Contents

For many years InterBase has had functionality that other databases still lack, or are only recently implementing. One of these features is Event Alerters. The Interbase Event Alerter mechanism provides a means whereby applications can respond to actions and database changes made by other, concurrently running applications, without having to resort to polling the database on a regular basis, or communicating directly with the other applications. Unfortunately, the ability to harness the power of Event Alerters via the Borland Database Engine (BDE) does not exist.

TIBEventAlerter

The TIBEventAlerter component allows a Delphi application to register interest in, and asynchronously handle, events posted by an InterBase server.

TIBEventAlerter Component

properties   methods   events

Unit
IBCtrls.pas

Description
TIBEventAlerter component allows a Delphi application to register interest in, and asynchronously handle, events posted by an InterBase server.   The Interbase event mechanism provides a means whereby applications can respond to actions and database changes made by other, concurrently running applications, without having to resort to polling the database on a regular basis, or communicating directly with the other applications.

In essence, the TIBEventAlerter allows an application to say 'I want to be informed when events X, Y and Z occur'.   The application then continues executing without polling the database checking for X, Y and Z.   When any of the requested events does occur, the InterBase server will notify the application and OnEventAlert will be called to allow the event to be processed.

The procedure for utilising InterBase events is as follows:

1. Create a trigger or stored procedure on your InterBase server which will post an event.
2. Add an IBDatabase and an IBEventAlerter to your form.
3. Register the events you wish to be notified about.
4. Write an OnEventAlert event handler to handle incoming event notifications.

It is important to remember that InterBase posts events within the context of transactions - you must commit any transaction that posts an event for the client to be notified.   In addition, InterBase consolidates events before posting them.   For example, if an InterBase trigger posts 20 x STOCK_LOW events within a transaction, when the transaction is committed these will be consolidated into a single STOCK_LOW event, and the client will only receive one event notification.

Events property

Applies to
TIBEventAlerter

Declaration
property Events: TStrings

Description
The Events property contains the list of events that an IBEventAlerter component will respond to.   A single IBEventAlerter can register interest in up to 15 events. If you need to respond to more that 15 events use more that one IBEventAlerter component.   An exception will be raised if you attempt to add too many events at runtime.   At design time the Events property editor will only allow a maximum 15 events to entered.

To add an event to the Events list use the following code
```
        IBEventAlerter.Events.Add( 'STOCK_LOW')
```

Note: event names are case-sensitive.

properties

Events
Database
Registered

Database property

Applies to
TIBEventAlerter

Declaration
property Database: TDatabase

Description
Database is a reference to the TDatabase component that will provide the InterBase connection required to register InterBase Events or to manipulate InterBase metadata.

Registered property

Applies to
TIBEventAlerter

Declaration
property Registered: boolean

Description
Registered indicates whether any events are currently registered or not.   Setting Registered to true will call RegisterEvents and register the events in the Events list.   At design time, no event notifications will be received even if Registered is true.

OnEventAlert event

Applies to
TIBEventAlerter

Declaration
procedure OnEventAlert: TEventAlert

```
TEventAlert = procedure( Sender: TObject; EventName: string;
                                    EventCount: longint; var CancelAlerts: Boolean)
```

OnEventAlert is called every time an InterBase event is received by an IBEventAlerter component.   The EventName variable contains the name of the event that has just been received.   The EventCount variable contains the number of EventName events that have been received since OnEventAlert was last called.

To cancel interest in any further events, set Cancelalerts = true.   If you later decide that you want to receive events again, call the QueueEvents method.   You cannot cannot call RegisterEvents, UnregisterEvents, QueueEvents or CancelEvents from within an OnEventAlert event handler.

OnEventAlert runs as a separate thread to allow for true asynchronous event processing, however, the IBEventAlerter component provides synchronisation code to ensure that only one OnEventAlert event handler executes at any one time.

events

OnEventAlert

methods

CancelEvents	RegisterEvents
QueueEvents	UnregisterEvents

CancelEvents method

Applies to
TIBEventAlerter

Declaration
procedure CancelEvents

Description
CancelEvents cancels interest in any pending InterBase events.   CancelEvents does not unregister the events and to restore interest in the events again simply call QueueEvents.

Note: You cannot call CancelEvents from within an OnEventAlert handler.

RegisterEvents method

Applies to
TIBEventAlerter

Declaration
procedure RegisterEvents

Description
RegisterEvents registers interest in the events listed in the Events property.   RegisterEvents will call the QueueEvents method to start receiving event notifications.

QueueEvents method

Applies to
TIBEventAlerter

Declaration
procedure QueueEvents

Description
QueueEvents is called to start receiving event notifications.

You must call RegisterEvents to specify which events you wish to receive before calling QueueEvents.   If RegisterEvents has not been called an exception will be raised.

UnregisterEvents method

Applies to
TIBEventAlerter

Declaration
procedure UnregisterEvents

Description
UnregisterEvents calls CancelEvents to cancel any pending event notifications, and then unregisters interest in the events in the Events list.   When the IBEventAlerter component is destroyed UnregisterEvents will be called automatically.

Data Definition Language
SQL statements which define database metadata
such as tables, indexes etc.

eg.   create table Employee ...