



SSF Tools: Enhanced Email Notifier User Guide

Document Revision History

Revision Date	Written/Edited By	Comments
Feb 26 2017	Christian Cairney	Initial release with SSD v3

© Copyright 2017 SailPoint Technologies, Inc., All Rights Reserved.

SailPoint Technologies, Inc. makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. SailPoint Technologies shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Restricted Rights Legend. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of SailPoint Technologies. The information contained in this document is subject to change without notice.

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Regulatory/Export Compliance. The export and reexport of this software is controlled for export purposes by the U.S. Government. By accepting this software and/or documentation, licensee agrees to comply with all U.S. and foreign export laws and regulations as they relate to software and related documentation. Licensee will not export or reexport outside the United States software or documentation, whether directly or indirectly, to any Prohibited Party and will not cause, approve or otherwise intentionally facilitate others in so doing. A Prohibited Party includes: a party in a U.S. embargoed country or country the United States has named as a supporter of international terrorism; a party involved in proliferation; a party identified by the U.S. Government as a Denied Party; a party named on the U.S. Government's Entities List; a party prohibited from participation in export or reexport transactions by a U.S. Government General Order; a party listed by the U.S. Government's Office of Foreign Assets Control as ineligible to participate in transactions subject to U.S. jurisdiction; or any party that licensee knows or has reason to know has violated or plans to violate U.S. or foreign export laws or regulations. Licensee shall ensure that each of its software users complies with U.S. and foreign export laws and regulations as they relate to software and related documentation.

Trademark Notices. Copyright © 2017 SailPoint Technologies, Inc. All rights reserved. SailPoint, the SailPoint logo, SailPoint IdentityIQ, and SailPoint Identity Analyzer are trademarks of SailPoint Technologies, Inc. and may not be used without the prior express written permission of SailPoint Technologies, Inc. All other trademarks shown herein are owned by the respective companies or persons indicated.

Table of Contents

Overview.....	4
Notifier Features	4
Installation	5
Enhanced Email Notifier Data Flow	6
Dynamic Template Naming Overview.....	7
Summary Email Processing Overview.....	7
Configuration	8
System Settings.....	8
Notifier Settings	8
Dynamic Template Naming.....	10
Dynamic Email Rule	10
Summary Emails	11
The Summary Template Map.....	11
Work Item Assignment.....	13
The Summary Rule.....	14
Enhanced Notifier Summary Processor Task.....	16
Advanced Summary Velocity Rendering	17

Overview

The Enhanced Email Notifier tool adds functionality to the standard email notification functionality found in IdentityIQ.

Notifier Features

The additional features of this tool are:

- Dynamically change the email template before an email gets sent
- Process emails and gather summary information to be sent later

Installation

The Enhanced Email Notification tools consists of the following files:

Java classes:

Filename	Description
EnhancedNotifier.class	Enhanced “Email Notifier” class which extends IdentityIQ functionality
EnhancedSummaryProcessor.class	AbstractTaskDefinition which processes summary information if it exists
VelocityEvaluator.class	A Velocity tool which allows for dynamic parsing of Velocity Templates inside Velocity. SailPoint IdentityIQ version of Velocity does not have this feature (Velocity v1.5)

Configuration objects:

Filename	Description
Configuration-SystemConfiguration-EnableEnhancedEmailNotification.xml	System Configuration merge file to enable the Enhanced Email Notifier
Configuration-EnhancedEmailNotifier_Example.xml	Example of a simple Enhanced Notification Config. Although this is an example it is deployed by default and is intended to be modified to meet requirements.
Configuration-EnhancedEmailNotifier_Example_Enhanced.xml	Example of a complex Enhanced Notification Config. This example is stored in a ‘Samples’ folder which means the SSD will not deploy it by default.

EmailTemplate object:

Filename	Description
EmailTemplate-SummaryEmail.xml	Basic Email Template for the Enhanced Notification class

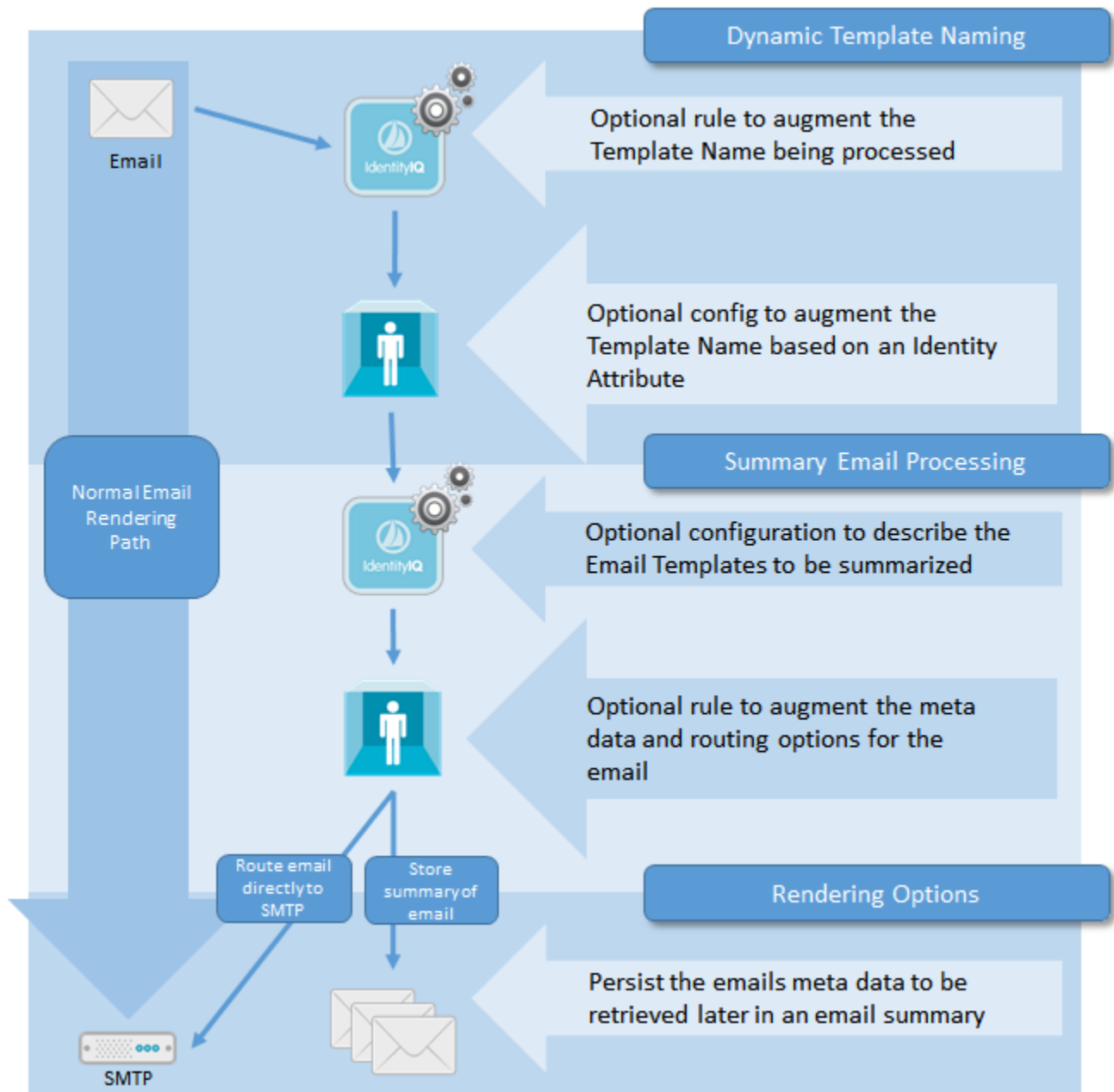
TaskDefinition object:

Filename	Description
TaskDefinition-EnhancedNotifierSummaryProcessorTask.xml	TaskDefinition template for running the Enhanced Summary Processor task

The Enhanced Email Notifier is included in the Services Standard Deployment (SSD) but it is not automatically deployed by default. Deployment can be switched on or off by use of the “deployEnhancedEmailNotifier” property in the build.properties file. Set this to “true” to enable deployment, or “false” to disable. Alternatively, the SSD Deployer tool can be used to enable or disable deployment (see the SSD Deployer User Guide). The default setting prevents the configuration files being imported, although the Java classes will still be added to the resulting build.

Enhanced Email Notifier Data Flow

The Notifier class has a specific flow of data; the diagram below describes these processes:



Dynamic Template Naming Overview

- The Notifier can be configured to execute a rule which returns the email template name used to render the email.

Use case example: Internal and External staff are to receive different notifications on an Access Request.

- The Notifier allows an identity attribute to be used to augment the template name, by appending “_” + attribute value to the end of the template name.

Use case example: Language settings are persisted in an Identity country code attribute; this can be appended to an email template name to allow for multi-language emails

Summary Email Processing Overview

- The Notifier can be configured to list all the Email Notification Template names which need to be summarized.
- The Notifier class will attempt to discover the basic rendering solution for the summary email if none is configured.
- An optional rule can be used to augment the meta data from the original Email Notification options passed to the email handler by IdentityIQ. This allows additional meta data to be present when the email summary is rendered.
- Finally, the summary email meta data is rendered using a Velocity email template.

Configuration

System Settings

To enable this Notifier, the class needs to be registered in the System Configuration object. When you deploy the Notifier using the SSD this will be done for you. A merge file is provided to automate this:

```
Configuration-SystemConfiguration-EnableEnhancedEmailNotification.xml
```

But this can also be done manually by adding the follow key to the SystemConfiguration Configuration Object.

```
<entry key="emailNotifierClass"
value="sailpoint.services.standard.emailnotifier.enhanced.EnhancedNotifier"/>
```

Notifier Settings

The Notifier has its own Configuration Object, named “Enhanced Email Notifier”. This needs to be properly implemented before this class is enabled.

The following keys are used in this Configuration Object.

Configuration Key	Type	Description
serverRootPath	String	OPTIONAL: Holds the serverRootPath of the server, so deep links can be enabled if required. This is exposed in the Velocity template as \$serverRootPath. NB: In IdentityIQ 7.0 and later the value of this is ignored, and the serverRootPath from the SystemConfiguration object is used instead.
dynamicEmailRuleName	String	OPTIONAL: The Rule that can return a new template name instead of the one used in the original email
dynamicEmailIdentityAttribute	String	OPTIONAL: The Identity attribute used to augment the template name being used for the email.
summaryTemplateMap	Map	OPTIONAL: A list of email templates which will be targeted for Summary Processing. This map can contain additional rendering information to allow the email to be auto generated without requiring additional development of a Velocity Template.
summaryRuleName	String	OPTIONAL: The Rule which can augment the summary information which is persisted, so meta data can be added or removed if required
summaryEmailTemplateName	String	Velocity Email Template name to render the

		summary results. This is normally set to “Summary Email” but if more advanced summary rendering is required, a bespoke Template can be selected.
--	--	--

Dynamic Template Naming

Dynamic Template Naming is a function of this Notifier class which can change the Email Template used to render the Email with another.

Two basic processes are available to provide this functionality:

1. Dynamic Email Rule Name
2. Dynamic Email Identity Attribute

Of these processes, one or both can be enabled so they augment each other. Each of these processes are executed in the following Order:

1. "Dynamic Email Rule Name"
 - a. The rule used to change the email notification template name.
 - b. If the template name is changed, then that is used for any future processing
 2. "Dynamic Email Identity Attribute"
 - a. Uses an Identity Attribute value to append on the end of the name of the Template Name.
- e.g.
- Template name from original email = "Work Item Assignment"
 - The Dynamic Email Identity Attribute is configured to the "language" Identity Attribute
 - The Identity the email is being sent for has a language attribute value of "GB"
 - The new template name to render the email is now "Work Item Assgnment_GB"
 - If the new template name is not found, it will revert to the original template name

Dynamic Email Rule

The Dynamic Email Rule has the following signature.

Variable	Type	Description
emailTemplate	sailpoint.object.EmailTemplate	The original email template selected to render the email.
emailOptions	sailpoint.object.EmailOptions	The meta data used to render the email
context	SailPointContext	The SailPoint Context
emailAddress	String	Email address
identity	sailpoint.object.Identity	The Identity who matches the email address

The return value must be a String, with a value of the new EmailTemplate name. If the template name does not exist, then the original Email Template will continue to be used.

Summary Emails

Summary Emails work by collecting the Email meta data stored in the Email Options and rendering them using an “Enhanced Notifier Summary Processor Task”.

To enable this feature the summary email template must be configured in the Notifier’s Configuration object:

```
<entry key="summaryEmailTemplateName" value="Summary Email"/>
```

Please note: The default “Summary Email” Email Template also has provision to dynamically render chosen email templates with minimum configuration. If the implementer wishes to enhance this template, there are recommendations in a later section on how to achieve this.

Additionally, a list of templates required to be summarized should be listed in the summaryTemplateMap key. Below is an example of two EmailTemplate names which will be summary processed: the “Work Item Assignment” and “LCM Identity Update” templates:

```
<entry key="summaryTemplateMap">
  <value>
    <Map>
      <entry key="Work Item Assignment"/>
      <entry key="LCM Identity Update"/>
    </Map>
  </value>
</entry>
```

Please note: The list of templates to be processed in the summaryTemplateMap is the name of the original email template sent to the Notifier *or* the name of the template returned in the Dynamic Email Rule.

The Summary Template Map

In the above example, summaryTemplateMap entries do not have any values; the value of these entries is a Map which contains hints on how the Summary Processor renders the emails. If no value exists for the summaryTemplateMap entry, then the EmailTemplateNotifier will create basic rendering hints on how to render them and store them back in this configuration object when the first email of that type has been sent.

After a “Work Item Assignment” email has been sent from IdentityIQ, the configuration will now contain:

```
<entry key="summaryTemplateMap">
  <value>
    <Map>
      <entry key="LCM Identity Update"/>
      <entry key="Work Item Assignment">
        <value>
          <Map>
            <entry key="columns">
              <value>
                <List>
                  <Map>
                    <entry key="label" value="WorkItem: item name"/>
                    <entry key="name" value="item.name"/>
                    <entry key="value" value="$item.name"/>
                  </Map>
                </List>
              </value>
            </entry>
          </Map>
        </value>
      </entry>
    </Map>
  </value>
</entry>
```

```

</Map>
<Map>
  <entry key="label" value="WorkItem: item description"/>
  <entry key="name" value="item.description"/>
  <entry key="value" value="$item.description"/>
</Map>
<Map>
  <entry key="label" value="WorkItem: item id"/>
  <entry key="name" value="item.id"/>
  <entry key="value" value="$item.id"/>
</Map>
<Map>
  <entry key="label" value="requesterName"/>
  <entry key="name" value="requesterName"/>
  <entry key="value" value="$requesterName"/>
</Map>
<Map>
  <entry key="label" value="workItemName"/>
  <entry key="name" value="workItemName"/>
  <entry key="value" value="$workItemName"/>
</Map>
</List>
</value>
</entry>
<entry key="title" value="Work Item Assignment"/>
</Map>
</value>
</entry>
</Map>
</value>
</entry>

```

The Configuration Object has now been changed to give a basic list of columns a title. The map now contains the following keys:

Entry	Type	Description
columns	List	Contains a list of the columns of data to be rendered in an email. Each column is a Map.
title	String	The title of the section to be rendered in an email

The “columns” entry is a List object which holds maps of each column to be rendered in an email. Each Column Map has the following entries:

Entry	Type	Description
label	String	Header for the column
name	String	Name of the item
value	String	Velocity template to render the value

The “value” is an actual Velocity Template, which will be executed on the fly for each email to resolve the data. This will be discussed in more detail later in this section.

When the email gets rendered it will look like the following:

John Smith,

Please find a summary communications sent to you since 25-Feb-2017 23:33

Work Item Assignment

WorkItem: item name	WorkItem: item description	WorkItem: item id	requesterName	workItemName
0000000298	Confirm User Name	40280c815a5f7e6a015a6101f3a20079	John Smith	0000000298
0000000298	Confirm Access Request	40280c815a5f7e6a015a6101f3a20079	John Smith	0000000298

We can tidy this email up by removing some unwanted detail and additional formatting. Altering the Summary Template Map's render items for the "Work Item Assignment" render hint for this template:

```
<entry key="Work Item Assignment">
  <value>
    <Map>
      <entry key="columns">
        <value>
          <List>
            <Map>
              <entry key="label" value="Date created"/>
              <entry key="name" value="dateCreated"/>
              <entry key="value" value="$spTools.formatDate($createdDate, 3, 3)"/>
            </Map>
            <Map>
              <entry key="label" value="Item ID"/>
              <entry key="name" value="itemId"/>
              <entry key="value" value="$item.name"/>
            </Map>
            <Map>
              <entry key="label" value="Description"/>
              <entry key="name" value="description"/>
              <entry key="value">
                <value>
                  <String><![CDATA[
<a
href="{serverRootPath}/workitem/commonWorkItem.jsf#/commonWorkItem/${item.id}">${item
.description}</a>
]]></String>
                </value>
              </entry>
            </Map>
          </List>
        </value>
      </entry>
      <entry key="title" value="Work Items which have been assigned"/>
    </Map>
  </value>
</entry>
```

The above render hint will create an email shown in the next caption. The things we have changed in this template are:

- Added a Date Created item and using the SailPoint spTools formatted the date.
- Changed the description so now it will also act like a deep link into IdentityIQ, going straight to the Workitem.
- Removed the workitem ID, Requester Name from the template.

John Smith,

Please find a summary communications sent to you since 25-Feb-2017 23:33

Work Items which have been assigned

Date created	Item ID	Description
25/02/17 23:33	0000000298	Confirm User Name
25/02/17 23:43	0000000298	Confirm Access Request

The Summary Rule

The Summary Rule when specified in the Notifier Class configuration item “summaryRuleName”, will execute a rule before the Summary meta data is to be persisted for later re-call by the Summary Processor. The Summary Rule can add additional meta data which can be rendered by the Summary Processor.

The input signature of this rule is as follows:

Variable	Type	Description
context	SailPointContext	The SailPoint Context
emailAddress	String	Email address
identity	sailpoint.object.Identity	The Identity who matches the email address
summaryMap	Map	A Map of values which can be persisted from the Email Options object.
inSummaryMap	Boolean	Email exists in the Summary Template Map.

The rule can return different results depending on how the implementer wants the rule to react.

- Return NULL
The Summary Processing will abort and not persist the email; the email will be sent to the STMP server instead.
- Return the summaryMap after processing, this will indicate that Summary Processing is to continue.
- If “sendImmediately” is added to the summaryMap, then the email will be sent to the STMP Server AND be processed by the Summary Processor.

Enhanced Notifier Summary Processor Task

The Enhanced Notifier Summary Processor Task will process existing summary emails and clear them down after the emails have been rendered.

The Task is expected to be run once per day, when the summary emails are expected to be delivered.

The Task has one option, to run in Debug mode. This mode allows you to test whether the summary emails render as expected. Although it will still be rendered out to the Email Settings on IdentityIQ, the Summary queues will not be deleted until the Debug mode is turned off.

Advanced Summary Velocity Rendering

The supplied Summary Email Template can be modified for bespoke rendering of each email template by modifying the Velocity template.

The Template has been designed to changed; an excerpt of the template is below:

```
#if ( $summaryKey.equals('PUT YOUR EMAIL TEMPLATE HERE') )

    <p>
        <h3>$summaryKey</h3>
        #set ($items = $summeries.get($summaryKey))
        <table><tr><th>Date/Time created</th></tr>
        #foreach ($itemMap in $items)

            ## Process the item map ... e.g.:
            <tr><td>$spTools.formatDate($itemMap.createdDate, 3, 3)</td></tr>

        #end
    </table>
</p>
## To process additional templates, then add
##elseif ( $summaryKey.equals('PUT YOUR EMAIL TEMPLATE HERE')) etc.
##
## do not change anything below this line
##

#else
##
## Auto render the email, as the template renderer was not found above
##

    #autoRender( $renderHints.get($summaryKey) )

#end
```

At the code section:

```
#if ( $summaryKey.equals('PUT YOUR EMAIL TEMPLATE HERE') )"
```

Add the name of the email template you wish to use to process a given Email Notification Type. There is supplied code to show how this can be achieved. Do not change anything below the line:

```
## do not change anything below this line
```