TAdrockAlarmList

<u>Properties</u>   <u>Methods</u>   <u>Events</u>   <u>Installation</u>

Unit

AdAlarm

Description

Use a TAdrockAlarmList component when you want to have alarms triggered in your programs.

To use this control just drop on your form. You then need to call the Add method to add an alarm,

There is **no** limit on the number of alarms the control can have.

With TAdrockAlarmList you set multiple alarms to go off (i.e., fire events) at various times during the day.   This control makes it easy to schedule events to happen at various times.   Just set the properties and wait for Alarm to notify you.

Alarm even tells you when the date has changed.   You can use this to remove all of the current alarms and set new ones for the day.   A must if youre writing a PIM.

The alarm time is very flexible because of the use of wildcards in the time format.

E.g. 10:??:05   This will trigger the alarm every minute from 10:00:00 to 10:59:59 but only when at 5 seconds past any minute in the hour.

E.g. 10:??:?5   This will trigger the alarm every minute from 10:00:00 to 10:59:59 but only when the seconds contain a 5 in the seconds: 05, 15, 25, 35, 45, and 55 seconds.

Any part of the time format can contain wildcards, including every part of the time format.

E.g. ??:??:??   This will trigger the alarm every second, no matter what the hour or minute.

The format is fixed and each segment (Hours, Minutes and Seconds) must be 2 digits long, either numbers or wildcards.

E.g. 10:05   This is **invalid** since the seconds must be specified, a EAlarmBadFormat exception will be raised in this situation.

Properties

This is the properties page for TAdrockAlarmList, please select a property.

[Enabled](#)

[Interval](#)

Name

Tag

[Version](#)

Methods

Add

AddExact

Delete

DeleteByRef

Events

[OnAlarm](#)

[OnNewDate](#)

[OnSecondsChange](#)

OnAlarm

Applies to TAdrockAlarmList

Declaration

Event   Alarm(Sender: TObject;   AlarmEvent: Integer; Reference: String);

Description

This event gets fired for every alarm that gets triggered. The AlarmEvent is the actual position of the alarm in the list of alarms. The reference is the reference you used when you added the alarm. You can use the reference to identify which of the alarms has been triggered.

OnNewDate

Applies to <u>TAdrockAlarmList</u>

<u>Declaration</u>

Event NewDate(Sender: TObject;   NewDate: TDateTime);

<u>Description</u>

This event gets fired when the date changes at midnight, you can use this to re-program the list of timer events for the new day. Or whatever you want to do with it.

Enabled

Applies to <u>TAdrockAlarmList</u>

<u>Declaration</u>

property Enabled : Boolean;

<u>Description</u>

Set this property to enable or disable the alarms. Normally you would just leave this set to TRUE.

Add

Applies to

TAdrockAlarmList components

Declaration

method Add(AlarmTime, Ref : String);

Description

Calling this method will add an alarm to the AdrockAlarmList component. The alarmTime that you set can contain wildcards to allow any time to work with the alarm. Use the AddExact method if you want to add an exact time using a TDateTime variable.

The Ref gets returned when the alarm is triggered, you can use this to know which alarm was triggered, it is a string property so can contain text or numbers.

If you supply an invalid AlarmTime format a EAlarmBadFormat exception will be raised.

Example

??:??:?2	Set the alarm to trigger for every time the seconds contains a 2,	E.g. 02, 12, 22, 32, 42, and 52 past the minute.
??:10:00	Set the alarm to trigger 10 minutes past any hour.	E.g. 1:10:00, 2:10:00, 3:10:00, 4:10:00, 5:10:00 etc…..
10:00:00	Set the alarm to trigger at 10 Oclock only.	
??:??:??	Set the alarm to trigger every second.	

As you can see the alarm is very flexible in its triggering

> **AdrockAlarmList1.Add(??:??:?2,Any Hour, Any Minute, and only X2 Seconds);**
>
> **AdrockAlarmList1.Add(??:??:?5,Any Hour, Any Minute, and only   X5 Seconds);**
>
> **AdrockAlarmList1.Add(??:?5:00,Any Hour, but only x5 Minutes and 0 seconds);**
>
> **AdrockAlarmList1.Add(??:?7:00, Any Hour, but only x7 Minutes and 0 seconds);**
>
> **AdrockAlarmList1.Add(10:11:23,10:11:23);**
>
> **AdrockAlarmList1.Add(10:11:23,10:11:23);**
>
> **AdrockAlarmList1.Add(10:11:22,10:11:22);**
>
> **AdrockAlarmList1.Add(10:11:18,10:11:18);**
>
> **AdrockAlarmList1.Add(15:11:18,3:11:18 PM);**

Delete

Applies to

TAdrockAlarmList components

Declaration

method Delete(ItemToDelete : Integer);

Description

Calling this method will delete the item specified in the ItemToDelete variable. The item to delete is the number of the item in the list when it was added to the list.

You can also call the method DeleteByRef to delete the item that matches the reference string. This reference string is the string that was passed to the add method.

If you try to delete an Alarm that is not found then the EAlarmItemNotFound exception is raised.

Example:

{ Add a new alarm item }

AdrockListBox1Add(10:24:00, Chris);

{ Delete the alarm by the position of the alarm in the list. }

AdrockListBox1.Delete(0); { Delete the first item In the list }

DeleteByRef

Applies to

TAdrockAlarmList components

Declaration

method DeleteByRef(Ref : String);

Description

Calling this method will delete the item that matches the Ref variable. You can use this method if you do not know the position of the item you want to delete. If you know the item position that you want to delete you can call the Delete method.

If you try to delete an Alarm that is not found then the EAlarmItemNotFound exception is raised.

Example:

{ Add a new alarm item }

AdrockListBox1Add(10:24:00, Chris);

{ Delete the alarm by using the reference Chris that was used when the alarm was setup. }

AdrockListBox1.DeleteByRef(Chris);

TAdrockAlarmList

Applies to

TAdrockAlarmList components

Click here for the **Contents Page**

Version

Applies to

<u>TAdrockAlarmList components</u>

<u>Declaration</u>

Property Version : String;

<u>Description</u>

This property displayed the current version of the component in the Object Inspector. The version is read-only and can not be set.

Interval

Applies to

<u>TAdrockAlarmList components</u>

<u>Declaration</u>

Property Interval : Integer;

<u>Description</u>

This property sets the interval between calls to the routines that check the alarms, and determine if any should be triggered. By default this property is set to 1000, which is 1 second. Some time this value might not be suitable.

Even if this property was set to 400 which would trigger approximately 2.5 times per second an alarm will only be triggered once per second. So there will not be any double triggering because of the timer frequency.

AddExact

Applies to

<u>TAdrockAlarmList components</u>

<u>Declaration</u>

method AddExtact(AlarmTime : TDateTime; Ref : String);

<u>Description</u>

Calling this method will add an alarm to the AdrockAlarmList component. The alarmTime is used to add a specific time instead of a variable or wildcard time. Use the <u>Add</u> method if you want to use wildcards in the time.

The Ref gets returned when the alarm is triggered, you can use this to know which alarm was triggered, it is a string property so can contain text or numbers.

Example

   AdrockAlarmList1.AddExact(Time,My Alarm);

OnSecondsChange

Applies to TAdrockAlarmList

Declaration

Event NewDate(Sender: TObject;   NewDate: TDateTime);

Description

This event gets fired once every second. You could use it to update a time or clock.

Installation Instructions

<u>**To install this component>**</u>

- **Getting Started**

  - We will assume that you are installing ADALARM………..
  - UNZIP the archive (eg: ADALARM.ZIP) to a directory temporally.
  - Create a directory to hold this component and any others that you will install in the future. (eg: C:\DELPHI\COMPONEN) It is easier to put new components in a separate directory since you can update them easily and if there is a new release of DELPHI it will not effect your component directory.

- **Installation of the HLP and KWF files.**

  - Exit DELPHI is you are running it, as you can not install the KWF file while delphi is running.
  - Move or copy help file (e.g.: AdALARM.HLP) to your Delphi Bin directory (suggest C:\DELPHI\BIN), it only really needs to be somewhere on the path, but the DELPHI bin directory is a good place.
  - Delphi has a program to add our help index to Delphis help index. What this does is allow you to search for both delphi commands and the commands and properties of the component you want to install. This program is called HELPINST and is found in the DELPHI\HELP sub directory. Copy the keyword file (ADALARM.KWF) to the delphi help directory.
  - Change the current directory to the Delphi Help directory (e.g.: CD C:\DELPHI\HELP)
  - Run the HELPINST program and load the DELPHI.HDX file from the DELPHI\BIN directory.
  - Add the .KWF file (e.g.: ADALARM.KWF) to the list of indexs.
  - Save the new index file and then exit the HelpInst program.

- **Installation of the DCU, DCR, HIS and TXT files.**

  - We next must add the component to the Delphi VCL Library.
  - Copy the DCU, DCR, HIS, and TXT files to the directory that you made for the components. (e.g.:   C:\DELPHI\COMPONEN)
  - Load delphi and selected the Options menu Item, and then selected the Install Components… option. You should get a screen as follows.

  - Add the directory where you have installed the components to the search path directory. All components that get installed must be in this search path.
  - Click the add button and then the browse button to navigate to the directory where you copied the .DCU file (e.g.: C:\DELPHI\COMPONEN)
  - Select the List Files Of Type and select Unit File *.DCU, you should then see the DCU file that you copied to the directory. Select the file and you will be returned to the Install

Components dialog box.

- Click on the OK button and the VCL library will be re-built.

- You will notice on your VCL Palette bar in DELPHI that there is now a new tab called ADROCK, in side here will be an icon to represent the component you just installed.

· **Other Files.**

- The ADALARM.HIS file contains a history of updates and changes made.

- The ADALARM.TXT file contains a brief description and ordering details.

- There is a Sample Program in the file DEMOALRM.ZIP which is in the distribution ZIP archive.

Note: We hope you have no trouble in using this component in your projects and that you like using it and other Adrock components.

We welcome feedback from you, with you we can make better upgrades and quicker fixes to any problems that you are experiencing.

Please contact us at   CompuServe 100353,110 or Internet: DHMCC@IBM.NET.