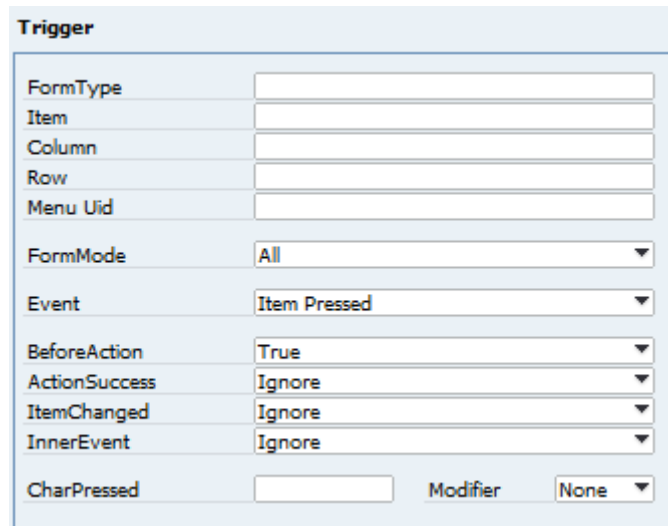


Guide - Understanding B1 Validation Triggers

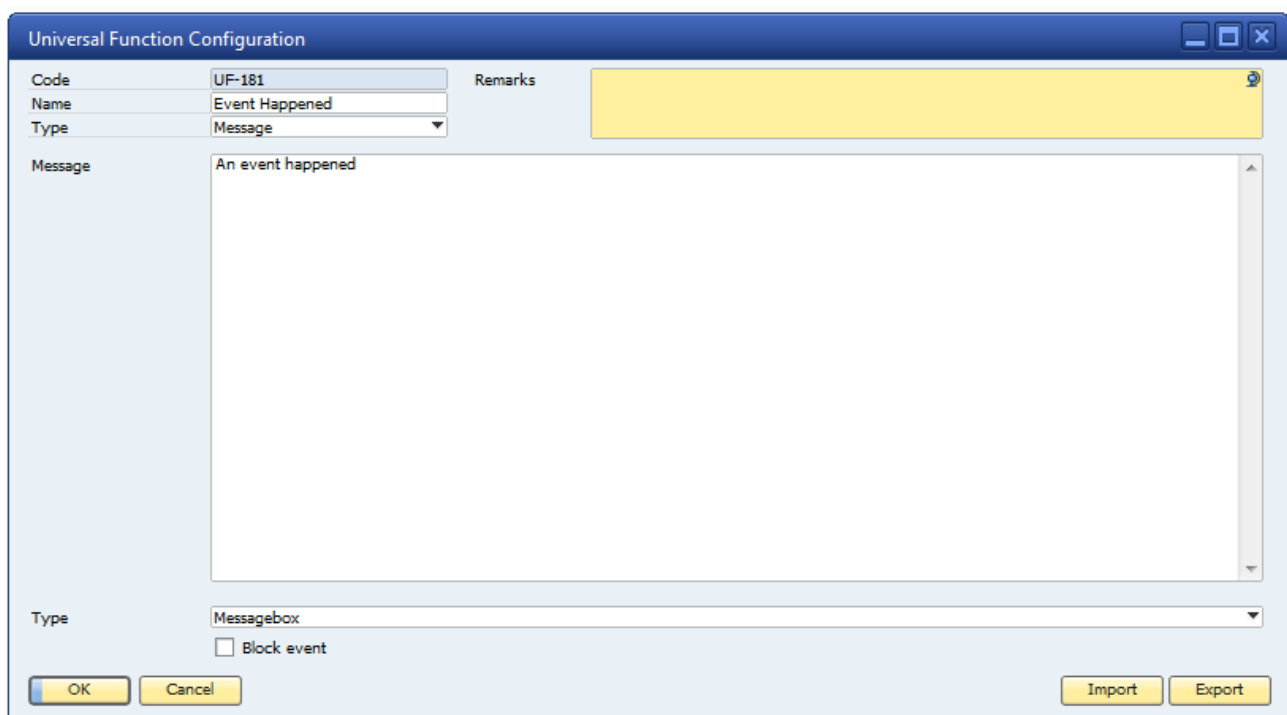
This document will go into details of how the B1 Validation System Trigger works.



The screenshot shows a 'Trigger' configuration window. It contains several input fields and dropdown menus. The fields are: FormType, Item, Column, Row, Menu Uid, FormMode (set to 'All'), Event (set to 'Item Pressed'), BeforeAction (set to 'True'), ActionSuccess (set to 'Ignore'), ItemChanged (set to 'Ignore'), InnerEvent (set to 'Ignore'), CharPressed, and Modifier (set to 'None').

Data for test

In order to test the event you might like to test it yourself. The best way to do this is by creating the following two configurations



The screenshot shows a 'Universal Function Configuration' window. It has a 'Code' field with 'UF-181', a 'Name' field with 'Event Happened', and a 'Type' dropdown set to 'Message'. There is a 'Remarks' field with a yellow background. The 'Message' field contains the text 'An event happened'. At the bottom, there is a 'Type' dropdown set to 'MessageBox', a 'Block event' checkbox, and buttons for 'OK', 'Cancel', 'Import', and 'Export'.

B1 Validation Configuration

Name: Sample of Event Sort: 100 Number: 142

Users (Empty = All) ☒ Include users ☐ Exclude users

Trigger

FormType: 134
Item:
Column:
Row:
Menu Uid:
FormMode: All
Event: Click
BeforeAction: True
ActionSuccess: Ignore
ItemChanged: Ignore
InnerEvent: Ignore
CharPressed: Modifier: None

Condition

☒ No condition, just execute function ⇒ UF-181
☐ Simple condition
When item | (column) | Comparison type Value | Type | Execute function
 Equals String (0)
☐ SQL Condition Common SQL structures

Condition result	Function to execute

Condition result Function to execute

Add line Remove line

☒ Active Add Cancel Import Export

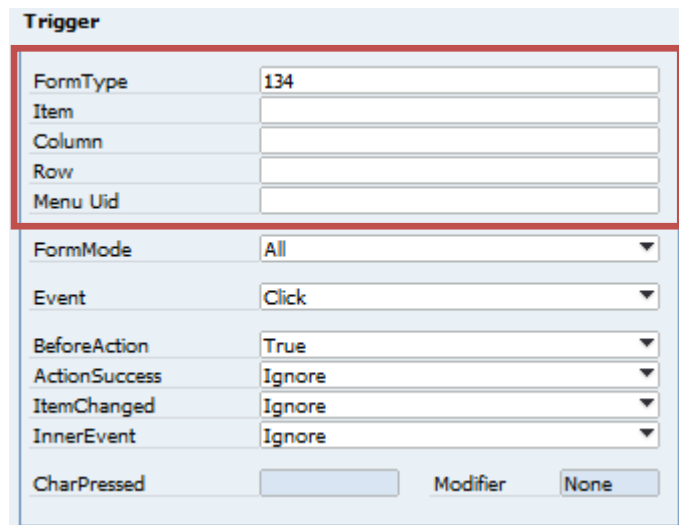
The above two configurations where we change the trigger-options of the B1 Validation configuration for each test will give you an idea of what goes on.

An overview

If we look at the SAP Business One Client in general we can see it as an interface that receive input from a user via a mouse and a keyboard, can we can see every little thing that the user does as a potential trigger.

The trigger system can be seen as filter of that certain user-actions we would like to add our additional functionality to, and configuring the B1 Validation System Trigger is taken these hundreds of events and specifying the ones we need.

The FormType, Item, Column, Row and MenuUid



Trigger	
FormType	134
Item	
Column	
Row	
Menu Uid	
FormMode	All
Event	Click
BeforeAction	True
ActionSuccess	Ignore
ItemChanged	Ignore
InnerEvent	Ignore
CharPressed	
Modifier	None

First part of this filtering is the formtype, item column, row and MenuUid.

SAP Business One consists of thousands of windows but by entering the formtype of the window we would like to work on we do the first major filtering.

Next thing we filter by is what item we would like to work with. If we didn't provide an item every single item of the whole window would trigger the event.

NB: A FEW EVENTS LIKE FORM LOAD AND FORM UNLOAD SHOULD NOT CONTAIN AN ITEM BUT 95 % OF ALL EVENTS SHOULD HAVE THE FORMTYPE/ITEM COMBINATION.

If the item that you are working with is on line level you will have to provide the column that you are working with (else again ever of the lines columns would react to the trigger).

Last and most rarely user is the row-field. This can be used again to make the filter even more strict to say that as example that you only should react to the first of 10 rows.

Alternativ if event MenuClick is used Item, Col and Row are ignored for MenuUid which check if a specific menuUid is clicked.

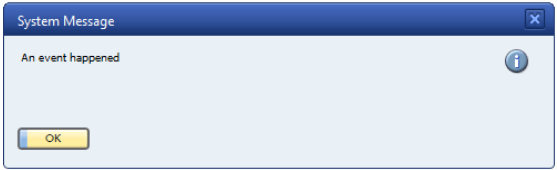
Samples

Trigger	Result
---------	--------

Trigger

FormType	134		
Item			
Column			
Row			
Menu Uid			
FormMode	All		
Event	Click		
BeforeAction	True		
ActionSuccess	Ignore		
ItemChanged	Ignore		
InnerEvent	Ignore		
CharPressed		Modifier	None

Every place on the business Partner master data that we press receive the event

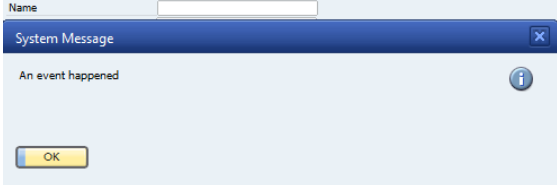


Trigger

FormType	134		
Item	7		
Column			
Row			
Menu Uid			
FormMode	All		
Event	Click		
BeforeAction	True		
ActionSuccess	Ignore		
ItemChanged	Ignore		
InnerEvent	Ignore		
CharPressed		Modifier	None

Now we reduce the event only to item 7 (The name field)

Name




Form Mode

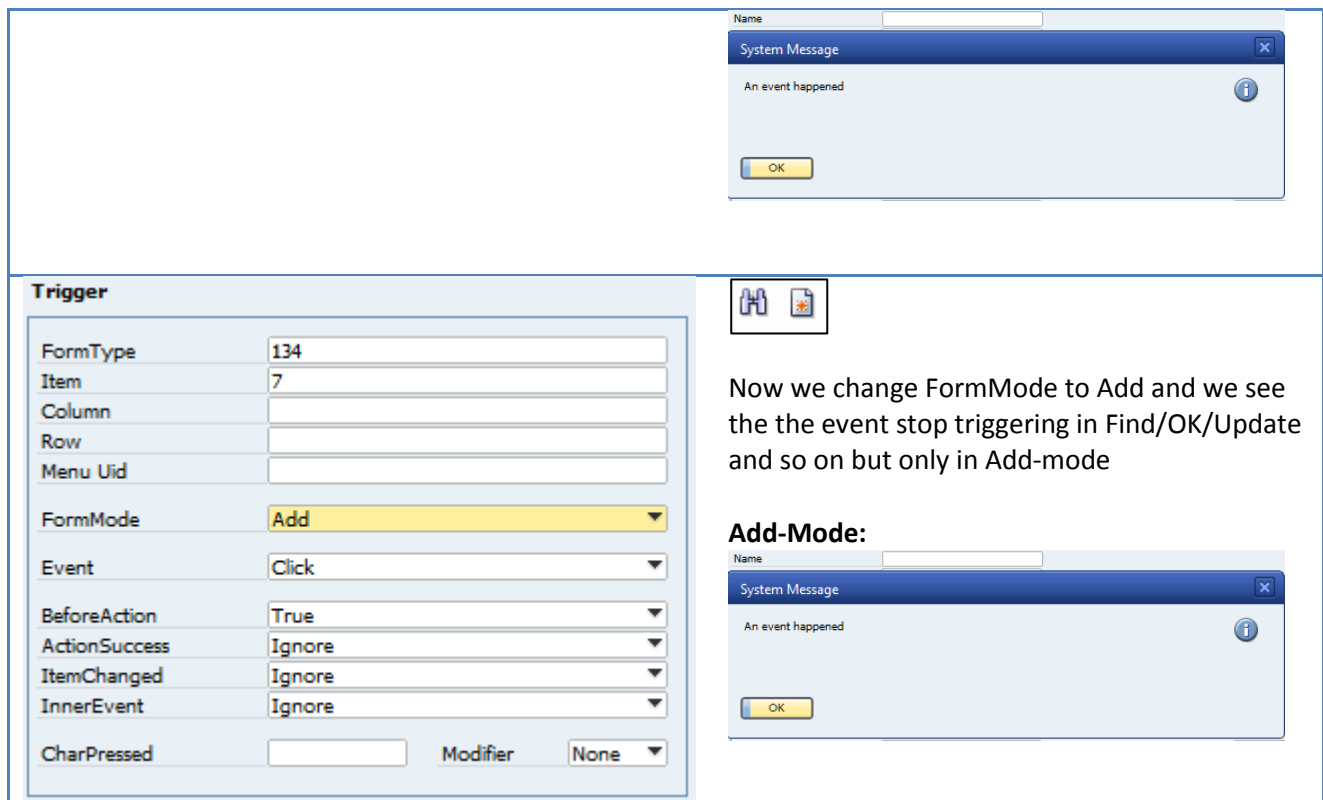
Next thing we can set is the form-mode

Trigger	
FormType	134
Item	7
Column	
Row	
Menu Uid	
FormMode	All
Event	0 - All
BeforeAction	1 - Add
ActionSuccess	2 - Find
ItemChanged	3 - Ok
InnerEvent	4 - Update
CharPressed	5 - OK/Update
	6 - Add/Update
	7 - Add/OK/Update
	8 - History (Changelog) mode
	9 - ReadOnly mode
	10 - OK/History
	11 - OK/ReadOnly
	12 - OK/History/ReadOnly
Comments	

This makes it possible to have a trigger only work on a certain window-mode. We can choose between the 6 different combinations

Samples

Trigger	Result																												
<table border="1"> <thead> <tr> <th colspan="2">Trigger</th> </tr> </thead> <tbody> <tr> <td>FormType</td> <td>134</td> </tr> <tr> <td>Item</td> <td>7</td> </tr> <tr> <td>Column</td> <td></td> </tr> <tr> <td>Row</td> <td></td> </tr> <tr> <td>Menu Uid</td> <td></td> </tr> <tr> <td>FormMode</td> <td>All</td> </tr> <tr> <td>Event</td> <td>Click</td> </tr> <tr> <td>BeforeAction</td> <td>True</td> </tr> <tr> <td>ActionSuccess</td> <td>Ignore</td> </tr> <tr> <td>ItemChanged</td> <td>Ignore</td> </tr> <tr> <td>InnerEvent</td> <td>Ignore</td> </tr> <tr> <td>CharPressed</td> <td></td> </tr> <tr> <td>Modifier</td> <td>None</td> </tr> </tbody> </table>	Trigger		FormType	134	Item	7	Column		Row		Menu Uid		FormMode	All	Event	Click	BeforeAction	True	ActionSuccess	Ignore	ItemChanged	Ignore	InnerEvent	Ignore	CharPressed		Modifier	None	<div>  </div> <p>Here we trigger in all modes.</p> <p>Find-Mode:</p> <div> Name <input type="text"/> </div> <div> System Message <div> An event happened </div> <div> OK </div> </div>
Trigger																													
FormType	134																												
Item	7																												
Column																													
Row																													
Menu Uid																													
FormMode	All																												
Event	Click																												
BeforeAction	True																												
ActionSuccess	Ignore																												
ItemChanged	Ignore																												
InnerEvent	Ignore																												
CharPressed																													
Modifier	None																												



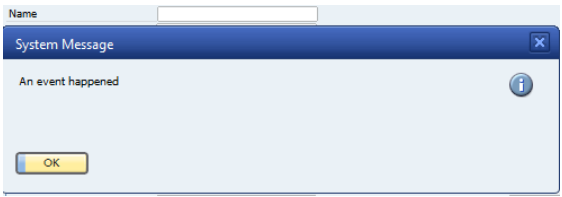
The screenshot shows the 'Trigger' configuration window on the left and a 'System Message' dialog on the right. The 'Trigger' window has the following settings:

Trigger	
FormType	134
Item	7
Column	
Row	
Menu Uid	
FormMode	Add
Event	Click
BeforeAction	True
ActionSuccess	Ignore
ItemChanged	Ignore
InnerEvent	Ignore
CharPressed	
Modifier	None

The 'System Message' dialog on the right shows the message: "An event happened" with an 'OK' button.

Now we change FormMode to Add and we see the the event stop triggering in Find/OK/Update and so on but only in Add-mode

Add-Mode:

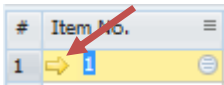
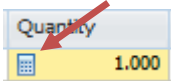


The 'System Message' dialog in Add-Mode shows the message: "An event happened" with an 'OK' button.

The different events

As of version 4.0.0.0 there are 19 types of events that are possible to catch

Event	Description
Click	<p>This event happens before all other events and basically happens every time you press the left mouse (even before you release the button and even if you does not press a specific item or items that are not normally something you press (an label as an example)).</p> <p>The click event is often confused with the much more common "item pressed" event.</p>
Combobox select	This event happens when a combobox selection is made. This event will only trigger on comboboxes on header or line-level.
Double click	<p>Double click is the same as the click event but if two clicks happens fast after each other.</p> <p>Please note that you will still get the first click event since at that point the system is unable to know that you will click again</p>
Form Load	<p>Form load happens one time when opening the a window and will not happen more on that window.</p> <p><u>NB: THIS EVENT ONLY NEED A FORMTYPE AND NOT AN ITEM SINCE IT WILL NEVER HAPPEN ALONG WITH AN ITEM; ONLY THE FORM</u></p> <p><u>TIP: THIS IS MOST USED BY THE UNIVERSAL FUNCTION CONTENT CREATOR.</u></p>

Form Closing	<p>Form Closing happens just before a window is about to be closed, but compared to form-closed happens before the windows is actually gone (This is handy if you want to prevent a window from closing)</p> <p><u>NB: THIS EVENT ONLY NEED A FORMTYPE AND NOT AN ITEM SINCE IT WILL NEVER HAPPEN ALONG WITH AN ITEM; ONLY THE FORM</u></p>
Form Closed (Form Unload)	<p>Form Unload it the equal to form load but happens instead when the window is closed.</p> <p>This event is very rarely used.</p>
Got Focus	This event happen when an edit text gets focus (aka it is marked with the yellow current field color)
Lost Focus	This event happens after a field is left. This is very similar to the more used Validate event, but the difference is that Lost Focus happens after you are out of the field (For that reason validation of entered data should not normally be Lost Focus but validate that happens just before you leave the field).
Item Pressed	This is the most often used event. The event happens when you press clickable items like buttons, tabs, checkboxes and linkedbuttons (Orange arrows)
Key Down	This event happens when a key is pressed (Use the Char Pressed and Modifier option for more on what key was pressed (ALT, CTRL or SHIFT))
Validate	This event is quite commonly used and happens when a field is about to lose focus (Often used when you want to trigger something based on what was entered in a field)
Menu Click	<p>This event happens when a menu-item is pressed. This menu-item can be form the toolbar, main menu, top-menu or right-click menu.</p> <p><u>NB: IF YOU USE A SHORTCUT KEY LIKE CTRL+F OR F1 A MENU-EVENT IT STILL FIRED.</u></p> <p><u>NB: IT IS NOT POSSIBLE TO USE A MENU CLICK EVENT WITHOUT ENTERING A MENUUID IN THE TRIGGER</u></p>
Data Load	This event sound similar to the Form Load but only happens once data is actually shown on the screen. Example if you open a Sales order a FormLoad will happen, but it is first when you navigate to an existing Sales Order that the Data Load event happens. Similar if you search or use a golden arrow to go to data a Data Load event happen.
Data Add	Similar to the Data Load event but happen everytime you add data to the screen (You press the Add-button and the data is saved)
Data Update	Similar to Data Add but when you update data
Data Delete	Similar to Data Add and Data Update but when data is removed (for the database, not simply switch to Find mode).
Column Link Pressed	<p>Happens if you press a golden arrow on cell level</p> 
Column Picker Pressed	<p>Happens if you press a picker (Choose From List, Calculator or Date-picker)</p> 

When B1UP is finished loading	As the name say, this event happen when B1 Usability Package is fully finished loading. This is handy if you wish some specific screen to be shown after startup.
--------------------------------------	---

There are tons of combinations here so please test this using your test-environment to see the different events in action.

Before Action, ActionSucess, Itemchanged and Inner Event

Trigger

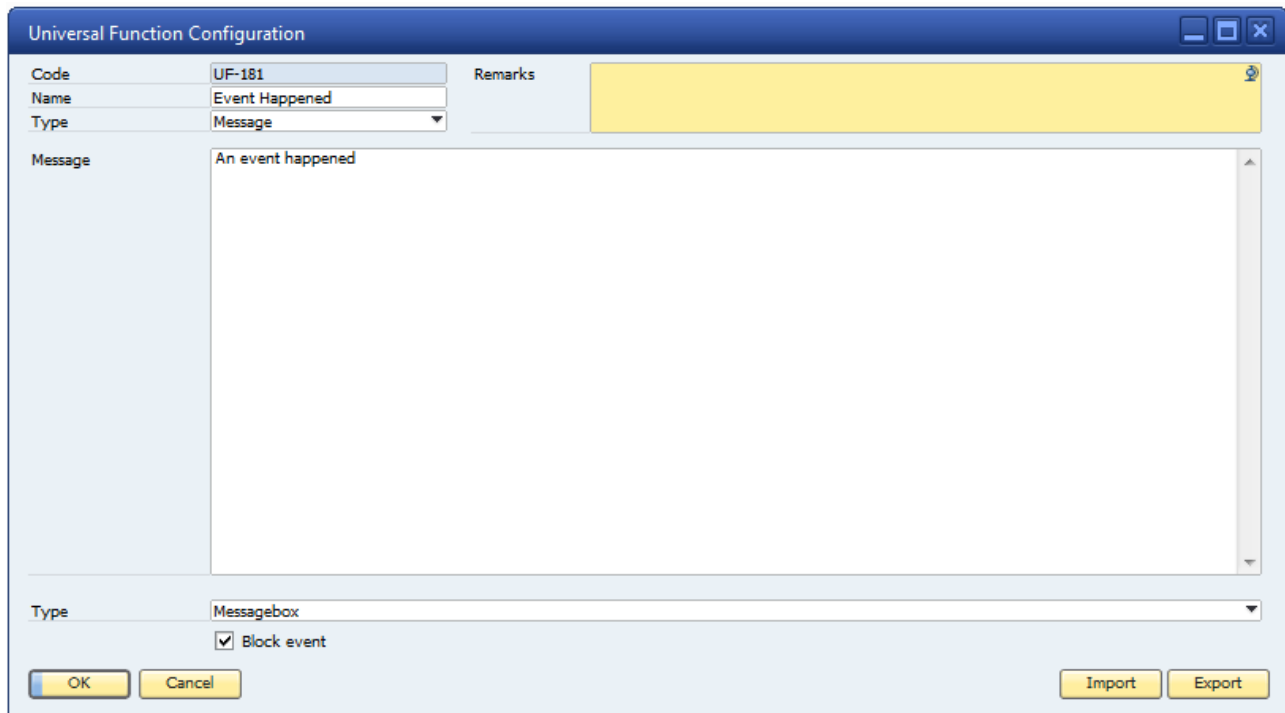
FormType			
Item			
Column			
Row			
Menu Uid			
FormMode	All		
Event	Item Pressed		
BeforeAction	True		
ActionSuccess	Ignore		
ItemChanged	Ignore		
InnerEvent	Ignore		
CharPressed		Modifier	None

These 4 advanced options can each be set to true, false or ignore.

Option	Description
BeforeAction + ActionSuccess	<p>The before action option and the Action Success properties, and let you determine when you wish to trigger the event compared to when SAP Business One react to the event.</p> <p>Best way to describe it better is by sample so please see below</p>
ItemChanged	This determines if you only want to react to the event it the item of the event was changed.
InnerEvent	User-events can in some rare cases cause other events to happen. These are called inner events. Sometimes this can lead to an endless loop of events. It is in general best to set this to ignore but in some rare cases you might need to change it.

Samples

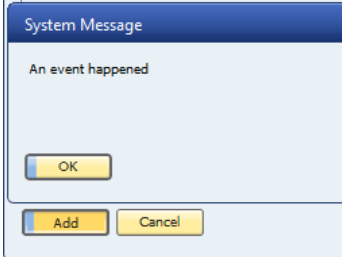
NB: PLEASE NOTE THAT FOR THESE SAMPLES WE NEED TO CHANGE THE TEST UNIVERSAL FUNCTION SLIGHTLY



The 'Universal Function Configuration' dialog box is shown. It has a blue title bar with standard window controls. The main area is divided into sections. The top section contains 'Code' (UF-181), 'Name' (Event Happened), and 'Type' (Message). To the right is a 'Remarks' field. Below this is a large text area for the 'Message' containing 'An event happened'. At the bottom, there is a 'Type' dropdown set to 'Messagebox' and a checked checkbox for 'Block event'. 'OK', 'Cancel', 'Import', and 'Export' buttons are at the bottom.

Code	UF-181	Remarks	
Name	Event Happened		
Type	Message		
Message	An event happened		
Type	Messagebox		
<input checked="" type="checkbox"/> Block event			
OK		Cancel	Import Export

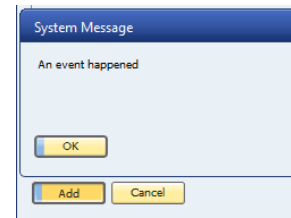
The block event property is now checked (Needed by the Before Action sample)

Sample	Result																										
<p>Trigger</p> <table border="1"><tr><td>FormType</td><td>134</td></tr><tr><td>Item</td><td>1</td></tr><tr><td>Column</td><td></td></tr><tr><td>Row</td><td></td></tr><tr><td>Menu Uid</td><td></td></tr><tr><td>FormMode</td><td>Add</td></tr><tr><td>Event</td><td>Item Pressed</td></tr><tr><td>BeforeAction</td><td>True</td></tr><tr><td>ActionSuccess</td><td>Ignore</td></tr><tr><td>ItemChanged</td><td>Ignore</td></tr><tr><td>InnerEvent</td><td>Ignore</td></tr><tr><td>CharPressed</td><td></td><td>Modifier</td><td>None</td></tr></table>	FormType	134	Item	1	Column		Row		Menu Uid		FormMode	Add	Event	Item Pressed	BeforeAction	True	ActionSuccess	Ignore	ItemChanged	Ignore	InnerEvent	Ignore	CharPressed		Modifier	None	<p>We here trigger clicking on the add-button in business partner.</p> <p>If we click on the button we see that the event happen before SAP add the business partner and we block the event so the add does not happen</p>  <p>The 'System Message' dialog box shows the message 'An event happened'. It has 'OK', 'Add', and 'Cancel' buttons.</p>
FormType	134																										
Item	1																										
Column																											
Row																											
Menu Uid																											
FormMode	Add																										
Event	Item Pressed																										
BeforeAction	True																										
ActionSuccess	Ignore																										
ItemChanged	Ignore																										
InnerEvent	Ignore																										
CharPressed		Modifier	None																								

Trigger	
FormType	134
Item	1
Column	
Row	
Menu Uid	
FormMode	Add
Event	Item Pressed
BeforeAction	Ignore
ActionSuccess	True
ItemChanged	Ignore
InnerEvent	Ignore
CharPressed	
Modifier	None

Now let us try to set the Before action to ignore and set Action success to true

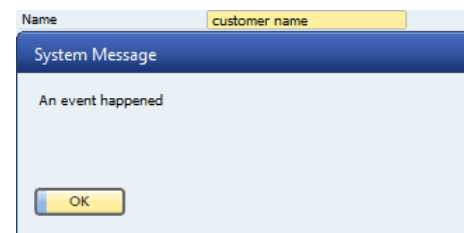
Now if we click on the button we see that the event happen after SAP add the business partner (After the success of SAP Business Ones work)



Trigger	
FormType	134
Item	7
Column	
Row	
Menu Uid	
FormMode	Add
Event	Validate
BeforeAction	False
ActionSuccess	Ignore
ItemChanged	Ignore
InnerEvent	False
CharPressed	
Modifier	None

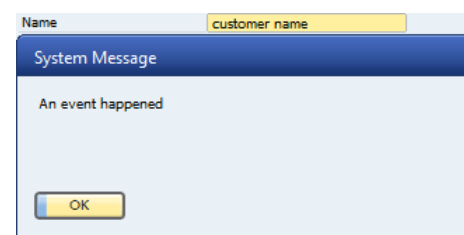
Now let's try something else. We are going to trigger validate of the customer name (REMEMBER TO SET BEFORE ACTION TO FALSE ELSE WE WILL BE BLOCKED AND BE UNABLE TO LEAVE THE FIELD)

If we go to the name field, enter a name and exit the field we get the event



Now let just try to enter the name field again and exit it without changing the name.

Note that we get the same event.



Trigger	
FormType	134
Item	7
Column	
Row	
Menu Uid	
FormMode	Add
Event	Validate
BeforeAction	False
ActionSuccess	Ignore
ItemChanged	True
InnerEvent	False
CharPressed	<input type="text"/> Modifier <input type="text"/> None

Now let's use the ItemChange option to True to change the above behavior

Now let just try to enter the name field again and exit it without changing the name.

Note that we get the event now.

Only if we enter the field, change the name and change it will we get the event.

Char pressed and Modifiers

Trigger	
FormType	
Item	
Column	
Row	
Menu Uid	
FormMode	All
Event	Item Pressed
BeforeAction	True
ActionSuccess	Ignore
ItemChanged	Ignore
InnerEvent	Ignore
CharPressed	<input type="text"/> Modifier <input type="text"/> None

Final part of the trigger is the char pressed and the Modifier. These are only used when you use the event Key Down (else they are ignored).


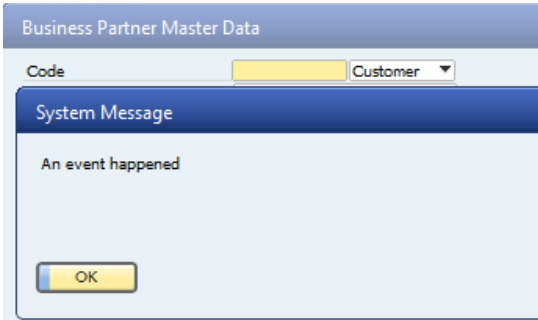

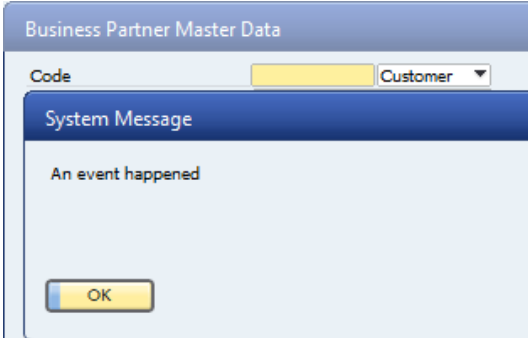
Basically what you can do is to enter the ASCII value of a keyboard char pressed and get an event for this.

NB: PLEASE NOTE THAT SAP ITSELF USES A LOT OF KEYBOARD SHORTCUTS AND IN ORDER FOR ADD-ONS NOT TO GET IN THE WAY NOT EVERY SHORTCUT COMBINATION WILL WORK

Char Pressed Table

Key	Value	Key	Value
ENTER	13	C	99
TAB	9	D	100
BACKSPACE	8	E	101
SPACE	32	F	102
PAGEUP	33	G	103
PAGEDOWN	34	H	104
END	35	I	105
HOME	36	J	106
DELETE	46	K	107
UPARROW	38	L	108
LEFTARROW	37	M	109
DOWNARROW	40	N	110
RIGHTARROW	39	O	111
0	48	P	112
1	49	Q	113
2	50	R	114
3	51	S	115
4	52	T	116
5	53	U	117
6	54	V	118
7	55	W	119
8	56	X	120
9	57	Y	121
A	97	Z	122
B	98		

Samples

Trigger	Result																										
<p>Trigger</p> <table border="1"> <tr><td>FormType</td><td>134</td></tr> <tr><td>Item</td><td>5</td></tr> <tr><td>Column</td><td></td></tr> <tr><td>Row</td><td></td></tr> <tr><td>Menu Uid</td><td></td></tr> <tr><td>FormMode</td><td>Add</td></tr> <tr><td>Event</td><td>Key Down</td></tr> <tr><td>BeforeAction</td><td>True</td></tr> <tr><td>ActionSuccess</td><td>Ignore</td></tr> <tr><td>ItemChanged</td><td>Ignore</td></tr> <tr><td>InnerEvent</td><td>Ignore</td></tr> <tr><td>CharPressed</td><td>36</td></tr> <tr><td>Modifier</td><td>None</td></tr> </table>	FormType	134	Item	5	Column		Row		Menu Uid		FormMode	Add	Event	Key Down	BeforeAction	True	ActionSuccess	Ignore	ItemChanged	Ignore	InnerEvent	Ignore	CharPressed	36	Modifier	None	<p>Char Press 36 Equals the Home-button and item 5 is the code, so if you enter the code field and press the Home-key the event will trigger.</p>  
FormType	134																										
Item	5																										
Column																											
Row																											
Menu Uid																											
FormMode	Add																										
Event	Key Down																										
BeforeAction	True																										
ActionSuccess	Ignore																										
ItemChanged	Ignore																										
InnerEvent	Ignore																										
CharPressed	36																										
Modifier	None																										
<p>Trigger</p> <table border="1"> <tr><td>FormType</td><td>134</td></tr> <tr><td>Item</td><td>5</td></tr> <tr><td>Column</td><td></td></tr> <tr><td>Row</td><td></td></tr> <tr><td>Menu Uid</td><td></td></tr> <tr><td>FormMode</td><td>Add</td></tr> <tr><td>Event</td><td>Key Down</td></tr> <tr><td>BeforeAction</td><td>True</td></tr> <tr><td>ActionSuccess</td><td>Ignore</td></tr> <tr><td>ItemChanged</td><td>Ignore</td></tr> <tr><td>InnerEvent</td><td>Ignore</td></tr> <tr><td>CharPressed</td><td>36</td></tr> <tr><td>Modifier</td><td>CTRL</td></tr> </table>	FormType	134	Item	5	Column		Row		Menu Uid		FormMode	Add	Event	Key Down	BeforeAction	True	ActionSuccess	Ignore	ItemChanged	Ignore	InnerEvent	Ignore	CharPressed	36	Modifier	CTRL	<p>Now we modify the trigger to use the modifier CTRL so now the event only happen if you hold down CTRL key while</p>  <p>CTRL +</p> 
FormType	134																										
Item	5																										
Column																											
Row																											
Menu Uid																											
FormMode	Add																										
Event	Key Down																										
BeforeAction	True																										
ActionSuccess	Ignore																										
ItemChanged	Ignore																										
InnerEvent	Ignore																										
CharPressed	36																										
Modifier	CTRL																										