



## SMS Event Reminders

A common Triggered SMS scenario is to trigger SMS event reminders from the Salesforce **EVENT** object. Often customers will have 3<sup>rd</sup> party calendar tools like Smart Cloud Connect, Revenue Inbox, Cirrus Insight, ZynBit, SF Lightning Sync, Calendly or TimeTrade sync'ing their Outlook/Gmail Calendars to the Salesforce **EVENT** object or they are using Salesforce itself to create the Event. In either case it is often quite costly if a customer misses an Appointment, thus SMS Reminders can greatly improve your appointment success rate.

While all the decent SMS apps offer **Scheduled SMS**, the process is manual. Most other SMS apps can of course trigger an SMS and use the Salesforce Process Builders "**Scheduled Actions**" feature but this creates a "black box" situation where the Salesforce User cannot see that the SMS is scheduled nor can the Scheduled Action be aborted easily.

360SMS is the **only** SMS app which provides a method to **programmatically** create the Scheduled SMS record. This method has the major advantage that the Scheduled SMS is visible in a Related List and can also be manually or programmatically deleted if the appointment is cancelled or you just don't want the reminders sent.

This document assumes you have first read the document [Sending SMS Through Process Builder](#). Most of this document talks about the idiosyncrasies of the Salesforce EVENT object more than the actual sending of the SMS which is the easy part.

### Oddities with the Salesforce Event Object

The biggest challenge with SMS Event Reminders has little to do with SMS and everything to do with the awkwardness of the Salesforce EVENT object. Two major challenges arise when working with the **EVENT** object.

1. The Event object is not a normal SF object in that it does not allow related lists and thus the SMS History cannot be linked to it.
2. The Event Object does not have direct links to the Contact/Lead or any object for that matter. The **Name** field and **Related To** fields, api names: WhoId and WhatId respectively, are polymorphic Id fields, meaning that they hold either Contact or Lead Ids and the WhatId can hold practically anything.
  - a. Unfortunately, because of this design there is no way to write formulas or use standard SF tools to traverse up to the Parent objects, such as getting a merge tag to the Contact.First\_Name
  - b. Luckily Salesforce provides Event.Who.FirstName and a few other fields and the 360SMS Merge Tag picker allows one to pick these merge tag fields.
  - c. If you need other custom fields from the Contact/Lead or the RelatedTo (WhatId) then there are fairly easy workarounds. Ask your Salesforce consultant or a 360SMS consultant for help.



## Event object linking challenge

Since you typically want an SMS Template or iText Survey Question to have both the FirstName and the Date/Time of the Event in the text message, you must make the Template or Survey based on the **EVENT** object in order to use merge fields of the EVENT. But this then presents the challenge that the SMS History cannot naturally link to the Contact or Lead because of the polymorphic WhoId.

As documented in the “*Relating Outbound SMS to an Alternate Object*” section of the [Sending SMS Through Process Builder](#) document, we must set the **Scheduled SMS Name** field to the Event.Id (shows as ActivityId in the picklist) and the **Related Object Id** to the Event.WhoId (shows as Name ID in the picklist). This is the precise use case for why the Scheduled SMS uses these two fields even though in normal triggers the ID’s are usually both the same Id. This awesome feature of 360SMS allows the template/survey to use the details of the Event (Date, Time, etc.) but then schedule the actual SMS to appear under the Contact/Lead.

The screenshot shows the configuration interface for a Scheduled SMS. It includes fields for Action Name, Record Type, and a table for Set Field Values. Two callouts provide context: a red one pointing to the [Event].Id value and a green one pointing to the [Event].WhoId value.

Action Name \* ⓘ  
Reminder 2 - Day Before

Record Type \*  
Scheduled SMS

Set Field Values

Field *	Type *	Value
Scheduled Sms Name	Field Reference	[Event].Id
Related Object Id	Field Reference	[Event].WhoId

The object of the ID used here must match the object of the Template or Survey

This links the SMS to the Contact/Lead



## Use SMS\_Reminder custom fields

We strongly recommend using formula fields on your Activity/Event object to hold the date/time that your reminders will fire. Create fields like SMS\_Reminder\_1, SMS\_Reminder\_2, SMS\_Reminder\_3 or perhaps SMS\_Reminder\_DayBefore, SMS\_Reminder\_15Minutes\_Before. Formulas have three major advantages:

1. You can more easily test/validate your date/time by looking at the resulting SMS\_Reminder field(s).
2. If you need to change your logic, you just change the formula without modifying the process builder
3. If using the Salesforce Scheduled Actions method rather than Scheduled SMS, then these fields are actually required as Salesforce does not allow formulas to calculate the dates within the Scheduled Actions interface.

**Important:** Salesforce does not allow you to add custom fields to the actual EVENT object as it is a virtual object, the real underlying object is the ACTIVITY object. Add your fields to the Activity Object. This article explains more if needed: <https://help.salesforce.com/articleView?id=000205377&type=1>

The screenshot shows the 'Edit Activity Custom Field' interface for 'SMS Reminder 3'. The field is defined with the following information:

- Field Label: SMS Reminder - Morning Of
- Field Name: SMS\_Reminder\_3
- Description: SR 12.7.19 - SMS Reminder 3 goes out on the morning of the appointment.
- Formula Return Type: Date/Time

The formula options section shows the 'Advanced Formula' tab selected. The formula is:

```
DATETIMEVALUE(LEFT(TEXT(ActivityDateTime),10) & " 15:00:00")
```

The formula is highlighted in yellow in the screenshot. The interface also includes a 'Simple Formula' tab and an 'Insert Field' button.

Figure 1 - A sample DateTime formula field for an SMS Reminder to be sent the **Morning Of** the appointment



For easy copy/pasting the formula is provided below with comments:

```

/*****
SF date math is tricky when wanting a specific time of day such as 8:00am. SF wants the
field in format: YYYY-MM-DD HH:MM:SS
Plus we have to compensate that SF returns everything in Greenwich Mean Time so you have to
adjust for your time zone.

1. Convert ActivityDateTime (shown in picklist as label TIME) into a text string
2. We just need the YYYY-MM-DD portion of it, so take left 10 characters
3. Now add the time component back adjusting for time zone Mountain Time = GMT minus 7 hrs.
4. Lastly convert it back into a datetimevalue

Note that if you are just wanting to calculate a few minutes or hours before the time you don't need this
complication. Sample formulas below:
1 hr before = ActivityDateTime - (1/24) (1 hour divided by 24 hrs in a day)
15 minutes before = ActivityDateTime - (15/60/24) (15 min divided by 60 min/hr divided by 24 hrs/day)
*****/

DATETIMEVALUE(LEFT(TEXT(ActivityDateTime - (7/24)),10) & " 15:00:00")

/* 15:00PM GMT (3:00 PM in UK) = 8:00am MST -7 hours difference*/

```

**Other common formulas:**

<b>24 hours before</b>
ActivityDateTime - 1 /*Salesforce date math is always in days*/

<b>10 minutes before</b>
ActivityDateTime - (10/60/24) /*Salesforce date math is always in days, so 10 min = 10 div 60min/hr div 24/hrs/day */

**Process Builder Itself**

The process builder itself is fairly trivial and better explained by reading the [Sending SMS Through Process Builder](#) document but for added clarity it is thoroughly documented below as a single screen capture does not fully show the formulas.

In this particular use case, we are using iText Survey Questions rather than Templates because we want to handle a possible CONFIRM or RESCHEDULE response. As noted in other documents, we feel strongly in always using Surveys rather than Templates because even if you don't have a call-to-action response, you never know when you might want to add one.

Again, this is redundant to the previous documentation but with Survey's it's an important and purposeful design element that we pass in a Question Id of the survey rather than the Survey Id. This is important because even though 99% of the time you will use the QuestionId of the first question, we have seen numerous use cases where it's useful to start a customer in the middle question of some particular survey such as when the survey is serving two purposes and the top of the survey asks questions that a particular customer has already answered or we



have the information but we don't want to create a whole new survey so we strategically send them a different question in the survey.

Event Reminder1 - Immediate Survey List [Create New Survey](#)

Survey: Event Reminder1 - Immediate Active Object: Event

Edit Clone Delete

Question: Hello {!Event.who.firstname} - {!Event.owner.firstname} here {!\$Organization.name} @ here. Confirming your appt for: + Answer  
 {!event.Activity\_Date\_as\_Text\_\_c} [Copy]

Reply CONFIRM or RESCHEDULE

Answer: CONFIRM || Yes || Y || 👍 || 🙏 || 🙌

Question: Great! See you then. Please arrive at least 15 minutes early. Label: Confirm Response + Answer

Answer: RESCHEDULE || I need to reschedule [Copy]

Question: No problem. Please suggest a new date/time and we'll get back to you to confirm in just a few minutes. Label: Reschedule Response + Answer

\*Answer: Exception Path [Copy]

Question: Sorry, I didn't understand that, I'm just a chatbot but I've alerted someone to answer you ASAP. Label: Exception Response + Answer

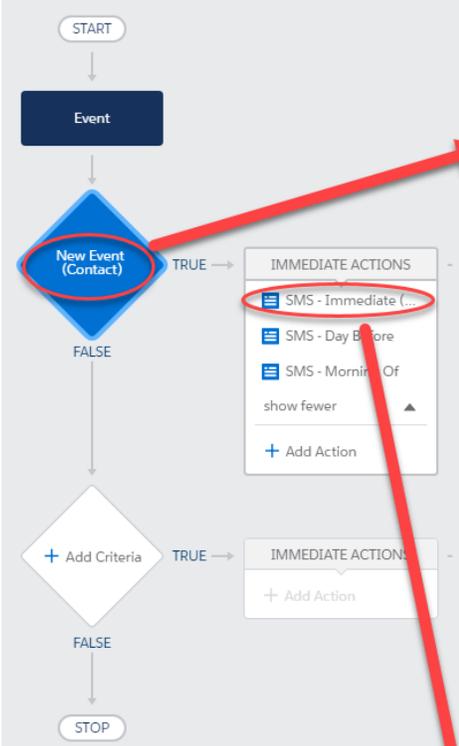
\* Other.

Get the QuestionId here or by drilling into the hyperlinked Question and getting it from the URL

Figure 2 – Important: the Template or Survey is based on the **EVENT** object in order to merge the date/time and AssignedTo. Because the Event.Whold (Name) is a polymorphic (Contact or Lead), Salesforce cannot traverse to either table but luckily they do have key formula fields like **FirstName**.

For your convenience here is a copy/pastable question with the merge tags. Note that a full set of custom ActivityDate formulas is on page 7 of this document, as Salesforce merges the ActivityDateTime value in a confusing **YYYY-MM-DDTHH:MM:SSZ** format which is also in Greenwich Mean Time, e.g. 2020-05-09T23:30:00Z similarly the Event.StartTime field is of no use because it too merges as a GMT time.

<b>Q1: EventReminderImmediate</b>
Hello {!Event.who.firstname} - {!Event.owner.firstname} here @ {!\$Organization.name}. Confirming your appt for: <span style="background-color: yellow;">{!event.Activity_Date_as_Text__c}</span>
Reply CONFIRM or RESCHEDULE
<b>Confirm/Yes Path</b>
CONFIRM    Yes    Y    👍    🙏    🙌



### Define Criteria for this Action Group

Criteria Name\* ?

Criteria for Executing Actions\*

Conditions are met

Formula evaluates to true

No criteria—just execute the actions!

Build Formula

Insert:

```

/*****
SR 5.5.20: A new event will schedule various SMS's to fire as reminders. We use formulas on
the actual ACTIVITY object such as SMS_Reminder_1, SMS_Reminder_2, ... so that we can
easily edit the dates and review them w/o updating this Process Builder.

Since EVENTS cannot have a child/related object we use the 360SMS technique of setting the
ScheduledSMS.Name to the EventId so that our template is based off of the EVENT object so
we can merge in the Event.Date fields. But we set the RELATED_OBJECT_ID to the Contact so
that it shows up on the Contact record and uses its Mobile number.

We use an iText (Survey Question) instead of a Template in case they write back with the
RESCHEDULE or CONFIRM keywords.
*****/
AND(
  ISNEW(), /* we'll have a more complex process for a meeting date/time changing */
  LEFT([Event].WhoId,3) = '003', /* we need separate code for a Contact vs. Lead */
)
    
```

### Create a Record ?

Action Name\* ?

Record Type\*

Set Field Values

Field*	Type*	Value*
Scheduled Sms Name	Field Reference	[Event].Id
Contact	Field Reference	[Event].WhoId
Related Object Id	Field Reference	[Event].WhoId
Scheduled Time	Field Reference	[Event].SMS_Remind...
Phone Api	String	MobilePhone
Question	ID	a1F1T00000HLrHmUAL

+ Add Row

**Event.Id so our Template or iText merges Event details**

**Relates the SMS to the Contact**

**Leave off if immediate but for consistency we like to see them all scheduled, even if just 1 minute from Now()**



## Use Activity Date Formulas for a richer experience

The standard Salesforce Event Date and Time fields are less than ideal when used in merge tags because they are shown in the Salesforce style **YYYY-MM-DDTHH:MM:SSZ** format. Below are some sample formulas you can use for better merge tags. Remember, the EVENT object does not allow custom fields so place these on the **ACTIVITY** object. All formulas below are data type = TEXT except where noted with the Activity\_Hour.

Refer to the next section for making the formulas customer Time Zone aware with our **Time Zone Anything™** product.

<b>Activity_Day</b>
<b>Description:</b> Simple formula to convert the Activity Date to the specific day of the month but as a text value for easier use in the larger Activity_DateTime_Text formula
<code>TEXT( DAY( ActivityDate ) )</code>

<b>Activity_Hour - Important: this is of type NUMBER</b>
<b>Description:</b> Simple formula that returns the Activity Hour as an <b>Integer</b> mostly to make the <b>Activity Time</b> formula easier to read and compile. Returns a number like 13 for 1:00PM or 0 for midnight
<pre>/* DateTime is shown in military time plus it is in GMT: 2020-05-09T23:00:00Z. */ /* 12th character and 2 of them gets the Hour. 23 in this case */ /* Then adjust for Mountain Daylight Time -6 hrs or Standard -7 */</pre>
<code>VALUE( MID( TEXT( ActivityDateTime - 6/24 ), 12, 2 ) )</code>

<b>Activity_Month</b>
<b>Description:</b> Simple formula to convert the Activity Date to the specific short English month value for easier use in the larger Activity_DateTime_Text formula
<pre>CASE(MONTH( ActivityDate ),   1, "Jan",   2, "Feb",   3, "Mar",   4, "Apr",   5, "May",   6, "June",   7, "July",   8, "Aug",   9, "Sept",   10, "Oct",   11, "Nov",   12, "Dec",   "Error")</pre>

<b>Activity_Year</b>
<b>Description:</b> Simple formula to convert the Activity Date to the year but as a text value for easier use in the larger Activity_DateTime_Text formula
<code>TEXT( YEAR( ActivityDate ) )</code>



### Activity\_Day\_Of\_Week

**Description:** Sometimes you want a merge tag to say. "Looking forward to our meeting on {Activity\_Day\_of\_Week}."

```
CASE( WEEKDAY(ActivityDate),
1, "Sunday",
2, "Monday",
3, "Tuesday",
4, "Wednesday",
5, "Thursday",
6, "Friday",
7, "Sunday",
"")
```

### Activity\_Time

**Description:** Creates a string like **5:00 PM** from Salesforce DateTime field that is in GMT and shown as a crazy long military time plus it is in GMT: 2020-05-09T**23**:00:00Z. See the Activity\_Hour\_\_c formula for extracting the hour and converting to Mountain time. In this case 23 – 6 = 17 (5:00PM MDT)

Ask Steve for help with dynamic daylight savings time formula or specific customer-based time zones.

```
/* Doesn't handle daylight savings so it's GMT-6 for MDT and GMT-7 for MST */
/* Datetime is like 2020-05-01T13:30:00Z so the Hour is the 12th character over plus 2 characters */

IF( OR( Activity_Hour__c = 0, Activity_Hour__c = 12 ),

  /*THEN - it's 12 noon or 12 midnight */
  "12",

  /*ELSE - get the HOUR and convert from military time: Hrs minus 0 before noon, Hrs - 12 afternoon*/
  TEXT(
    Activity_Hour__c - IF(Activity_Hour__c < 12, 0, 12 )
  )
) /*end of the hrs IF statement*/

& ":" &

/*MINUTES */
MID( TEXT( ActivityDateTime - 6/24 ), 15, 2 )

& " " &

/*AM or PM*/
IF(Activity_Hour__c < 12, "AM", "PM" )
```

And finally the big boy formula that combines all of the previous components together.

### Activity\_DateTime\_Text

**Description:** Salesforce ActivityDateTime field will merge in the unusable SF format, 2020-01-09T**23**:00:00Z

This formula converts the datetime into a friendlier text value: **Wednesday, Jan 13, 2020 @ 5:00 PM**

Note that this is using all the other formula field examples for easier management.

```
Activity_Day_of_Week__c & ", " & Activity_Month__c & " " & Activity_Day__c & ", " &
Activity_Year__c & " @ " & Activity_Time__c
```



The resulting Event Page Layout with all the various formulas is shown below:

Event

360SMS/

Location	Start	End
https://www.gotomeet.me/SteveRoch	5/19/2020 5:00 PM	5/19/2020 5:30 PM

**Details**
Attendees
Related
Chatter

▼ Calendar Details

Subject	360SMS/Traction	Assigned To	Steve Roch
Location	https://www.gotomeet.me/SteveRoch	Stop Drip	<input type="checkbox"/>
Start	5/19/2020 5:00 PM	Created By	Steve Roch, 5/14/2020 12:21 PM
End	5/19/2020 5:30 PM		
Type	Meeting		

▼ Related To

Name		SMS Reminder - Day Before	5/18/2020 5:00 PM
Related To		SMS Reminder - Morning Of	5/19/2020 9:00 AM
Attendees	Accepted or Maybe: 3, Declined: 0, No Response: 0 ✓ Steve Roch (Organizer), ✓	SMS Reminder - 10 min Before	5/19/2020 4:50 PM

▼ Dates as Text Strings for Merge Tags

Activity Day	19	Activity Day of Week	Tuesday
Activity Month	May	Activity Time	5:00 PM
Activity Year	2020	Activity DateTime Text	Tuesday, May 19, 2020 @ 5:00 PM
Activity Hour	17		

Figure 3 - The SMS Reminder formulas and Text-Based Date formulas for easy merge tags are shown in the page layout

Page 9 of 16



## Time Zone Anything™

The optional Time Zone Anything™ module from Bolder CRM handles the challenge of offering up the date/time to the customer in **their time** rather than your time. The formulas above fixed the GMT problem with date/times by hard-coding your time zone adjustment into the formulas. But what if you do business across multiple time zones and want to offer the time to the customer in his/her local time to avoid confusion?

By purchasing the Time Zone Anything™ product you can then add a lookup field to your Contact/Lead records or in this case the Event object (Activity). Then make minor adjustments to these formulas below. It even handles Daylight Savings Time:

<p><b>Activity_Hour - OPTIONAL Time Zone Anything™</b></p> <p><b>Description:</b> Simple formula that returns the Activity Hour as an <b>Integer</b> mostly to make the <b>Activity Time</b> formula easier to read and compile. Returns a number like 13 for 1:00PM or 0 for midnight</p> <p>If using the optional <b>Time Zone Anything™</b> product, change the manual time zone adjustment to dynamically reference the <b>Time_Zone.Offset_DST_Adjustment</b> field</p> <pre>/* DateTime is shown in military time plus it is in GMT: 2020-05-09T23:00:00Z. */ /* 12th character and 2 of them gets the Hour. */ /* Now using Time Zone Anything adjust to the customers Time */</pre> <p>VALUE( MID( TEXT( ActivityDateTime + Time_Zone_r.Offset_DST_Adjusted_c/24 ), 12, 2 ) )</p>
---

<p><b>Activity_Time - OPTIONAL Time Zone Anything™</b></p> <p><b>Description:</b> Creates a string like <b>5:00 PM</b> from Salesforce DateTime field that is in GMT and shown as a crazy long military time plus it is in GMT: 2020-05-09T<b>23</b>:00:00Z. See the Activity_Hour__c formula for extracting the hour and converting to Mountain time. In this case 23 – 6 = 17 (5:00PM MDT)</p> <p>Ask Steve for help with dynamic daylight savings time formula or specific customer-based time zones.</p> <pre>/* Datetime is like 2020-05-01T13:30:00Z so the Hour is the 12th character over plus 2 characters */ IF( OR( Activity_Hour__c = 0, Activity_Hour__c = 12 ),     /*THEN - it's 12 noon or 12 midnight */     "12",     /*ELSE - get the HOUR and convert from military time: Hrs minus 0 before noon, Hrs - 12 afternoon*/     TEXT(         Activity_Hour__c - IF(Activity_Hour__c &lt; 12, 0, 12 )     ) /*end of the hrs IF statement*/     &amp; ":" &amp;     /*MINUTES - some countries like India are 11.5 hrs difference from say Mountain time*/     MID( TEXT( ActivityDateTime + (Time_Zone_r.Offset_DST_Adjusted_c /24) ), 15, 2 )     &amp; " " &amp;     /*AM or PM*/     IF(Activity_Hour__c &lt; 12, "AM", "PM" )</pre>
--



And finally the big boy formula that combines all of the previous components together.

<b>Activity_DateTime_Text - OPTIONAL Time Zone Anything™</b>
<b>Description:</b> Salesforce ActivityDateTime field will merge in the unusable SF format, 2020-01-09T23:00:00Z This formula converts the datetime into a friendlier text value: <b>Wednesday, Jan 13, 2020 @ 5:00 PM</b> Note that this is using all the other formula field examples for easier management and compile size limits.
Activity_Day_of_Week__c & ", " & Activity_Month__c & " " & Activity_Day__c & ", " & Activity_Year__c & "@ " & Activity_Time__c & " " & <b>Time_Zone__r.Name</b>

The resulting Event Page Layout when Time Zone Aware now looks like this:

**Event** 360SMS/Traction C + Follow

Location: <https://www.gotomeet.me/SteveRoch> | Start: 5/19/2020 5:00 PM | End: 5/19/2020 5:30 PM

**Details** Attendees Related Chatter

**Calendar Details**

Subject	360SMS/Traction C	Assigned To	<a href="#">Steve Roch</a>
Location	<a href="https://www.gotomeet.me/SteveRoch">https://www.gotomeet.me/SteveRoch</a>	Stop Drp	<input type="checkbox"/>
Start	5/19/2020 5:00 PM	Created By	<a href="#">Steve Roch</a> , 5/14/2020 12:21 PM
End	5/19/2020 5:30 PM		
Type	Meeting		
Time Zone	Pacific		

**Related To**

Name		SMS RemInder - Day Before	5/18/2020 5:00 PM
Related To		SMS RemInder - Morning Of	5/19/2020 9:00 AM
Attendees	Accepted or Maybe: 3, Declined: 0, No Response: 0 ✓ <a href="#">Steve Roch</a> (Organizer), <a href="#">Show More</a>	SMS RemInder - 10 mln Before	5/19/2020 4:50 PM

**Dates as Text Strings for Merge Tags** w/ **360 Time Zone Anything** - customer time

Activity Day	19	Activity Day of Week	Tuesday
Activity Month	May	Activity Time	4:00 PM
Activity Year	2020	Activity DateTime Text	Tuesday, May 19, 2020 @ 4:00 PM Pacific
Activity Hour	16		



## Scheduled SMS versus Salesforce Scheduled Actions

Salesforce process builder offers a native **Scheduled Actions** feature in addition to the **Immediate Actions**. The actions available are the same as Immediate but the developer supplies either a DateTime field or there are some light capabilities to trigger 1 hour before or after. Most customers find Scheduled Actions to limiting for three reasons:

1. Not enough granularity to the date/time that the action should fire – it’s either 1 hr or 1 day.
2. The scheduled actions are stored in Salesforce **Jobs** which are not visible to end users – so they might not know that an action is even scheduled, unlike a Scheduled SMS which appears in the Related List.
3. Difficult to cancel – once an action is scheduled it is nearly impossible for an end user to cancel it
  - a. However, there are some techniques which you can use to make the scheduled action **fail** should some criteria change where you do not want the SMS to fire at its scheduled time. An example would be a homemade drip campaign where one triggers the creation of several scheduled actions over multiple days but if the status changes to some value, you want the messages to no longer send. Since Scheduled Actions cannot be stopped once started, we typically set the **PhoneAPI** fields as a formula such as:  
IF LEAD.Status = 'Stop Drip' THEN "" ELSE "MobilePhone"
    - i. So even though the action still fires, there is no phone field designated so it will just fail.



## Event Reminders with Salesforce Flows

For an even more robust event reminder solution combine the techniques above but instead of using Process Builder exclusively move all your logic to a Flow which is called by the EVENT process builder. Several screen captures below.

The screenshot displays the Salesforce Process Builder interface for a process named "Event Reminder Via SMS". The process flow starts with a "START" node, followed by an "Event" object. A decision diamond labeled "New Event" branches into three paths: "TRUE", "FALSE", and "CANCELLED". Each path leads to an "IMMEDIATE ACTIONS" group. The "TRUE" path includes an action "Event Reminder - N...". The "FALSE" path includes an action "Update - Event Re...". The "CANCELLED" path includes an action "Cancel - Event Rem...".

The "Define Criteria for this Action Group" panel for the "New Event" action is shown, with the following configuration:

- Criteria Name: New Event
- Criteria for Executing Actions:
  - Conditions are met
  - Formula evaluates to true
  - No criteria—just execute the actions!
- Build Formula:
 

```

      /*****
      This is the simple Event.WhoId method that most customers can use rather than the harder
      EventRelations method. We handle the NEW different than an Updated DateTime because the
      Update has to delete existing Scheduled SMS
      *****/
      ISNEW() &&
      NOT(ISBLANK([Event].WhoId ))
      
```

The "Launch a Flow" configuration panel is also shown, with the following details:

- Action Name: Event Reminder - New
- Flow: Event Reminder
- Set Flow Variables:
 

Flow Variable *	Type *	Value *
vEventId	Field Reference	[Event].Id
vNewOrUpdated	String	NEW

A red callout bubble points to the "vEventId" variable in the "Set Flow Variables" section, containing the text: "The flow exposes variables of type 'Input' so that the PB can pass in the values."

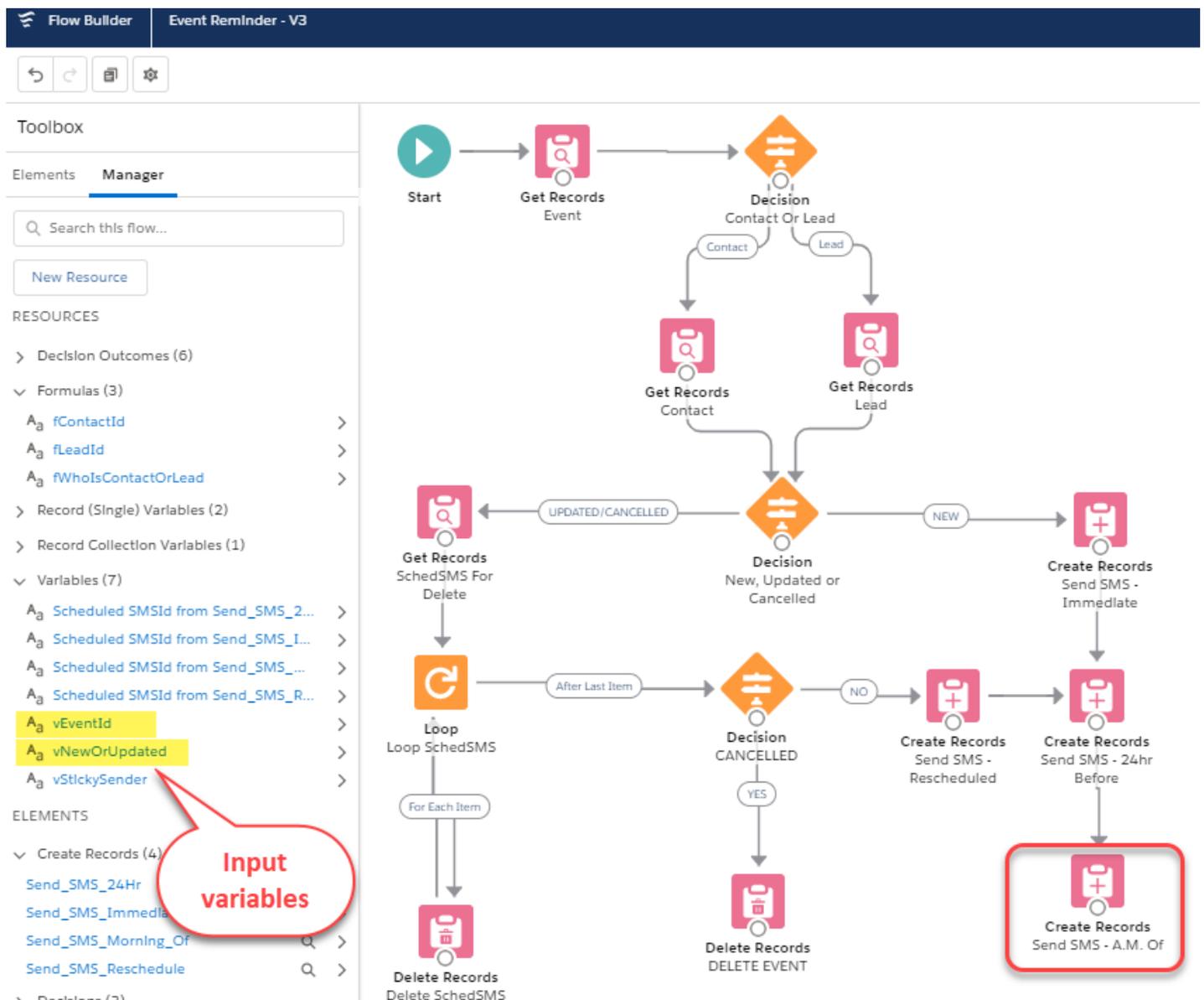
Figure 4 - You'll still want a Process Builder on the EVENT Insert or Update so that it can call the flow. Note that we have separate Criteria for the NEW, UPDATED Date/Time and CANCELLED events as the flow accepts an input parameter of one of these three values and acts accordingly.



The flow accepts the EventId and an input variable named **vNewOrUpdated** which the process builder passes in string values of NEW, UPDATED or CANCELLED.

- **NEW** - simply creates the various Scheduled SMS records.
- **UPDATE** - first looks for all unsend Scheduled SMS for this Event and deletes them since presumably the date/time changed. It then sends a “Your meeting has been rescheduled” SMS and reschedules the SMS’s for the new date/time.
- **CANCELLED** – simply deletes all the Scheduled SMS and then optionally deletes the actual Event.

The flow also has the advantage of being able to do a real Lookup on the Contact/Lead to get for instance the **Sticky\_Sender** whereas the Process Builder cannot do this because of the limitation on the Event.Whold.





Edit Create Records

Create Salesforce records using values from the flow.

\*Label:  \*API Name:

Description:

**How Many Records to Create**

One  
 Multiple

**How to Set the Record Fields**

Use all values from a record  
 Use separate resources, and literal values

**Create a Record of This Object**

\*Object:

**Set Field Values for the Scheduled SMS**

Field	Value
<input type="text" value="Name"/>	<input type="text" value="{!recEvent.Id}"/>
<input type="text" value="tdc_tsw__Contact__c"/>	<input type="text" value="{!fContactId}"/>
<input type="text" value="tdc_tsw__Lead__c"/>	<input type="text" value="{!fLeadId}"/>
<input type="text" value="tdc_tsw__Phone_Ap1__c"/>	<input type="text" value="MobllePhone"/>
<input type="text" value="tdc_tsw__Question__c"/>	<input type="text" value="a23f4000001fPI2AAE"/>
<input type="text" value="tdc_tsw__Related_Object_Id__c"/>	<input type="text" value="{!recEvent.WhoId}"/>
<input type="text" value="tdc_tsw__Scheduled_Time__c"/>	<input type="text" value="{!recEvent.SMS_RemInder_3__c}"/>
<input type="text" value="tdc_tsw__Sender_Number__c"/>	<input type="text" value="{!vStlckySender}"/>

Manually assign variables (advanced)

Figure 5 - The creation of the Scheduled SMS is the same as a Process Builder – but you can COPY/PASTE components in FLOWS!!!! And you can use the DESCRIPTION in every component to document your code!



## About the Author

Steve Roch, CEO of Bolder CRM is an SMS Industry expert having worked or consulted with the top three SMS Apps on the Salesforce AppExchange and also having built the popular Salesforce app [ActionGrid™](#), acquired by Conga in April-2016 and now renamed [Conga Grid™](#). Bolder CRM is the exclusive distributor of 360 SMS in the United States, Canada and the United Kingdom.

Learn more about Steve and Bolder CRM at <https://boldercrm.com/360SMS> and <https://www.linkedin.com/in/steveroch/>

Call/Text: [+1 720.605.0632](tel:+17206050632) or [+44 1234 480 564](tel:+441234480564) or [+61 488 845 944](tel:+61488845944)

WhatsApp: [+1 303.800.3258](tel:+13038003258) or [click here](#) to initiate a WhatsApp convo with Steve

Email: [steve@boldercrm.com](mailto:steve@boldercrm.com)