

# Notify Guide

---

# Installed Components

---



**Note:** *This article applies to Fuji and earlier releases. For more current information, see [Notify](http://docs.servicenow.com)<sup>[1]</sup> at <http://docs.servicenow.com>*  
**The ServiceNow Wiki is no longer being updated. Visit <http://docs.servicenow.com> for the latest product documentation.**

## Overview

Activating the Notify plugin adds or modifies tables, user roles, script includes, and other components.



**Note:** *Notify replaces NotifyNow starting with the Eureka release. The basic functionality has remained the same, but has been extended.*

## Tables

Notify adds or modifies the following tables.

Display Name [Table Name]	Description
NotifyNow Conference Call [notifynow_conference_call]	Information for conference call records.
Participant [notifynow_participant]	Information regarding individual conference call participants.
SMS Messages [notifynow_message]	Information for the actual SMS messages sent. Starting with the Eureka release.
Answer [notifynow_answer]	Information for possible answers to the SMS question sent. Starting with the Eureka release.
Conversation [notifynow_conversation]	Information for the conversation thread. Starting with the Eureka release.
Question [notifynow_question]	Information for the possible questions in the SMS. Starting with the Eureka release.
Response_action [notifynow_response_action]	Information for the notify response action associated with the question. Starting with the Eureka release.
Response_choice [notifynow_response_choice]	Information for the notify response choice associated with the question. Starting with the Eureka release.
NotifyNow Participant Session [notifynow_participant_session]	Information regarding conference call sessions for individual conference call participants.

## Properties

Notify adds the following system properties.

Property	Description
Number of frequent participants to display [glide.notifynow.frequent_participant_count]	The number of frequent conference call participants to be displayed.
Voice [glide.notifynow.voice]	The voice used for Notify communications.
Twilio AccountSID [glide.notifynow.twilio.accountsid]	The Twilio account AccountSID. See Associating a Twilio Account for more details.
Twilio AuthToken [glide.notifynow.twilio.token]	The Twilio account AuthToken. See Associating a Twilio Account for more details.
Twilio SMS and voice enabled phone number [glide.notifynow.twilio.phonenumber]	The Twilio number that Notify will use.
Twilio answering machine detection [glide.notifynow.twilio.answering_machine_detection]	A boolean flag to enable or disable answering machine detection. Set to <b>true</b> to use answering machine detection. Set to <b>false</b> to continue the call. This property is available starting with the Eureka release.
Conversation thread release age [nn.thread.release.age]	The number of days after which the conversation thread release time runs out. Defaults to 10 days. This property is available starting with the Eureka release.
glide.notifynow.conference_call.code.pattern	The pattern used to generate a conference call code for public conference calls. Number signs (#) in the pattern are replaced with random numbers when a code is generated. This property is available starting with the Fuji release.
glide.notifynow.fix_invalid_phone_number	A boolean flag to enable or disable automatic correction of invalid phone numbers. Twilio might incorrectly prefix non-US phone numbers with <i>1</i> . When this property is true, the instance automatically removes the <i>1</i> for non-US calls. When this property is false, you might be incorrectly identified when reconnecting to a call using SMS. This property is available starting with the Fuji release.

## User Roles

Notify adds the following user roles.

Role	Contains Roles	Description
notifynow_admin	None	Administrator with privileges for Notify functionality.

## UI Actions

Notify adds the following UI actions.

UI Action	Table	Description
Initiate Conference Call	Incident Alert [incident_alert]	Adds a link to the Incident Alert form, if that plugin is activated, which displays a dialog box for starting a conference call with selected participants.
Invite to Conference Call	NotifyNow Conference Call [notifynow_conference_call]	Adds a link to the Notify Conference Call form which displays a dialog box for adding more users to an active conference call.

## Script Includes

Notify adds the following script includes.

Name	Description
IncidentAlertConferenceCall	A utility JavaScript Prototype class allowing users to initiate the conference call and add other users to the conference call. Should be used in conjunction with the slushbucket UI page (notifynow_participant).

## Business Rules

Notify adds the following business rules.

Business Rule Name	Table	Description
SMS on new Incident Alert	Incident Alert [incident_alert]	Sends an SMS to any default contacts added when an incident alert record is created.
Conference Call Allowed	Incident Alert [incident_alert]	Displays or hides the <b>initiate conference call</b> UI action by storing <b>true</b> or <b>false</b> in <code>g_scratchpad.conferenceCallAllowed</code> .
Update Conference Call Started IA Activity	NotifyNow Conference Call [notifynow_conference_call]	Logs when a conference call started by writing to an incident alert's comment field if the source record is from the incident_alert table.
Update Conference Call Finished IA Activity	NotifyNow Conference Call [notifynow_conference_call]	Logs when a conference call ended and what actions the conference call participants took by writing to an incident alert's comment field if the source record is from the incident_alert table.

## Workflow Activities

Notify adds the following workflow activities.

Activity Name	Activity Category	Description
Send Notify SMS	Notify	Sends an SMS message (maximum 1600 Characters) to an E.164 <sup>[2]</sup> compliant mobile phone number of the selected recipients and/or groups.
Send Notify SMS Question	Notify	Sends predefined SMS Question message (maximum 1600 Characters) to an E.164 compliant mobile phone number of the selected recipients and/or groups. This is one of two main activities for workflow On-Call Assign by Acknowledgement. In a predefined message, the recipient is asked to assign himself to a newly created incident. The same message also contains predefined answers. Starting with the Eureka release.
Send Email Question	Notify	Sends email generated from email template containing notification that recipient is the current On-call resource for a newly created task. This is one of two main activities for the workflow On-Call Assign by Acknowledgement. In a predefined message, the recipient is asked to assign himself to a newly created incident. The email contains two links that enable the user to accept or reject assignment. Starting with the Eureka release.

## References

- [1] [https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/product/notify2/concept/c\\_Notify2.html](https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/product/notify2/concept/c_Notify2.html)  
[2] <http://en.wikipedia.org/wiki/E.164>

# Notify API

---



**Note:** This article applies to Fuji and earlier releases. For more current information, see *Scripting Notify*<sup>[1]</sup> at <http://docs.servicenow.com> **The ServiceNow Wiki is no longer being updated. Visit <http://docs.servicenow.com> for the latest product documentation.**

## Overview

The Notify API provides functionality for sending emails, sending SMS messages, and setting up conference calls.

## Where to Use

Use this when you want to use Notify functionality with applications on your system. See *Using Notify with Incident Alert Management* for details on how this is implemented for incident alert management.

## Method Summary

### Fuji Release

The following methods are available starting with the Fuji release.

Method	Description
<code>mute</code>	Mutes a participant on a conference call.
<code>unmute</code>	Unmutes a participant on a conference call.
<code>kick</code>	Removes a participant from a conference call.

### Eureka Release

The following methods are available starting with the Eureka release.

Method	Description
<code>isSMSCapable</code>	Checks if the telephone number associated with the Twilio account is capable of sending SMS messages.
<code>isVoiceCapable</code>	Checks if the telephone number associated with the Twilio account is capable of setting up phone calls.
<code>sendSMSQuestion</code>	Sends an SMS question.
<code>sendEmailQuestion</code>	Sends an email question.
<code>getPreferredE164VoiceNumber</code>	Gets the preferred E.164 phone number of the user for sending a voice message.
<code>getPreferredE164SMSNumber</code>	Gets the preferred E.164 phone number of the user for sending an SMS message.
<code>getPreferredEmailAddress</code>	Gets the preferred email address of the user for sending an email.

## Dublin Release

The following methods are available starting with the Dublin release.

Method	Description
sendSMS	Sends an SMS message to a mobile phone number.
initiateConferenceCall	Initiates a new conference call with a number of participants.
addConferenceCallParticipant	Adds ad-hoc users to an existing conference call.
getConferenceCallParticipants	Returns all participants for a conference call.
isCallable	Determines whether a user is callable or not.
getFrequentlyCalledUsers	Returns frequently called users in alphabetical order.
getStatus	Returns the current status of the NotifyNow plugin.
getReadyState	Returns a boolean indicating whether Notify is correctly set up to use the Twilio account.
getPhoneNumber	Returns the phone number used for calls and SMS messages.
convertLocalPhoneNumberToE164	Converts a local phone number into an E.164 <sup>[2]</sup> -compliant number.

## Method Detail

### sendSMS

Sends an SMS message to an E.164 <sup>[2]</sup>-compliant mobile phone number (maximum 1600 characters).



**Note:** Notify supports international numbers starting with the Eureka release. Using this method with a number that does not support sending SMS messages results in an error being logged. Also see the `isSMSCapable` API call.

#### Parameters

- `phoneNumber` - (String) An E.164-compliant phone number.
- `smsBody` - (String) The body of the SMS message. The SMS body is truncated if the maximum length (1600) is exceeded. The maximum number of characters is determined by Twilio starting with the Eureka release.
- `sourceRecord` - (GlideRecord) [Optional]The source record to associate to the SMS message.

#### Returns

`void`

#### Example 1

```
var source = new GlideRecord("my_table");
source.query("my_field", "my_value");

if (source.hasNext() && source.next()) {
    // send a text message
    var nn = new SNC.NotifyNow();
    var message = "this is just a test";
    var number = "+31612345678";
    nn.sendSMS(number, message, source);
}
```

**Example 2 (by dotwalking)**

```
new SNC.NotifyNow().sendSMS("+31612345678", "this is a test", current);
```

**Example 3 (by dotwalking)**

```
new SNC.NotifyNow().sendSMS("+31612345678", "this is a test without  
source record");
```

**Example 4 (testing too large message body)**

```
new SNC.NotifyNow().sendSMS("+31612345678", Array(1601).join("a"));
```

**initiateConferenceCall**

Initiates a new conference call for an (optional) source record with a number of participants by passing a JavaScript array containing `sys_ids` from the `sys_user` table and/or E.164-compliant phone numbers.



**Note:** Notify supports international numbers starting with the Eureka release. Using this method with a number that does not support making outgoing phone calls results in an error being logged. Also see the `isVoiceCapable` API call.

**Parameters**

- `conferenceCallParticipants` - (String[]) One or more users, conference call participants, identified by the `sys_ids` from the `sys_user` table or E.164-compliant phone numbers.
- `conferenceCallTitle` - (String) Title of the conference call. This parameter has a maximum length of 40 characters.
- `sourceRecord` - (GlideRecord) [Optional] Source record to associate to the conference call. For example, an incident or problem number.
- `private` - (boolean) [Optional] Value to control if a conference call is private. This value defaults to false. This parameter is available starting with the Fuji release.

**Returns**

GlideRecord - the record of the created conference call, or null if something went wrong.

**Example 1**

This initiates a conference call with participants that have a E.164-compliant phone number and participants from the `sys_user` table and sends conference call details via SMS and email to all participants.

```
// define phone number participants
var participants = ['+31205655548', '+31205655552', '+31652825393'];

// we also want to add two Dutch sys_user participants
var user = new GlideRecord('sys_user');
user.addNotNullQuery('mobile_phone');
user.addQuery('mobile_phone', 'STARTSWITH', '+316');
user.setLimit(2);
user.query();

// add users to the participant array
while (user.hasNext() && user.next()) {
    gs.log('adding user ' + user.getValue('name') + ' with phone
```

```

number ' +
        user.getValue('mobile_phone') + ' to the participant
array');
    participants.push(user.getUniqueValue());
}

// define a source record to associate with the conference call
var source = new GlideRecord("cmdb_ci");
source.query("asset_tag", "P1000167");
if (source.hasNext() && source.next()) {
    // set up conference call
    var nn = new SNC.NotifyNow();
    var conferenceCall = nn.initiateConferenceCall(participants,
"testing 1 2", source);

    // check if the conference call was successfully created
    if (conferenceCall != null) {
        gs.log('started conference call: ' +
conferenceCall.getUniqueValue());
    } else {
        gs.log('could not start the conference call :(');
    }
}
}

```

### Example 2 (no source record)

This initiates a conference call with E.164-compliant phone numbers for participants, without the optional `source record` parameter and does not send any conference call details via SMS or email.

```

var participants = ['+31205655548', '+31205655552', '+31652825393'];

// set up conference call
var nn = new SNC.NotifyNow();
var conferenceCall = nn.initiateConferenceCall(participants,
"testing12");
gs.log('started conference call: ' + conferenceCall.getUniqueValue());

```

## addConferenceCallParticipant

Adds ad-hoc users to an ongoing conference call.

### Parameters

- `conferenceCall` - (GlideRecord or String) Conference call record or the `sys_id` of an active conference call.
- `participant` - (GlideRecord or String) User record or `sys_id` for a user with an E.164-compliant phone number, or an E.164-compliant phone number to call.



**Note:** When the method is called with a phone number for the *participant* parameter and there is exactly one `sys_user` record that matches the phone number, that `sys_user` record will be related to the participant. The participant's phone number field will be left blank because the phone number is in the `sys_user` record. If there are several `sys_user` records that match the phone number, or if there are no results, the participant's phone number field will be filled in, and there will be no stored reference to `sys_user` because the user is not known.



## Returns

GlideRecord - the participant record of the new participant that was added to the conference call.

## Example 1

```
// add a new participant by conference call sys_id (string) and phone
number (string)
var nn = new SNC.NotifyNow();
gs.log(nn.addConferenceCallParticipant('d193b242eb020100a04d4910f206fe39',
'+31612345678'));
```

## Example 2

```
// add a new participant by conference call sys_id (string) and user
record (GlideRecord)
var user = new GlideRecord('sys_user');
user.query('user_name', 'myUserName');
if (user.hasNext() && user.next()) {
    var nn = new SNC.NotifyNow();

gs.log(nn.addConferenceCallParticipant('d193b242eb020100a04d4910f206fe39',
user));

    // you could have added the user by sys_id as well:
    //
nn.addConferenceCallParticipant('d193b242eb020100a04d4910f206fe39',
user.getValue('sys_id'));
} else {
    gs.log('no such user');
}
```

## Example 3

```
// add a new participant by conference call record (GlideRecord) and
phone number (string)
var conferenceCall = new GlideRecord('notifynow_conference_call');
conferenceCall.query('title', 'IA0001001');
if (conferenceCall.hasNext() && conferenceCall.next()) {
    var nn = new SNC.NotifyNow();
    gs.log(nn.addConferenceCallParticipant(conferenceCall,
'+31612345678'));
} else {
    gs.log('no such conference call');
}
```

## getConferenceCallParticipants

Returns all participants for a conference call. By passing the `isCallable(participant)` parameter, you can limit the results to return only the participants you can call or only the participants you cannot call.

### Parameters

- `conferenceCallId` - (String) The ID of the conference call.
- `isCallable` - (boolean) [Optional] A flag to either return only the users you can call (if true) or those you cannot call (if false).

### Returns

GlideRecord - the participants.

### Example 1

```
var nn = new SNC.NotifyNow();
var user =
nn.getConferenceCallParticipants('c2e91710eb120100f34087b9d106fe37');

while (user.hasNext() && user.next()) {
    if (user.getValue('participant')) {
        gs.log('user: ' + user.getValue('sys_id'));
    } else {
        gs.log('phone number: ' + user.getValue('phone_number'));
    }
}
```

### Example 2

```
var nn = new SNC.NotifyNow();
var user =
nn.getConferenceCallParticipants('c2e91710eb120100f34087b9d106fe37',
true);

while (user.hasNext() && user.next()) {
    if (user.getValue('participant')) {
        gs.log('user: ' + user.getValue('sys_id'));
    } else {
        gs.log('phone number: ' + user.getValue('phone_number'));
    }
}
```

### Example 3

```
var conferenceCallId = '32b11430eb1201003cf587b9d106feb8';

// get all participants
gs.log('all conference call participants:');
var nn = new SNC.NotifyNow();
var user = nn.getConferenceCallParticipants(conferenceCallId);
gs.log(user);
```

```
// get all callable participants
gs.log('all conference call participants we can call:');
user = nn.getConferenceCallParticipants(conferenceCallId, true);
gs.log(user);

// get all un callable participants
gs.log('all conference call participants that are already in an active
session and whom we cannot call:');
user = nn.getConferenceCallParticipants(conferenceCallId, false);
gs.log(user);
```

## isCallable

Determines whether a user is callable or not. A user must have a valid phone number to be callable. A user who is already in an active session is not callable.

### Parameters

- `participant` - (GlideRecord or String) A `sys_user` or `notifynow_participant` record, or an E.164-compliant phone number.

### Returns

boolean - whether this participant can be called or not.

### Example

```
var nn = new SNC.NotifyNow();
gs.log('by number: ' + nn.isCallable('+31612345678'));

var user = GlideRecord('sys_user');
user.query('sys_id', '13d39544eb5201003cf587b9d106fea9');
if (user.hasNext() && user.next())
    gs.log('by user: ' + nn.isCallable(user));

var participant = GlideRecord('notifynow_participant');
participant.query('sys_id', '33b11430eb1201003cf587b9d106feb9');
if (participant.hasNext() && participant.next())
    gs.log('by participant: ' + nn.isCallable(participant));
```

## getFrequentlyCalledUsers

Returns a number of users, up to the `limit` parameter, in alphabetical order.

### Parameters

- `limit` - (int) The maximum number of results.

### Returns

GlideRecord - the frequently called users in alphabetical order.

### Example

```
var nn = new SNC.NotifyNow();
var fc = nn.getFrequentlyCalledUsers(10);
```

```
while (fc.hasNext() && fc.next()) {
    gs.log("got user " + fc.getValue('name') + ' - ' +
fc.getValue('sys_id'));
}
```

## getStatus

Gets the current status of Notify configuration.



**Note:** This method can only be accessed by administrators or users with the `notifynow_admin` role. Users with all other roles get the message `Unauthorized` when trying to run the function in a script.

### Parameters

- None

### Returns

String - one of the possible status messages shown in the table.

Possible responses include:

Status	Description
NO_NUMBER_MESSAGE	The account does not have a telephone number set up. Ensure that you set up the telephone number for the account. The status NO_NUMBER is used in versions prior to Eureka.
NO_ENDPOINTS_MESSAGE	The account does not have its endpoints set up correctly. Ensure that you set up the endpoints for the account. The status NO_ENDPOINT is used in versions prior to Eureka.
ACCOUNT_OK_MESSAGE	The account is active and ready for use. The status ACCOUNT_OK is used in versions prior to Eureka.
ACCOUNT_NO_AUTH	The Twilio AuthToken is not valid.
ACCOUNT_NOT_CONFIGURED	The Twilio AccountSID or AuthToken is not valid.

### Example

```
var nn = new SNC.NotifyNow();
gs.log(nn.getStatus());
```

## getReadyState

Indicates whether Notify is set up correctly or not.



**Note:** This method can only be accessed by administrators or users with the `notifynow_admin` role. Users with all other roles get the message `False` when trying to run the function in a script.

### Parameters

- None

### Returns

boolean - true if Notify is set up correctly, otherwise false.

### Example

```
var nn = new SNC.NotifyNow();
gs.log((nn.getReadyState()) ? "OK" : "NOT OK");
```

## getPhoneNumber

Gets the phone number used to make and receive calls, and to send and receive SMS messages.



**Note:** This method can only be accessed by administrators or users with the `notifynow_admin` role.

### Returns

String - E.164-compliant phone number. For example, +1123456789.

### Example

```
var nn = new SNC.NotifyNow();
gs.log(nn.getPhoneNumber());
```

## convertLocalPhoneNumberToE164

Converts a local phone number to an E.164-compliant phone number based on the user's location.

### Parameters

- `userID` - (String) The `sys_id` of a user to get location information from.
- `phoneNumber` - (String) The user's phone number.

### Returns

String - the E.164-compliant phone number.

### Example 1

```
var E164Number = new SNC.NotifyNow().convertLocalPhoneNumberToE164('...',
    '02012345678');
```

### Example 2

```
var localPhoneNumber = '01784 221600';
var userName = 'Heath Vanalphen';

var user = new GlideRecord('sys_user');
user.get('name', userName);
var E164Number = new
SNC.NotifyNow().convertLocalPhoneNumberToE164(user.getUniqueValue(),
localPhoneNumber);
gs.log('converted: ' + localPhoneNumber + ' to ' + E164Number + ' based
on ' + user.getValue('name') +
    '\s location (' + user.getValue('location') + ')');
```

## isSMSCapable

- When no argument is given: check if the telephone number associated with the Twilio account is capable of sending SMS messages.
- When a user `sys_id` is passed as an argument: check if the user is able to send SMS messages.

### Parameters

- `userID` - (String) [Optional]The `sys_id` of the user you want to check for an SMS-capable phone number.

### Returns

`boolean` - whether the telephone number associated with the Twilio account is capable of sending SMS messages.

#### Example 1

```
gs.log('the twilio number is SMS capable: ' + ((new
SNC.NotifyNow().isSMSCapable()) ? 'yes' : 'no'));
```

#### Example 2

```
gs.log('the user is able to send SMS messages (e.g. has a SMS device):
' + ((new SNC.NotifyNow().isSMSCapable('someuserid')) ?
    'yes' : 'no'));
```

## isVoiceCapable

- When no argument is given: check if the telephone number associated with the Twilio account is capable of setting up phone calls.
- When a user `sys_id` is passed as an argument: check if the user is able to set up voice calls.

### Parameters

- `userID` - (String) [Optional]The `sys_id` of the user you want to check for a voice-call capable phone number.

### Returns

`boolean` - whether the telephone number associated with the Twilio account is capable of setting up phone calls.

#### Example 1

```
gs.log('the Twilio number is Voice capable: ' + ((new
SNC.NotifyNow().isVoiceCapable()) ? 'yes' : 'no'));
```

#### Example 2

```
gs.log('the user is able to send SMS messages (e.g. has a SMS device):
' +
    ((new SNC.NotifyNow().isVoiceCapable('someuserid')) ? 'yes' :
    'no'));
```

## sendSMSQuestion

Sends an SMS question.

### Parameters

- `phoneNumber` - (String) An E.164-compliant phone number to send the message to.
- `question` - (GlideRecord or String) The question record to send or the `sys_id` of a question record.
- `sourceRecord` - (GlideRecord) [Optional] The source record to associate to the SMS question, such as an incident.

### Returns

String - conversation `sys_id` or null if the SMS was not successful.

### Example 1

```
var question = new GlideRecord("notifynow_question");
question.query();

// get the first question
if (question.hasNext() && question.next()) {
    // send the sms question
    var number = "+31612345678";
    var nn = new SNC.NotifyNow();
    nn.sendSMSQuestion(number, question.getUniqueValue(), current);
}
```

### Example 2 (by dotwalking)

```
new SNC.NotifyNow().sendSMSQuestion("+31612345678",
"b6071733bf1111003cf585ce2c07390f", current);
```

## sendEmailQuestion

Sends an email question to an email address, and is similar to `sendSMSQuestion`. The `sendEmailQuestion` method produces a question body and requires users to click a link to indicate their choice.

### Parameters

- `emailAddress` - (String) Email address to send the question to.
- `question` - (GlideRecord or String) The question record to send or the `sys_id` of a question record.
- `sourceRecord` - (GlideRecord) [Optional] The source record to associate to the SMS question, such as an incident.
- `emailSubject` - (String) [Optional] Text to override the default email subject.

### Returns

String - the `sys_id` of the conversation that contains the generated question.

### Example 1 (no email subject argument so use default)

```
var user = GlideRecord("sys_user");
user.get("email", "someone@somedomain.com");

new SNC.NotifyNow().sendEmailQuestion(user.getValue('email'),
"b6b34500bf3111003cf585ce2c0739ce", user);
```

**Example 2 (by dotwalking)**

```
new SNC.NotifyNow().sendEmailQuestion("someone@somedomain.com",
    "b6071733bf1111003cf585ce2c07390f", current,
    "Please answer this question");
```

**Example 3 (by dotwalking)**

```
new SNC.NotifyNow().sendEmailQuestion("someone@somedomain.com",
    "b6071733bf1111003cf585ce2c07390f",
    "Please answer this question");
```

## getPreferredE164VoiceNumber

Get the preferred E.164<sup>[2]</sup>-compliant phone number for setting up a voice call at this moment in time.

**Parameters**

- **User** - (GlideRecord or String) The user record or the sys\_id of a user to get the E.164-compliant phone number from.

**Returns**

String - the E.164-compliant phone number or null.

**Example**

```
var userID = "some user sys id";
var E164Number = new
SNC.NotifyNow().getPreferredE164VoiceNumber(userID);
gs.log('the preferred phone number for setting up voice calls is ' +
E164Number + ' for user with id: ' + userID);
```

## getPreferredE164SMSNumber

Get the preferred E.164<sup>[2]</sup>-compliant phone number for sending SMS notifications at this moment in time.

**Parameters**

- **User** - (GlideRecord or String) The user record or the sys\_id of a user to get the E.164-compliant SMS number from.

**Returns**

String - the E.164-compliant phone number or null.

**Example**

```
var userID = "some user sys id";
var E164Number = new SNC.NotifyNow().getPreferredE164SMSNumber(userID);
gs.log('the preferred phone number for sending SMS notifications is ' +
E164Number + ' for user with id: ' + userID);
```



## getPreferredEmailAddress

Get the preferred email address for sending emails at this moment in time.

### Parameters

- `User` - (GlideRecord or String) The user record or the `sys_id` of a user to get the email address from.

### Returns

String - the email address or null.

### Example

```
var userID = "some user sys id";
var email = new SNC.NotifyNow().getPreferredEmailAddress(userID);
gs.log('the preferred email address for sending email notifications is ' + email + ' for user with id: ' + userID);
```

## mute

Mutes a participant on a conference call. This method is available starting with the Fuji release.

### Parameters

- `participant` - (GlideRecord) the conference call participant to mute.

### Returns

boolean - true if the participant was muted, otherwise false.

### Example

```
var participantId = "<participant sys_id>";
var participant = new GlideRecord('notifynow_participant');
participant.get(participantId);
if (participant.isValid()) {
    // mute participant
    result = new SNC.NotifyNow().mute(participant);
    gs.log('participant muted: ' + result);
}
```

## unmute

Unmutes a participant on a conference call. This method is available starting with the Fuji release.

### Parameters

- `participant` - (GlideRecord) the muted conference call participant to unmute.

### Returns

boolean - true if the participant was unmuted, otherwise false.

### Example

```
var participantId = "<participant sys_id>";
var participant = new GlideRecord('notifynow_participant');
participant.get(participantId);
if (participant.isValid()) {
    // unmute participant
    result = new SNC.NotifyNow().unmute(participant);
    gs.log('participant unmuted: ' + result);
}
```

```
}
```

## kick

Removes a participant from a conference call. This method is available starting with the Fuji release.

### Parameters

- `participant` - (GlideRecord) the conference call participant to remove from the call.

### Returns

`boolean` - true if the participant was removed, otherwise false.

### Example

```
var participantId = "<participant sys_id>";
var participant = new GlideRecord('notifynow_participant');
participant.get(participantId);
if (participant.isValid()) {
    // kick participant
    result = new SNC.NotifyNow().kick(participant);
    gs.log('participant kicked: ' + result);
}
```

## References

- [1] [https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/product/notify2/concept/c\\_ScriptingNotify.html](https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/product/notify2/concept/c_ScriptingNotify.html)

# Administration

---

## Overview

Notify administrators can:

- View and edit Notify properties.
- Configure Notify for Twilio use.
- View conference call information and messages made by all applications using Notify on the instance.
- View or create questions and answers via SMS or email for bi-directional communication.

The name of the module has changed from NotifyNow to Notify starting with Eureka. To navigate in versions prior to Eureka, you can replace **Notify** with **NotifyNow**. In places where the whole path name has changed, the previous version path is provided.

---

# Viewing and Editing Notify Properties

- 1. Navigate to **Notify > Administration > Properties.**

🔔 Notify Properties

Account Status

Your Twilio AccountSID or AuthToken are not valid.

Notify Properties

Twilio AccountSID.

Twilio AuthToken.

.....

Twilio SMS and voice enabled phone number.

Number of frequent conference call participants to be displayed.

10

The voice used for Notify communications.

alice

- 2. Fill in the fields (see table).
- 3. Click **Submit**.

Field	Description
Account Status	Information showing the status of the associated Twilio account. See Account status messages.
Twilio AccountSID	The Twilio account AccountSID, acting as the user name for that account. See Associating a Twilio Account.
Twilio AuthToken	The Twilio account AuthToken, acting as the password for that account. See Associating a Twilio Account.
Twilio SMS and voice enabled phone number.	The Twilio number that Notify will use. Telephone numbers must be entered in the E.164 format and need to exist under the Twilio account. See Associating a Twilio Account.
Number of frequent conference call participants to be displayed	The number of people to display in the frequently called list.
Voice	The voice used for Notify communications. Select <b>woman</b> or <b>man</b> to provide female or male voice support for English, Spanish, French, German, and Italian. Select <b>alice</b> to provide female voice support for a wider range of languages.

## Account Status Messages

Message	Description
Your account is ready for use	Notify is correctly configured and ready for use.
Your Twilio AccountSID or AuthToken are not valid	An incorrect value has been entered in the <b>AccountSID</b> or <b>AuthToken</b> fields in the Notify properties.
Your Twilio phone number is not valid	Incorrect telephone number information is defined in the Notify properties.
Your Twilio phone number does not have properly configured endpoints	The <b>Request URL</b> endpoint settings have not been properly configured.
Your Twilio account is not configured properly	One or more of your Twilio account settings is incorrect. Open the Twilio dashboard and check the settings.

## Configuring Notify to Use Twilio

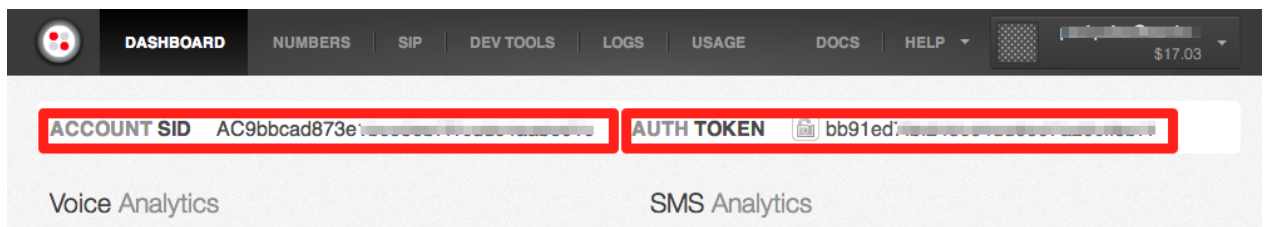
Each ServiceNow instance using Notify requires a valid Twilio <sup>[1]</sup> account and telephone number. Refer to the Twilio documentation for detailed instructions.

After the Twilio account is set up, perform the following to use that account with Notify:

- Associate the account with Notify.
- Configure Twilio with ServiceNow endpoints.

## Associating a Twilio Account

1. Navigate to **Notify > Administration > Properties**.
2. Enter the **AccountSID**, **AuthToken** and **telephone number** values. These values can be obtained from the Twilio dashboard:



When the Twilio account details are entered in the Notify Properties page, the account status is updated.

3. [Optional] To use Notify on multiple ServiceNow instances, activate Notify on each instance and create a separate Twilio account and telephone number for each instance.



**Note:** Some telephone numbers are voice capable, but not SMS capable. This is shown in the Notify properties with relevant messages. Two methods are available to check this starting with the Eureka release: `isSMSCapable` and `isVoiceCapable`. For more information, see *Notify API*.

## Configuring Twilio with ServiceNow Endpoints

In order for Twilio to receive Notify commands, the **Request URL** field value for every Twilio telephone number must point to the ServiceNow instance that uses Notify with that Twilio telephone number.

To set these values, do one of the following:

- Define them manually in the Twilio dashboard.
- Ensure the fields are blank in the Twilio dashboard, then open the Notify Properties page. Notify automatically configures the correct endpoints if the **Request URL** fields are blank.

Set these values from within the Twilio dashboard.

The screenshot shows the Twilio dashboard interface for configuring a phone number. The top navigation bar includes 'DASHBOARD', 'NUMBERS', 'SIP', 'DEV TOOLS', 'LOGS', 'USAGE', 'DOCS', and 'HELP'. Below this, a sub-navigation bar shows 'TWILIO NUMBERS', 'VERIFIED CALLER IDS', 'SHORT CODES', and 'PORTING REQUESTS'. The main content area is for a specific phone number, '+1 858-... (858) ...'. The 'Voice' section is selected, showing 'View Calls (Inbound | Outbound)' and 'URLs'. It contains four fields: 'Voice Request URL' (https://notifyusa.service-now.com/NotifyNowCallProcessor.do), 'Voice Fallback URL' (empty), 'Status Callback URL' (https://notifyusa.service-now.com/NotifyNowCallStatusProcessor.do), and 'Caller Name Lookup' (Disabled). Each URL field has a 'POST' button and a dropdown arrow. Below the 'Voice' section is the 'Messaging' section, which has a 'Messaging Request URL' field (https://notifyusa.service-now.com/NotifyNowSmsProcessor.do) and a 'POST' button. At the bottom, there is a 'Save Changes' button and a 'Release Number' link.

The **notifyusa** values underlined in the image should be replaced with your ServiceNow instance name, for:

- Voice Request URL: https://notifyusa.service-now.com/NotifyNowCallProcessor.do
- Status Callback URL: https://notifyusa.service-now.com/NotifyNowCallStatusProcessor.do
- Messaging Request URL: https://notifyusa.service-now.com/NotifyNowSmsProcessor.do

## Using Conference Calls

Applications that use Notify, such as incident alert management, can launch and maintain conference calls between involved parties.

To see a list of conference calls activated in the last week:

1. Navigate to **Notify > Conference Calls**.

You can also access conference call information from the relevant record, such as the **Conference Calls** related list in an incident alert record.

2. Click a conference call in the list to view details.

**Notify Conference Call**

Number:	CONF000001001	Source:	Incident Alert: IA0001002
Title:	IA0001002	Started at:	2013-10-29 08:27:26
Initiator:	System Administrator	Finished at:	2013-10-29 08:28:22

**Notify Conference Call Participants** Go to Participant

Conference call = CONF000001001

	Participant	Phone Number	Call duration	Active
<input type="checkbox"/>	Charles Taylor (Duty Manager)	+31 6 12345678	11 Seconds	false
<input type="checkbox"/>	John Smith (Duty Director)	+31 6 12345678	11 Seconds	false
<input type="checkbox"/>	Lucy Barnes (Incident Manager)	+31 6 12345678	30 Seconds	false

**Participant Sessions** Go to Response   1 Notify Conference Call Participant Session

Notify Conference Call Participant Sessions

	Response	Joined at	Left at
<input type="checkbox"/>	completed	2013-10-29 08:27:52	2013-10-29 08:28:22

Actions on selected rows...

1 to 3 of 3

- Click the arrow beside the check box to expand a **Participant** entry. This shows participant session information, including details of the conference calls that the participant has been involved in.
- Click a **Participant** name to see more detailed information about that participant.

**Notify Conference Call Participant**

Conference call:	CONF000001001	Call duration:	30 Seconds
Participant:	Lucy Barnes (Incident Ma	Active:	<input type="checkbox"/>
Phone Number:	+31 6 12345678	Netherlands	

**Participant Sessions** Go to Response   1 to 1 of 1

Notify Conference Call Participant Sessions

	Response	Joined at	Left at
<input type="checkbox"/>	completed	2013-10-29 08:27:52	2013-10-29 08:28:22

Actions on selected rows...

1 to 1 of 1

## Managing Conference Call Participants

You can mute, unmute, and kick participants on a conference call starting with the Fuji release. Muting a participant prevents that person from speaking on the conference call but does not notify any participants, including the muted participant. Kicking a participant removes that person from the conference call and sets the **Response** field for the participant session to **kicked**.

To mute, unmute, or kick a conference call participant:

- Navigate to **Notify > Conference Calls**.
- Select an active conference call.
- In the **Notify Conference Call Participants** related list, select a participant.
- Click the button for the action you want to perform, such as **Mute** to mute the participant.

You can perform these actions for multiple participants as a single operation using the **Actions on selected rows** choice list on the **Notify Conference Call Participants** related list.

## Using Public Conference Calls

A conference call can be public, starting with the Fuji release. The **Private** field on the conference call record indicates if a call is private (selected) or public (not selected). By default, all calls created automatically, such as those created through Incident Alert Management, are *public*. You can create private calls using the Notify API `initiateConferenceCall` method.

When created, public conference calls generate an associated **Code** which is sent via SMS to all invited participants. These participants can distribute the code to allow other users to join the public call. An administrator can control the format of the code using the property `glide.notifynow.conference_call.code.pattern`.

Anyone with the code can connect to a public conference call using one of these methods:

- By calling the Twillio phone number and entering the code for that conference call.
- By sending the code in an SMS message to the Twillio phone number. Participants that join a public call this way are considered *ad-hoc* participants, indicated on the participant record.

## Viewing Messages

Applications implementing Notify, such as incident alert management, can send Short Message Service (SMS) text messages to relevant contacts under predefined conditions, such as when a new incident alert has been created.

To see a list of SMS messages sent over the past week:

1. Navigate to the appropriate module for your version:
  - Eureka: **Notify > Messages**
  - Dublin: **NotifyNow > SMS Messages**
2. Click a message to see the message details.

The screenshot shows the 'SMS Outbox' interface. At the top, there's a header bar with a back arrow, 'SMS Outbox', and buttons for 'Update', 'Delete', and a search icon. Below the header, there are several input fields and buttons. The 'Table:' field is set to 'Incident Alert [incident\_alert]'. The 'Source:' field is set to 'Incident Alert: IA0001002'. The 'To:' field is set to '+31646293484'. The 'Body:' field contains the text 'An incident alert with number IA0001002 has been created.'. The 'State:' field is set to 'Sent'. At the bottom, there are 'Update' and 'Delete' buttons.

## Creating and Viewing Questions

You can define questions that contain a response action and a set of response choices. Use these questions to simplify communications. Then, you can monitor the resulting communication thread. Questions can be created starting with the Eureka release.

For example, the process can be used to ask members of a specific on-call group to acknowledge or reject an incident. Depending on the answer from the first team member, the incident is assigned to that person or an SMS question is sent to the next on-call member. In this example, the questions are used as part of a workflow.

To see a list of predefined questions both Notify and On-Call Scheduling need to be active:

1. Navigate to **Notify > Questions**.
2. Click the question **On-Call responsibility Accept/Reject** to see the question details.
3. Click the **Notify Response Choices** related list to view or edit the choices.

**Notify Question - On-Call responsibility Accept/Reject**

\* Name

\* Question

Params  Response Action

---

**Notify Response Choices**  Go to

Question = On-Call responsibility Accept/Reject

<input type="checkbox"/> <input type="button" value="Q"/> Code	<input type="checkbox"/> Text	<input type="checkbox"/> Order ▲
<input type="checkbox"/> <input type="button" value="ACC"/> ACC	Accept	1
<input type="checkbox"/> <input type="button" value="REJ"/> REJ	Reject	2

☐ Actions on selected rows...    to 2 of 2

The person who receives the SMS question can either choose to accept or reject the assignment. If the assignment is accepted, the **Response action** can be that the incident is actually assigned to that person. The response action is a script that you can modify or replace.

If the assignment is rejected, an SMS question may be sent to the next person in the escalation chain, and so on. You can combine questions with a workflow to automate a process like on-call scheduling escalation.

To create a new question:

1. Navigate to **Notify > Questions**.
2. Click **New**.
3. Fill in the fields, as appropriate (see table).
4. Click **Submit**.

Field	Description
Name	Enter a description of the question.
Question	Enter the question you want to send. You can enter variables in the question with the { } tags, for example, {0}. When using more than one variable, use consecutive numbering. When using multiple variables, reference them by specifying their position in the parameter list, starting from {0}. The variable is picked up by the script and translated into, for example, an incident or a problem number.
Params	Enter parameters which map to the fields on the record that the workflow was triggered for. This enables the message to show dynamic data. Multiple parameters should be separated by a comma. For example, <i>number</i> , shows the incident number in case the workflow was triggered from an incident. A second parameter could be, for example <i>short_description</i> .
Response Action	Select the action that must be taken. This is a script that can be modified, or you can create a new script by clicking the search button and selecting <b>New</b> . Several examples of scripts are shown. Write your own script and click <b>Submit</b> .

To create a new response choice:

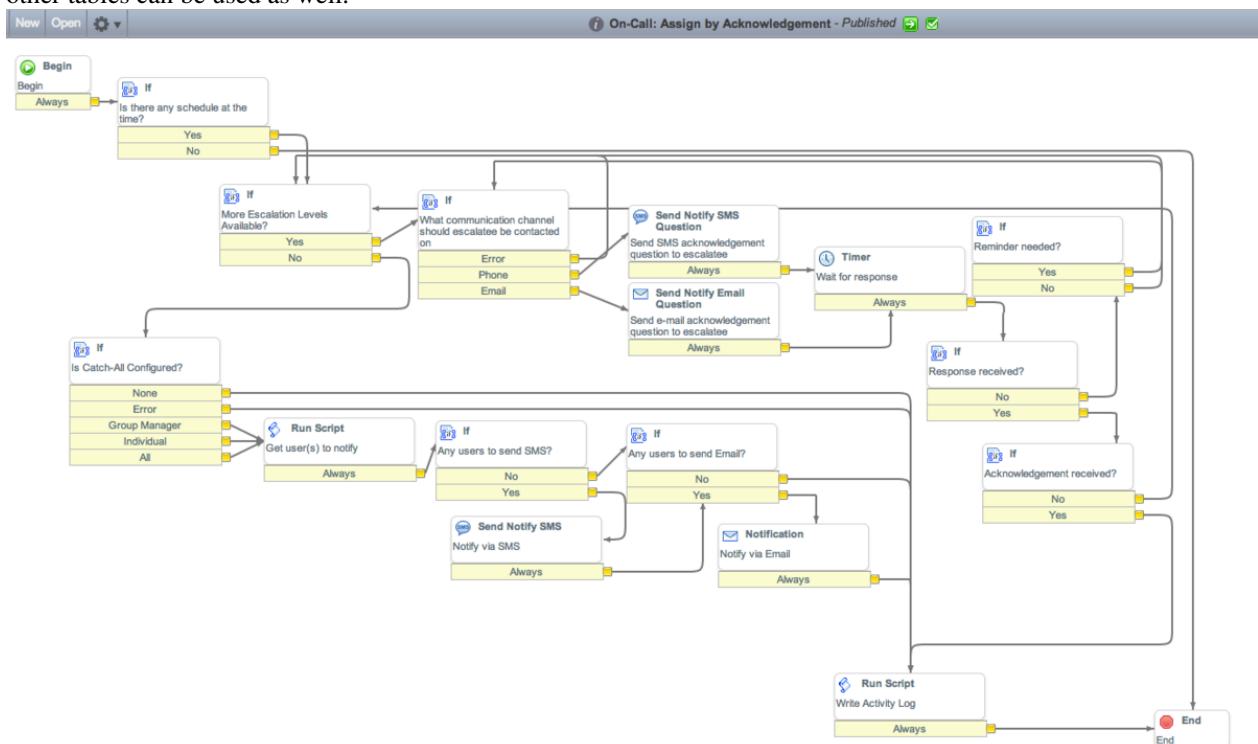
1. Navigate to **Notify > Questions**.
2. Open the notify question you have just created.
3. Click **New** to create a new notify response choice.
4. Fill in the fields, as appropriate (see table).
5. Click **Submit**.



Field	Description
Code	The abbreviation for the response choice in the text message. For example, <i>ACC</i> for accept or <i>REJ</i> for reject.
Text	The text displayed for this response choice.
Order	A number indicating the order in the list of response choices. Choices with lower order numbers are listed before choices with higher order numbers.

## On-Call: Assign by Acknowledgement Workflow

The On Call:Assign by Acknowledgement workflow, which is part of on-call scheduling, enables sending a question whether a person who is on-call accepts assignment to an incident that has just been created or not. The person can accept, reject or not respond to the assignment request that was sent through SMS or email. Depending on the answer or reply time-out, the workflow branches or loops until it is finished. If no user accepts assignment in time, then the catch-all person is notified, if configured. The workflow is used with a trigger rule based on the incident table, but other tables can be used as well.



To define notify questions, see [Creating and Viewing Questions](#). An example question, **On-Call responsibility Accept/Reject**, is provided in the demo data.

To view notify conversations, see [Viewing Conversations](#).

To view the emails sent, navigate to **System Logs > Emails**.

## Viewing Conversations

The Notify Conversation [notifynow\_conversation] table tracks bi-directional communications and adds the ability to have more than one active conversation with the same phone number at the same time. The thread number is combined with the phone number to link the SMS or email response to the correct conversation. Conversations remain active until a response is received or until they are automatically deactivated after the thread release time runs out. The default thread release time is 10 days. To change this value, add the `nn.thread.release.age` property.

To see a list of all active and inactive conversations:

1. Navigate to **Notify > Conversations**.
2. Click a conversation record to see the conversation details.

The **State** of the conversation can have one of the following values:

Value	Description
Ended	The conversation has been completed.
Failed	The conversation has failed.
Pending	The SMS or email is waiting to be sent.
Sent	The SMS or email has been sent.
Unanswered	The SMS or email has not been answered.



**Note:** The mode of conversation cannot be changed during one conversation thread. For example, you cannot switch between SMS and email during one conversation.

## References

- [1] <https://www.twilio.com/>

# Article Sources and Contributors

**Installed Components** *Source:* <http://wiki.servicenow.com/index.php?oldid=250642> *Contributors:* David.Bailey, Fuji.publishing.user, John.ramos, Joseph.messerschmidt, Ludwig.adriaansen, Phillip.salzman

**Notify API** *Source:* <http://wiki.servicenow.com/index.php?oldid=250754> *Contributors:* Cheryl.dolan, David.Bailey, Fuji.publishing.user, John.ramos, Joseph.messerschmidt, Ludwig.adriaansen

**Administration** *Source:* <http://wiki.servicenow.com/index.php?oldid=239744> *Contributors:* Fuji.publishing.user, Ludwig.adriaansen, Phillip.salzman

# Image Sources, Licenses and Contributors

**Image:Warning.gif** *Source:* <http://wiki.servicenow.com/index.php?title=File:Warning.gif> *License:* unknown *Contributors:* CapaJC

**Image:Notify-Properties.png** *Source:* <http://wiki.servicenow.com/index.php?title=File:Notify-Properties.png> *License:* unknown *Contributors:* David.Bailey

**File:TwilioDashboard.png** *Source:* <http://wiki.servicenow.com/index.php?title=File:TwilioDashboard.png> *License:* unknown *Contributors:* Paul.palse

**File:TwilioTelephoneNumberDashboard.png** *Source:* <http://wiki.servicenow.com/index.php?title=File:TwilioTelephoneNumberDashboard.png> *License:* unknown *Contributors:* David.Bailey, Paul.palse

**Image:NotifyNow\_confcall\_details.png** *Source:* [http://wiki.servicenow.com/index.php?title=File:NotifyNow\\_confcall\\_details.png](http://wiki.servicenow.com/index.php?title=File:NotifyNow_confcall_details.png) *License:* unknown *Contributors:* David.Bailey

**Image:NotifyNow\_confcall\_participants.png** *Source:* [http://wiki.servicenow.com/index.php?title=File:NotifyNow\\_confcall\\_participants.png](http://wiki.servicenow.com/index.php?title=File:NotifyNow_confcall_participants.png) *License:* unknown *Contributors:* David.Bailey

**Image:SMS\_Details.png** *Source:* [http://wiki.servicenow.com/index.php?title=File:SMS\\_Details.png](http://wiki.servicenow.com/index.php?title=File:SMS_Details.png) *License:* unknown *Contributors:* David.Bailey

**Image:NotifyQuestion.png** *Source:* <http://wiki.servicenow.com/index.php?title=File:NotifyQuestion.png> *License:* unknown *Contributors:* Ludwig.adriaansen

**File:OnCallSchedulingByAcknowledgement2.png** *Source:* <http://wiki.servicenow.com/index.php?title=File:OnCallSchedulingByAcknowledgement2.png> *License:* unknown *Contributors:* Ludwig.adriaansen

**Image:Conversation\_Details.png** *Source:* [http://wiki.servicenow.com/index.php?title=File:Conversation\\_Details.png](http://wiki.servicenow.com/index.php?title=File:Conversation_Details.png) *License:* unknown *Contributors:* Georgi.ivanov