Page 1

#Criando Web Applications with Delphi - uniGUI

Page 2

#Criando Web Applications with Delphi - uniGUI

summary

Knowing the uniGUI Components Presentation
Presentation of the components of palettes Standard, Additional, Data Controls and Extra's uniGUI.

Exploring the Demos
Exploring the Demos uniGUI.

Knowing the Ext JS
Presentation of Sencha Ext JS framework.

licenses
Understanding uniGUI licenses.

Working with Digital Certification Deploying Projects

Application Direct Access Bank
Creating an application with direct Bank. to

Consuming Web Services
Creating an application that consume services of a web service.

Migrating an Application
Desktop migrating an application to with uniGUI.

#Criando Web Applications with Delphi - uniGUI

Page 4

#Criando Web Applications with Delphi - uniGUI

Presentation

The Framework uniGUI

The acronym means uniGUI unified Graphical User Interface. That is, it is an interface unified graphics for development applications.

It is a framework for developing Ajax Web applications using Delphi. With the uniGUI is possible to develop applications RAD. What is it? According to wikipedia:

Rapid Application Development (RAD) or Application Rapid development (in Portuguese) is a process model development of interactive software and incremental what one emphasizes cycle extremely short development. RAD considers the use of fourth generation techniques, It works with the reuse of components existing program when possible, or create components Reusable. Are used automated tools to facilitate software construction.

That is, you continue to build your Applications with good old Delphi WYSIWYG. But what is this Note the definition given by wikipedia:

WYSIWYG is an acronym of the English "What You See Is What You translation refers to something like what you get "(OQVVEOQVO). It *capacity of a computer program allowing a document, and manipulated on the screen, has the of use. ppearance*

It is clear then that you will draw the program in Delphi and it will appear in browser in the same way for the user, and that the program will be converted an Ajax Web application.

Page 5

#Criando Web Applications with Delphi - uniGUI

Presentation

The Framework uniGUI

With uniGUI components, we can say the barrier between development Desktop and web at Delphi was finally broken.

There are web applications that run "Client side", i.e., are completely to be written

Web can become a nightmare of those.

To browse the Internet you must have a system installed on your computer, browser or *browser*. The pages we access when we surf the Internet They are also systems.

Some say that a simple static page made using only HTML is not a system, but let's assume that each and any application that runs in a *browser* is a web application.

It provides the site.

For example, when you access the site bank (financial institution), so that data from your current account appear screen, it is necessary for the queries on other database financial, carrying out the necessary security, and then display the data requested.

Page 6

#Criando Web Applications with Delphi - uniGUI

Presentation

The Framework uniGUI

This part of the application which the consultations Bank in data it is not interpreted by the *browser*. Such consultations are held in the company's server, which returns the result already in HTML format It is displayed by the *browser*. We say that this part of the application runs on the "server side".

And then we arrived in the "bad". take that desktop application made in Delphi and pass web is not possible. The principle is need to create another application using the "soup in letters" what we mentioned previously: HTML, CSS, Java Script, PHP, etc.

Some frameworks have been developed to Delphi to make it possible to break this barrier. The uniGUI is one of them. Whereby seems is the best!

THE uniGUI promise is Web application development for a new dimension. in that application made with uniGUI is Same time! the Web

Of that form, the can build and debug your application op like a normal desktop application. or Later choose one of the options available to publish your application in web.

With uniGUI you can create an VCL for desktop or web with the same code.

Presentation

The Framework uniGUI

For the uniGUI work, it
use other frameworks.

To build the "face" of
web application is used
JavaScript framework Ext JS.

To perform the conversion between
Pascal and JavaScript is used to
ExtPascal.

O final result It is an
high-level web application
not far behind others
developed with code
native.

In the next picture we can
see the same application running
on the desktop and the web.

Page 8

#Criando Web Applications with Delphi - uniGUI

Installation

Procedure for Installation

Before installing a new version of uniGUI,
it is important to remove the version of the package
current, if there is already an installed. Soon
then uninstall uniGUI by Panel
Windows Control.

Follow the step by step procedure for
installation of a new version of uniGUI.

2-After download to your machine,
run the installer and accept the
license to use, as the following image.

site. You must register in the forum uniGUI
and access the downloads section. It will be possible
then perform the download of the trial version. Up until
the time of writing this book, this is
standard procedure to download the uniGUI.

Page 9

#Criando Web Applications with Delphi - uniGUI

Installation

Procedure for Installation

3-Select the folder for installation. The standard
is *[ProgramFiles] \ Fmsoft \ Framework *.

4-Select versions of Delphi where you
You want to install uniGUI. Be sure
Delphi is not running during the
installation process. It is possible that
screen does not appear during the
process.

Page 10

#Criando Web Applications with Delphi - uniGUI

5-Press the Install button to complete the installation process.

6-Run Delphi and open in the folder the project for your version of Delphi. Example: uniGUI2016PlusGroup (Delphi XE6).

Page 11

#Criando Web Applications with Delphi - uniGUI

Installation

Procedure for Installation

7-In the Project Manager there are 11 packages. Proceed with Build in all packages starting with *SynEdit_Rxxxx.bpl.*

8-After building all packages, add BPLs in Delphi, the following order:

- SynEdit_D20xx.bpl
- uniGUIxxdcl.bpl
- uniGUIxxChartdcl.bpl

Several components will be installed. O Delphi will now have four new palettes uniGUI: Standard, Additional, Data Extra.ols and

Watch the messages relating to the

Page 12

#Criando Web Applications with Delphi - uniGUI

Installation

Procedure for Installation

Page 13

#Criando Web Applications with Delphi - uniGUI

Installation

Procedure for Installation

customer portal to.

Page 14

#Criando Web Applications with Delphi - uniGUI

Knowing the uniGUI Components

Standard palette

The Standard palette uniGUI has 26 components as the image below. She It has some components equivalent to Delphi.

Other components are uniGUI's own. Following is a list of components palette uniGui Standard.

TUniCheckBox. CheckBox utilized at the applications built with uniGUI. To migrate a desktop application, the component TCheckBox It should be replaced by this.

TUniComboBox. ComboBox used in applications built with uniGUI. To desktop application, the component It should be replaced by this.

Page 15

#Criando Web Applications with Delphi - uniGUI

Knowing the uniGUI Components

Standard palette

TUniNumberEdit. Component own uniGUI no equivalent standard in Delphi. This component allows only the input numbers, both integers as decimals.

TUniLabel. Label used in applications made with uniGUI. When migrating an application desktop, TLabel component should be replaced by this.

TUniMemo. Memo used in applications made with uniGUI. When migrating an application desktop, the component must be TMemo replaced by this.

TUniFormattedNumberEdit. Component no standard equivalent uniGUI own in Delphi. This component allows only entry numbers, either as whole decimals. Unlike TUniNumberEdit, the number here can be formatted by FormattedInput property.

TUniPanel. Equivalent to TPanel. project is executed, this component to a with a <table> inside.

TUniTabControl. TabControl used in applications built with uniGUI. TTabControl.

TUniToolBar. Toolbar's uniGUI. Equivalent to TToolBar. the added are the type TUniToolButton.

TUniTreeView. TreeView uniGUI. To migrate a desktop application, the TTreeView should be replaced by this.

TUniStatusBar. UniGUI status bar. when migrating an ap TStatusBar component must be for this.1

Page 16

#Criando Web Applications with Delphi - uniGUI

Knowing the uniGUI Components

Standard palette

TUniTimer. Component equivalent to TTimer. However, this has a more properties. With the property

TUniButton. Button used in made with uniGUI. Equivalent to TButton.

TUniRadioGroup. RadioGroup used in applications built with uniGUI. To migrate a desktop application, the component TRadioGroup It should be replaced by this.

TUniRadioButton. Radio button used in applications built with uniGUI. when migrating a desktop application, component TRadioButton should be replaced by this.

TUniProgressBar. Bar in Progress used in applications built with uniGUI. Equivalent to TProgressBar.

remembering what TUniNumberEdit It has ShowTrigger call that activates a "spin" component.

TUniPageControl. PageControl the migrate a desktop application, the TPageControl should be replaced by this.

TUniMainMenu. MainMenu of migrate a desktop application, the TMainMenu should be replaced by this.

Page 17

#Criando Web Applications with Delphi - uniGUI

Knowing the uniGUI Components

Standard palette

TUniGroupBox. Equivalent component to TGroupBox.

TUniListBox. ListBox used in applications made with uniGUI. When migrating an application desktop, the component must be TListBox replaced by this.

TUniHiddenPanel. Component own uniGUI. All inserted visual component this panel will be hidden from the user. Useful in several as cases we'll see posteriorly.

TUniImageList. ImageList used in applications built with uniGUI. Equivalent to TImageList.

Knowing the uniGUI Components

Additional palette

The Additional palette uniGUI has 18 components as the image below.

There are similar components to those Delphi and other uniGUI's own.

TUniBitBtn. BitBtn used in applications made with uniGUI. When migrating an application desktop, the component must be TBitBtn replaced by this.

TUniDateTimePicker. Component of uniGUI which is equivalent to TDateTimePicker.

TUniImage. UniGUI component that It allows you to display an image to the standard TImage component equivalent Delphi.

TUniSpeedButton. Speedbutton used applications built with uniGUI. To application desktop, TSpeedButton should be replaced by this.

Page 19

#Criando Web Applications with Delphi - uniGUI

Knowing the uniGUI Components

Additional palette

TUniTrackBar. TrackBar utilized at the applications built with uniGUI. equivalent TTrackBar.

TUniSplitter. Splitter used in applications made with uniGUI. When migrating an application desktop, the component must be TSplitter replaced by this.

TUniURLFrame. Component own uniGUI no equivalent in Delphi. Like component you can display a website, browser, for example.

TUniFileUpload. UniGUI component equivalent in Delphi. With this it is possible to upload a file It will be properly handled by the

in the window, this component allows a dialog (window) with a timetable to be called by the user.

performed.

TUniHTMLFrame. Component to create a frame html.

Page 20

#Criando Web Applications with Delphi - uniGUI

Knowing the uniGUI Components

Additional palette

TUniPopupMenu. PopupMenu used in applications built with uniGUI. equivalent TPopupMenu.

TUniHTMLMemo. UniGUI component that It lets you create an HTML editor.

TUniScrollBox. scrollbox used in applications built with uniGUI. equivalent TScrollBox.

TUniCanvas. Own Component uniGUI no equivalent in Delphi. It's possible manipulate the screen to insert shapes and objects or even allow the user to draw something.

TUniMenuButton. Cool Component no equivalent in Delphi. He joins a button with a PopupMenu and creates a common button attachment menu. Very nice and helpful.

TUniGenericControl. Generic Control It will be rendered in the browser as a .

Page 21

#Criando Web Applications with Delphi - uniGUI

The Data Controls palette has uniGUI 16 components as the image below.

There are similar components to those Delphi and other uniGUI's own.

TUniDBEdit. Equivalent to TDbEdit.

TUniDBNumberEdit. There is no equivalent default in Delphi. Would be equivalent to TUniNumberEdit.

TUniDBGrid. Equivalent to TDBGrid.

TUniDBImage. Equivalent to TDBImage.

TUniDBMemo. Equivalent to TDBMemo.

TUniDBHTMLMemo. There is no default in Delphi. Would be equivalent TuniHtmlMemo.

TUniDBNavigator. TDBNavigator.

Page 22

#Criando Web Applications with Delphi - uniGUI

Knowing the uniGUI Components

Palette Data Controls

TUniDBListBox. Equivalent to TDBListBox.

TUniDBFormattedNumberEdit. There is no standard equivalent in Delphi. Would be equivalent to TUniFormattedNumberEdit.

TUniDBComboBox. Equivalent to TDBComboBox.

TUniDBCheckBox. Equivalent to TDBCheckBox.

TUniDBText. Equivalent to TDBText.

TUniDBLookupListBox. Equivalent to TDBLookupListBox.

TUniDBLookupComboBox. Equivalent to TDBLookupComboBox.

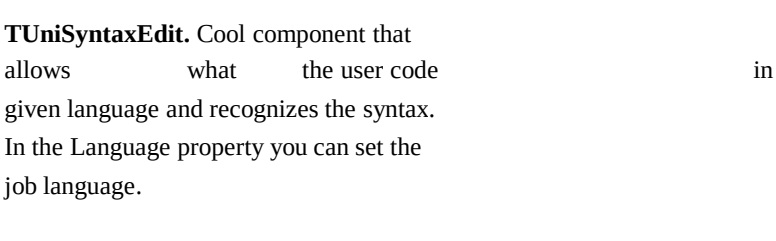
Page 23

#Criando Web Applications with Delphi - uniGUI

Knowing the uniGUI Components

palette Extra

The Extra palette uniGUI has 4 components as the image below.



The components are very interesting, as we shall see later.

TUniSyntaxEdit. Cool component that allows what the user code in given language and recognizes the syntax. In the Language property you can set the job language.

TUniCalendarPanel. How about creating a schedule for your application? Like this component this task will be facilitated by extreme.

TUniThreadTimer. Made to facilitate Work with Threads.

TUniChart. It would be the equivalent of TChart. It contains many graphics as we will see later.

Page 24

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

Introduction

two subfolders called "Desktop" and "AllFeaturesDemo" that brings in a "Touch".

application, the operation of all components.

The folder "Desktop" contains 108 subfolders, or is, are 108 demo applications for testing. Learning through applications demo is an excellent way of understand how things really work.

This demo has 145 forms. We will not explore all here, but we will give those who will help us to understand as uniGUI works.

Page 25

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo

Page 26

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - UniServerModule

By default this creates uniGUI
form inherit from
TuniGUIServerModule. O
goal is to set up several
property of server,
as we can see
on the image, which
shows the Object Inspector.

Properties such as the timeout
session and port
server are configured
on here.

Get used with at
properties and try changes
them. Soon you realize that
You can not change some
of them, as I mentioned
previously. this is
because some properties
They can not be changed in
Trial version.

Page 27

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - UniMainModule

Another form created by
by default when uniGUI
starts a project. The goal is

You can change the picture
background, monitor the use of
keys, change O theme,
disable button click
right of mouse, among
other things.

Analysis with calm each
property and perform the tests
needed to understand
how things work.

Page 28

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - MainForm

Form main gives
application. We can observe
all components
visual are of uniGUI:
UniImageList, UniClockTimer,
UniMainMenu, etc.

This one Form inherits from
TuniForm.

This form is pretty cool
because it creates a tree
with the menu and will create the
windows flaps. Many
wish your system
behave exactly this
way.

If you examine the source code
this form realize the
following details:

Exploring the Demos

AllFeaturesDemo - MainForm

The function will ShowSource
display in the window SourceForm
source code form
is selected (and .DFM
.PAS).

we have The Function
ConstructNavigator what go
build the tree with all
at the options available
application. Hence there
functions accessory for
navigate between tabs, closed
them, etc.

To ramp up an new
application, it would be interesting
enjoy that form
principal, adapting to their
reality.

Page 30

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Types of Forms

The UniFormTypes form brings
eight buttons. The goal here is
understand how the
form in relation to the style of
edge, the edge and display icons.

Edge Style:
1) DialogForm
2) Borderless Form

View:
1) Open Form (nonmodal)
2) Open Form (Modal)

Each option will display a form
different with the characteristic
selected.

Page 31

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Buttons

In the application there are four
forms that display
types of buttons available in
uniGUI:

1) UniButtonMenu
2) UniIconButtons
3) UniSimpleButtons
4) UniToogleButton

There is nothing special in
coding forms. THE
Most of them have nothing
implemented.

The goal is to see which
are the buttons and how they
behave. Analysis with
calm the forms to
how you will use the
buttons on your applications.

Page 32

#Criando Web Applications with Delphi - uniGUI

This one demo contains nine forms showing features graphics.

To work with the graphics, you must add series. O UniChart component contains a property call SeriesList.

Look, per example, O form UniChartArea. Select O component UniChart1 and watch The property SeriesList. She It contains three series.

Now analyze the source code of this form. the three series are handled with random values in the method ApplyChanges.

Page 33

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Charts

Then you ask, "but where I select the type of chart? I searched all properties and not found. "

True, you will not find a property UniChart the component to change the type of graphic. In fact, you will select the type in time to add a new series.

There are eight types available series:

1) Area
2) Bar
3) Gauge

Analyze each sample form for
understand how each type of area and works
how data is displayed to the user.

Page 34

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - TUniFrame

Let's open a parenthesis
here to explain a detail
important.

All the forms
graphics and various other such
application inherit in
TUniFrame.

this is for what O
form can be inserted
the main form of the tab,
because O method
TMainForm.NavTreeClick
instantiates an object TUniFrame
and adds in the tabsheet
Main PageControl.

Study O method
TMainForm.NavTreeClick to
understand how it
it works.

Page 35

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Client Events

And if you want to include a code JavaScript at application? It is possible? Yes!

To understand how this works, let's analyze the examples what were available. Six, according observed at next image.

Page 36

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Client Events | UniButtonClientClick

Open it O form UniButtonClientClick.
Select the UniButton1 button.
Watch what exist an property call ClientEvents.

If you click the "+" sign this property will see the three subproperties what let you add events in side client: Enabled, ExtEvents and UniEvents.

When selecting the Enabled, you You will be activating events.

Adding events?

Page 37

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Client Events | UniButtonClientClick

The following window will appear with a list of possible events. Note that the UniButton1 is with two events added: click and dblclick. Beside you can see the deployed JavaScript code. Analyze other forms to learn how to use the various types of available events. Do it other tests to understand how it really works.

Page 38

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Client Events | UniClientDynamic

You will find UniClientDynamic it button works,

Analysis th
form. ou
e
e
See
You rameCre will
JavaScript being
dynamics. order

Page 39

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Client Events | UniClientAjax

And Ajax calls? Has
how to make? Yes of course!

For understand as
works, you will study the
UniClientAjax form.

Select the first button and
watch what do not exist
events implemented in
ExtEvents. However, see the
which is implemented in
UniEvents. that property
We have access to events
beforInit, afterCreate,
ajaxRequest and ajaxCallback.

In the case of UniButoon1 which
occurs is as follows:

Exploring the Demos

AllFeaturesDemo - Client Events | UniClientAjax

ajaxRequest

```
OnAjaxRequest function (sender, url, date)
{
    sender.setText ( 'Ajax request in progress ...');
    sender.setDisabled (true);
}
```

An ajax request is made (ajaxRequest function "Sender" which is the its amended text for request in progress ... ". button is disabled.

TUniClientAjax.UniButton1Click

```
Sleep (3000);
```

in that time run code that is implemented in the unit UniButton1Click seen system will seconds.

ajaxCallback

```
OnAjaxCallback function (sender, response)
{
  sender.setDisabled (false);
  sender.setText ( 'UniButton1');
}
```

Soon after the code is callback (ajaxCallback side). ion to view

This function enables button and change its "UniButton1".

Page 41

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Cookies

The form UniCookiesDemo
It shows how to work with
this feature.

It is a component for
work with cookies?
No. Remember that we have not seen
none component in this
sense.

still restart or terminate
application, access data
session, access parameters,
get customer information
(IP, browser, system
operating), etc.

Page 42

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Database

That's the part that matters to
most people. As
working with database
using uniGUI?

The uniGUI brings seven demos for
what O developer
understand O what It is
need to to use the
framework with banks
data.

Let's look at aspects
major these examples.

This screen we are seeing
side can be studied in
UniDelayedSearch form.

Page 43

#Criando Web Applications with Delphi - uniGUI

To Open O form UniDelayedSearch you go to perceive what exist one ClientDataSet and one DataSource, components standards of Delphi.

So comes the components visual: UniLabel1, UniEdit1, UniDBGrid1 and UniContainerPanel1.

Where the data come from?

Well, is DataSource1 linked to ClientDataSet1. But does not ClientDataSet1 one Provider. The property FileName is also empty.

To understand how it works, let's look the source code.

Exploring the Demos

AllFeaturesDemo - Database | UniDelayedSearch

Open the unit and realize that there are two methods:

UniEdit1Change

This method is performing the According to filter what user enters the Edit.

UniFrameCreate

It is this method that the base data is created. The author takes the file "customer.cds" which

use.

Page 45

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Database | UniDBControls

In the image below we look at the UniDBControls form.

Page 46

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Database | UniDBControls

With this example you can see the use of components of the DB Aware uniGUI:

- UniDBEdit
- UniDBFormattedNumberEdit
- UniDBComboBox
- UniDBNavigator
- UniDBGrid
- UniDBText
- UniDBCheckBox
- UniDBDateTimePicker
- UniDBListBox

Page 47

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Database | UniFishFacts

The following example is UniFishFacts form.

The way to load data follows the pattern seen above, using a file ".CDS".

What we have different this example is to use images on a field type Blob. You can climb a image using O UniFileUpload component. THE Implementation It is good simple. Open the unit and review O method UniFileUpload1Completed.

This method is bound

Page 48

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Database | UniLiveCombo

This example is very interesting.

Note that this form has a ClientDataSet and UniComboBox.

This is not a DB component Aware.

When you run the example, you will notice an hourglass will occur if the search to be a little time consuming and soon then the data will be presented in ComboBox.

So that everything works the way presented you must use the method OnRemoteQuery the UniComboBox.

This method has two parameters: "QueryString: string " and "Result: TStrings. "

The QueryString is the text that the user types in the which will serve as a filter for the query that will be Result is the String list that is loaded in Combo. In the case of this example, a loop is performed ClientDataSet and country names are added, according to the filter.

Page 49

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Database | UniDBLookUp and UniMasterDetail

The following examples continue with scheme to load data files ".CDS". Recalling that the file ".CDS" is the

Already UniMasterDetail loads the data files "Customer.cds" and "Order.cds" that has binding. Are used two ClientDataSets. To perform the master / detail just set the properties MasterSource and MasterFields the ClientDataSet2.

Study the two forms carefully and notice how the implementation is easy.

Page 50

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Database | UniSQLDemo

This example creates an editor It allows you to run queries SQL.

That example no uses ClientDataSet. THE property DataSource gives grid it is configured as follows:

UniMainModule.DataSource5

We can then see that there is a DataModule with some DataSources within it.

When you open the UniMainModule will observe that the connection to the database is being realized with components ADO.

Exploring the Demos

AllFeaturesDemo - Database | UniSQLDemo

We have four connection components: ADOConnection1, ADOConnection2, ADOConnection3 and ADOConnection4.

In the case of UniSQLDemo form, the connection being used is ADOConnection1. The database is used The "fddemo.mdb". Note that the connection is held in the source code of the unit MainModule. The file "fddemo.mdb" is in folder "Files" project "AllFeaturesDemo".

We can also see the use of component UniSyntaxEdit. He is set to SQL language.

We also have a UniTreeView where the name the tables are displayed. When the user select a table, the code "select * from [Table_name] "appears in UniSyntaxEdit.

When the user clicks the Play button, SQL entered code is passed to the property "SQL.Tex" "UniMainModule.ADOQuery6".

Then just open the "ADOQuery6" and They will be displayed.

You can now learn a lot of things with example.

In a later chapter we will see how to an example CRUD with direct access DBMS.

Page 52

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Download and Upload

We have four examples involving download and uploads:

UniAdvancedDownload: We've talked about TUniGUIApplication. This class contains the TUniGUISession. It is through TUniGUISession we have access to the data session. To AdvancedDownload analyze the code, we have to send a file is very easy:

At this point, do the following: below line "UniSession.SendFile FilesFolderPath + 'unipdf.pdf'); "type "UniSession." And wait for *Code* That way you will know exactly what may cause Session of uniGUI.

Get this habit whenever you want to about a class. It's one of the best

Page 53

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Download and Upload

UniDownload: This example is very simple. So much that is called "Simple Download". Recalling that "UniDownload" is the name of form. The name of the unit is "download" and caption that was reported in the menu tree is "Simple Download".

In this example we can see content the memo is saved in a local file previously defined. Only that.

UniUpload: That example uses O UniFileUpload component. It is possible to send an image or file any. If you send a picture, it will be displayed in UniImage component.

If you choose to send any file, the it will be saved on the server. Note the event OnCompleted of component UniFileUpload.

UniSendFile: same AdvancedDownload using the session.

Page 54

#Criando Web Applications with Delphi - uniGUI

Notice in the figure that you can meet many uniGUI components navigating the examples "Form Controls ".

You must take time to analyze each forms to understand how the components work. Note the use of the properties and events. Follow this section forms list:

- UniCalendarDialogDemo
- UniCharEOL
- UniCheckBoxDemo
- UniComboBoxDemo
- UniDateTimeDemo
- UniEditMethods
- UniMaskEdit
- UniSimpleEdits
- UniEditTypes
- UniFormattedEdit
- UniGroupBoxDemo
- UniHTMLEditor
- UniLabelDemo
- UniLayoutManagementDemo
- UniListBoxDemo
- UniMemoDemo
- UniProgressBarDemo
- UniRadioButtonDemo
- UniStatusBarDemo
- UniSyntaxArea
- UniTrackBarDemo

Page 55

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Grids

That section It contains 24 forms.

IT IS one of the sections important because we need work hard with grids.

Study calmly each forms for learn such things as:

- Using CheckBox columns.
- Fix column.
- Creating grouped headers.

UniSendFile: same AdvancedDownload using the session.

Take the time needed
to study these examples.

Page 56

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Mask

The term Mask can fool the developer.
You may think it has to do with
masks such as "## ## 0.00". It is
this type of mask.

The goal here is to show the user that
some task is being performed.

Study the three examples: UniMaskClient,
UniMaskDemo and UniDataMask.

See the UniMaskDemo form the
UniButton4 It contains
call ScreenMask, where
enable this feature, change the
define the message and the target
which in this case is UniDBGrid1.

Study the examples to understand how
using this feature in your applications.

Page 57

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Various

each of the forms.

In the examples next to you will learn how uniGUI works with the HTML canvas, images, menus, messages and panels.

It also has an example running a jQuery remote.

The first step is to see if there is any thing implemented in the code. if you do not there is nothing there, explore the properties of components with calm and attention to find out how things work. After try changing something and run the project to see the result. This is one of the best ways of learning.

Page 58

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Various

In the next picture we can see several other Examples available in the project.

You must run each of them and analyze each of the forms.

In the examples next to you will learn how uniGUI works with:

- Prompts - message requesting a given of user.
- Scroolbox.
- Page Controller.
- Threads and Timers.
- Toolbars - Toolbar.
- TreeView.
- Unicode.
- URLFrames.

Page 59

#Criando Web Applications with Delphi - uniGUI

Exploring the Demos

AllFeaturesDemo - Miscellaneous

In the miscellany section we found 8 forms with several examples.

What we are seeing below is the UniGoogleMaps form. The map is loaded in the component UniHTMLFrame1. Study the ClientEvents property of this component. In the source code form you will observe how the data is loaded into the grid and how the map changes position according to the coordinates.

Other examples are very interesting: Calendar Panel (schedule), Client Info, Curve3D, HTML Frame, HTML5 Audio, Server Statistics and Splitter. Study each form with calm and attention.

If you study the project carefully AllFeaturesDemo have dominated much of the uniGUI.

Page 60

#Criando Web Applications with Delphi - uniGUI

Knowing the Ext JS

Sencha Ext JS

Ext JS is a library used for building interactive web applications using AJAX, DHTML and DOM.

Behind the Ext JS is Sencha. That company has more than 10,000 of the Fortune 100 rely on Sencha solutions

You do not need to learn how to use
because uniGUI does the "dirty work"
convert the application made in
Delphi.

Page 61

#Criando Web Applications with Delphi - uniGUI

Knowing the Ext JS

ExtPascal

"You said I
I'm not going need
learn to use
Ext JS and that
uniGUI does O
'Dirty work' but
as that
it works?".

O uniGUI use the
ExtPascal for this.
The ExtPascal converts
Pascal objects
for objects Ext JS
(JavaScript).

This is a ExtPascal
free project under
GPLv3 license
It is available from
GitHub.

Behind the Ext JS is Sencha. That
company has more than 10,000
of the Fortune 100 rely on
Sencha solutions.

You can get more information on the
of Sencha. Having a company like this
behind the framework gives us a
safety in their use.

You do not need to learn how to use
because uniGUI does the "dirty work"
convert the application made in
Delphi.

Knowing the Ext JS

licenses

The information is from July 2015. It is good check how is the situation on the site of Sencha.

Note that the lowest value is $ 4,340.00. IT IS a minimum package of five licenses. And not It contains all framework resources!

This value is prohibitive for most developers. If we convert the value for our currency to the dollar at R $ stay at R $ 13,454.00!

And now? As the uniGUI can work with Sencha Ext JS?

Page 63

#Criando Web Applications with Delphi - uniGUI

licenses

uniGUI

The company behind the uniGUI is FMSoft Co. Ltd. This company is an official partner of Sencha.

Because of this partnership, can FMSoft distribute OEM copies of Ext JS.

The uniGUI distributes a partial copy of Ext JS. His uniGUI license grants you use and

The address where this information The following is:

http://forums.unigui.com/index.php?/73-unigui-editions /

The concept of licensing and editions still niGUI it is in There will probably be a final

And how much it costs to license uniGUI? Let's go find the answer to that question in the forum the uniGUI. Remember that this information is July 2015. It is good to consult the forum to know if the information still proceeds.

Ext JS.

2) Complete uniGUI
It includes the desktop version based Ext JS. Also, include the mobile based on Sencha Touch.

Page 64

#Criando Web Applications with Delphi - uniGUI

licenses

uniGUI

Each category will be divided in two editions: standard and professional.

- Standad: only accompanies DCUs. No It has source code.
- Professional: Contains part of the source code and in the future may contain all the code source.

According to uniGUI forum, it is possible that there is still an Enterprise version with more resources.

Therefore, we have:

- uniGUI Standard
- uniGUI Professional
- uniGUI Complete Standard
- uniGUI Complete Professional

Currently (July 2015) we are in the uniGUI vendor calls beta. During this phase being the sold in two editions, namely:

uniGUI Pro 680 Dollars

- desktop development for web.
- Sencha Ext JS OEM.
- source code (Units available in version 0.99 and uniG UIxx.
- One-year subscription.

uniGUI Complete Pro 890 Dollars

- desktop development for web and Mobile.
- Sencha Ext JS OEM.
- Sencha Touch Complete OEM Grids and Charts).
- source code (Units uniGUIxxm.dpk - available in 0.99 and following). uniG UIxx.
- One-year subscription.

Page 65

#Criando Web Applications with Delphi - uniGUI

There some paths (paths) what
They must be configured in UniServerModule
before the developer to make the *deployment* of
application.

ExtRoot: the web application needs to know where
are the Ext JS files. The default value is
"[Ext] \", ie they are in the archive
way "<InstallFolder> \ FMSoft \
Framework \ uniGUI \ ext ". This path points
for uniGUI. Since you will not install
uniGUI on the client machine, you must
set this property.

The easiest way to configure this
property is to leave it blank and copy
folder "ext" into the folder where
are executable files of your
application. The blank value on the property
"ExtRoot" will be automatically converted
for "EXT \".

Page 66

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

Configuring Paths

However, for security reasons, it is
better to copy this folder to another location and
leave files in the folder "ext" as
only reading.

ServerRoot: Sets the main way to
all relative paths. If the value stay
blank, the system will point to the folder
application.

For example, in "ExtRoot" you inform
"\ Ext_js" and "ServerRoot" you inform
"C: \ Test", the system will search for files
Ext JS on the path "C: \ test \ ext_js".

Cached data property.

Page 67

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

VCL application

A unique feature and very interesting
the uniGUI is the ability to create applications
VCL that they are both a server
web. This means that each VCL application
created with the uniGUI library contains a
Web server (Web Server).

This server allows multiple sessions
same application through a browser
(Browser). Therefore, you can run the
application usually via desktop and also
the browser and perform the appropriate tests.

This feature should be used only as
the project is being developed for testing
and even to approve the application with
to the client.

If the VCL application is closed, the server
It will also be aborted.

Page 68

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

VCL application

Web server (Web Server).

This server allows multiple sessions
same application through a browser
(Browser). Therefore, you can run the
application usually via desktop and also
the browser and perform the appropriate tests.

This is the standard way to create an application
uniGUI through the wizard. This feature should
only be used while the project is
It is developed for testing and even
to approve the application with the customer.

If the VCL application is closed, the server
It will also be aborted.

[...]

The concept of licensing and editions
still niGUI it is in
There will probably be a final
when released version 1.0 of uniGUI.

Overall, there will be two main
categories:

1) uniGUI
It includes the desktop version based
Ext JS.cha

2) Complete uniGUI
It includes the desktop version based
Ext JS. Also, include the mobile
based on Sencha Touch.

Page 69

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

Standalone server

The standalone server is similar to the application
VCL. By using this option, the form
main application will not be visible on the screen.
However, an icon appears in the taskbar
Windows.

In the case of AllFeaturesDemo, see how the
icon is on the taskbar:

If you double-click on the icon will
a window, as seen below.

The File menu contains only the Close
close the form, but does not end the
application. It "returns" to taskbar.

Page 70

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

Standalone server

The Manage menu has two options: Server Monitor and Shutdown. The second option is obvious, the server will be shut down.

If you select the first option, Server Monitor, notice that the uniGUI will display in browser some information as image below.

The window has three tabs: Status, and license. We can see the status flap we have an open session. It's possible also observe the maximum number of amount of bytes sent, etc.

Page 71

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

Standalone server

In the second tab you can analyze the features: internal memory, CPU load, etc.

Since I am using the trial version,
the screen trial@fmsoft.net email.

Finally, the last tab, we have
information about the product license.

Page 72

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

Standalone server

This type of server is a good choice
when you are not too worried about
availability. As well?

For example, the system runs only within the
business and everyday local server
company is off the night and rewired by
morning. In this case, the standalone server
solves.

From then need not double click on
icon, when restarting the server is
We need to create a shortcut in the folder
boot windows or even on record
windows. Thus, whenever the
computer is turned on, the standalone server
It will run automatically.

And where you configure the application so that it
behave as a standalone server? At the
ServerModule, StandAloneServer property.

Deploying Projects

ISAPI module

What is ISAPI? Let's look at the definition given by wikipedia:

The Internet Server Application Programming Interface - ISAPI is a protocol used by Windows-based computers to run a dynamic application loading a external module in the address space the webserver process.

To use ISAPI, you must create a DLL (Dynamic link library - dynamic link library) that exports some symbols called by the webserver.

These dlls are loaded in the PC memory when starting the server (IIS, Apache or other), making them available to the client.

Deploy your application as a uniGUI ISAPI module is probably the best way of let it available use.

You can use all web servers that support ISAPI extensions.

The uniGUI was tested with the servers: ig

- IIS 5.1
- IIS 6.0
- IIS 7.0
- Apache 2.2

Page 74

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

ISAPI module - IIS 5

You must create a Virtual Directory (Virtual Directory).

Select Next.

Page 75

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

ISAPI module - IIS 5

Enter a name (alias) to the directory virtual, as the following image.

Enter a folder for the new *alias* created.

Page 76

#Criando Web Applications with Delphi - uniGUI

Grant the appropriate permissions. See the permission "Execute (such at ISAPI applications or CGI) "needs to be selected.

After creating the virtual directory, necessary files into the folder informed. Make sure the user IIS has privileges to access the folder informed and others that will be during the execution of your application.

The way to run the application browser is as follows:

http: // localhost / <virtualdirectory> / Love> .dll

See the URL of the demo in uniGUI page:

http://prime.fmsoft.net/demo/ucdemo.dll

Page 77

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

ISAPI module - IIS 6

You must create a pool of applications (Application Pool).

Enter a name for the new pool.

Page 78

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

ISAPI module - IIS 6

In the pool properties created, go to the tab Recycling and uncheck the "Recycle worker processes (in minutes) ".

Also in properties in the Performance uncheck the "Shutdown worker processes after being idle for (time in minutes) ".

Page 79

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

ISAPI module - IIS 6

Page 80

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

ISAPI module - IIS 6

Enter a name (alias) to the directory virtual, as the following image.

Enter a folder for the new *alias* where uniGUI ISAPI module files created by you are located.

Page 81

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

ISAPI module - IIS 6

Grant the appropriate permissions. See the permission "Execute (such at ISAPI applications or CGI) "needs to be selected.

Access the properties of the new virtual set and change the selecting what you created earlier, as the following image.

Page 82

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

ISAPI module - IIS 6

To finish, add your ISAPI extension in the allowed extensions list.

status to Allowed ".

Page 83

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

ISAPI module - IIS 7

The pimeiro step
in IIS 7 is to create
Pool.

Deploying Projects

ISAPI module - IIS 7

Enter a name for the Pool and select "option in Managed Code "in the combobox.

Open Advanced Options (Advanced Settings) and the pool perform the following settings:

- Select "Enabled 32-Bit Applications" to True. That option is available in 64-bit versions of Windows.
- Check "Disable Overlapped Recycle" to True.
- Enter "Regular Time Interval" to 0 (zero).

Page 85

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

ISAPI module - IIS 7

The step is now add an new application, as image next to.

Page 86

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

ISAPI module - IIS 7

Enter a name for the application and set the path (Physical path). This path is where find the your ISAPI module files created in uniGUI.

Page 87

#Criando Web Applications with Delphi - uniGUI

Select the menu Manage Application and click the option advanced Settings as next image.

Page 88

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

ISAPI module - IIS 7

Point the Application Pool option for one that you created in the first step.

Note that you can also adjust the physical path in this window: Physical Path.

Page 89

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

ISAPI module - IIS 7

The next step is to adjust the "Handler Mappings "for the application you just created. Image below.

In the "Handler Mappings", click the button right-click "ISAPI-dll" and select "Edit Feature Permissions ". Image below.

Check the "Run" and press OK.

Page 90

#Criando Web Applications with Delphi - uniGUI

Deploying Projects

ISAPI module - Apache 2.2

be carried out in httpd.conf. Such
file usually gets in the way "... \ Apache
Software Foundation \ Apache2.2 \ conf "

The first step is to uncomment the line

LoadModule isapi_module modules / mod_isapi.so

Then you have to associate the file ".dll"
with the ISAPI module. Add the following line
below gives line what uncommented
previously:

AddHandler isapi-handler .dll

Options Indexes
AllowOverride None cCGI
Order allow, deny
Allow from all
</ Directory>

Finally create a new *alias* for the
created:y

Alias / mywebapps " C: / webapps "

After making changes to the file
httpd.conf, just restart Apache for
the same take effect.

Page 91

#Criando Web Applications with Delphi - uniGUI

Digital certification

SSL

Secure Socket Layer (SSL) is a global standard
in developed security technology
by Netscape in 1994. It creates a channel
encrypted between a web server and a
browser (browser) to ensure that
all transmitted data is confidential
and insurance.

The uniGUI supports SSL in
Standalone servers.

In ISAPI so you must configure the
SSL directly to the chosen server: IIS,
Apache, etc.

The first step is to get the file
certificate required:

- root.pem
- cert.pem

Types of certificate

Digital certificates are grouped in
following types:

digital signature certificates (A1, A2,
and A4) are certificates used for
confirmation of identity on the web,
electronic, online transactions, private
virtual, electronic transactions,
electronic, session key encryption and
signing documents with verification
integrity of their information.

confidentiality certificates (S1, S2, S3
the certificates used for encryption
documents, databases, messaging, and
other electronic information.

Page 92

#Criando Web Applications with Delphi - uniGUI

Digital certification

Types of Certificates

Certificate type A1 and S1

It is the certificate that the generation of the keys encryption is performed by software and its Storage can be done in hardware or repository password protected, encrypted by software. Its maximum validity is one year, and the frequency of publication of the LCR maximum of 48 hours and the maximum period admitted to complete the process Revocation 72 hours.

Certificate type A2 and S2

It is the certificate that the generation of the keys encryption is performed by software and they are stored in Card Intelligent or *Token* , both without capacity generation key and password protected. Cryptographic keys are at least 1024 bits. The maximum validity of the certificate is two years.

Certificate type A3 and S3

It is the certificate that the generation storage of cryptographic keys are made smart card or *token* , both with key generation and protected per pass Cryptographic approved by ICP- Cryptographic keys are at least 1024 bits. The maximum validity of the three years.[5]

Certificate A4 and S4 type

It is the certificate that the generation storage of cryptographic keys are made smart card or *token* , both with key generation and protected per pass Cryptographic approved by ICP- Cryptographic keys are at least 2048 bits. The maximum validity of the three years.[5]

Page 93

#Criando Web Applications with Delphi - uniGUI

Digital certification

Certificates formats

PKCS - *Public-Key Cryptography Standards*

Specifications produced by Laboratorios RSA, in cooperation with developers

First published in 1991 as result of meetings with a small group of public key technology pioneers, the documents

		the RSA algorithm, covering the following aspects: cryptographic encryption schemes, digital signature schemes with appendix, for representing keys.
PKCS # 2	revoked	
PKCS # 3	Diffie-Hellman Key Agreement *Standard*	It describes a method for the implementation of the agreement key Diffie-Hellman. Applied Protocols for establishing secure connections.
PKCS # 4	revoked	
PKCS # 5	Password-Based Cryptography Standard	Provides recommendations for the implementation of password-covering key derivation functions, encryption schemes and message authentication.
PKCS # 6	Extended-Certificate Syntax Standard	Describes the syntax for the extended licenses, consisting of a set of attributes, collectively signed by the certificate issuer. The extend the certification process by providing more information the public key., as well

Page 94

#Criando Web Applications with Delphi - uniGUI

Digital certification

Certificates formats

Pattern	Name	Description
PKCS # 7	Cryptographic Message Syntax Standard	Describes the general syntax for data that can be encrypted, such as digital signatures and digital envelopes.
PKCS # 8	Private-Key Information Syntax Standard	Describes the syntax for information of the private key. It also for encrypted private keys.
PKCS # 9	Selected Attribute Types	Defines the types of selected attributes for use in standards: PKCS PKCS # 8 and PKCS # 10.
PKCS # 10	Certification Request Syntax Standard	Describes the syntax for an application for certification of a public and possibly a set of attributes.
PKCS # 11	Cryptographic Token interface Standard	Specifies an API, called Cryptoki for devices that have information Cryptographic and perform cryptographic functions. Used to *tokens* smart (*smart card*).
PKCS # 12	Personal Information Exchange Syntax Standard	Specifies a portable format for storing or transporting private keys a user, certificates, etc.
PKCS # 13	Elliptic Curve Cryptography *Standard*	Under development.
PKCS # 14	Pseudo-random Number Generation	Under development.
PKCS # 15	Cryptographic Token Information Format Standard	It establishes a standard that allows users to use *tokens* identify themselves to multiple applications.

First published in 1991 as result of meetings with a small group ey technology pioneers, the documents ed. Below is a table that brings a

Digital certification

Certificates formats

PEM format - ***Privacy enhaced Mail***

It is the most common format provided by a B.C. Usually have the following extensions: ".pem", ".crt", "Cer" and ".key". They are Base64-encoded in ASCII format and contains statements "----- BEGIN CERTIFICATE ----- "And" ----- END CERTIFICATE ----- ". The certificates in server, certificates intermediaries, and private keys can all be placed in PEM format. IT IS one DER encoded in Base64.

DER format - ***Distinguished Rules ling***

It is a binary form of a certificate Unlike the PEM ASCII format. It has the extension ".der" but often the extension is used ".cer". as there certificates with the extension ".cer" DER format when in PEM format, the way to know what type it is is open file and check if it is an ASCII or a torque. All types of certificates and private keys can be encoded in DER format. The RSP is usually used the Java platform.

Page 96

#Criando Web Applications with Delphi - uniGUI

Digital certification

File Types - Extensions

The following types refer to extensions of files in Digital certificates:

- CER - Certified CER encoded. Sometimes it is a sequence of certificates;
- DER - DER encoded certificate;
- PEM - encoded Certificate

may contain certificates and public private keys (protected password);
- PFX - Even if P12.

Page 97

#Criando Web Applications with Delphi - uniGUI

Digital certification

obtainment

Here are the steps to get a digital certificate:

1. Choose a Certification Authority (CA) ICP-Brazil;

2. To request the very portal of the AC Internet chosen the digital certificate issuance of individuals (eg e-CPF) and / or legal (eg e-CNPJ). The most traded types are: A1 (Valid for one year - stored in computer) and A3 (valid for up to three years - stored on card or cryptographic token). The AC can also report on applications, costs, payment methods, equipment, necessary documents and other requirements;

3. For the issue of a digital certificate required that the applicant personally a Registration Authority (RA) of Authority Certifier validate the data filled in the request. This process is called validation face and will be scheduled directly AR that will instruct the applicant on necessary documents. Who choose certificate type A3 may receive in AR card or token with certificate digital;

4. The CA and / or RA will notify the procedures to download the certificate;

5. When your digital certificate is close to maturity, it may be renewed electronically, once, without the need for further validation presential.

Page 98

#Criando Web Applications with Delphi - uniGUI

First, it should be remembered that the certificate digital is the "identity" of the person in the world virtual. Thus, the adoption of some is necessary care to prevent anyone else use your digital certificate:

1. The private key password and own private key should not be shared with no one;

2. If the computer that generated the key pair cryptographic be shared with several users, storage is not recommended in private key on the hard disk, for all users They will have access to it, being better storage floppy, smart card or token;

3. If the private key is stored on disk Hard a computer, you should protect it from access not authorized, keeping the physically safe. Never leave the room open when you leave and you You need to leave your computer on. also use a screen saver password. Beware of virus computer, they can damage your key private;

4. If the pair generation software keys allows a choice between whether or protect the private key, it is for access through password. Do not means that anyone who has access to computer can go through the key private, signing contracts and moving bank. In general, it is much easier to than a physically secure computer;

5. Use a sizable password, intercalating letters and numbers, once programs with the function of avoid the use of personal data such as spouse or children, birth dates, adresses, phones, related to the person. Password never should to be noted, and memorization.

Page 99

#Criando Web Applications with Delphi - uniGUI

Digital certification

SSL - uniGUI

Let us recall the certificate files requested by uniGUI: root.pem, cert.pem and key.pem.

Note that all contain the extension ".pem". The uniGUI makes use of Indy and the file ".pem" which It will be used must contain only one certificate. Below we can see the contents of a file ".pem".

----- BEGIN CERTIFICATE -----
MIIB8jCCAV + gAwIBAgIQfjGd2Py0qZJGqdkPiRlDdjAJBgUrDgMCHQUAMBAxDjAM
BgNVBAMTBWVsaXRlMCAXDTEzMDYwMjE3NTA0OFoYDzIxMTMwNTA5MTc1MDQ
MQ4wDAYDVQQDEwVlbGl0ZTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAt3p
pMYHNzUueLZBb1eMrPop6Emta / KLyLaK94vlM1lV / 6ITiuFtuSs9gq0s516s2th7
FUkBpgfQvrb 3b9h10WMca8MTbYrLGL + + dHqRk4jGt / 8GAUeYHkKddk /
3aMdURpTgE2iK / d86C1YdsxqXTxP + UAX /
DAYKKwYBBAGCNwoDBDAtBgNVHREEJjAkoCIGCisGAQQBgjcUAgOgFAwSZWxpdG
RUxJVEUtUFJJTUUAMAkGA1UdEwQCMAAwCQYFKw4DAh0FAAOBgQCDSHm54tMH

Page 100

#Criando Web Applications with Delphi - uniGUI

Digital certification

SSL - uniGUI | Setting the Parameters

Copy the three files (root.pem, and cert.pem
key.pem) to the same folder that
find the executable of your server.

Additionally, copy the DLLs in OpenSSL:
libeay32.dll and ssleay32.dll for that
folder.

Sometimes these DLLs are already present
in the Windows system folder. Keep
the latest version.

You'll find these DLLs with the uniGUI
in the following folders:

[UniGUI Installation Folder] \ .. \ Framework \ uniGUI \ SSL \ dll \ x86
[UniGUI Installation Folder] \ .. \ Framework \ uniGUI \ SSL \ dll \ x64

You will need to perform some changes in
Application ServerModule.

Ssl> Enabled = True
Ssl> SSLOptions-> certfile = cert.pem
Ssl> SSLOptions-> KeyFile =
Ssl> SSLOptions-> RootCertFile =
Ssl> SSLPassword = [password]

Page 101

#Criando Web Applications with Delphi - uniGUI

Digital certification

SSL - uniGUI | Setting the Parameters

In this case you would access the application follows:

https: // localhost: 8077

If you wish, you can use the default port SSL is 443. It does not need to inform the port explicitly in the browser:

https: // localhost

Sometimes you need to have a safe area and another released. Some sites do this: part the site is accessed via "http" and other means "Https". You can do this with uniGUI also.

It would be possible to access the 8077 port normally (http) and the safe "would be accessed through port (Https).

Recalling that should be analyzed if a server using port 443, because it is standard SSL port.

Page 102 ***#Criando Web Applications with Delphi - uniGUI***

Application Direct Access Bank

Development

Let's create step by step an application access a MySQL database.

The first step is to create the database. Open the access tool to the bank MySQL data to create the database. In my case I'm using SQLyog. this example It can also be made with other banks data.

I will create a database called "test".

This database will create a table called "person".

```
CREATE TABLE 'person' (
    'Id' int (11) NOT NULL AUTO_INCREMENT,
    'Name' varchar (50) DEFAULT NULL,
    PRIMARY KEY ( 'id')
) ENGINE = InnoDB
```

Ready. The goal here is to create fast an application that can see the data and also change, delete and add new records. To this end, just have a table simple as this we create.

Page 103

#Criando Web Applications with Delphi - uniGUI

Application Direct Access Bank

Development

Since we are with the bank and created table, we will create our application.

Open Delphi and select: File / New / Other.

Then select the uniGUI group for Delphi Application Wizard option.

Page 104

#Criando Web Applications with Delphi - uniGUI

Application Direct Access Bank

Development

In the wizard you must enter the project name, the folder where the file is saved and select the kind of project among the four available: Standalone Server, Standalone Server / ISAPI Module, ISAPI Module and Windows Service Application.

Page 105

#Criando Web Applications with Delphi - uniGUI

Application Direct Access Bank

Development

The project will be created with the three forms patterns: MainForm, UniMainModule and UniServerModule.
Let's create our application in MainForm, putting that form all components
needed. Following is the list of components that you must enter at MainForm:

- SQLConnection
- SQLTable
- DataSetProvider
- ClientDataSet
- DataSource
- UniDBGrid
- UniDBNavigator
- UniImage

Note the image.

Let's analyze what
properties shall
be configured.

Application Direct Access Bank

Development - Properties - SQLConnection

- Driver: MySQL
- Database: test
- HostName: localhost
- Password: root (see what the password of your server)
- UserName: root
- LoginPrompt: False

If a problem occurs when attempting to activate the component, copy the files and dbxmys.dll libmysql.dll to the application folder.

Page 107

#Criando Web Applications with Delphi - uniGUI

Application Direct Access Bank

Development - Properties - SQLTable

- Active: True
- SQLConnection: SQLConnection1
- TableName: person

We use TSQLTable for convenience to this example. Consider using TSQLQuery in their applications for production.

Page 108

#Criando Web Applications with Delphi - uniGUI

Application Direct Access Bank

Development - Properties - DataSetProvider

DataSet: SQLTable1

It will bridge the gap between SQLTable (which is unidirectional) and ClientDataSet.

Page 109

#Criando Web Applications with Delphi - uniGUI

- Active: True
- ProviderName: DataSetProvider1

Required to load the data into
memory. The DataSource will be bound to it
so that data appears in the grid.

It is the only component which implemented
an event: AfterPost with the following code:

ClientDataSet1.ApplyUpdates (0);

This will cause the data to be updated
the DBMS.

Page 110

#Criando Web Applications with Delphi - uniGUI

Application Direct Access Bank

Development - Properties - DataSource

- DataSet: ClientDataSet1

It could be linked directly
SQLTable. However, it could not display the
data on the components DB Aware (DBGrid,
DBNavigator). Because? Why is SQLTable
unidirectional.

That's why we need to use the Provider and
ClientDataSet.

Page 111

#Criando Web Applications with Delphi - uniGUI

Application Direct Access Bank

Development - Properties - UniDBGrid

DataSource: DataSource1

Just set the DataSource to the data already appear in the grid.

Page 112

#Criando Web Applications with Delphi - uniGUI

Application Direct Access Bank

Development - Properties - UniDBNavigator

run.

Recalling that inserted a UniImage only to "Perfuming" the application. Just I pointed to the AlbertEije logo.

Run the application and see the result!

Page 113

#Criando Web Applications with Delphi - uniGUI

Application Direct Access Bank

Development - Application Runtime

Here is the application running in the browser. Exercise is now enrich application. Test the knowledge gained from the study of AllFeaturesDemo and work on that application.

Page 114

#Criando Web Applications with Delphi - uniGUI

Consuming Web Services Application

project DAV

To show how to consume Web Services with uniGUI, we will create an application that works as a DataSnap server.

We will take advantage of the PAF-ECF windows that displays a list DAV - Document Assistant Sales. Thus it will be possible to observe the operation of a master-application detail. View the form in the image.

Page 115

#Criando Web Applications with Delphi - uniGUI

Consuming Web Services Application

project DAV

What I did was take advantage of the PAF project, remove all other windows, controllers and VOs that had no relationship with the DAV and left the window below as the main because it was the only one He left.

- DAV uniGUI: the project has migrated to uniGUI as a standalone server by accessing the application server to view, enter and change data.
- DAV VCL: the PAF original design, modified to become only the DAV window.
- Server: is the DataSnap server applications prepared for the DAV window.

We will not go into detail about the operation itself of the window and the DAV server applications. If you want to learn how to develop an application server that DataSnap returns the JSON data to client applications, consider making the T2Ti ERP training available in T2Ti.COM site.

We will study this chapter the project DAV uniGUI.

Page 116

#Criando Web Applications with Delphi - uniGUI

Consuming Web Services Application

Project DAV uniGUI

In the picture we can observe the form DAV already converted uniGUI for running in the browser.

We have the following components:

- 2 UniGroupBox
- 2 UniDBGrid
- 3 UniBitBtn
- 1 UniImage
- 2 ClientDataSet
- 2 DataSource

And things Work? It is what let's go to see now.

Consuming Web Services Application

Project DAV uniGUI

The first step is to create the tables. In fact the tables already exist, but within the bank T2TiERP data. So I spent these tables to the database "test" that we created previously. The following images with the structure of the tables for you to create in your bank data. Below we analyze the structure of the table "dav_cabecalho".

Page 118

#Criando Web Applications with Delphi - uniGUI

Consuming Web Services Application

Project DAV uniGUI

How is a master-detail design, we need the child table: "dav_detalhe". In the image below you can analyze the structure of it. Create the two tables in the database "test". If you give another name to the database, change the "MySQL_DBExpress_conn.txt" file in the folder server.

Page 119

#Criando Web Applications with Delphi - uniGUI

Consuming Web Services Application

Project DAV uniGUI

Ready. With the tables created, you can run the server: "T2Ti_ServidorAplicacao.exe". Note that the server name refers to T2Ti ERP Project. It's the same server, but it was changed to only deal with the DAV. I left fewer files so you do not miss to try understand the project. Recalling that the focus is not to understand the DataSnap Server.

Just you realize that this server is returning JSON objects to the client application calling his methods. This is what happens with the DAV_uniGUI Project.

When you run the server, you will see a window similar to the image.

Page 120

#Criando Web Applications with Delphi - uniGUI

We will now study the design methods
DAV_uniGUI.

```
TMainForm.UniFormCreate procedure (Sender: TObject);
begin
   JanelaAtiva: = False;

   ConfiguraCDSFromVO (CDSMestre, TDavCabecalhoVO);
   ConfiguraGridFromVO (GridMestre, TDavCabecalhoVO);

   ConfiguraCDSFromVO (CDSDetalhe, TDavDetalheVO);
   ConfiguraGridFromVO (GridDetalhe, TDavDetalheVO);
end;
```

You will see in this project we have
a variable called JanelaAtiva serving
to test whether the window is really active.
In this case, the event may AfterScrool
load the data in the detail grid without
problems.

At first we leave the inactive window to
you can configure the fields of
ClientDataSets and Grids of headers
According to the information of VOs.

The methods Config
ConfiguraGridFromVO are in unit
Library.

```
TMainForm.CarregaDados procedure;
var
   Filter: String;
begin
   try
      Filter: = 'SITUATION =' +
      TDavController.SetDataSet
      (CDSMestre);

TObjectList <TDavCabecalhoVO>
DAVController.TDAVController '
[Filter], 'GET'));

TController.TratarRetorno
ecalho, True, True, CDSMestre);Cab

      JanelaAtiva: = True;
      CDSMestre.First;
      GridMestre.SetFocus;
   finally
   end;
end;
```

The CarregaDados procedure will
method of server query.

#Criando Web Applications with Delphi - uniGUI

Consuming Web Services Application

Project DAV uniGUI

First we create a filter to load
DAVs whose status is equal to "P"
pending.

Hence we trigger the BuscarLista method
TController, passing as TDAVController
parameter. Everything works via RTTI for the
application can run at the same time
in two or three layers. This already works
so for T2Ti ERP Project. I just
I took the whole structure for the project here
the uniGUI.

After we made the query, the system
will handle the return, displaying data

```
procedure                TMainForm.CD
TDataSet);               (DataSet:
begin
   Then if JanelaAtiva
   begin

TController.TratarRetorno
alho.Items [CDSMestre.RecNo-1abec
True, True, CDSDetalhe);
      GridDetalhe.Refresh;
   end;
end;
```

The following is observed that when
is activated call the CarregaDados
method.

```
TMainForm.UniFormActivate
begindure (Sender: TObject);
```

header, the detail data is loaded automatically.

methods are triggered.

Page 122

#Criando Web Applications with Delphi - uniGUI

Consuming Web Services Application

Project DAV uniGUI

Recalling that the project contains DAV_uniGUI DAVController one, but it is not he who is running.

We are in an application in three layers. Thus, being DAVController run is the server. You can even even open Server and debugs using it to understand how things work.

But let us return our focus to the Project DAV_uniGUI.

```
TMainForm.botaoExcluirClick procedure (Sender: TObject);
begin
    JanelaAtiva: = False;

TController.ExecutarMetodo ( 'DAVController.TDAVControlle
r ',' Delete ', [CDSMestre.FieldByName (' ID '). AsInteger]
'DELETE', 'Boolean');
    CarregaDados;
    JanelaAtiva: = True;
end;
```

The previous method tries to delete a You will get an error, because the excluding DAV returns a message stating that a VAD can not be deleted.

Note that the error message is not properly treated. As an exercise, try treat that message and display it to the user.

For this, consider how we handling exceptions in both Client and server.

Let's look at the codes Enter buttons and Change.

Page 123

#Criando Web Applications with Delphi - uniGUI

Consuming Web Services Application

Project DAV uniGUI

```
    ObjetoVO: = TDavCabecalhoVO.Create;

    TDavCabecalhoVO (ObjetoVO) .IdEmpresa: = 1;
     TDavCabecalhoVO (ObjetoVO) .NomeDestinatario: = 'TEST
Recipient '+ DateTimeToStr (Now);
          TDavCabecalhoVO (ObjetoVO) .CpfCnpjDestinatario: =
'00000000000191';
    TDavCabecalhoVO (ObjetoVO) .Situacao: = 'P';
    TDavCabecalhoVO (ObjetoVO) .TaxaDesconto: = 0;
    TDavCabecalhoVO (ObjetoVO) .Desconto: = 0;
    TDavCabecalhoVO (ObjetoVO) .SubTotal: = 5;
    TDavCabecalhoVO (ObjetoVO) .Value: = 5;
    TDavCabecalhoVO (ObjetoVO) .DataEmissao: = Date;
                TDavCabecalhoVO (ObjetoVO) .HoraEmissao : =
FormatDateTime ( 'hh: mm: ss' Now);

    DAVDetalhe: = TDavDetalheVO.Create;
    DAVDetalhe.IdProduto: = 1;
    DAVDetalhe.DataEmissao: = date;
    DAVDetalhe.Item: = 1;
    DAVDetalhe.Quantidade: = 1;
    DAVDetalhe.ValorUnitario: = 5;
    DAVDetalhe.ValorTotal: = 5;
    DAVDetalhe.Cancelado: = 'S';
    DAVDetalhe.Persiste: = 'S';
```

```
TDavCabecalhoVO (ObjetoVO)
AIL);DavDetalheVO.Add (DAVDet

TController.ExecutarMetodo
er ',' Insert ', [TDavCabecalhoVO
'List');oVO)],' PUT '

    CarregaDados;
    JanelaAtiva: = True;
end;
```

See we have two objects to work with the header and detail. Inside header we have already implemented new.

The work here is to instantiate objects assign values. We are doing it statically, but you could bring such Edits values or other visual controls.

Page 124

#Criando Web Applications with Delphi - uniGUI

Consuming Web Services Application

Project DAV uniGUI

```
TMainForm.botaoAlterarClick procedure (Sender: TObject);
var
    ObjetoVO: TDAVCabecalhoVO;
    DAVDetalhe: TDAVDetalheVO;
begin
    JanelaAtiva: = False;

    ObjetoVO: = TDavCabecalhoVO.Create;

                    TDavCabecalhoVO (ObjetoVO) .id : =
CDSMestre.FieldByName ( 'ID') AsInteger.;
    TDavCabecalhoVO (ObjetoVO) .IdEmpresa: = 1;
          TDavCabecalhoVO (ObjetoVO) .NomeDestinatario : =
'MODIFIED - RECIPIENT TEST' + DateTimeToStr (Now);
          TDavCabecalhoVO (ObjetoVO) .CpfCnpjDestinatario: =
'00000000000191';
    TDavCabecalhoVO (ObjetoVO) .Situacao: = 'P';
    TDavCabecalhoVO (ObjetoVO) .TaxaDesconto: = 0;
    TDavCabecalhoVO (ObjetoVO) .Desconto: = 0;
    TDavCabecalhoVO (ObjetoVO) .SubTotal: = 5;
    TDavCabecalhoVO (ObjetoVO) .Value: = 5;
    TDavCabecalhoVO (ObjetoVO) .DataEmissao: = Date;
                TDavCabecalhoVO (ObjetoVO) .HoraEmissao : =
FormatDateTime ( 'hh: mm: ss' Now);

    DAVDetalhe: = TDavDetalheVO.Create;
                                DAVDetalhe.Id : =
CDSDetalhe FieldByName ( 'ID') AsInteger ;
    DAVDetalhe.ValorUnitario: = 5;
    DAVDetalhe.ValorTotal: = 5;
    DAVDetalhe.Cancelado: = 'S';
    DAVDetalhe.Persiste: = 'S';
    DAVDetalhe.GtinProduto: =
    'DAVDetalhe.NomeProduto: =
'+ DateTimeToStr (Now);JCT TEST
    DAVDetalhe.UnidadeProduto: =
    DAVDetalhe.TotalizadorParcial: =
    'TOT';

TDavCabecalhoVO (ObjetoVO)
AIL);DavDetalheVO.Add (DAVDet

TController.ExecutarMetodo
er ',' Change ', [TDavCabecalhoVO
'Boolean');))],' POST '

    CarregaDados;
    JanelaAtiva: = True;
end;
```

The change of method is quite similar inclusion. We just spent the IDs, They exist and are known for

Page 125

#Criando Web Applications with Delphi - uniGUI

Migrating an Application

project DAV

To facilitate the analysis of migration applications for uniGUI, we will use the same example of DAV Project.

The DAV Project is a normal VCL application. It can work in two or three layers changing the file parameter Layers conexao.ini.

Open DAV VCL project and study your code source. Run the project and perform tests.

Remember that to run this project, you must create the tables views previously. Also, you must run the application server.

Some components used in the project part of the suite Jedi. I made sure to use third-party components to study also this aspect of migration application.

The components used in this example the following:

- 2 TGroupBox
- 2 TJvDBUltimGrid
- 3 TJviBitBtn
- 1 TImage
- 2 ClientDataSet
- 2 DataSource

Have you noticed what has to be done
For components?
first components above shall be replaced by uniGUI components.

And as I did this migration?

First I created a new application for uniGUI as we have seen in the
Direct access to the Bank. " Hence the
properties Height, Width and the
MainForm.

Page 126

#Criando Web Applications with Delphi - uniGUI

Migrating an Application

project DAV

After that I inserted the components:

- UniGroupBox
- 2 UniDBGrid

Note that inside the folder "DAV
is our DAV_uniGUI.dpr project,
we have three subfolders: Controller,
and Util.lule

The ClientDataSets and DataSources I copied and
I glued the DAV VCL project. At that point I
I had two instances of Delphi open,
one with PafEcf.dpr project (DAV VCL) and
another with DAV_uniGUI.dpr project.

After changing the properties of
components remained implement the code
source.

But before that, the other had to include
project files. What files?

It is likely what
comprising the file structure and
folders.

Page 127

#Criando Web Applications with Delphi - uniGUI

Migrating an Application

project DAV

Because the client (VAD VCL) also
one Controller folder? Because both
as the client has a class
TControler for ac
operations, but these classes are
to the client and the server.

To migrate this application to the
I did not have to change anything on
Of course it still works
as an application server, allowing
consumption of its methods and
JSON objects for those who run them,
one VCL application Mobile, a call
directly in the browser or a web
with uniGUI.ade

The idea is the following:

Using the same controller and VOs both
application client and the server.

For this there are folders and the controller
VO within the Common folder. Both the server
and the client will use these files.

All those files that have been
I needed some small details to
that the application ran smoothly
with uniGUI.

Migrating an Application

project DAV

Remember FormCreate method?

```
TMainForm.UniFormCreate procedure (Sender: TObject);
begin
   JanelaAtiva: = False;

   ConfiguraCDSFromVO (CDSMestre, TDavCabecalhoVO);
   ConfiguraGridFromVO (GridMestre, TDavCabecalhoVO);

   ConfiguraCDSFromVO (CDSDetalhe, TDavDetalheVO);
   ConfiguraGridFromVO (GridDetalhe, TDavDetalheVO);
end;
```

Here we take the opportunity to set up ClientDataSet and The Grid. methods ConfiguraCDSFromVO and ConfiguraGridFromVO They are in the library unit.

However, the application I use a DAV VCL TJvDBUltimGrid and uniGUI know we need to use a TUniBDGrid. Of that so, I needed to change this method in Library to be able to use it with uniGUI.

Another change I had to perform was AfterScroll method. See how it is implemented in DAV VCL:

```
procedure TFCarregaDAV.CDS
TDataSet); (DataSet:
begin
   Then if JanelaAtiva

TController.TratarRetorno
alho.Items [CDSMestre.RecNo-1abec
True, True, CDSDetalhe);
end;
```

Now return to page 121 and see how it It is implemented for uniGUI. I refresh the detail grid.

At this point you should undertake a exercise. Try to migrate the DAV for uniGUI following the steps I mentioned in this chapter.

Page 129

#Criando Web Applications with Delphi - uniGUI

Final considerations

Conclusion

After everything we've seen, it is worth using the uniGUI?

Well, the answer to this question you already know who will give it? It is yourself.

But we can get some clues to facilitate this response.

It is very important that you perform tests. Put AllFeaturesDemo inside out. Master the framework. Do not start and do not buy the framework before it. Just take this step when you are absolutely that this is the way.

After dominating in fact demos, try migrate one implementation,