
 C:\Program Files (x86)\FMSoft\Framework\unigui\uni-
 1.90.0.1496\falcon\css\preloader.css
 该信息提供了一个路径,表示需要将Falcon支持文件复制到特定的目录中。

- 安装UniFalcon 1.2.0.72\packages目录下的组件包。

时间: [十月 27, 2020](#) 没有评论:

标签: [Delphi](#), [uniGUI](#)

2020年2月21日星期五

uniGUI中使用自定义CSS

1. 在serverModule.CustomCSS中填写css内容,如下:

```
.BtnStyle
{
  background:#d9534f !important;
  color:#fff !important;
  border-color:#d9534f;
  -webkit-transition: background 0.5s ease-in-out;
  transition: background 0.5s ease-in-out;
}
```

2. 在button.ClientEvents.ExtEvents.added添加绑定,如下:

```
function added(sender, container, pos, eOpts)
{
  sender.addCls('BtnStyle')
}
```

时间: [二月 21, 2020](#) 没有评论:

标签: [Delphi](#), [uniGUI](#)

2018年6月12日星期二

页面模式时窗体居中显示

在ServerModule.MainFormDisplayMode=mfPage时,居中显示窗体页面的方法:

- 1.添加容器控件,如TUniPanel。
- 2.修改Panel.Anchors = [],然后将所有组件放置到Panel中。

时间: [六月 12, 2018](#) 没有评论:

标签: [uniGUI](#)

2018年5月24日星期四

允许用户调整表格TuniStringGrid的列宽并持久化

表格TuniStringGrid比较常用,但uniGUI默认不允许用户调整宽度,设计时的默认宽度对用户不友好,需要让表格支持用户自己调整列宽。步骤如下:

1. 修改uniStringGrid单元的SetColumnAttribs函数,允许用户调整列宽。

```
JSConfig('resizable', [(goColSizing in FOptions) and (FColWidths[Index]>0)], Col);
```
2. 添加uniStringGrid.ClientEvents.ExtEvents.columnresize事件监听。

```
function columnresize(ct,column,width,eOpts){
  ajaxRequest(MainForm.Grid1, 'event_colresize', ['idx='+column.dataIndex,'w='+width])
}
```
3. 添加uniStringGrid.OnAjaxEvent事件处理,将列宽同步到服务器。

```
uniStringGrid.Columns[idx].Width := w;
```
4. 将uniStringGrid.Columns列宽写入配置文件

时间: [五月 24, 2018](#) 没有评论:

标签: [uniGUI](#)

2018年5月22日星期二

修改异常时的消息提示

修改MainModule.ServerMessagers属性中的参数。

ServerMessages	(TUniServerMessages)
ExceptionTemplate	(TStrings)
InvalidSessionMessage	Invalid session or session Timeout
InvalidSessionTemplate	(TStrings)
LoadingMessage	Loading...
TerminateMessage	Web session terminated.
TerminateTemplate	(TStrings)
UnavailableErrMsg	Communication Error

ServerMessages	(TUniServerMessages)
ExceptionTemplate	(TStrings)
InvalidSessionMessage	没有登录系统 或 登录超时.
InvalidSessionTemplate	(TStrings)
LoadingMessage	正在读取
TerminateMessage	您已退出系统.
TerminateTemplate	(TStrings)
UnavailableErrMsg	通讯错误: 数据无法正确传输.

时间: 五月 22, 2018 没有评论:

标签: uniGUI

Class TServerControlPanelForm not found

点击服务器窗口菜单Manage.Server Monitor时报错: Class TServerControlPanelForm not found, 在主单元引用uniGUIRegClasses, 注册TServerControlPanelForm类就可以了。

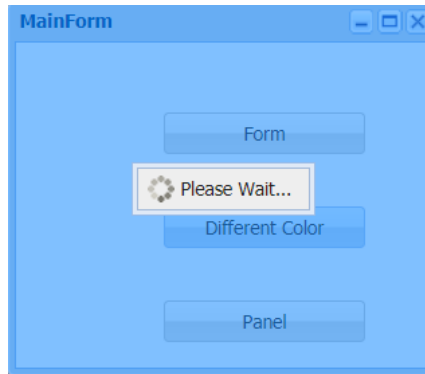
时间: 五月 22, 2018 没有评论:

标签: uniGUI

2018年2月6日星期二

uniGUI进度指示(ScreenMask)用法

当进行某些耗时的操作时, 需要显示ScreenMask(sm)来告知用户, 避免产生假死的错觉。sm进度指示效果如下:



sm的三种用法:

- uniGUI界面组件自带有ScreenMask属性。
设置ScreenMask属性:
Enabled := True;
Message := 'Please Wait';
Target := MainForm;
Target属性用于决定sm的显示位置。
- 使用TUniScreenMask组件。
设置TUniScreenMask属性:
AttachedControl := ButtonLoad; //触发控件
Enabled := True;
DisplayMessage := 'Please Wait';
TargetControl := MainForm; //显示区域
- 使用界面组件客户端事件(ClientEvents.ExtEvents)脚本。
设置ExtEvents脚本内容, 如Ext.button.button.click:
function OnClick(sender, e)
{
 sender.showMask('Please Wait');
 //sender.hideMask(); //隐藏sm
 ajaxRequest(sender, 'BtnClick');
}

关联OnAjaxEvent事件, 处理BtnClick事件:

```
procedure TMainForm.UniButton1AjaxEvent(Sender: TComponent; EventName: string;
Params: TUniStrings);
begin
    if EventName='BtnClick' then
    begin
        Sleep(1000);
    end;
end;
```

时间: [二月 06, 2018](#) 没有评论:

标签: uniGUI

2018年2月1日星期四

uniGUI的窗体切换(2)

属性uniGUI.MainModule.EnableSynchronousOperations默认为False, 像模式窗体或对话框, 代码在服务端和客户浏览器分开执行, 就需要使用回调。如下:

```
procedure TMainForm.DCallBack1(Sender: TComponent; Res: Integer);
begin
    UniMemo1.Lines.Add('DResult-1: '+IntToStr(Res));
end;

procedure TMainForm.UniBtn1Click(Sender: TObject);①
begin
    MessageDlg('msg', mtWarning, [mbOK], DCallBack1);
    ShowMessage('done' + IntToStr(nInt));②
end;
```

用户在浏览器点击触发UniBtn1Click, 服务器开始执行①处代码。虽然MessageDlg是模式对话框, 但该对话框在用户端是模式, 并不会阻塞②的执行。

若uniGUI.MainModule.EnableSynchronousOperations=True, 则可以使用与C/S相同的写法:

```
procedure TMainForm.UniBtn1Click(Sender: TObject);①
var nInt: Integer;
begin
    nInt := MessageDlg('msg', mtWarning, mbOK);
    ShowMessage('done' + IntToStr(nInt));②
end;
```

服务器执行①处代码时, 阻塞MessageDlg, 在用户端弹出模式对话框, 待用户处理后, 返回nInt值执行②处代码。经测试, Form.ShowModal也可以使用服务端阻塞方式顺序执行, 和C/S用法一致。

时间: [二月 01, 2018](#) 没有评论:

标签: uniGUI

2018年1月8日星期一

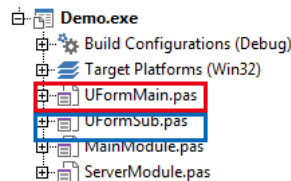
uniGUI的窗体切换

C/S模式下, 窗体切换很简单, 当前FormMain要切换到FormSub, 可以用:

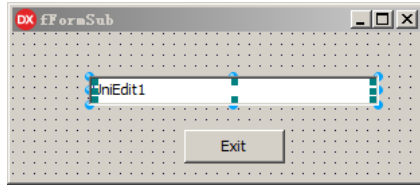
- FormSub.Show: 主窗体和子窗体可以同时获得焦点。
- FormSub.ShowModal: 子窗体获得焦点后, 主窗体无法操作, 直到子窗体关闭。又称为模式窗体。

在uniGUI下, 子窗体依然可以使用这两种方式显示。演示如下:

1. 使用uniGUI → Application Wizard生成项目, 并添加一个子窗体。



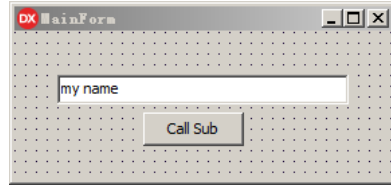
2. 子窗体代码如下:



```
function fFormSub: TfFormSub; //①
begin
  Result := TfFormSub(UniMainModule.GetFormInstance(TfFormSub));
end;

procedure TfFormSub.UniBtnExitClick(Sender: TObject);
begin
  ModalResult := mrok;
end;
```

3. 主窗体代码如下:



```
function MainForm: TMainForm; //①
begin
  Result := TMainForm(UniMainModule.GetFormInstance(TMainForm));
end;

procedure TMainForm.BtnCallClick(Sender: TObject);
begin
  Visible := False; //②
  fFormSub.UniEdit1.Text := UniEdit1.Text; //③
  fFormSub.ShowModal(
    procedure(Sender: TComponent; Result: Integer) //④
    begin
      if not MainForm.Showing then
        MainForm.Show(); //②
      //xxxxx
    end
  );
end;
```

4. 源码分析:

- ①标记: 窗体对象的创建, 需要使用UniMainModule.GetFormInstance函数。如果是Applicaion Form, 向导会自动生成窗体创建函数。
- ②标记: 这是一个用户交互的优化。子窗体弹出后, 主窗体在背景上显示, 会导致界面很乱。可以先隐藏掉, 然后在子窗体关闭后再显示出来。该写法与C/S不同, 是因为C/S用的阻塞模式, 如下:
 Visible := False;
 fFormSub.ShowModal(); //阻塞, 窗体关闭后执行下一条语句
 Visible := True;
- ③标记: 调用子窗体时, 传递参数。
- ④标记: 当子窗体关闭时, 触发回调函数。这里用的匿名函数。

总结: 与C/S的调用方法基本一样, 使用Show、ShowModal方法显示新窗体。由于uniGUI是异步调用, C/S顺序执行的代码, 需要放在回调函数中。

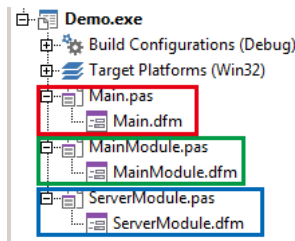
时间: 二月 08, 2018 没有评论:

标签: uniGUI

2018年1月5日星期五

uniGUI系统模块分析

由uniGUI For Delphi的Applicaion Wizard生成的项目, 包含三个模块:



如上图，三个模块中包含三个类：

```
TMainForm = class(TUniForm)
TUniMainModule = class(TUniGUIMainModule)
TUniServerModule = class(TUniGUIServerModule)
```

- **ServerModule.TUniServerModule**

由构造函数可以看出，该模块的类会生成一个全局使用的唯一化实例对象，系统启动时创建，系统退出时释放，存在于程序整个运行周期，所有客户端共用该对象。构造函数如下：

```
function UniServerModule: TUniServerModule;
begin
  Result := TUniServerModule(UniGUIServerInstance);
end;
```

- **MainModule.TUniMainModule**

该模块是客户端的控制器，每个客户端连接成功时创建，连接断开后释放，存在于客户端整个会话周期。每个客户端使用各自独立的MainModule模块。

- **Main.TMainForm**

该模块是第一个窗体类，构造函数如下：

```
function MainForm: TMainForm;
begin
  Result := TMainForm(UniMainModule.GetFormInstance(TMainForm));
end;
```

所有uniGUI的页面窗体，都是通过GetFormInstance创建。第一个注册的TUniForm类，作为系统的主页面窗体。

- 各模块生命周期测试

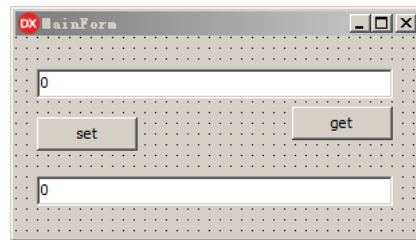
1、SerModule:

```
TUniServerModule = class(TUniGUIServerModule)
public
  { Public declarations }
  Counter: Integer;
end;
```

2、MainModule:

```
TUniMainModule = class(TUniGUIMainModule)
public
  { Public declarations }
  Counter: Integer;
end;
```

3、Main:

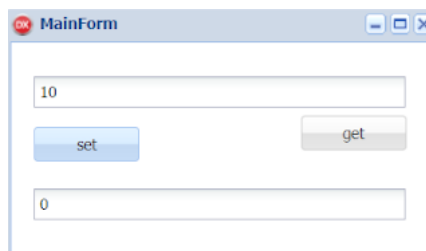


```
procedure TMainForm.BtnSetClick(Sender: TObject);
begin
  UniMainModule.Counter := StrToInt(UniEdit1.Text);
  UniServerModule.Counter := StrToInt(UniEdit1.Text);
end;
```

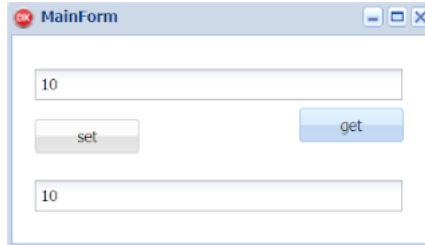
```
procedure TMainForm.BtnGetClick(Sender: TObject);
begin
  UniEdit1.Text := IntToStr(UniMainModule.Counter);
  UniEdit2.Text := IntToStr(UniServerModule.Counter);
end;
```

测试结果：

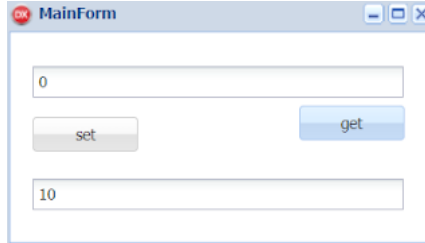
- 1、浏览器登录系统，设置(set)MainModule与ServerModule变量的值。



- 2、读取模块变量，正常。



3、重新登录系统，读取模块变量。MainModule重置为0，表示原会话的MainModule已释放，重新创建了新的模块；ServerModule值不变。



时间: [二月 05, 2018](#) 没有评论:

标签: uniGUI

uniGUI日志输出

- 使用uniGUIApplication.UniSession.Log

输出的日志与会话Session有关，可以存放在指定的目录中，格式如：Demo.exe: 00001904: 17:24:43 [msg]:s2

- 使用UniServerModule.Logger.AddLog

输出全局日志，存放在exe所在的log目录下，格式如：Demo.exe: 00001704: 14:33:26 [category]:message
该方法的第三个参数，用于指定日志存放的目录。

- 使用UniServerModule.ServerLogger

该属性控制系统的日志输出，包括：是否记录Indy产生的异常、是否记录Session产生的异常

时间: [二月 05, 2018](#) 没有评论:

标签: uniGUI

2018年1月4日星期四

使用uniGUI的程序发布方式

uniGUI是Delphi开发B/S系统的神器，虽然开发环境自带IntraWeb，但混合HTML/CSS来开发B/S，代码看起来很怪异。uniGUI隐藏了Web页面的细节，开发过程和C/S开发没有区别，编译后也可以以压缩包的方式发布，解压即可使用。

uniGUI.TUniUIServerModule默认设置使用了几个环境变量：

- ExtRoot: **[ext]**表示使用uniGUI安装目录下的ext-xx目录。
- UniRoot: **[uni]**表示使用uniGUI安装目录下的uni-xx目录。

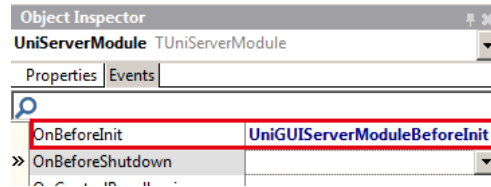
这两个目录中存放的是uniGUI运行时所需的js文件。

部署时，这两个目录需要和编译后的exe打包在一起，例如：



如果开发时设定ExtRoot=ext-js,UniRoot=uni-js，则需要把这两个目录复制到项目的输出文件夹中（如：Bin），和uniGUI.exe在同一目录。

一般开发时，使用默认设置就可以了。在代码中动态指定目录，如下：



代码:

```
procedure TUniServerModule.UniGUIServerModuleBeforeInit(Sender: TObject);
var nStr: string;
begin
  nStr := ExtractFilePath(ParamStr(0)) + 'ext-js';
  Logger.AddLog('TUniServerModule', nStr);

  if DirectoryExists(nStr) then
  begin
    ExtRoot := 'ext-js';
    UniRoot := 'uni-js';
  end;
end;
```

系统启动时，检测同目录下是否有所需的目录。没有，视为开发环境；有，则视为运行环境。将exe和所有目录打包，就可以发布上线了。

时间: [二月 04, 2018](#) 没有评论:

标签: [uniGUI](#)

[主页](#)

[较旧的博文](#)

订阅: [博文 \(Atom\)](#)

“炫酷模板”主题背景. 主题背景图片创建者: [Jason Morrow](#). 由 [Blogger](#) 提供支持.