

Where Event Code is Stored

The OpenInsight Form Designer provides a way to add BASIC+ code to an event on a control by using the Form Designer's Event Editor. The source code, if you use the Form Designer, is stored in the **SYSREPOSEVENTS** table, and the executable code (the compiled event code) is stored in the **SYSREPOSEVENTEXES** table. In both tables, the record ID contains four parts, as shown below:

APPNAME*EVENTTYPE*FORMNAME.CONTROLNAME

Event Code for a Particular Control in a Window

For example, in the **EXAMPLES** application, the source code for the **GOTFOCUS Event** of the **CUST_ENTRY** window's **CUST_NAME** control is stored in **SYSREPOSEVENTS** (and the compiled code in **SYSREPOSEVENTEXES**) with the following record ID:

EXAMPLES*GOTFOCUS*CUST_ENTRY.CUSTNAME

To "promote" an event is to make it apply to a larger set of controls and situations. The promotion is done through a naming convention in which if a level of qualification is omitted, the event is assumed to apply to all controls at that level. The promotion applies to the compiled code only (in **SYSREPOSEVENTEXES**). The source code for the event can be stored in any OpenInsight row.

Event Code for All Controls in a Window

One event handler could be written, typically as a stored procedure, that would handle all **GOTFOCUS events** for the window, instead of writing an event handler for each control. As long as the compiled code is copied into the **SYSREPOSEVENTEXES** table with the correct record ID, the code would be run for each **GOTFOCUS event** in the window. The result is to "promote" the event handler to apply to all controls in **CUST_ENTRY**, instead of being restricted to just the **CUSTNAME** control.

To accomplish this, the compiled code for an event handler designed for the **GOTFOCUS Event** of **all** control in the **CUST_ENTRY** window is stored in **SYSREPOSEVENTEXES**, with the following record ID:

EXAMPLES*GOTFOCUS*CUST_ENTRY.

Event Code for All Controls of a Particular Type for a Particular Event

This promotion can be generalized to an even higher level. Suppose we want to code **one** event handler for **all GOTFOCUS events** in the entire application, for a particular control type (such as an **EDITFIELD**). To give an application a uniform "look and feel", perhaps on the **GOTFOCUS event** for an edit field, we would like the background color to be yellow. Some applications contain hundreds of windows with dozens of edit fields in each window. Without promoting the event to apply to all edit fields, it would be necessary to write a separate event handler for each window's edit fields.

The naming convention for this type of promotion is:

ApplicationName*EventType.ControlType.OIWIN*

where ControlType is the control's type (such as **EDITFIELD**), and **OIWIN*** is a special code that indicates "all forms". So, the record ID in **SYSREPOSEVENTEXES** for code to handle **all GOTFOCUS events** for **any** edit field in **any** window in the **EXAMPLES** application is:

EXAMPLES*GOTFOCUS.EDITFIELD.OIWIN*

Event Code for All Controls for a Particular Event in All Windows

To have the event code apply to all controls in all windows of a particular application, do not code the **control type**. You could code one handler for all **GOTFOCUS events** in all windows of the **EXAMPLES** application, promoting the event to a higher level of generality. In the code, you could test the control type by retrieving the **TYPE** property using **Get_Property()** and then use a **CASE statement** to handle the behavior when that type of control receives focus. The record ID in **SYSREPOSEVENTEXES** would then be:

EXAMPLES*GOTFOCUS..OIWIN*

Event Code for All Controls in All Windows

Why not, then, have **one** subroutine that controls all the events for **all** the windows in an application? In that way, instead of many short, fragmented event handlers, there is one centralized "repository" for **all** event coding. The record key in **SYSREPOSEVENTEXES** for an event handler handling **all** events in **all** windows of the **EXAMPLES** application would be:

EXAMPLES*..OIWIN*

In summary, to create promoted events, you will need to use the System Editor to write and compile the code, and then copy it to the correct ID in the **SYSREPOSEVENTEXES** table, at the level you desire.