Emails Using Notification Templates and Communication Manager

Jason Plaisted

Senior Customer Success Manager

3/10/2016

Accela

Agenda

- Demonstrations
  - Email with Report Generated from Inspection Result
  - Email Generated form Workflow Task Update
- How to develop a Custom Functions for Sending Notifications with Embedded Reports (yes, even Ad Hoc Reports!)
- Setting up Communication Manager
- How Configure Notification Templates
- How to setup Template Variables from Records, Inspections, etc.
- How to configure the Communications Tab on the Record to show the generated emails

Custom Send Notification Functions

Variables Needed

Variable	Example Value	Notes
acaURL	aca.demo.accela.com/csm	INCLUDES_CUSTOM_GLOBALS
agencyReplyEmail	noreply@accela.com	INCLUDES_CUSTOM_GLOBALS
contactTypesArray	"Applicant", "Property Owner", "Business Owner"	Any contact type that you want the notification to be sent
notificationTemplate	MESSAGE_PERMIT_ISSUED_REPORT	V360 Admin > Communication Manager > Notification Templates
reportName	Building Permit	V360 Admin > Report Manager
rptParams	rptParams.put("p1Value", capIDString);	For Ad Hoc use p1Value, p2Value etc.

Add Custom Script Call to Standard Choice or 3.0 Event Script

Example –

WTUA:Permits/*/*/*

```
matches(wfTask,"Permit Status","Permit Issuance") &&
matches(wfStatus, "Issued", "Permit Issued") ^
sendPermitIssuedReportNotification();
```

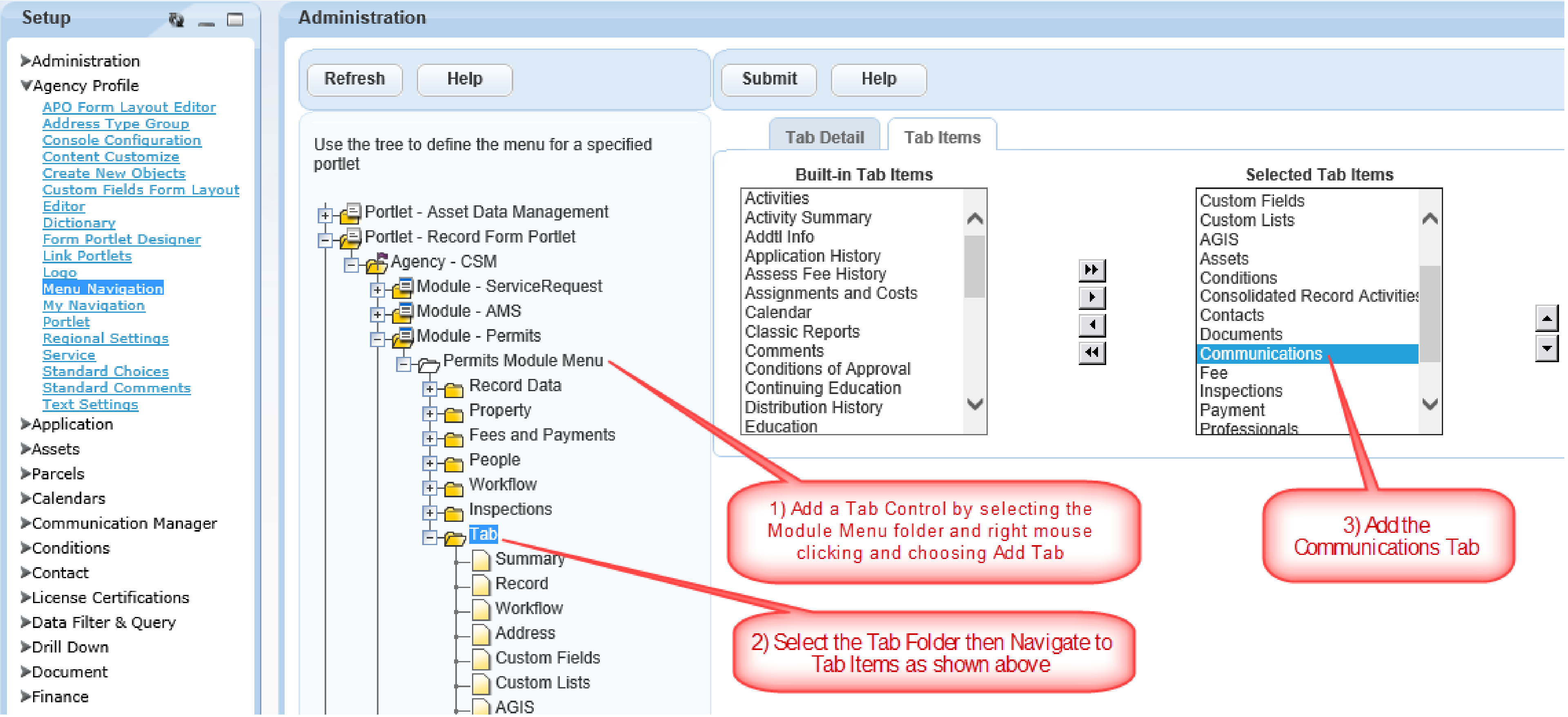
Enabling Communication Manager

8.0 Communication Manager

Feature	FID
Account Settings	0360-Account Settings
Alert	0151-Admin Alert and Rules
Alert Rule	0151-Admin Alert and Rules
Announcements	0134-Admin-Message-General 0135-Admin-Message-Supervisor
General Settings	0361-General Settings
Notification Templates	0184-Admin Content Customization
Communication List	8492-Communication List
Record Communication List	8494-Record Communication List

Configure Communications Tab

Agency Profile > Menu Navigation > Portlet – Record Form Portlet



Use the tree to define the menu for a specified portlet

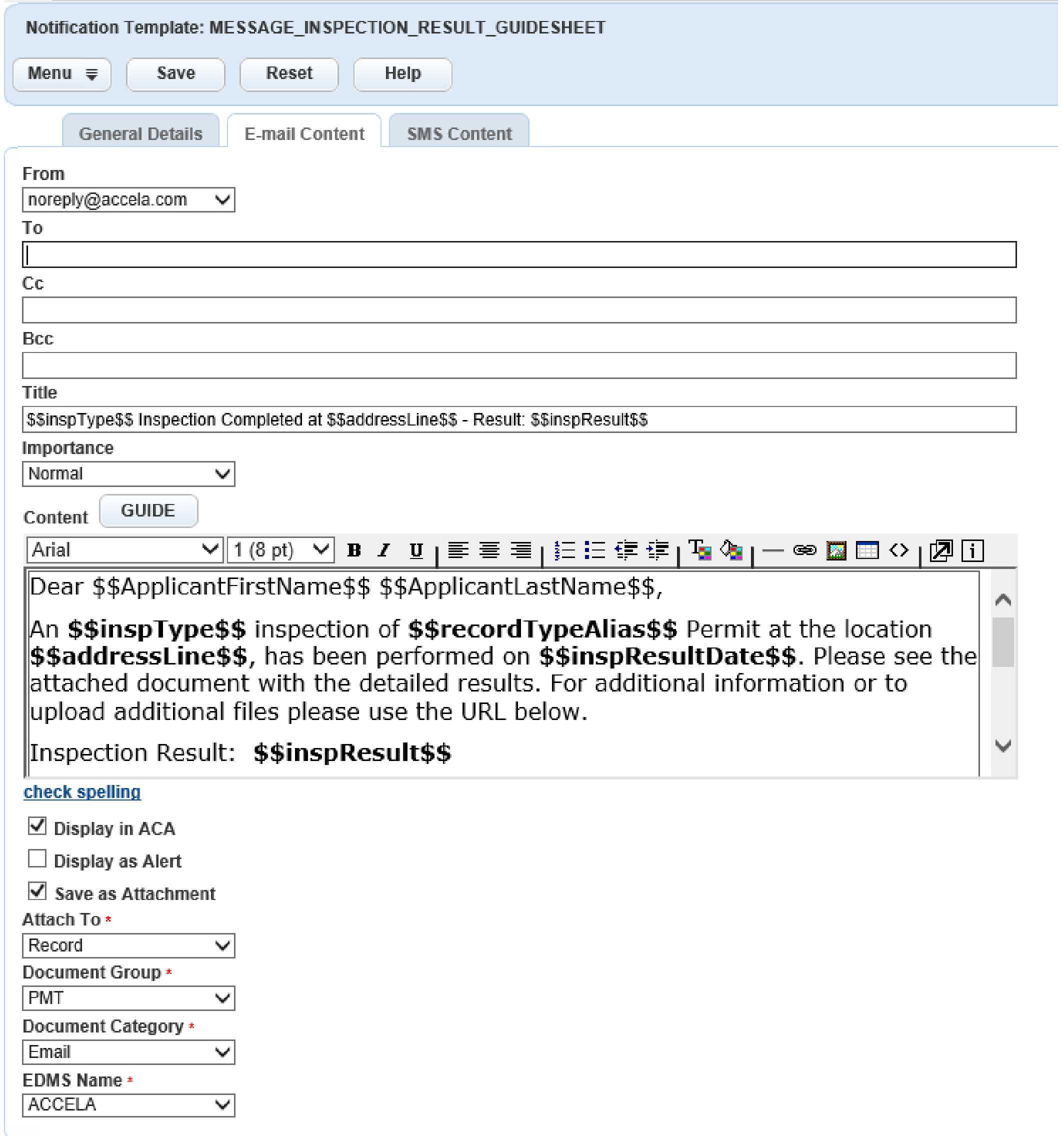
1) Add a Tab Control by selecting the Module Menu folder and right mouse clicking and choosing Add Tab

2) Select the Tab Folder then Navigate to Tab Items as shown above

3) Add the Communications Tab

Configure Notification Templates

Administration > Communication Manager > Notification Templates



Notification Template: MESSAGE_INSPECTION_RESULT_GUIDESHEET

From: noreply@accela.com

To:

Cc:

Bcc:

Title: $$inspType$$ Inspection Completed at $$addressLine$$ - Result: $$inspResult$$

Importance: Normal

Content

Arial | 1 (8 pt)

Dear $$ApplicantFirstName$$ $$ApplicantLastName$$,

An **$$inspType$$** inspection of **$$recordTypeAlias$$** Permit at the location **$$addressLine$$**, has been performed on **$$inspResultDate$$**. Please see the attached document with the detailed results. For additional information or to upload additional files please use the URL below.

Inspection Result: **$$inspResult$$**

check spelling

☑ Display in ACA

□ Display as Alert

☑ Save as Attachment

Attach To *: Record

Document Group *: PMT

Document Category *: Email

EDMS Name *: ACCELA

Notification Template Parameter Functions

Included in Master Scripts 3.0 or EMSE Notification Template Functions Article

- getRecordParams4Notification(param)
  - $$altID$$
  - $$capName$$
  - $$capStatus$$
  - $$fileDate$$
  - $$workDesc$$
  - $$balanceDue$$
  - $$capTypeAlias$$
- getPrimaryAddressLineParam4Notification(params)
  - $$addressLine$$
- getPrimaryOwnerParams4Notification(params)
  - $$ownerFullName$$
  - $$ownerPhone$$
- getACARecordParam4Notification(params ,acaURL) –Deep link to ACA Record
  - $$acaRecordUrl$$
- getACADocDownloadParam4Notification(params, acaUrl, docModel)
  - $$acaDocDownloadUrl$$

Notification Template Parameter Functions

Included in Master Scripts 3.0 or EMSE Notification Template Functions Article

- “Applicant” Contact Type would return $$ApplicantFullName$$
- getContactParams4Notification(paramsArray, conType)

```
"$$" + conType + "LastName$$“
"$$" + conType + "FirstName$$“
"$$" + conType + "MiddleName$$“
"$$" + conType + “FullName$$"
"$$" + conType + "BusinesName$$"
"$$" + conType + "ContactSeqNumber$$"
"$$ + conType + "$$"
"$$" + conType + "Relation$$"
"$$" + conType + "Phone1$$“
"$$" + conType + "Phone2$$“
"$$" + conType + "Email$$“
```

```
"$$" + conType + "AddressLine1$$“
"$$" + conType + "AddressLine2$$“
"$$" + conType + "City$$“
"$$" + conType + “State$$“
"$$" + conType + “Zip$$"
"$$" + conType + “Fax$$“
"$$" + conType + “Notes$$“
"$$" + conType + “Country$$“
```

Notification Template Parameter Functions

Included in Master Scripts 3.0 or EMSE Notification Template Functions Article

- “Electrician” License Type would return $$ElectricianFullName$$
- licProfObj. getEmailTemplateParams(paramsArray);

```
"$$" + licType + "LastName$$“
"$$" + licType + "FirstName$$“
"$$" + licType + "MiddleName$$“
"$$" + licType + “FullName$$"
"$$" + licType + "BusinesName$$“
"$$" + licType + "BusinesName2$$“
"$$" + licType + "BusinesLicense$$“
"$$" + licType + "LicSeqNbr$$"
"$$ + licType + "$$“
"$$" + licType + "LicenseState$$"
"$$" + licType + "LicenseExpirationDate$$“
"$$" + licType + "LicenseInsuranceExpDate$$"
"$$" + licType + "LicenseIssueDate$$"
```

```
"$$" + licType + "Phone1$$“
"$$" + licType + "Phone2$$“
"$$" + licType + "Phone3$$“
"$$" + licType + "Email$$“
"$$" + licType + "AddressLine1$$“
"$$" + licType + "AddressLine2$$“
"$$" + licType + "City$$“
"$$" + licType + “State$$“
"$$" + licType + “Zip$$"
"$$" + licType + “Fax$$“
"$$" + licType + “Notes$$“
"$$" + licType + “Country$$“
```

Notification Template Parameter Functions

Event Specific Functions – New!

- WorkflowTaskUpdateAfter
  - getWorkflowParams4Notification(paramsArray)
    - $$wfTask$$
    - $$wfStatus$$
    - $$wfDate$$
    - $$wfComment$$
    - $$wfStaffUserID$$
    - $$wfHours$$
- InspectionScheduleAfter
  - getInspectionScheduleParams4Notification(params)
    - $$inspId$$
    - $$inspInspector$$
    - $$InspectorFirstName$$
    - $$InspectorMiddleName$$
    - $$InspectorLastName$$
    - $$inspGroup$$
    - $$inspType$$
    - $$inspSchedDate$$

Notification Template Parameter Functions

Event Specific Functions – New!

- InspectionResultSubmitAfter
  - getInspectionResultParams4Notification(params)
    - $$inspId$$
    - $$inspResult$$
    - $$inspComment$$
    - $$inspResultDate$$
    - $$inspGroup$$
    - $$inspType$$
    - $$inspSchedDate$$

- You can develop your own by looking at the Event Master Script Variables, INCLUDES_ACCELA_GLOBALS, and adding your own INCLUDES_CUSTOM_GLOBALS

Notification Template Functions

Generating Reports & Sending Notifications

runReport4Email(itemCap, reportName, conObj, rParams, eParams, emailTemplate, module, mailFrom)
Description: Runs a report and sends it in an email using a Notification Template

runReport4EmailOrPrint(itemCap, reportName, conObj, rParams, eParams, emailTemplate, module)
Description: Runs a report and sends it in an email and displays the report on screen

runReportAttach(itemCapId, aaReportName)
Description: Runs a report and attached it to the Documents tab on a record
NOTE: See Saving Reports using Scripting

sendNotification(emailFrom, emailTo, emailCC, templateName, params, reportFile)
Description: Sends a email using a Notification Template

generateReport(itemCap, reportName, module, parameters)
Description: Generates and returns the report file which can be attached to an email

Example Functions

Generating Reports & Sending Notifications

sendInspectionResultGuidesheetReport()

Description: Runs an inspection detail report and sends it in an email to all contacts on the record using a Notification Template

sendPermitIssuedReportNotification()

Description: When a permit is issued, this function generates the permit, attaches it to the record, and sends it in an email to all contacts on the record using a Notification Template.

sendPermitIssuedReportWithDocList()

Description: When a permit is issued, this function generates the permit, attaches it to the record, generates a list of documents and hyperlinks, and sends it in an email to all contacts on the record using a Notification Template.

sendPermitReadyToIssueNotification()

Description: When a permit is ready to be issued, this function sends it in an email to all specified Contacts, and Licensed Professionals on the record using a Notification Template.

Resources – Knowledge Articles

- 8.0 Communication Manager - How to enable and configure Communication Manager
- Accela Civic Platform 8.0.1 Configuration Guide – Communication Manager Section
- Communication Manager - Auto Attach Email to a Record or Auto Reply to Sender
- Notification Template EMSE Functions
- On Demand Webinar: Upgrading to Master Scripts 3.0 and the EMSE Tool

Saving Reports using Scripting

Enhancing Security Control on EMSE Methods (14ACC-10181)

Affects 7.3.3 Service Pack 1 (11/06/2014) or higher

IMPORTANT Notice: This feature impacts the behavior of the existing EMSE scripts that call to the writeToFile, deleteFile, and/or newInstance methods. If your agency uses such scripts, make sure to perform the configurations described in Administrator Configuration Details.

This release provides the following security control enhancements related with EMSE methods:

- Restricting the folder that the writeToFile and deleteFile methods can access to on the server.
- Enforcing a class white list. The newInstance method can only create instances of the classes that belong to the class white list.

Administrator Configuration Details

To implement the security control, you need to perform the following steps:

1. On the application server machine, navigate to the \av.biz\conf\av\ path, and open the ServerConfig.properties file.

2. Add the following code to the ServerConfig.properties file:

```
av.emse.file.path= {Specify a file path in the application server}
```

`av.emse.file.path` defines the file path on the server that the writeToFile method adds files to or the deleteFile method deletes files from.

For example, if you want to set the file path as D:\emse\temp, add the following code:

```
# UtilScript file path
av.emse.file.path=D:\\emse\\temp
```

Note: Even though the script that calls the writeToFile or deleteFile method may contain the file path, Accela Automation ignores the path in the script, and always uses the path defined in ServerConfig.properties.

Note: If you do not specify av.emse.file.path in the ServerConfig.properties file, the script that calls the writeToFile or deleteFile method now throws exception.

Saving Reports using Scripting

3. Create a file named “EMSESecurityConfig.properties” under \av.biz\conf\av\. Note that the EMSESecurityConfig.properties file name is case-sensitive.

4. Add the following code in the EMSESecurityConfig.properties file:

```
emse.class.whitelist={class #1, class#2,...}
```

`emse.class.whitelist` defines the classes that can be instantiated through the newInstance method.

For example, if you add the following code in the file, the newInstance method can create

com.accela.* and java.util.* instances:

```
emse.class.whitelist=com.accela.*, java.util.*
```

> **Note:** *If the EMSESecurityConfig.properties file does not exist in \av.biz\conf\av\, the newInstance method can only be used for creating “com.accela.*” instances.*
>
> **Note:** *If EMSESecurityConfig.properties file exists in \av.biz\conf\av\,the newInstance method can only be used for creating instances of the classes that belong to the class white list in the file. Note that if the file does not have any class in the white list, the newInstance method cannot create any instances, not even “com.accela.*” instances.*

5. Restart the server services for the changes to take effect.

Questions?

Accela