TMS IntraWeb iPhone Controls Pack
DEVELOPERS GUIDE

April 2015
Copyright © 2015 by tmssoftware.com bvba
Web: http://www.tmssoftware.com
Email : info@tmssoftware.com

Table of contents

Availability 3
Screenshots 4
Tutorial 7
List of included components 9
TTIWiPhoneButton 11
TTIWiPhoneEdit 12
TTIWiPhoneEmailLabel, TTIWiPhoneLocationLabel, TTIWiPhonePhoneLabel & TTiWiPhoneSMSLabel 14
TTIWiPhoneGeolocation 16
TTIWiPhoneHeader & TTIWiPhoneFooter 17
TTIWiPhoneList 19
TTIWiPhoneMenu 22
TTIWiPhoneOnOffButton 24
TTIWiPhonePageFlip & TTIWiPhonePageTransition 25
TTIWiPhonePopup 27
TTIWiPhoneRegion 29
TTIWiPhoneScrollRegion 30
TTIWiPhoneSpinner 31
TTIWiPhoneStyle 32
TTIWiPhoneTrackbar 33

Availability

TMS IntraWeb iPhone Controls Pack is available as VCL components for Delphi and C++Builder.

TMS IntraWeb iPhone Controls Pack is available for
Delphi 2009, 2010, XE, XE2, XE3, XE4, XE5, XE6, XE7, XE8
C++Builder 2009, 2010, XE, XE2, XE3, XE4, XE5, XE6, XE7, XE8

IntraWeb 10.0.x, 11.0.x, 12.x or 14.0.x is required

TMS IntraWeb iPhone Controls Pack has been designed for and tested with: Windows 98, 2000, XP, Vista, 7, 8 on IntraWeb 10.x or higher

Current version of TMS IntraWeb iPhone Controls Pack has been designed for and tested with iPhone, iPad, Android.
Please note the TMS IntraWeb iPhone Controls Pack is not intended to be used in common desktop browsers like Internet Explorer, FireFox, Chrome, Safari …

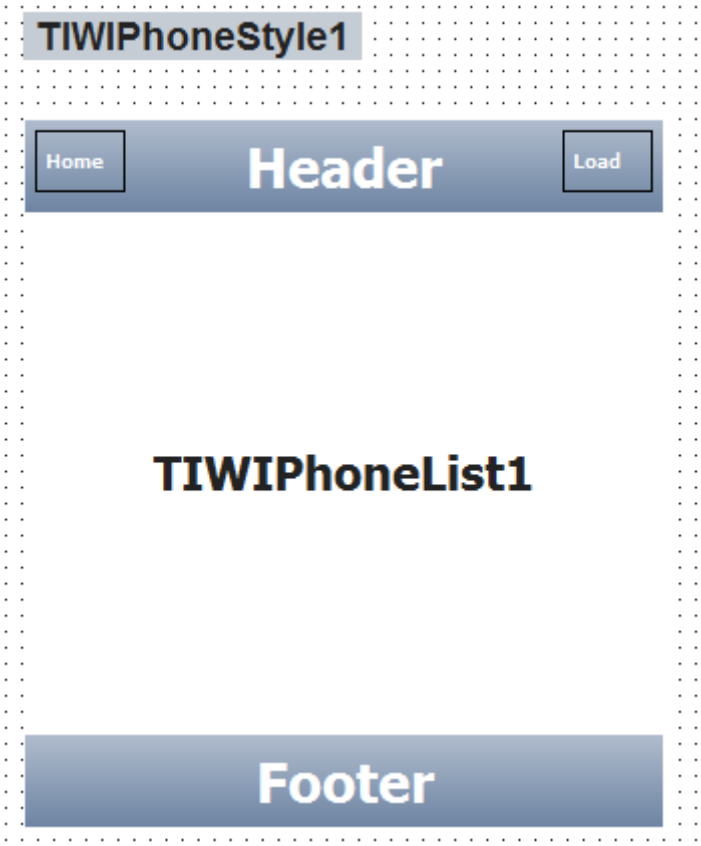
Screenshots


BEL PRO... 15:56
Stats
France 2
Spain 3
UK 2
Ukraine 3
USA 4
Scores
Not Rated 7
1 - 25 0
Models
Stats
Settings
BEL PRO... 15:57
Settings
Models list:
Display images: ON
Enable details: ON
Rounded corners: OFF
Save
Models
Stats
Settings


Proximus
09:27
Not Charging
Model Agency
About
Details
Update
Liza Kane
Spain
100
Mariya Zakharova
Ukraine
100
Melissa Minton
USA
0
Ming Xi
China
0
Mona Rahmati
Denmark
0
Paola Lopera
Spain
85
Rachael Murphy
USA
84
Sian Williams
UK
0
Tara Shakti
France
57
Tintaisha Stafford
UK
0
Load more items
Paola Lopera
Email:
paola.lopera@domain.com
SMS:
3606
Tel:
62156675
Comment:
Score:
85
Address:
Carrer del Bruc 17, Barcelona, Spain
Set current location
Biography:
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce vel quam tempus lorem auctor pharetra at sed nisi. Pellentesque a mi quis lectus semper volutpat vitae et turpis. Etiam elementum gravida enim, ut pulvinar quam pellentesque ornare. Etiam sed.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce vel quam tempus lorem auctor pharetra at sed nisi. Pellentesque a mi quis lectus semper volutpat vitae et turpis. Etiam elementum gravida enim, ut pulvinar quam pellentesque ornare. Etiam sed.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce vel quam tempus lorem auctor pharetra at sed nisi. Pellentesque a mi quis lectus semper volutpat vitae et turpis. Etiam elementum gravida enim, ut pulvinar quam pellentesque ornare. Etiam sed.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce vel quam tempus lorem auctor pharetra at sed nisi. Pellentesque a mi quis lectus semper volutpat vitae et turpis. Etiam elementum gravida enim, ut pulvinar quam pellentesque ornare. Etiam sed.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce vel quam tempus lorem auctor pharetra at sed nisi. Pellentesque a mi quis lectus semper volutpat vitae et turpis. Etiam elementum gravida enim, ut pulvinar quam pellentesque ornare. Etiam sed.
10
Models
Stats
Settings

Tutorial

1. Start by adding a TTIWiPhoneStyle control on the form.
   This way we make sure the application will use the default iPhone/iPad web page configuration, background color and font settings.
2. Then add a TTIWiPhoneRegion on the form.
   The form size is automatically adjusted to the width and height of the iPhone/iPad screen size.
3. Select the TTIWiPhoneRegion and add a TTIWiPhoneHeader, TTIWiPhoneFooter and a TTIWiPhoneList control. These controls will automatically be correctly aligned. The header at the top, the footer at the bottom and the list in the middle.
4. Change the TTIWiPhoneHeader1.RightButton.Caption value to “Load”.
5. This is what your design time form should look like:

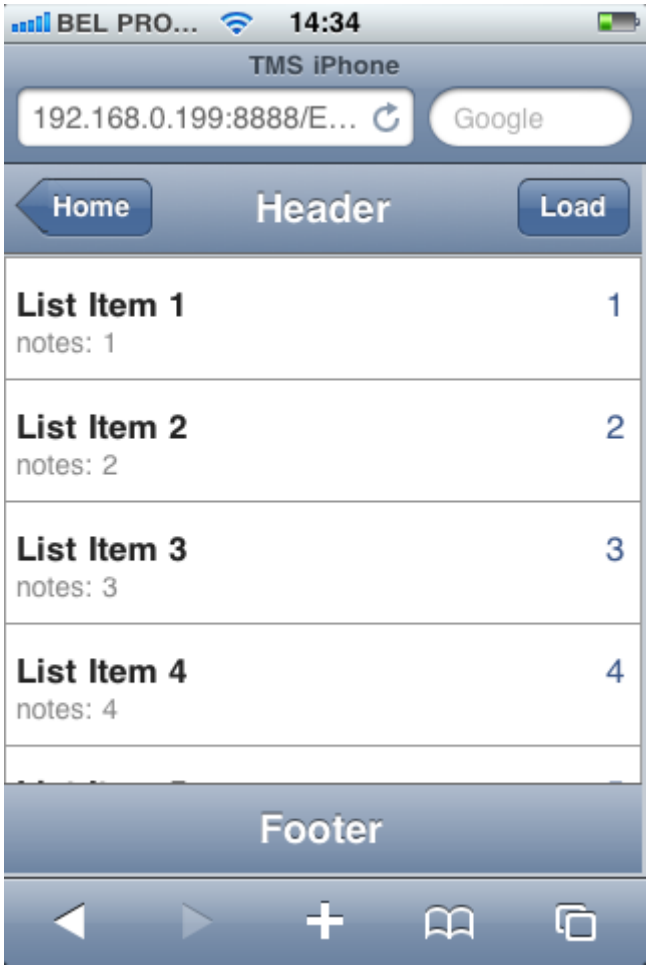


6. Now it’s time to add some items to the TTIWiPhoneList. Add the following code to the TTIWiPhoneHeader.OnRightButtonClick:

```
procedure TIWForm1.TIWIPhoneHeader1RightButtonClick(Sender: TObject);
var
  I: integer;
begin
  TIWIPhoneList1.Items.Clear;
  TIWIPhoneList1.ItemAppearance.Height := 50;
  TIWIPhonelist1.ItemAppearance.NotesMargins.Left := 0;
```

```
  for I := 0 to 10 - 1 do
  begin
    with TIWIPhoneList1.Items.Add() do
    begin
      Caption := 'List Item ' + IntToStr(I+1);
      Value := IntToStr(i+1);
      Notes := 'notes: ' + IntToStr(I+1);
    end;
  end;
end;
```

7. That's all there is to it! Hit the Run (F9) button and then type in the correct URL in your iPhone Safari browser (don't forget to add the port number if you're using a standalone application).
8. Press the Load button and watch the list items appear on the screen:



List of included components


TTIWiPhoneButton


TTIWiPhoneEdit


TTIWiPhoneEmailLabel


TTIWiPhoneFooter


TTIWiPhoneGeolocation


TTIWiPhoneHeader


TTIWiPhoneList


TTIWiPhoneLocationLabel


TTIWiPhoneMenu


TTIWiPhoneOnOffButton


TTIWiPhonePageFlip


TTIWiPhonePageTransition


TTIWiPhonePhoneLabel


TTIWiPhonePopup


TTIWiPhoneRegion


TTIWiPhoneScrollRegion


TTIWiPhoneSMSLabel


TTIWiPhoneSpinner


TTIWiPhoneStyle


TTIWiPhoneTrackbar

TTIWiPhoneButton

TTIWiPhoneButton description

Fully customizable iPhone style button.

TTIWiPhoneButton features

- Button in iPhone style with rounded corners
- Full webkit based rendering, no images used
- Asynchronous events & client-side events
- Optionally add image in button

TTIWiPhoneButton use

Appearance: Change the button background and border color settings. The button background color consists of 2 gradients, a top and bottom (mirror) gradient and this color can be set for normal, hot and disabled state. The default color values resemble the standard iPhone button appearance.

ButtonType: Set the button display type (btBack, btDefault, btRound, btSquare). btDefault sets the type as a standard iPhone button. btRound & btSquare set the button as a rounded rectangle and straight rectangle respectively. The btBack type is a button with arrow shape on the left side.

Caption: Sets the text for the button

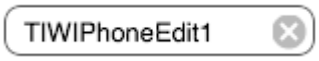
ImagePosition: Indicate if the image should be displayed before or after the caption text

ImageURL: Specify the image to display in the button

ScriptEvents.Click: allows adding JavaScript code that will be executed when the button is clicked

The button provides two server side events: OnButtonClick and OnAsyncButtonClick. When the button is clicked and only the OnAsyncButtonClick event is assigned, it will not cause a full page refresh.

TTIWiPhoneEdit



TTIWiPhoneEdit description

Fully customizable iPhone edit with action button.

TTIWiPhoneEdit features

- Edit in iPhone style with rounded corners
- Full webkit based rendering, no images used
- Configurable action button
- Asynchronous events & client-side events

TTIWiPhoneEdit use

AutoCapitalize: When true, enables the browser's auto-capitalization feature (iOS only)

AutoComplete: When true, enables the browser's auto-completion feature (if available)

AutoCorrect: When true, enables the browser's auto-correction feature (iOS only)

ButtonClick: Allows adding Javascript code that will be executed when the button is clicked

ButtonColor: Sets the color of the button

ButtonType: Set the button display type.

By default the ButtonType is set to ebtDelete. The delete button is only displayed when the edit is not empty. Clicking the delete button clears the text in the edit unless the OnButtonClick or OnAsyncButtonClick event is assigned.

- ebtBookmark: Displays a star shaped button
- ebtCustom: Displays the image defined in ButtonImageURL
- ebtDelete: Displays a delete button
- ebtRefresh: Displays a refresh symbol
- ebtSubmit: Displays a submit symbol

ButtonImageURL: Specify the image to display as the button, when ButtonType is set to ebtCustom

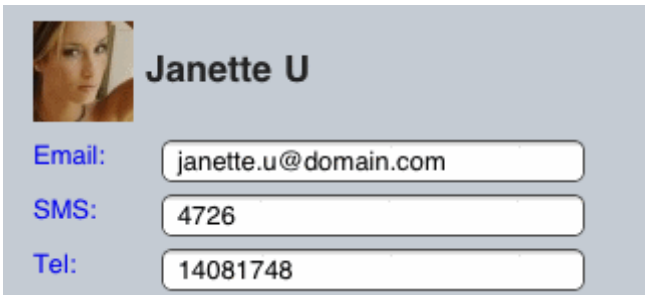
KeyboardType: Sets the keyboard type that is displayed when the Edit receives focus (iOS only)

- ktDefault: Displays the default keyboard
- ktEmail: Displays an email keyboard
- ktNumeric: Displays a numeric keyboard
- ktPhone: Displays a telephone keypad
- ktURL: Displays an URL keyboard

The button provides two server side events: OnButtonClick and OnAsyncButtonClick. When the

button is clicked and only the OnAsyncButtonClick event is assigned, it will not cause a full page refresh.

TTIWiPhoneEmailLabel, TTIWiPhoneLocationLabel, TTIWiPhonePhoneLabel & TTiWiPhoneSMSLabel



TTIWiPhoneLabels description

Various label controls that invoke phone, SMS, email and maps functions on the iPhone.

TTIWiPhoneLabels features

TTIWiPhonePhoneLabel

- Label starting iPhone dialer app with predefined phone number

TTIWiPhoneLocationLabel

- Label starting iPhone maps app with predefined location and optional predefined destination

TTIWiPhoneEmailLabel

- Label starting iPhone email app with predefined email and optional predefined subject, body text, CC email and BCC email

TTIWiPhoneSMSLabel

- Label starting iPhone SMS app with predefined phone number

TTIWiPhoneLabels use

TIWiPhonePhoneLabel: invokes the phone dialer with the value set by IWiPhonePhoneLabel.TelephoneNumber

TIWiPhoneLocationLabel: invokes the maps app with the value set by IWiPhoneLocationLabel.Location (or IWiPhoneLocationLabel.Latitude and IWiPhoneLocationLabel.Longitude) and optionally the IWiPhoneLocationLabel.Destination (or IWiPhoneLocationLabel.DestinationLatitude and IWiPhoneLocationLabel.DestinationLongitude).
Note:
The Location/Destination typically contains a street address or country name, etc... The Latitude and Longitude contain the geographic coordinates of the location.

TIWiPhoneSMSLabel: invokes the iPhone SMS app with the value set by IWiPhoneSMSLabel.SMSNumber

TIWiPhoneEmailLabel: invokes the iPhone email app with the email address set by IWiPhoneEmailLabel.EmailAddress and optionally the IWiPhoneEmailLabel.CCAddress,

IWiPhoneEmailLabel.BCCAddress. In addition, the email subject and body can optionally be preset with IWiPhoneEmailLabel.Body and IWiPhoneEmailLabel.Subject.

If no caption text is specified, the predefined phone number (TTIWiPhonePhoneLabel, TTIWiPhoneSMSLabel), the predefined email address (TTIWiPhoneEmailLabel) or the predefined location (TTIWiPhoneLocationLabel) will be displayed instead.

Example:

```
IWiPhonePhoneLabel.Caption := 'My friend';
IWiPhonePhoneLabel.TelephoneNumber := '001 800 123456';
```

This will display as "My friend" and when clicked will start the phone dialer with number 001 800 123456.

TTIWiPhoneGeolocation

TTIWiPhoneGeolocation description

Retrieves the current geographic location.

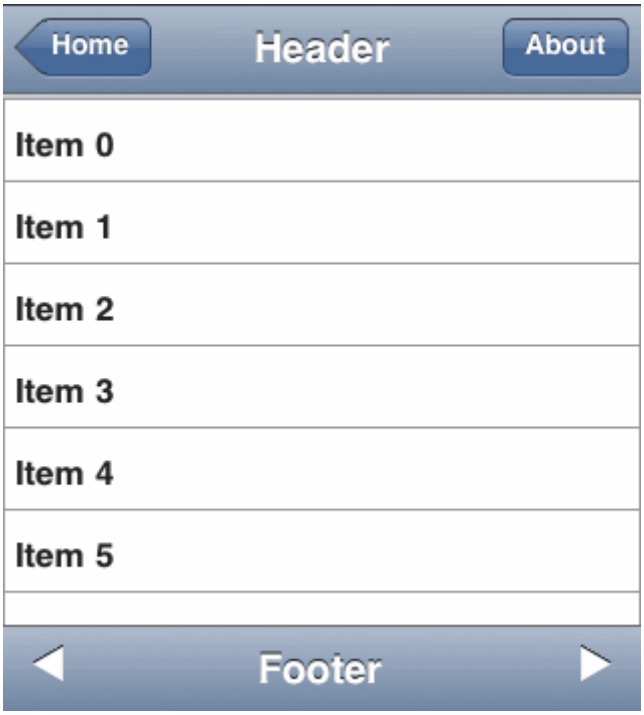
TTIWiPhoneGeolocation features

- Non visual component to retrieve the current geographic location
- Retrieve location as a street address or as latitude/longitude coordinates
- Works asynchronously

TTIWiPhoneGeolocation use

- Call the TIWiPhoneGeolocation.RetrieveLocation method (for example through a button's OnAsyncClick event).
- When the location has been found, the TIWiPhoneGeolocation.OnAsyncLocationRetrieved event is triggered. The current geographic location is now available from the event's parameters: Location (street address), Latitude and Longitude (geographic coordinates).

TTIWiPhoneHeader & TTIWiPhoneFooter



TTIWiPhoneHeader & TTIWiPhoneFooter description

Customizable iPhone application header control & iPhone application footer control. The header typically consists of up to two buttons and a header text. The footer typically consists of a graphical element on the left and right and a footer text.

TTIWiPhoneHeader & TTIWiPhoneFooter features

TTIWiPhoneHeader

- Asynchronous updates, asynchronous events
- Optional button left & right with text and/or image
- Optional arrow shape back button
- No images used for rendering
- Clientside events for button clicks

TTIWiPhoneFooter

- Optional graphic element left & right
- Asynchronous updates, asynchronous events

TTIWiPhoneHeader & TTIWiPhoneFooter use

TTIWiPhoneHeader

BGColor, BGColorTo: set the gradient background start and end color.

Caption: sets the header text

LeftButton / RightButton: Configure the left hand side and right hand side button respectively. See the TTIWIPhoneButton section for further information.

ScriptEvents: contain the JavaScript code that can be specified that is executed when either the left or right button is clicked.

The header provides following events:

OnAsyncLeftButtonClick, OnAsyncRightButtonClick: event triggered when one of the buttons is triggered without causing a full page refresh.

OnLeftButtonClick, OnRightButtonClick: event triggered when one of the buttons is triggered with a full page refresh.

TTIWiPhoneFooter

BGColor, BGColorTo: set the gradient background start and end color.

Caption: sets the footer text

LeftImageURL / RightImageURL: Specify the image to use on the left hand side and right hand side respectively.

ScriptEvents: contains the JavaScript code that can that is executed when either the left or right graphic is clicked.

OnAsyncLeftImageClick, OnAsyncRightImageClick: event triggered when one of the images is triggered without causing a full page refresh.

OnLeftImageClick, OnRightImageClick: event triggered when one of the images is triggered with a full page refresh.

TTIWiPhoneList

TTIWiPhoneList description

Fully customizable iPhone style list control.

TTIWiPhoneList features

- Supports standard list mode & settings mode
- In settings mode, items can be organized in sections
- Image, value, caption and notes per item
- Smooth iPhone style scrolling & scroll indicator
- Asynchronous updates, asynchronous events
- Asynchronously inserting and removing items
- Can show detail in connection with TIWiPhonePageFlip
- No images used for rendering
- Extensive control over appearance: colors, margin, font
- Standard iPhone look & feel colors
- Client-event for item clicks

TTIWiPhoneList use

ListType ltNormal: Display the list as a normal list.
ListType ltSettings: Display the list as a settings list with rounded borders and (optional) sections

Full control to position the Image (1), Caption (2), Notes (3), Value (4), Detail indicator (5) elements by using the ItemAppearance.*Margins properties.

The TIWiPhoneList.ItemAppearance controls how each item in the list looks. BGColor/BGColorTo set the gradient start and end color for an item. The SelectedBGColor, SelectedBGColorTo set the gradient start and end color for a selected item. The fonts for caption, notes and value are also controlled by CaptionFont, NotesFont and ValueFont under TIWiPhoneList.ItemAppearance. The ImageHeight and ImageWidth properties define the size of the Items[].Image. If the size is set to 0, the image is displayed in its full size.

The items in the TIWiPhoneList are organized via a collection: TIWiPhoneList.Items. The item in this collection has following properties:

TiPhoneListItem:
Caption: sets the caption text
Data: additional data associated with the item maintained in a TStringList
Detail: when true, an indicator '>' is shown that indicates a detail screen is available
DeviceOrientation: Enables support to use the IWiPhoneList in the ContentRegion of the IWiPhonePopup control. Set to doAutoHandle to automatically resize the control when the device orientation changes or set to doIgnore to ignore this when used in the IWiPhonePopup ContentRegion
Image: sets the URL of the image to be shown in the item
Name: sets the name of the item
Notes: sets the optionally displayed notes
Section: when true, the item represents a section title in a TIWiPhoneList with ListType = ltSettings
Tag: holds an integer value associated with the item
Value: sets the value of the item as string

The TIWiPhoneList can display a typical "See more" button below the last item if not all collection items are currently displayed in the list (The initial number of items displayed can be set with InitialItemCount). After clicking this button, additional items will be inserted in the list asynchronously (Configure the number of items that are inserted after every click with AsyncLoadCount).

The TIWiPhoneList exposes following methods:

AsyncClear: clears all items from the collection and removes all items from the client-side browser window asynchronously
AsyncItemAdd: inserts the next item from the collection in the displayed items list
AsyncItemsAdd: inserts all remaining items from the collection in the displayed items list
AsyncItemRemove(ItemIndex: integer): removes an item from the displayed list
AsyncItemUpdate(ItemIndex: integer): refresh a single item on the displayed list
For example: after changing the Caption text for item number 3, call AsyncItemUpdate(3) to update the relative item in the list.
AsyncRefresh: refresh the items that are currently displayed in the list
ScrollToTop: programmatically scroll the list to the first item asynchronously

The TIWiPhoneList exposes following events:

OnAsyncExtraItemsLoaded: event asynchronously triggered when additional items have been added after clicking the "See more" button.
OnAsyncImageClick: event asynchronously triggered when the item image is clicked
OnAsyncItemClick: event asynchronously triggered when the item is clicked
OnAsyncLoadExtraItems: event asynchronously triggered when the "See more" button is clicked. If this event is assigned, no additional items will be added this allows for adding custom code.
OnImageClick: synchronous event for click on the item image

OnItemClick: synchronous event for click on the item
OnRenderListItem: event triggered every time an item is ready to be rendered in the browser. The e item properties (Caption, Notes, Value, Image and Detail) can be changed here.

Example:

This adds via code an item to the list:

```
with TIWiPhoneList.Items.Add do
begin
  Caption := 'New item';
  Value := 'value';
  Notes := 'This is the description of the item';
  Detail := true; // show the detail indicator
end;
```

TTIWiPhoneMenu

TTIWiPhoneMenu description

Fully customizable iPhone application menu control.

TTIWiPhoneMenu features

- Collection of menu items with text
- iPhone style status indicator per item
- Asynchronous updates, asynchronous events
- Client-side JavaScript events

TTIWiPhoneMenu use

TTIWiPhoneMenu consists of a horizontal series of menu items. The menu item has an image and a caption text. Optionally, a menu item can show a status indicator (red circular indicator). The status indicator is fully asynchronously updatable. The menu can have one selected menu item, set by TTIWiPhoneMenu.SelectedIndex. The menu items are organized in a collection: TTIWiPhoneMenu.Items. This is a collection of TiPhoneMenuItem instances:

TiPhoneMenuItem
Caption: sets the text for the menu item
Image: sets the URL of the image to be shown in the item
IndicatorCaption: sets the value of the optional status indicator
IndicatorVisible: when true, the indicator is shown at the top right for the menu item
Name: sets the name of the menu item
SelectedImage: alternate image shown for a menu item in selected state
Tag: holds an integer value associated with the item

The TTIWiPhoneMenu exposes following events:

OnAsyncItemClick: event asynchronously triggered when the menu item is clicked
OnItemClick: synchronous triggered event when the menu item is clicked
ScriptEvents.ItemClick: this sets the JavaScript code that will be executed when the menu item is clicked (the item index parameter is available through a JavaScript variable called “index”)

Example:

This code snippet shows how a click on the 2nd menu item asynchronously sets the indicator on the first menu item:

```
procedure TIWForm1.TIWIPhoneMenu1AsyncItemClick(Sender: TObject;
        EventParams: TStringList; ItemIndex: Integer);
```

```
begin
   if itemindex = 1 then
   begin
      TIWIPhoneMenu1.Items[0].IndicatorCaption := '*';
      TIWIPhoneMenu1.Items[0].IndicatorVisible := true;
   end;
end;
```

TTIWiPhoneOnOffButton

TTIWiPhoneOnOffButton description

iPhone style toggle button.

TTIWiPhoneOnOffButton features

- On/off toggle button in iPhone style with rounded corners
- Animation when toggling state
- Three built-in styles: normal, system, custom
- States can be represented by text
- Asynchronous updates, asynchronous events
- No images used for rendering

TTIWIPhoneOnOffButton use

TTIWiPhoneOnOffButton is an iPhone style toggle button with two states. The text for the on and off state is set with properties TTIWiPhoneOnOffButton.OnCaption and TTIWiPhoneOnOffButton.OffCaption. The colors of the button can be configured under TTIWiPhoneOnOffButton.Appearance. The default settings of the TTIWiPhoneOnOffButton represent the look and feel of the standard iPhone toggle button. This is also the style set when TTIWiPhoneOnOffButton.ButtonType = btDefault. Alternatively, the ButtonType = btSystem selects the iPhone system toggle button with orange thumb. Custom colors can be set and will be used when ButtonType is btCustom.

Notification when the button state changes is done via three methods:

ScriptEvents.Click: this sets the JavaScript code that will be executed when the button state changes
OnAsyncButtonClick: asynchronous triggered event when button is clicked to change state
OnButtonClick: synchronous triggered event causing full page refresh when button state changes

TTIWiPhonePageFlip & TTIWiPhonePageTransition

TTIWiPhonePageFlip & TTIWiPhonePageTransition description

Controls that allow switching between two associated regions using various animation types. While the TTIWiPhonePageFlip is limited to switching between two regions, the TTIWiPhonePageTransition can make a transition between any number of regions.

TTIWiPhonePageFlip & TTIWiPhonePageTransition features

- Webkit based animation between regions
- Client-side animation, asynchronous updates
- Different animation types configuration

TTIWiPhonePageFlip & TTIWiPhonePageTransition use

TTIWiPhonePageFlip

Configure the TTIWIPhonePageFlip control:

Add two TTIWIPhoneRegion controls to the form.
Add the required controls to both regions.
Assign the front (first) region to the FrontRegion property.
Assign the back (second/detail) region to the BackRegion property.

AnimationSpeed: Change the duration (in ms) of the animation.
AnimationType: Change the type of animation.
ActiveRegion: Sets if the FrontRegion or BackRegion is displayed on the first page render.

Switch between FrontRegion and BackRegion:

Starting the animation to switch from one assigned region to another region can be done in JavaScript code or via handling an asynchronous event:

Using an async event from any control on your form:
Call TTIWIPhonePageFlip.SlideToBack to switch to the BackRegion.
Call TTIWIPhonePageFlip.SlideToFront to switch to the FrontRegion.

Using a ScriptEvent from any control on your form:
Add the string CONTROLIDback(); to your script event to switch to the BackRegion.
Add the string CONTROLIDfront(); to your script event to switch to the FrontRegion.
(Where “CONTROLID” is the HTMLName of the TTIWIPhonePageFlip control in uppercase)

TTIWiPhonePageTransition

Configure the TTIWIPhonePageTransition control:

Add two or more TTIWIPhoneRegion controls to the form.
Add the required controls to the regions.
Add an item to the Regions collection for every region and assign the respective region to the IPhoneRegion property.

TransitionSpeed: Change the duration (in ms) of the transition.
TransitionType: Change the type of transition.
ActiveRegion: Set the index of the region that is displayed on the first page render.

Switching between regions:

Starting the transition to switch from one assigned region to another region can be done in JavaScript code or via handling an asynchronous event:

Using an async event from any control on your form:
Call TTIWIPhonePageTransition.TransitionToNext to switch to the next region in the collection.
Call TTIWIPhonePageTransition.TransitionToPrevious to switch to the previous region in the collection.

Using a ScriptEvent from any control on your form:
Add the string CONTROLIDnext(); to your script event to switch to the next region.
Add the string CONTROLIDprevious(); to your script event to switch to the previous region.
(Where “CONTROLID” is the HTMLName of the TTIWIPhonePageTransition control in uppercase)

TTIWiPhonePopup

TTIWiPhonePopup description

iPhone / iPad style popup control

TTIWiPhonePopup features

- Asynchronous updates & events
- Full webkit based rendering, no images used
- Built-in configurable collection of iPhone / iPad style buttons
- Can host custom content by adding a region that can contain any control
- Extensive control over positioning

TTIWiPhonePopup use

Initializing the TTIWiPhonePopup control:

- Add the required buttons with their Caption text to the Buttons collection
- Optionally configure a popup Caption text
- Optionally configure the positioning ArrowPosition and PositionType properties
- Initially the TTIWiPhonePopup will be hidden. Set the ShowPopup property to true using an asynchronous or synchronous event of any other control on the form to display the popup. Set the ShowPopup to false to hide the popup.

ArrowPosition:
Used to configure the position of the arrow on the side of the TTIWiPhonePopup.
This property also determines on what side of the control set in the PositionControl property the TTIWiPhonePopup is displayed.

PositionType:
Used to determine if the TTIWiPhonePopup is positioned based on it's absolute Top and Left values (ptAbsolute) or based on the position of the PositionControl (ptAtControl)

Example:
- Set PositionType to ptAtControl
- Set PositionControl to an IWEdit control on your form
- Set ArrowPosition to apTopLeft to display the popup at the bottom left corner of the IWEdit control and the arrow will be positioned in the top left corner of the popup, pointing at the IWEdit control.

ContentRegion:
Select any IWRegion on your form to display it's content in the TTIWiPhonePopup.

Note: the buttons from the Buttons collection will be hidden when a ContentRegion is assigned.

Example:
Add a TTIWiPhoneSpinner control on the ContentRegion to create a DatePicker or TimePicker control.

OnAsyncButtonClick, OnButtonClick, ScriptEvents.ButtonClick:
These events will fire when a button has been clicked and provide a button index parameter.
(For the ScriptEvens.ButtonClick clientevent the button index parameter is available through a JavaScript variable called "index")

Tip: Show or hide the TTIWiPhonePopup by using ScriptEvents:
You can use JavaScript to show or hide the TTIWiPhonePopup with the following calls:

```
POPUPIDShow();
POPUPIDHide();
```

(Where "POPUPID" is the Name of the control in uppercase)

Note: when using asynchronous or JavaScript events to display the TTIWiPhonePopup, remember to set the RenderInvisibleControls property of the form or region the TTIWiPhonePopup is placed on to true.

TTIWiPhoneRegion

TTIWiPhoneRegion description

This region has the size of the standard iPhone/iPad screen. This allows designing the region in the Delphi IDE taking the sizes of the target screen in account.

TTIWiPhoneRegion features

- Region for easy design time configuration of iPhone/iPad size screens

TTIWiPhoneRegion use

Using TTIWiPhoneRegion is similar to using a standard IntraWeb TIWRegion. The advantage of the TTIWiPhoneRegion is that it shows in the IDE the size of the region as it will show on the iPhone/iPad as well as allowing specifying a runtime only client-alignment of the region with a non-aligned design time behavior. This non-aligned design time behavior allows putting multiple regions on a single Delphi form at design time and at runtime switch between client-aligned regions.

Specify the TTIWiPhoneRegion size by setting TTIWiPhoneRegion.Device to diPad or diPhone and TTIWiPhoneRegion.DeviceOrientation to doHorizontal or doVertical.

TTIWiPhoneScrollRegion

TTIWiPhoneScrollRegion description

Region for displaying scrollable content.

TTIWiPhoneScrollRegion features

- Smooth iPhone style scrolling & scroll indicator

TTIWiPhoneScrollRegion use

Using TTIWiPhoneScrollRegion is similar to using a standard IntraWeb TIWRegion. The advantage of the TTIWiPhoneScrollRegion is that it allows clipping of the content and provides smooth iPhone style scrolling and scroll indicator.
It's typically used client aligned inside a TTIWiPhoneRegion which can already contain a TTIWiPhoneHeader and TTIWiPhoneFooter or TTIWiPhoneMenu. Thanks to the scrolling functionality the Header and Footer or Menu will remain at a fixed position.

EnableHorizontalScrolling: Vertical scrolling is always enabled, while horizontal scrolling is disabled by default but can be activated by setting the EnableHorizontalScrolling property to True.

The TTIWiPhoneScrollRegion exposes following method:

ScrollToTop: programmatically scroll to the top of the region

TTIWiPhoneSpinner

TTIWiPhoneSpinner description

The TIWIPhoneSpinner is a web implementation of the native iOS date/time selector wheel control.

TTIWiPhoneSpinner features

- Asynchronous updates & events
- Full webkit / HTML5 based rendering, no images used
- Configurable collection of Slots
- Built-in date and time selector
- Smooth iPhone style scrolling

TTIWiPhoneSpinner use

Initializing the TTIWiPhoneSpinner control:

- Add Slots to the Slots collection and add the required Items for each Slot
- Set the SelectedIndex for each Slot
- The Slots are automatically sized relative to the content of the Slot itself and the width of the TTIWiPhoneSpinner control

OnAsyncScrolled, ScriptEvents.Scrolled:
These events will fire when a slot has scrolled and provide a slot index and item index parameter. (For the ScriptEvens.Scrolled clientevent the parameters are available through JavaScript variables called “slotIndex”, “itemIndex” and “itemValue”)

Date and time selector:
Other than a general purpose single or multi slot spin control, the TIWIPhoneSpinner can also be configured to be used as a date or time selector. This can be done with the TIWIPhoneSpinner.Mode property that offers following presets: smDateDMY, smDateMDY, smDateYMD, smTimeHM, smTimeHMS. Without using any code, the spinner is this way immediately set up as a day, month, year or month day year or hour, minute or hour, minute, seconds spinner.

TTIWiPhoneStyle

TTIWiPhoneStyle description

Holds meta tags for controlling if an application runs full screen, defines the optional application icon on the iPhone and the optional splash screen.

TTIWiPhoneStyle features

- Non visual component to define different global iPhone application settings
- Can define iPhone/iPad application button icon
- Can define iPhone/iPad application splash screen

TTIWiPhoneStyle use

Drop the TTIWiPhoneStyle component on the form. Following control is possible:

BGColor, BGColorTo: set the application gradient background start and end color.
Font: the font settings will be applied to all other controls on the form, except for controls where the font settings are already configured on the control itself.
FullScreen: Boolean; when true, the application will run full screen if started from an iPhone button
HideAddressBar: Boolean; when true, the browser address bar is hidden
IconImage: sets the URL for the image that will be used for the iPhone button from where the web application can be started.
StartupImage: set the URL for the image that can be used as splash screen to start the application
StatusBarStyle: allows selecting a default color or semitransparent black statusbar on the iPhone
UsePrecomposedIcon: Boolean; when true, the image specified in IconImage will not be changed. When false, the image specified in IconImage will have the typical iPhone button rounded borders and glow effect applied.

Notes:
In IntraWeb 10 fullScreen can only be used when the IntraWeb application is deployed as ISAPI dll. This is because only in this configuration, the URL from where the application is started can remain fixed. This is the URL that the iPhone decides to use when it is selected to add the web application to the iPhone start buttons. The URL is typically set after the application has started and when the standalone server is used, the URL changes from the initial URL to URL/EXEC.
In IntraWeb XI this issue has been resolved.

FullScreen, IconImage, StartupImage and StatusBarStyle can only be used when the application is started from an iPhone button (i.e. this will not work when the application is started from a link or bookmark in the iPhone browser).

HideAddressBar can only be used when Fullscreen is false.

TTIWiPhoneTrackbar

TTIWiPhoneTrackbar description

iPhone style trackbar.

TTIWiPhoneTrackbar features

- Asynchronous updates & asynchronous track events
- Full webkit based rendering, no images
- Display the dynamically updated position value

TTIWiPhoneTrackbar use

The TTIWiPhoneTrackbar is very similar to a standard VCL TTrackBar. It has a MinimumValue and MaximumValue and a ThumbPosition. The thumb can be moved between the minimum and maximum. Optionally a automatically updated label with the position value can be displayed when ShowValue is true, the value will be formatted and positioned using the settings from ValueFormat and ValuePosition respectively. The thumb position can be updated synchronously and asynchronously. Changes in the trackbar at runtime trigger several events:

JavaScript events with event handlers defined under ScriptEvents:
Drag: holds the JavaScript executed while the thumb is being dragged.
EndDrag: holds the JavaScript executed when the thumb is released
StartDrag: holds the JavaScript executed when the thumb is first clicked to start dragging

Asynchronous events:
OnAsyncDrag: asynchronous event triggered while the thumb is being dragged.
OnAsyncEndDrag: asynchronous event triggered when the thumb is released
OnAsyncStartDrag: asynchronous event triggered when the thumb is first clicked to start dragging