Single Email Helper Plug-In – User Guide

Version 1.8

Contents

Overview

The Single Email Helper Plug-In was designed to assist in a common business use cases for leveraging email functionality within Process Builder:

1. When using Process Builder and using an 'Email Alerts' action, you need to ensure that an Email Template and Email Alert are configured for that object in order to leverage that action type.

2. When using Email Templates, you can't use cross object merge fields. In order to solve this, you need to use a custom formula field to pull the field value into the specific object and then it can be used within your email template.

3. When using standard Email Alerts within Process Builder, the Email Alert can only be sent to an existing users, groups or roles within your Salesforce instance.

The Single Email Helper Plug-In assists in each of these business use cases

1. The package includes an Apex Class that can be used as an action within your process (regardless of object) and allows you to set the following fields on your email: To, CC, Display Name, Subject & Body.

2. The inputs for your email message within your process accept merge fields from not only the record that is firing the process but also records that are related to that record (up to 5 relationships away). Simply enter merge field syntax within variables and the email content will be dynamic based on the record.

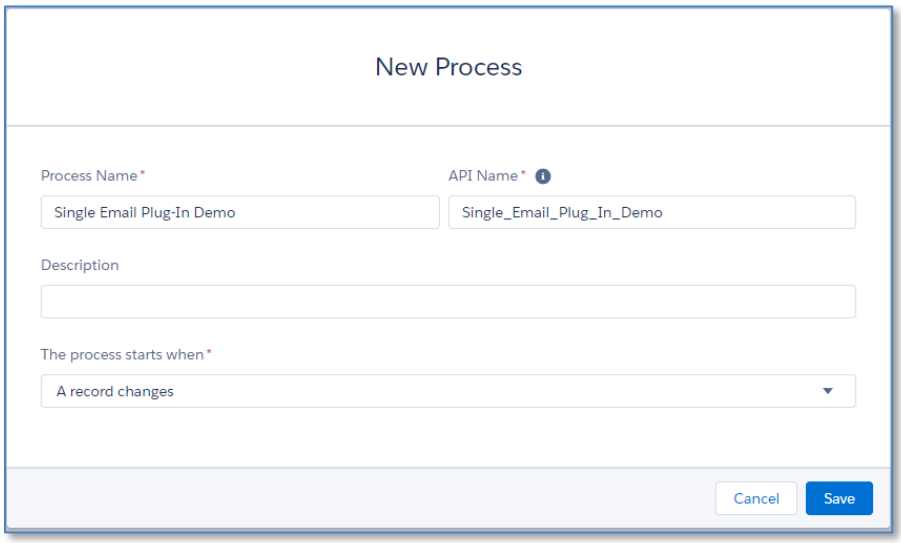
3. The plug-in is designed with a free-form text field for the email address which will be the primary and CC'ed recipients of the email. Additionally, both allow for multiple email addresses to be entered using commas as delimiters.

Implementing the Single Email Helper Plug-In in Process Builder

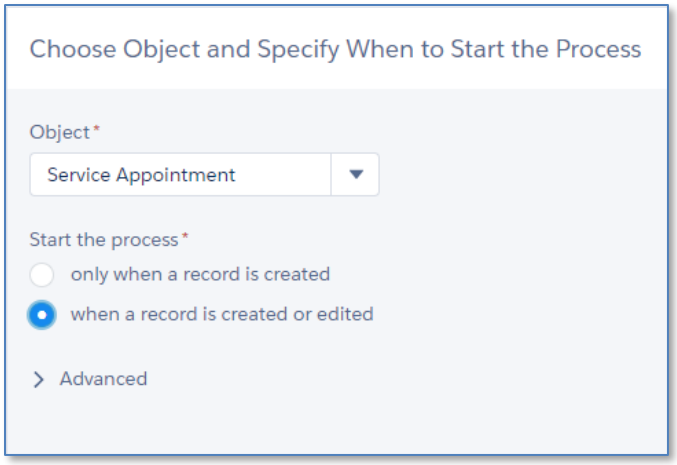
The Single Email Helper Plug-In delivers one Apex class that can be used via Process Builder. The class is to be used in an immediate or scheduled action within your new or existing Process Builder processes. Depending on your use case, you should determine the appropriate timing for the Single Email Helper Plug-In to be run.

In order to use the Single Email Helper Plug-In in your Process Builder process, follow the steps below:

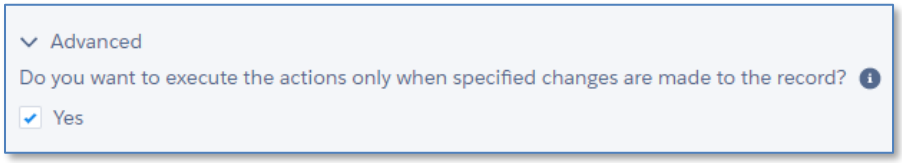
1. Navigate to Process Builder via the Setup menu (Build → Create → Workflow & Approvals → Process Builder).
2. If there is already an existing process where you want to add the Single Email Helper Plug-In call to, skip to step 10, otherwise, click on the 'New' button from the Process Builder screen.
3. Enter a name for the process and determine how this process will be called. For this example, we'll create a new process that is invoked when a record is changed.



4. Click the '+ Add Object' shape.
5. Select the object you want to have the Single Email Helper Plug-In run on.
    - In this example, the Service Appointment object will be used but any object that you can build a process on can be used in its place.
6. Next, select when the process should start.
    - The Single Email Plug-In can be used both when a record is created and when a record is updated. For this example, we'll select the "when a record is created or edited" option.
7. Once you have selected the object and when to start the process, click 'Save'.

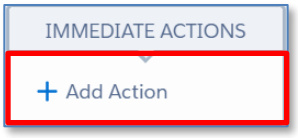


8. Next, click the '+ Add Criteria' shape and enter the criteria for which you want your assignment rules to be run.
    - In this example, we are going to enter no criteria however, implement the correct criteria for your scenario in which you want the email generated.
    - Additionally, if you want the Single Email Plug-In to run as a Scheduled Action as opposed to an Immediate Action, be sure to select the checkbox under the Advanced section.

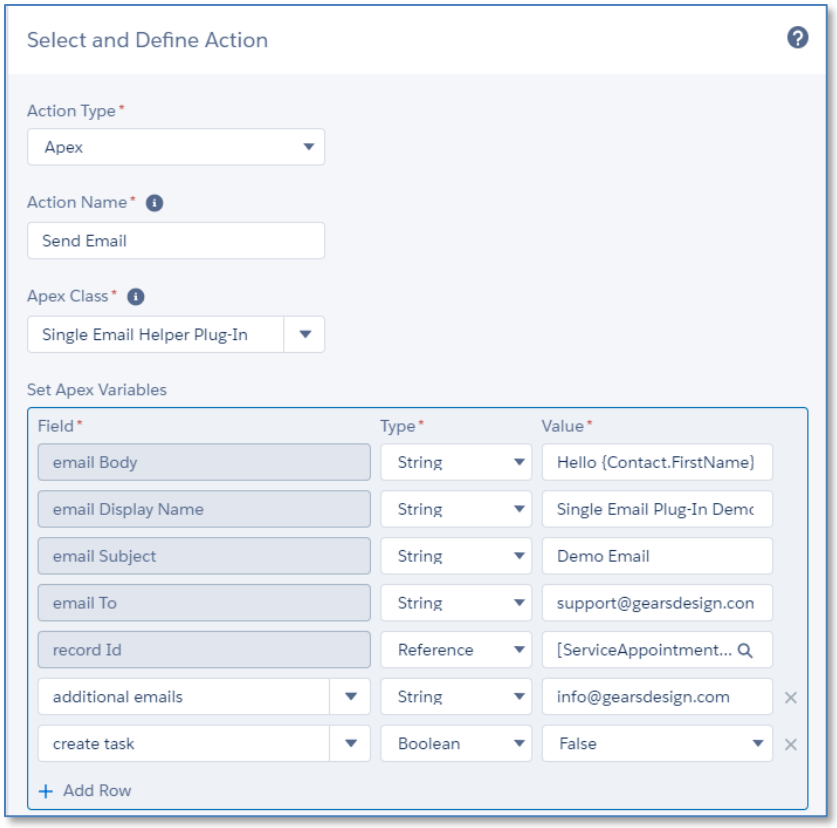


9. After entering the criteria, click 'Save'.

10. Depending on whether you are leveraging the Single Email Plug-In as an immediate or scheduled action, click the '+ Add Action' under the appropriate shape
    - In this example, an Immediate Action will be used.



11. In the resulting Action screen, select the following and then click 'Save':
    - Action Type = Apex
    - Action Name = <Intuitive Name for Action> (Ex. Send Email)
    - Apex Class = *Single Email Helper Plug-In*
    - Under 'Set Apex Variables' the following immediately appear:
        - **email Body** = Enter the body of the email that is being sent.
            - Be sure to use the "String" type variable and not "Formula".
            - If using merge fields in your email body, see the expected formatting for this feature under the Using Merge Fields in 'email Body' & 'email Subject' Variables section.
        - **email Display Name** = Enter the name/value that should appear on the inbound email.
            - Be sure to use the "String" type variable and not "Formula".
        - **email Subject** = Enter the Subject value for the email message.
            - Be sure to use the "String" type variable and not "Formula".
            - If using merge fields in your email subject, see the expected formatting for this feature under the Using Merge Fields in 'email Body' & 'email Subject' Variables section.
        - **email To** = Enter the email address(es) which the email should be sent to
            - Be sure to use the "String" type variable and not "Formula".
            - If sending to multiple email addresses, separate each with a comma (Ex. emailaddress1@demo.com, emailaddress2@demo.com...).
        - **record Id** = Use a "Reference" type variable and select the object ID for which the process is built for.
    - Additionally, there are two optional Apex Variables that can be added. To add these, simply click the '+ Add Row' link under the default Apex Variables:
        - **additional emails** = Enter any email addresses that should be CC'ed on the email message.
            - Be sure to use the "String" type variable and not "Formula".
            - If sending to multiple email addresses, separate each with a comma (Ex. emailaddress1@demo.com, emailaddress2@demo.com...).
        - **create task** = Select either "True" or "False" to determine whether a corresponding task record should be created for the email.



12. Finally, click the Activate button on your process to enable your new/updated process

Using Merge Fields in 'email Body' & 'email Subject' Variables

The Single Email Plug-In uses some common Salesforce syntax along with basic HTML tags in order to generate the body or subject of the email message. The tables below outline the expected syntax for the 'email Body' & 'email Subject' variables. Additionally, a few examples have been provided to show the resulting outputs.

Salesforce Syntax

Action	Input	Example
Retrieve field from object which process is built on	{<Object API Name>.<API Field Name>}	{ServiceAppointment.DueDate} / {Widget__c.Priority__c}
Retrieve field from parent object of record triggering process	{<Related Object API Name>.<API Field Name>}	{Account.Name} / {Contact.FirstName}
Retrieve field from parent of parent object of record triggering process	{<Related Object API Name>.<Parent Related Object API Name>.<API Field Name>}	{Account.Parent.Name}
Retrieve Base URL of record triggering process	{getSalesforceBaseURL}	https:// {getSalesforceBaseURL}/{Account.Id}

Common HTML Tags

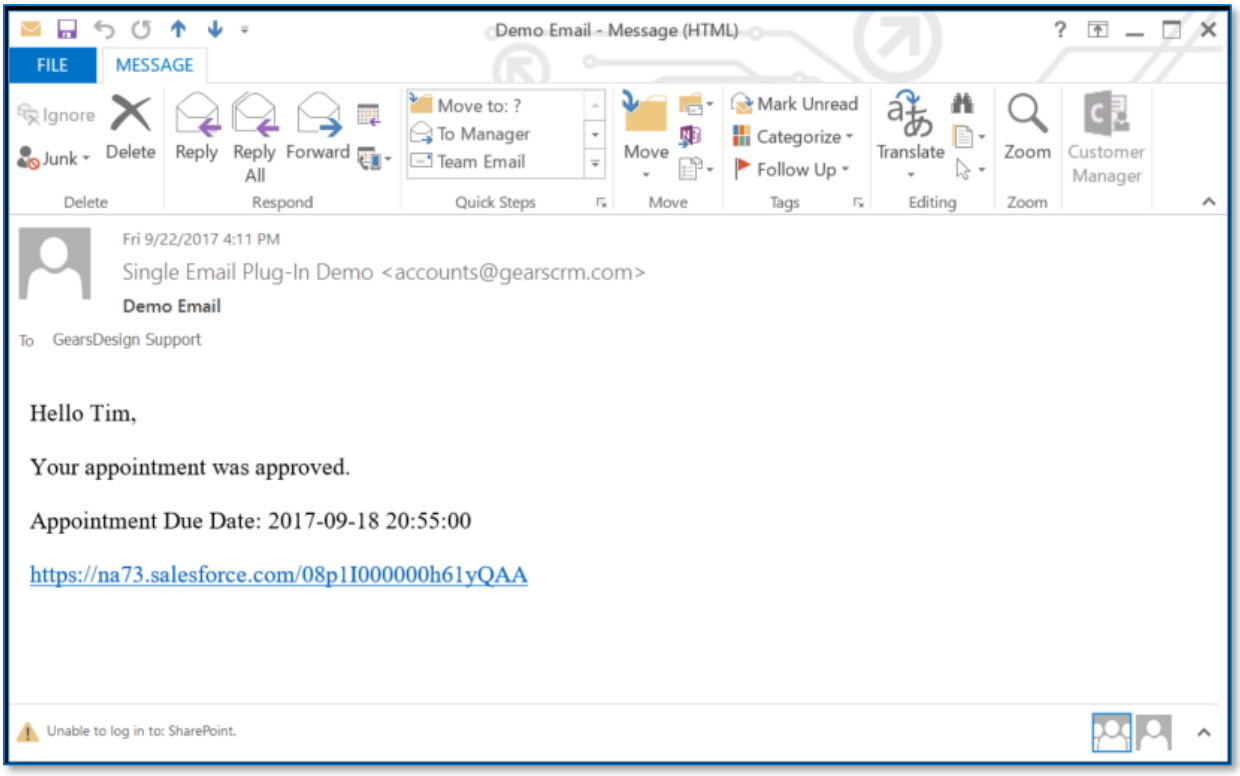
Action	HTML Tag	Example	Output
Line Break / Carriage Return	 	First Line Second Line	First Line Second Line
New Paragraph	<p>	…finally.<p>The End	…finally. The End
Bold		Bold	**Bold**
Italics		Italics	*Italics*
Underline	<u> </u>	<u> Underline </u>	Underline

Examples

Scenario 1:

*Hello {Contact.FirstName},
*
*
*
*Your appointment was approved.
*
*
*
*Appointment Due Date: {ServiceAppointment.DueDate}
*
*
*
https://{getSalesforceBaseURL}/{ServiceAppointment.Id}

Result:



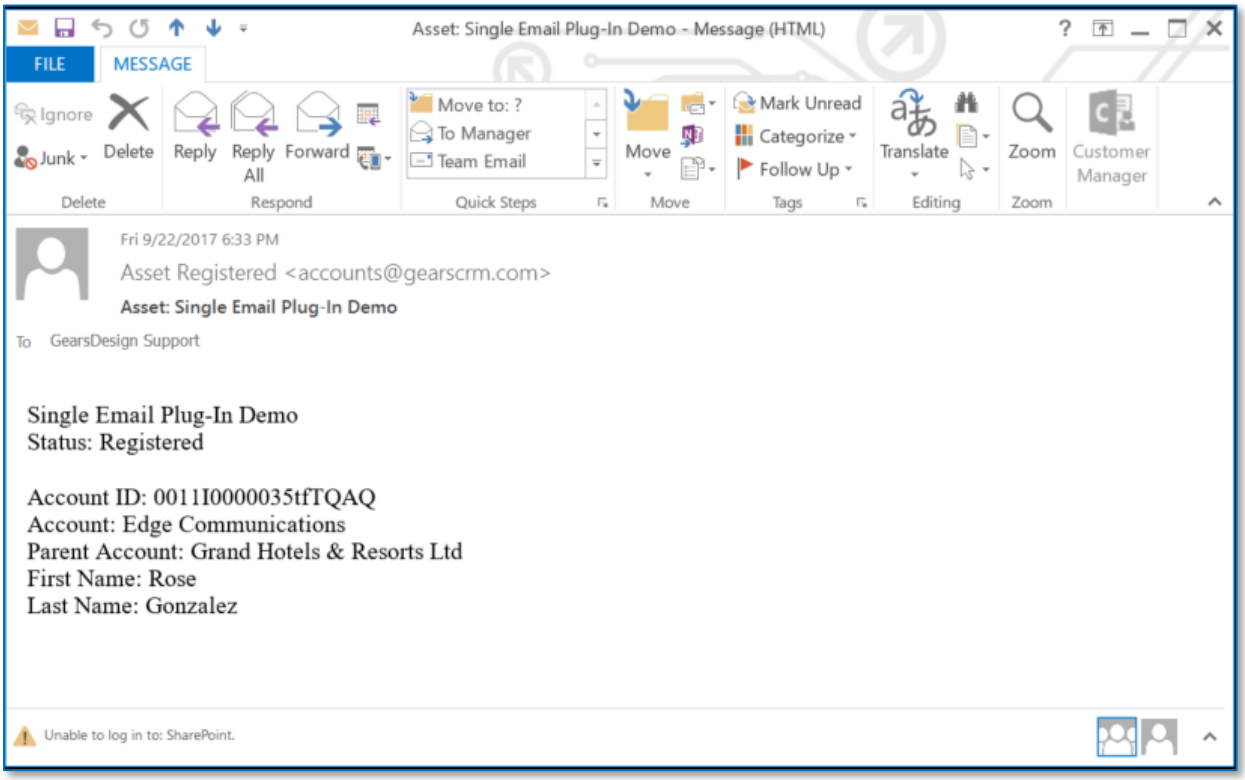
Scenario 2:

email Subject = *Asset: {Asset.Name}*

email Body =

*{Asset.Name}
*
Status: {Asset.Status} <p>
*Account ID: {Account.Id}
*
*Account: {Account.Name}
*
*Parent Account: {Account.Parent.Name}
*
*First Name: {Contact.FirstName}
*
Last Name: {Contact.LastName}

Result:



Salesforce Syntax - Advanced Functions

In some instances, the output format for specific field types need to be customized for your specific business use case. Specifically, the following data types have some additional functions you can apply to them which alters the appearance in your email message:

Date

When using a standard Date field via the Single Email Helper, the output is in a MM-DD-YYYY HH:MM:SS format – even though in the user interface, there is no time component. Additionally, for users in countries where the standard date format is DD-MM-YYYY, this output does not meet the desired outcome. As a result, the Single Email Helper app will allow you to pass variables as part of your field in order to tell the application how to correctly output your date value.

The variables for Date fields are as follows: {<Date Field>, <Data Type>, <Output Format>}. So for example, if you have a Date field on your Lead object called My_Date__c and the desired output in your email is DD-MM-YYYY format, then you'd input the following:

{Lead.My_Date__c, Date, dd/MM/YYYY}

As you can see, the first value is the date field in question, the second is the data type and the last is the desired format for your date value.

Date/Time

When using a standard Date/Time field via the Single Email Helper, the output is in a MM-DD-YYYY HH:MM:SS format. For users in countries where the standard date/time format is DD-MM-YYYY HH:MM:SS, this output does not meet the desired outcome. As a result, the Single Email Helper app will allow you to pass variables as part of your field in order to tell the application how to correctly output your date/time value.

The variables for Date/Time fields are as follows: {<Date Field>, <Data Type>, <Output Format>, <Time Zone>}. So for example, if you have a Date/Time field on your Lead object called My_DateTime__c, the desired output in your email is DD-MM-YYYY hh:mm:ss format and you want it in America/New York time zone, then you'd input the following:

{Lead.My_DateTime__c, DateTime, dd/MM/YYYY hh:mm:ss, America/New_York}

As you can see, the first value is the date field in question, the second is the data type, the third is the desired format for your date value and the last is the time zone in which the time value should be returned.

Important Note: If you do not want to specify the time zone and you'd rather use the running user's time zone, simply do not provide a fourth parameter value. Using the same example above, that would look like this:

{Lead.My_DateTime__c, DateTime, dd/MM/YYYY hh:mm:ss}

Percent

When using a standard Percent field via the Single Email Helper, the output is in a decimal format – for example, 25% would show as ".25". In order to correctly format this value in your email, the Single Email Helper app will allow you to pass variables as part of your field in order to tell the application how to correctly output your percent value.

The variables for Percent fields are as follows: {<Percent Field>, <Data Type>, <Number of Decimal Places>, <Percent to Multiply By>}. So for examples, if you have a Percent field on your Lead object called My_Percent__c which is currently set to 55.75% and you want that format to come through on your email, then you'd input the following:

{Lead.My_Percent__c, Percent, 2}

Inversely, if you wanted to round that same value to the nearest whole percent, then you'd input the following:

{Lead.My_Percent__c, Percent, 0}

As you can see, in both of these examples, the last parameter value was not passed. This is truly to be used in the case that you have a number field acting as a percentage. For example, you have a field called My_PercentNumber__c formatted as a Number(1, 2) so you have values like .25, .75, etc which are the equivalent of percentages. The fourth parameter can take that value and convert it into a true percent value. In this example, the My_PercentNumber__c field is set to ".45" and the desired output is "45". To do this, you'd pass the following:

{Lead.My_Percent__c, Percent, 0, 100}

The last parameter value takes the ".45" value and multiplies by 100 to get a final output of "45".