Contents - TDBPhDialer

How do I install this component?

1. Copy all files to the \DELPHI\LIB directory or choose a directory of your choice.
   The source files are as follows and are <u>only included with the registered version</u>:
   configdg.pas
   dbphdial.pas
   dialdlg.pas
   ini.pas (freeware)
   statbaru.pas (freeware)
   The following files are required for installation:
   ccube.dcu
   configdg.dcu
   configdg.dfm
   dbphdial.dcr
   dbphdial.dcu
   dbphdial.res
   dialdlg.dcu
   dialdlg.dfm
   ini.dcu
   registr1.dcu
   registr2.dcu
   registr2.dfm
   statbaru.dcu
   The remaining files are for documentation:
   dbphdial.hlp
   dbphdial.kwf
   readme.txt

2. While running Delphi, select menu OPTIONS / INSTALL COMPONENTS...

3. Click the ADD button - to add the TDBPhDialer component.

4. In the ADD MODULE dialog box, click the BROWSE... button.
   (make sure the search path is \DELPHI\LIB or what ever directory you placed the files)

5. Change the LIST BY FILE TYPE to *.dcu.

6. Select file **dbphdial.dcu**, and click the OK button.

7. Click OK to rebuild the component library. Wait until link completes successfully.

8. The TDBPhDialer Component will be placed in the "**Data Controls**" palette.

9. To install the help file **"dbphdial.hlp"** and the key word file **"dbphdial.kwf"** do the following:
   Move the "dbphdial.hlp" to the \delphi\bin directory.
   Move the "dbphdial.kwf" to the \delphi\help directory.
   To incorporate the Key Word File into the Delphi help system you must go to your Delphi Group
   and double-click on the **"HelpInst"** icon to bring up the Help File Installer.
   Go to file/open and go to the \delphi\bin directory and look for the "delphi.hdx" file.
   Select it and click OK. Now click the Plus Sign "+" to open the "dbphdial.kwf",
   select it and click OK. Now go to file/save. This will save and recompile the

"delphi.hdx" file.   The help system will now be available for the TDBPhDialer component.

What does this component do?

TDBPhDialer:

For starters, it does exactly what it sounds like, "it dials a telephone number for you".   But because it's data-aware, you can assign a data field (string) and data source to it. This gives you edit capability (Update) or ReadOnly capability (Browse) per data field and data record displayed by your application.

Features Include:

Data-aware edit field. (Update or ReadOnly).

Attached button to activate dialing process.

Popup menu (property configuration) at run-time:
This feature may be turned on or off with the **asAllowPopUp** property, default is true. Activated by using the right-mouse button over the phone dialer edit field.   (to override right-mouse click)
All configuration information is stored in the "application-name".ini file in the same directory as the executable.
Note - the INI file is created the first time you use the popup configuration at run-time.   Any design time changes
to the specialize Phone Dialer properties will not take effect if made after the INI file has been created. To make
property changes you must there-after use the popup configuration or delete the INI file or delete the section in
the INI file were the DBPhDialer is listed.   If the INI file doesn't exist then the properties you set at design-time
will be effective.   * The following properties are the only ones effective by this rule. asComPort, asDialType,
asInitString, asOutsideNumber, asBusyRedial, asBusyRetries, asDialOne, asUSFormat, asEditMask,
asLocalAreaCode, asWaitRedial and asConfirm.

Formatted phone numbers with user defined edit mask:
By setting the **asEditMask** property and using the specified delimiters the phone dialer can accommodate
virtually any phone number format.

Elapse time on phone:
Dialer dialog gives the user the current date and time as well as the total elapsed time of the telephone call.

Busy Redial: (**asBusyRedial**)
Redials x-number of times, with wait-redial (**asBusyRetries**) x-number of seconds.   Gives status of dialing process as well as any modem messages such as "Lost Carrier", "No Dial Tone", "Line Busy",   etc.........

Complete Modem Access: (**asInitString**)
You can specify any valid modem command(s) string to have your modem reset, speed dial, set loudness of speaker, etc........

What properties are available?

This component inherits from TDBEdit and TBitBtn, with the following additional properties added.

Properties:

asAllowPopUp:
If set to true, with right mouse click over edit field, allows popup configuration at run-time.
Default is true. (to override right-mouse click)

asBusyRedial:
If set to true, the phone dialer will redial (**asBusyRetries**) or (x) number of times, when busy.
Default is true.

asBusyRetries:
Integer value = redials (x) number of times.
Default is 10.

asComPort:
COM1, COM2, COM3 or COM4.
Default is COM1.

asInitString:
Modem initialization string. Modems sometimes can be improperly initialized, causing unusual results.
The phone dialer may not function properly because of this. If this happens, you can try the following
command: "ATZ" (leave off double tics!) , this will usually reset a modem to its default settings.
You may want to set the modem with special features such as speed dialing with the speaker turned up
loud, then issue the following command "ATS11=55L3". This field will accepts any valid modem command.
Default is blanks.

asDialType:
dtTone or dtPulse.
Default is dtTone.

asLocalAreaCode:
Integer value = a three digit number which indicates the users local area code.
The phone dialer will not use the area code when dialing if its local.
The **asUSFormat** property must be true for this feature to take effect.
Default is Blank.

asWaitRedial:
Integer Value = the amount of time in seconds to wait before redialing after a busy signal.
Default is 60.

asOutsideNumber:
Integer value = an outside number if required to dial out.

Default is blank.

asConfirm:
If set to true, it will confirm the number you are about to dial.
Default is false.

asEditMask:
By setting the asEditMask property and using the following delimiters the phone dialer can accommodate virtually any phone number format.
The **delimiters** are: blanks " ", hyphen "-", left-paren "(", right-paren ")", plus sign "+", and the numbers 0 thru 9.
For example: a normal US format may look like this: 999-999-9999 or (999) 999-9999.
Calling an International number from the US may look something like this: 011-99-999-999999.
All literals are saved along with the phone number.
Default is 999-999-9999.
Note - If the edit mask is left blank, the phone number will not be formatted, meaning any input phone number will not be altered.
(except when the **asUSFormat** property is true). What goes in is what comes out. And there will not be any length constraints.
Also If the EditMask property of a TStringField is supplied for the phone number field, then it will override the asEditMask property and the user will not be able to use any user defined edit mask. This maybe the preferred method to utilize any edit mask.

asUSFormat:
If set to true, United States phone format assumed: 999-999-9999.
If used in combination with **asEditMask**, it will use the user defined edit mask instead.
Must be true for the **asLocalAreaCode** property to function as indicated.
Default is true.

asDialOne:
If set to true, the phone dialer will always dial a "1" before the phone number.
Exception to this is if **asLocalAreaCode** supplied then it will not use a one "1" when the local area code equals the area code of the phone number to be dialed or if **asUSFormat** set to true, any number less then 8 characters will not use a one "1".
Default is true.

asButtonHint:
String = the hint message displayed when mouse pointer passes over dialer button.
Default: "click to dial phone number"

asButtonShowHint:
If set to true, the **asButtonHint** will be displayed otherwise it will not.
Default is true.

asTabOnEnterKey:
If set to true, the enter key will tab to the next control otherwise the cursor will remain in position.
Default is true.

Copyright/License/Warranty.

TDBPhDialer (a Delphi component) v1.0 (Shareware Version)
Copyright (C) 1996 ArcticSoft
All Rights Reserved

License Agreement

You should carefully read the following terms and conditions before using this software. Unless you have a different license agreement signed by ArcticSoft, your use of this software indicates your acceptance of this license agreement and warranty.

Evaluation and Registration

This is not free software. You are hereby licensed to use this software for evaluation purposes without charge for a period of 21 days. If you use this software after the 21 day evaluation period a registration fee of $16 is required plus $2 postage/handling.
Payments must be in US dollars drawn on a US bank, and should be sent to:

ArcticSoft, P.O. Box 26582, San Diego, CA 92196-0582.
e-mail: ArcticSoft@juno.com

When payment is received you will be sent a registered copy of the latest version of TDBPhDialer with source code.
Please allow three weeks for delivery.

Unregistered use of TDBPhDialer after the 21-day evaluation period is in violation of U.S. and international copyright laws.

Distribution

Provided that you verify that you are distributing the Shareware Version, you are hereby licensed to make as many copies of the Shareware version of this software and documentation as you wish; give exact copies of the original Shareware version to anyone; and distribute the Shareware version of the software and documentation in its unmodified form via electronic means. There is no charge for any of the above.

You are specifically prohibited from charging, or requesting donations, for any such copies, however made; and from distributing the software and/or documentation with other products (commercial or otherwise) without prior written permission, with one exception: Disk Vendors approved by the Association of Shareware Professionals are permitted to redistribute TDBPhDialer, subject to the conditions in this license, without specific written permission.

Registered Version

One registered copy of TDBPhDialer may either be used by a single person who uses the software personally on one or more software applications for development, or installed on a single workstation used non-simultaneously by multiple people on one or more software applications for development, but not both. You will be allowed to distribute the software with other products (commercial or otherwise)
but only as part of another **software application *.EXE (executable file).**
Modifications made to the following source code files, **dbphdial.pas, configdg.pas and**

dialdlg.pas will be allowed for making enhancments for the purpose of applying those enhancments to a application for which you are developing and not for the purpose of resale of the TDBPhDialer or a copy of said component.
Any source file (*.pas) that is marked as "FREEWARE" does not and will not have any restrictions of any type and can
be distributed in any way or form.
There are no Royalties fees.

Governing Law

This agreement shall be governed by the laws of the State of California.

Disclaimer of Warranty

THIS SOFTWARE AND THE ACCOMPANYING FILES ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED.
Because of the various hardware and software environments into which TDBPhDialer may be put,
NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED.

Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it.
The user must assume the entire risk of using the program.
ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE.

How do I use it?

Simply drop the component on any form to use.
Set the **DataSource** and **DataField** properties.
Set the property **AsComPort** to the proper modem COMPort.
Make any changes to any other property by choosing preference.
Compile, link and run your application.

While your application is running, right-click your mouse over the Phone Dialer edit field.
A popup menu: **Configure...** will appear. Click on it to activate the Configuration form.
Make any changes if desired, click the **OK button** to save or **Cancel button** to abort.
Type a phone number or use whats there.
Click **Phone Dialer button** to dial number.
Observe status messages and pick up the phone when person answers, then click the **Answer button**.
Click **Cancel button** when call is complete or to abort dialing process.

asCOMPort

Property type: TComPort
Possible values are: COM1, COM2, COM3 or COM4.
Default is COM1.

Ordering Information.

TDBPhDialer 1.1   (Includes all source code)

Ordering by check: To order by check send this order form and a check to:

ArcticSoft, P.O. Box 26582, San Diego, CA   92196-0582.
e-mail: ArcticSoft@juno.com

Payments must be in US dollars drawn on a US bank, or you can send international postal money orders in US dollars.

TDBPhDialer Single Copy   ____   copies at $16 each   ______
Calfornia residents add 7.75% sales tax   ______
Postage & Handling: add $2.00 per copy.   ______

Total payment   ______

3.5" disks are sent unless a 5.25" disk is requested.
Please allow up to three weeks for delivery.

* If e-mail address supplied, delivery can be made sooner.
  $2 handling fee still required.

Name: ______________________ Date:_________________________

Company: ____

Address: ____

City, State, Zip: ____

Country: ____

Day Phone: ______________   Eve: ____________________________

Electronic Mail address: _____________________________________

How did you hear about TDBPhDialer? _________________________

Please send me your comments:

asAllowPopUp

Property type: boolean
If set to true, with right mouse click over edit field, allows popup configuration at run-time.
Default is true.

asInitString

Modem initialization string.   Modems sometimes can be improperly initialized, causing unusual results.
The phone dialer may not function properly because of this. If this happens, you can try the following
command: "ATZ" (leave off double tics!) , this will usually reset a modem to its default settings.
You may want to set the modem with special features such as speed dialing with the speaker turned up
loud, then issue the following command "ATS11=55L3".   This field will accepts any valid modem command.
Default is blanks.

asBusyRetries

Property type: integer
Redials (x) number of times.
Default is 10,   meaning the Phone Dialer will redial 10 times.

asBusyRedial

Property type: boolean
If set to true, the phone dialer will redial (asBusyRetries) or (x) number of times, when busy.
Default is true, meaning the Phone Dialer will automatically redial when busy.

asEditMask

Property Type: string
By setting the asEditMask property and using the following delimiters the phone dialer can accommodate virtually any phone number format.
The **delimiters** are: blanks " ", hyphen "-", left-paren "(", right-paren ")", plus sign "+", and the numbers 0 thru 9.
For example: a normal US format may look like this: 999-999-9999 or (999) 999-9999.
Calling an International number from the US may look something like this: 011-99-999-999999.
All literals are saved along with the phone number.
Default is 999-999-9999.

Note - If the edit mask is left blank, the phone number will not be formatted, meaning any input phone number will not be altered.
(except when the **asUSFormat** property is true). What goes in is what comes out. And there will not be any length constraints.
Also If the EditMask property of a TStringField is supplied for the phone number field, then it will override the asEditMask property and the user will not be able to use any user defined edit mask. This maybe the preferred method to utilize any edit mask.

asUSFormat

Property type: boolean
If set to true,   United States phone format assumed: 999-999-9999.
If used in combination with **asEditMask**, it will use the user defined edit mask instead.
Must be true for the **asLocalAreaCode** property to function as indicated.
Default is true.

asLocalAreaCode

Property type: integer
A three digit number which indicates the users local area code.
The phone dialer will not use the area code in the dialing process if its a local area code.
The **asUSFormat** property must be true for this feature to take effect.
Default is Blank.

asDialOne

Property type: boolean
If set to true,   the phone dialer will always dial a "1" before the phone number.
Exception to this is if **asLocalAreaCode** supplied then it will not use a one "1" when the local area code equals the area code of the phone number to be dialed or if **asUSFormat** set to true, any number less then 8 characters will not use a one "1".
Default is true.

asDialType

Property type: TMethod
Dial type is either tone or pulse.
Possible values are: dtTone or dtPulse.
Default is dtTone.

asWaitRedial

Property type: integer
The amount of time in seconds to wait before redialing after a busy signal.
Default is 60, meaning the Phone Dialer will wait 60 seconds before attemping to redial the telephone number.

asOutsideNumber

Property type: integer
An outside number if required to dial out. Some businesses require this functionality.
Default is blank.

asButtonHint

Property type: string
This is the hint message displayed when the mouse pointer passes over dialer button.
Default: "**click to dial phone number**"

asButtonShowHint

Property type: boolean
If set to true,   the **asButtonHint** will be displayed otherwise it will not.
Default is true.

asConfirm

Property type: boolean
If set to true,   a confirmation dialog box will appear so you may confirm the number you are about to dial.
Default is false.

asTabOnEnterKey

Property type: boolean
If set to true, the enter key will tab to the next control otherwise the cursor will remain in position.
Default is true.

Override Popup Menu

Set the **asAllowPopUp** property to False.
To bypass the (right-mouse click) popup menu and call the popup menu or configuration form directly, the following code can be utilized:

```
{ to call the popup menu ( can be any event! ) }
procedure TForm1.FormClick(Sender: TObject);
var
   P1 : TPoint;
   x, y: integer;
begin
     X := 0;
     y := 0;
     P1.X := X; P1.Y := Y;
     P1 := DBPhDialer1.ClientToScreen(P1);
     DBPhDialer1.MakeAMenu(P1);
end;

{ to call the configuration form from a button click event }
procedure TForm1.Button1Click(Sender: TObject);
begin
     DBPhDialer1.OpenConfigDialog(Sender);
end;
```