



University of Colorado

Boulder | Colorado Springs | Denver | Anschutz Medical Campus



## OnBase Guide - Workflow - Sending SMS Notifications

---

**Goal:** To send SMS/text message notifications from workflow

**Complexity Level:** Departmental Workflow Developers

7/2/2020

---

## Table of Contents

Background .....	3
Prerequisites.....	3
Steps to Complete in Studio .....	3
Example.....	4



---

## Background

In some cases, it may be beneficial to send notifications by SMS/text message instead of or in addition to an email. This can be accomplished using OnBase workflow by sending notifications to the email address provided by mobile phone carriers, ex [10 digit phone number]@[carrier address].

In order to send an SMS notification, you will need the recipients U.S. mobile phone number and their carrier.

There are a variety of ways you can look up the format for the carrier email addresses. For example, [this Digital Trends article](#) has a list and basic instructions.

Keep in mind that in non-production, notifications will go to the IMS inbox and will not be sent out. Refer to the [Expectations and Tips for Certified Admins guide](#) for more information on viewing notifications in non-production.

---

## Prerequisites

To configure SMS notifications, you will need to have Studio installed and be familiar with workflow configuration, in addition to having the necessary permissions to do so. Please reference other [UIS guides](#) for installation instructions for Studio.

To collect phone numbers and carrier information, you may also need to be familiar with including fields for that information in a Unity form template.

The Workflow MRG and Unity Forms MRG include further details. Contact [UIS\\_DM\\_Support@cu.edu](mailto:UIS_DM_Support@cu.edu) for assistance if needed.

---

## Steps to Complete in Studio

Set the recipient value

- The recipient address will be [10 digit phone number]@[carrier address]
- Set this value to a property (session property bag only) or keyword value depending on how you configured the notification.
- The *Set Property to Expression* action type is helpful for setting the property value depending on the carrier. The property value can be used as-is or stored to a keyword value.

Configure your notification.

- Keep the contents brief (less than 160 characters).
- Use a plain text notification for SMS notifications instead of a formatted notification.
- Add a recipient using either a property value (session property bag only) or a keyword value that will contain the phone number's email address.

Configure an action to send the notification.



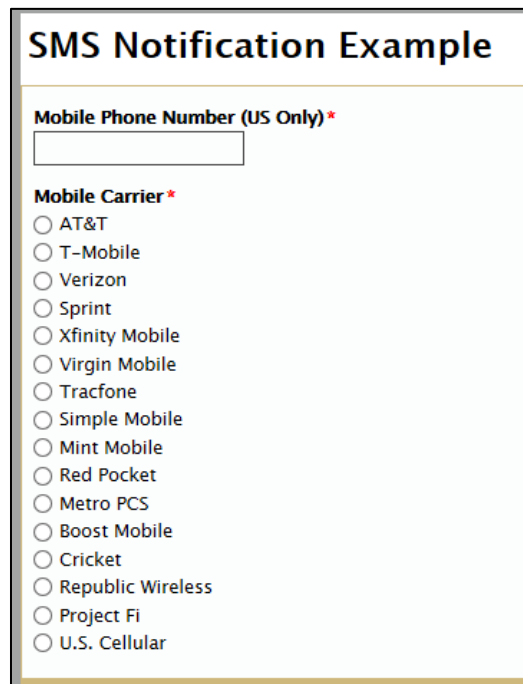
- Enter a sender address that you would like to appear as the sender, any replies will be directed to this address as well.

Replies to the SMS notification will not include the subject line, even if one was added to the SMS notification. In order to identify who a reply is from, you'd need to store the address as a keyword value so that you can search on that value.

---

## Example

While there are a variety of ways you could accomplish the same thing, in this example, the phone number and carrier are collected using a Unity form with two non-keyword fields.



The image shows a screenshot of a web form titled "SMS Notification Example". The form contains two main sections. The first section is labeled "Mobile Phone Number (US Only) \*" and features a single-line text input field. The second section is labeled "Mobile Carrier \*" and contains a list of radio button options for various carriers: AT&T, T-Mobile, Verizon, Sprint, Xfinity Mobile, Virgin Mobile, Tracfone, Simple Mobile, Mint Mobile, Red Pocket, Metro PCS, Boost Mobile, Cricket, Republic Wireless, Project Fi, and U.S. Cellular.

The radio button is configured to display the carrier name to the user, but the value saved is the carrier address extension:

### Radio Button Group Configuration

Configure Radio Button Group

**Data Type:**  
Text

**Orientation:**  
Vertical

**Radio Buttons:**

Label	Value	Default	Delete
AT&T	@txt.att.net	<input type="radio"/>	<input checked="" type="checkbox"/>
T-Mobile	@tmomail.net	<input type="radio"/>	<input checked="" type="checkbox"/>
Verizon	@vtext.com	<input type="radio"/>	<input checked="" type="checkbox"/>
Sprint	@messaging.sprintpcs.com	<input type="radio"/>	<input checked="" type="checkbox"/>
Xfinity Mobile	@vtext.com	<input type="radio"/>	<input checked="" type="checkbox"/>
Virgin Mobile	@vmobl.com	<input type="radio"/>	<input checked="" type="checkbox"/>
Tracfone	@mmst5.tracfone.com	<input type="radio"/>	<input checked="" type="checkbox"/>
Simple Mobile	@smtxt.com	<input type="radio"/>	<input checked="" type="checkbox"/>

When the form enters workflow, these values are copied from the form using the *Copy Property to/from Form Field* action type.

**Action Type**  
Copy Property to/from Form Field

**General** | Documentation

**Property Name**  
propPhoneNum

**Form Type**  
S - UIS - SMS Notification

**Form Field Name**  
phone

Copy To Form Field  
 Copy From Form Field

Then the values are concatenated:

**Action Type**  
Set Property to Expression

**General** | Documentation

**Property Name**  
propSMSAddress

**Expression**  
%VpropPhoneNum+%VpropPhoneCarrier

And stored to a keyword value which is configured as the recipient on the notification:

The image shows a configuration window titled "Action Type" with a dropdown menu set to "Set Keyword from Property Value". Below the dropdown are two tabs: "General" and "Documentation". Under the "General" tab, there are several fields and options:

- Keyword Type:** A dropdown menu set to "Email Address".
- Property Name:** A text input field containing "propSMSAddress".
- Options:** A group of radio buttons and checkboxes:
  - Add as new keyword value
  - Delete all existing keyword values
  - Replace existing keyword value
  - Constant value
  - Find value from another property
  - Delete Keyword if Property does not exist or is blank