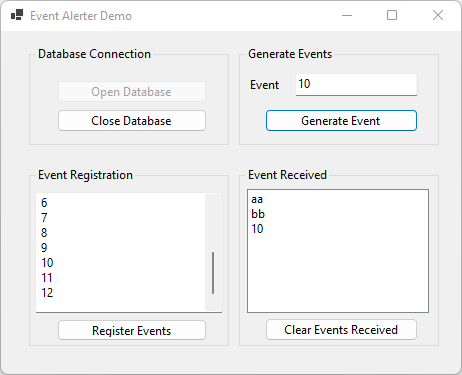
# Event Demo Readme

The is just a quick overview of the events demo. It does not require a specific version of InterBase, but if the EventsDemo.ib file is not found when run it will go ahead and create it and put in the stored procedure used to generate the events. Note that this will be in the ODS for the current server so moving that one to an older server will result in connection failures, so just delete it and let it be regenerated.



**Overview of controls**

Open Database – Connects to the eventdemo.ib and if not found in the same directory as the executable will create it and install the stored procedure used to generate events.

Close Database – Unregisters the events and closes the connection.

Register Events – Will first unregister and previously registered events. Then takes the items in the memo and line by line. The InterBase API accepts ASCII characters. .NET is all Unicode so two characters or more makes sure that it is being properly translated into ANSI before being registered.

Event text box – for entering the event you want to fire. Event names are case sensitive.

Generate Event – Will take the text in the event text box and pass it to a stored procedure in the DB that will fire the event and the transaction will be committed. This is important as no one listening will get an event alert until a commit is performed.

To demonstrate this, you can use IBConsole and connect to the EventsDemo.ib. With the Event demo running and the events registered open a wisql window and execute ‘execute procedure EventDemo('bb')’. You will not see the event in the demo, but this is not a bug, the transaction that fired the event has not been committed. Commit the wisql transaction and now the event shows in the demo.

Events Received list box – when events are fired, they will be placed into the list box

Clear Events Received – Clears the list box.

**Interesting code**

Creation – After creating an instance you need to assign the connection property to the connection to be used and then an event handler for when events are fired.

The event fired signature looks like



Finally, to register Events you call RegisterEvents passing a List<String> of the strings you want to watch. No Unicode characters allowed. The list you pass will be converted to ASCII character strings before being registered. You cannot call RegisterEvents if there are currently Registered events. You must first Unregister the events and then call RegisterEvents. It is safe to call UnregisterEvents when nothing is registered.

