|  |  |  |  |
| --- | --- | --- | --- |
| **Change Date** | **SNOW Release** | **Nature of change** | **Changed By** |
| 09-11-2014 | Base Framework | Email Test Group | Barb Wagner |

|  |  |
| --- | --- |
| Update Set Name: | GEN-POC-01 |
| Account: | Misc |
| Business Requirement No | BUS600 |
| System Requirement No |  |
| Component Requirement No | CMP-6105 |
| Instance | https://projectwotsitpov.service-now.com/ |

**Reference Documentation**

* DCD 112 System Requirements Specification
* APP 402 Requirements Traceability Matrix

**Summary Requirement Description**

Modify the out of box email testing function to allow for an email testing group instead of the out of the box functionality of providing individual email addresses for testing – this provides the ability for this functionality to be maintained at the group level instead of in the email properties

**Usage:**

When in test mode – enable the business rule created in this document

When not in test mode – disable the business rule (then standard email functions will be followed)

**This solution requires 3 Configuration Changes to be completed**

1. Create a new group – Email Test Group (this is not part of the update set and must be manually created)
2. Create a new system property – glide.email.test.group.custom
3. Create a new business rule – Email Test Group

**Test Configurations**

**Setup or predecessor data:**

Test Case ID:

**Assumptions**

1. This is used for testing environments only
2. Email is enabled

**Detailed Design Steps**

| **Step #** | **Description** | **Expected Results** |
| --- | --- | --- |
| 1 | **Create a New Group** |  |
| 2 | Navigate to Groups by typing “groups” in the filter navigation |  |
| 3 | Select – User Administration - Groups |  |
| 4 | Select – “New”  Name: Email Test Group  Group email – leave blank  Parent – leave blank  Manager – leave blank  Description - Group to use for email testing prior to production - members in this group will receive email notifications generated during testing |  |
| 5 | Press “Save” |  |
| 6 | Add team members that will be testing and are expecting to receive the test email messages  Note: this will generate a lot of email to the group assignees so be sure they know this |  |
|  | **Create a new system property** |  |
| 1 | Navigate to System properties by typing sys\_properties.LIST in the filter navigation | System properties list will display |
| 2 | Select “new”  Name – glide.email.test.group.custom  Description - Only this groups members can receive emails. Used for non-production systems for testing purposes (Use the group's name)(clear the "Email address to which all emails will be sent." property).  Choices – leave blank  Type – String  Value – Email Test Group  Ignore cache – not selected  Private – not selected  Read roles – leave blank  Write roles – leave blank |  |
|  | Press “Save” |  |
|  |  |  |
|  | **Create a new business Rule** |  |
| 1 | Navigate to Business Rules by typing “Business” in the filter navigation |  |
| 2 | Select System Definitions – Business Rules |  |
| 3 |  |  |
| 4 | Create the New Business Rule with the following attributes (Note you may have to save it to see all the fields.)  **Header**  Name – Email Test Group  Table – Email [sys\_email]  Active – selected  Advanced – selected  **When to Run tab**  When – Before  Insert – selected  Update – selected  Delete – not selected  Query – not selected  Leave all other options blank  **Actions tab**  Leave all options blank  **Advanced Tab**  Condition - current.type.changesTo('send-ready') && gs.getProperty('glide.email.test.group.custom','') != ''  Script (cut and paste full information in next line) |  |
|  | function onBefore(current, previous) {  //This function will be automatically called when this rule is processed.  /\* --Framework update --  This Business Rule looks for the system property "glide.email.test.group.custom" to be populated and if true verifies the recipients  of all outbound emails are only those that exist in a matching group which by default is "Email Test Group".  If the User that created the record referenced on the email is in the group, they will be added to the recipients as well.  All original recipients are noted in the bottom of the email body.  The process flow is:  1. Get the current email recipients from the email record.  2. Get an array of all valid users from the email group.  3. Go through the current email recipients and remove any that are not in the email group.  4. Add the user that last updated the email's referenced record if they are in the email group.  5. Write the valid recipients to the email.  6. Write a comment in the body of the email at the bottom that includes a note and all of the original recipients email addresses.\*/    glideEmailTestJRE();    function glideEmailTestJRE(){  var table = current.target\_table;  var record = current.instance;  var rec = current.recipients;  var bod = current.body;  var group = gs.getProperty('glide.email.test.group.custom','');  var grpMembers = [];  var validEmails = [];  var email = '';  var lastUpdate = '';    //Find the sys\_id of the email group.  var grp = new GlideRecord('sys\_user\_group');  if(grp.get('name', group)) {  var grp\_sys\_id = grp.sys\_id; //Sets the groups sys\_id  } else {  current.type = 'send-ignored'; //We did not find a good match on the provided group so do not send emails.  gs.log('Email Test Group "' + group + '" provided but does not match a valid group. notification not sent. Please check the "Email Test Group" property under System Properties > Email.');  return;  }    //Look through all members of the email group.  var grpUsr = new GlideRecord('sys\_user\_grmember');  grpUsr.addQuery('group',grp.sys\_id);  grpUsr.query();  while(grpUsr.next()){  email = grpUsr.user.email.toString();  if(rec.indexOf(email) >=0){  validEmails.push(grpUsr.user.email.toString()); //Create an array of emails from the original recipients that are allowed to receive emails  }  grpMembers.push(grpUsr.user.email.toString()); //Create an array of all test group members  }    //Look to see who last updated the referenced record if there is one and there are no valid recipients.  if (record && !validEmails) {  var gr = new GlideRecord(table);  if(gr.get(record)){  lastUpdate = gr.sys\_updated\_by;  }  }    //Check to see if the last updated user is in the current validEmails array and add them if they are not and valid.  //Only do this if there are no valid recipients.  if(lastUpdate){  var usr = new GlideRecord('sys\_user');  usr.addQuery('user\_name',lastUpdate);  usr.addQuery('notification',2); //Check to see if the user has notification set to "Email" which is 2.  usr.query();  if(usr.next()){  if(grpMembers.toString().indexOf(usr.email) >= 0){ //Are they a valid user.  if(validEmails.toString().indexOf(usr.email) == -1){ //Are they already in the validEmails array.  validEmails.push(usr.email.toString());  }  }  }  }  //If the current recipients field is blank then set the text to NONE.  if(rec == '')  rec = 'NONE';  current.body = bod + '<div><br/><br/><hr/></div><div>Testing mode on via system property "glide.email.test.group.custom". Intended recipients: ' + rec + '</div>';  //If there are no intended recipients then set the email to send-ignored and clear the recipients field.  if(validEmails.length == 0){  current.type = 'send-ignored';  current.recipients = '';  }else{  current.recipients = validEmails.toString();  }  }  } |  |
|  | Press “Save” |  |
|  | Testing the change |  |
| 1 | Create a new Incident |  |
| 2 | Review results in the System email log to ensure that   1. email is sent to individuals included in Email Test Group 2. each email body has the following information on the bottom of it: “Testing mode on via system property "glide.email.test.group.custom". Intended recipients:xxxx@xx.com |  |
| 3 |  |  |

**Tester Details**

|  |  |
| --- | --- |
| Designer Name: |  |
| Completion Date: |  |
| Tester Name: |  |
| Test Date: |  |
| Test Pass / Fail | Reason: |
| Reviewer Name | If needed |
| Reviewer Comments |  |