Working with the Notifications Framework

Introduction

The Notifications Framework provides a consistent, extensible communication mechanism that Campus Solutions product areas ("consumers") can use to enable communications from their areas. Customers can also use the framework for their own needs.

Notification support is a requirement of many existing and new Campus Solutions features including Delegated Access and other initiatives (referred to as Notifications Framework Consumer applications). Notifications encompass a broad area of communications between the Campus Solutions system and users and also between users themselves. A notification may take these forms (referred to as "channels"):

1. Email
2. SMS
3. Alerts (for example, Informational message that appears on a portal homepage)
4. Worklist Items (an actionable hyperlink that appears on a portal homepage)

The first iteration of the framework was the email channel, released to support Delegate Access/New User Registration in October 2012 (bundle#27). In the April 2013 bundle (bundle#29), the whole framework has been significantly enhanced to extend the email channel and add new channels for SMS, alerts, and worklist items.

The Notifications framework:

1. Enables the new developments in Campus Solutions, such as Campus Access (mobile) and Evaluations, to easily include notifications, plus
2. You can craft your own solutions to use the framework without needing to reinvent the lower level coding that enables emails, alerts, SMS and worklists to be sent and tracked.

How It Works

This framework:

A. Delivers an extensible, light-weight PeopleCode framework that can adapt to future needs.
B. Provides options to co-exist with, consume or be utilized by the existing COMMGEN processing.
C. Supports both real-time and batch notifications.
D. Provides a generic configurable notification solution that is not bound to any particular consuming application or to any particular UI technology.
E. Provides a Notification code level API that is flexible enough to be useful to all consuming applications.

At the deeper, technical level, the solution architecture uses the Entity Registry to generalize all notifications into a single structure. The architecture is modeled on a pluggable channel-based approach. Each notification type is supported by a dedicated channel that supports the idiosyncrasies of the particular notification type. All notification channels conform to the standard interface of a Notification Channel. As new notification types are identified, it will be a matter of creating a new channel that conforms to the standard Channel interface and then registering that channel with the framework.

Core goals are to:

1. Provide a generic extensible framework which meets the real-time notification needs of current features.
2. Not overlap with CommGen functionality (but to compliment and co-exist with it).
3. Provide a clear code level API that is usable by all forms of consumers (for example, Application Classes, Components, Application Engine programs, web services and so on).

The architecture has been implemented using vanilla PeopleTools application classes and Object Oriented design principles.

A vital point to understand is that the framework is not directly exposed to system users, but is used by consuming applications, also known as “consumers” (for example Delegate Access, New User Registration), which in turn interface with system users. PeopleSoft developers, rather than regular users, would directly interact with the framework, aside from the set up pages described below.

Technical Overview

The Notifications Framework is used to send notifications to interested parties (determined by the consumer setup), via a notification channel (email, SMS, alerts or worklists) based on some transaction performed by a consuming application (“the consumer”). For example, if an email needs to be sent to a person with whom an employee wants to share some data, and hence grants access to some of his/her data (as is the case with Delegated Access functionality), then the consumer can make use of Notifications Framework to achieve the purpose of sending email notifications to the person.

The content that needs to be sent to the concerned parties has to be created as a Generic template, which are then used to create the notification. These templates are grouped together under a common consumer. Based on the consumer setup, valid templates for a transaction are selected, and a notification item is created against each valid template.

Finally, the items are checked for any errors or/warnings, and if the items are found to be eligible for sending, then they are sent using the People Tools Notification Classes provided (PT_WF_NOTIFICATION.Notification App Class). The following diagram illustrates the relationships between the components of the framework:

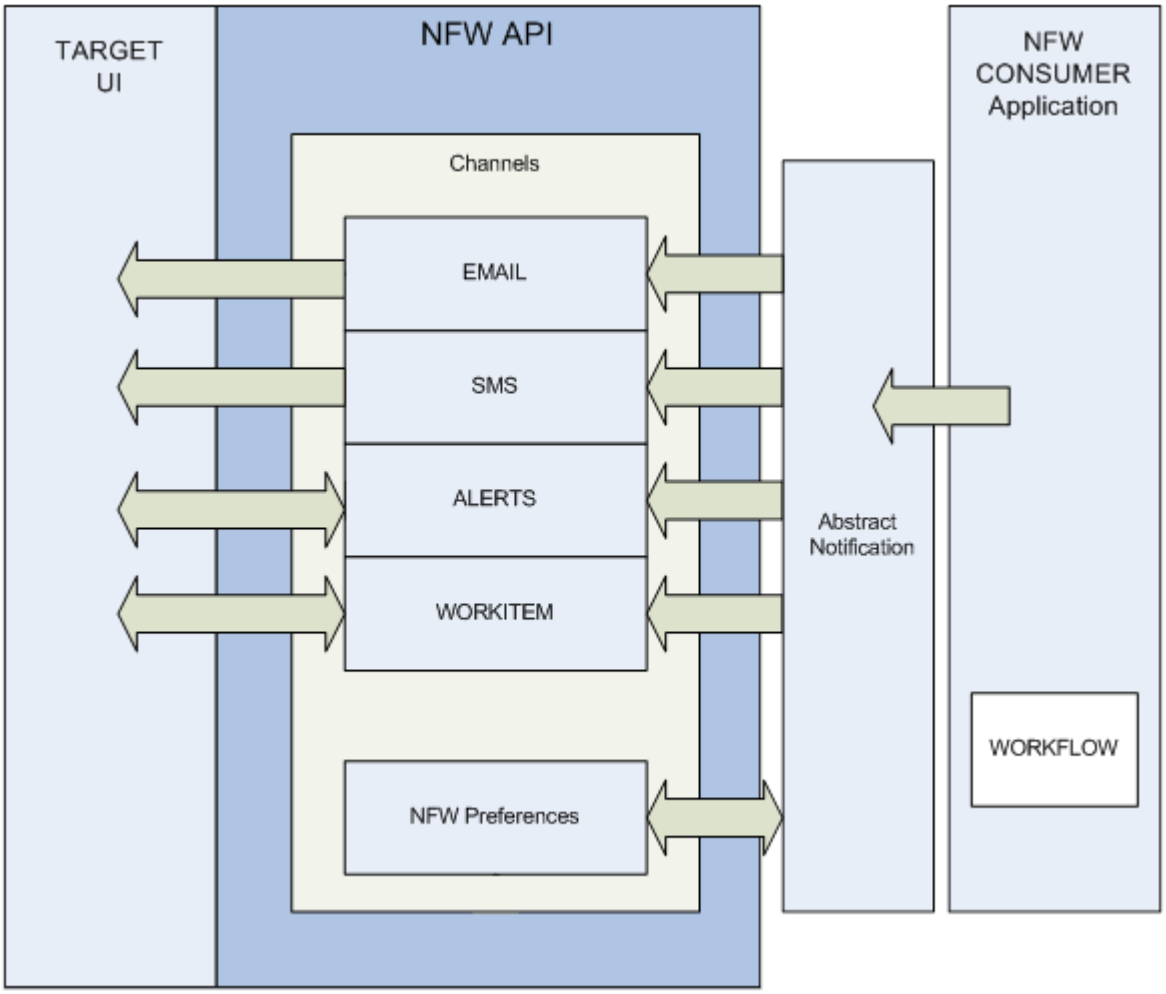


Figure 1 Notifications Framework High Level Architecture

In the diagram above, the AbstractNotification class acts as a layer of encapsulation over the NotificationManager, which is the gateway for all consumers using the Notifications Framework. This is the Abstract Base class that consumers of the Notifications Framework should extend and implement in order to minimize consumer effort in sending notifications. After all the processing is completed the NotificationRouter calls the applicable Notification Channel to deliver the notification type item/items created.

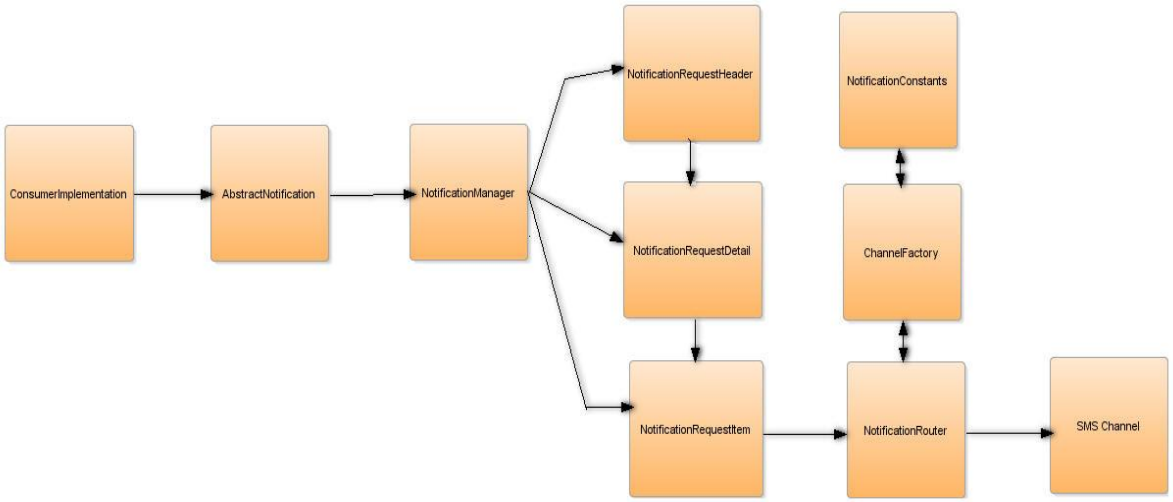


Figure 2 Notifications Framework Communications Diagram (Illustration for SMS Channel, same for all other channels)

Notifications Framework Entities:

Entity	Record
Notification Request Header	SCC_NTFREQ_HDR
Notification Request Detail	SCC_NTFREQ_DTL
Notification Request Item	SCC_NTFREQ_ITM
Notification Recipient	SCC_NTFREQ_RECP
Notification Data	SCC_NTFREQ_DATA

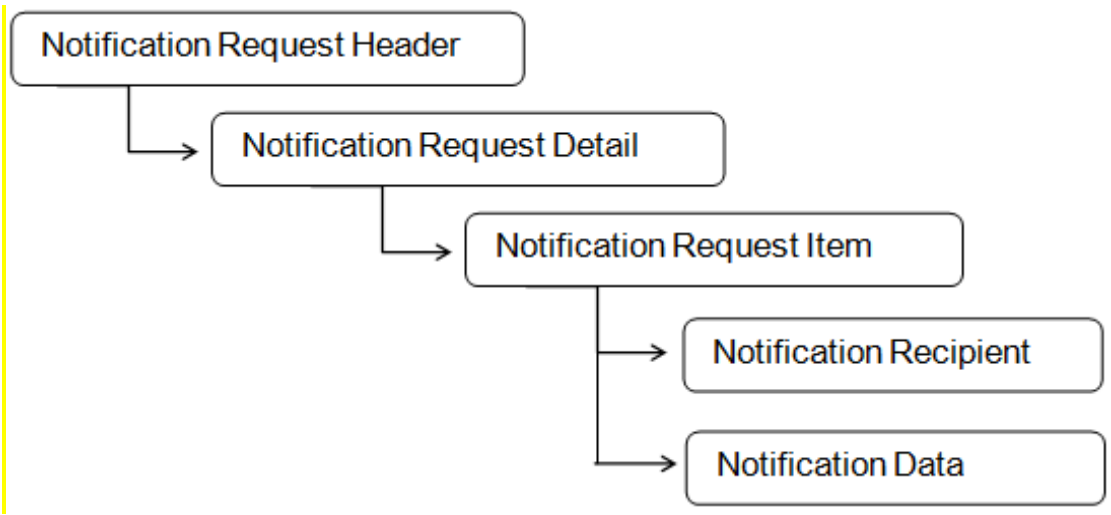


Figure 3 Entity Hierarchy

Setting Up and Consuming the Notifications Framework

This section discusses how to set up the Notifications Framework for sending ad hoc email notifications using the email channel. Where other notification channels differ from the email channel, the differences are highlighted.

The Notifications Framework uses the generic templates of the PeopleTools Workflow Technology, and the reader is urged to review the information in the PeopleTools People Book: "Workflow Technology - Using Notification Templates", for a more comprehensive understanding.

For each of the templates, the Notification Setup page is used to determine the email address to use for sending notifications to the recipients. From there you can define what email addresses to use in the 'To' section of the email message as well as in the CC (carbon copy) or BCC (blind carbon copy) sections. Email addresses of CC recipients are made publicly aware of the notification. Email addresses of BCC recipients are made aware of the notification privately.

If you know the recipients have an EMPLID in the system, you can select to use their email address you have on file that is marked as *Preferred* (subject to the user's notification preferences, if any – see below). If instead, the email address should be obtained from a specific place, than *custom logic* can be used. For example, within Delegated Access, an email notification is sent to the proxies at the

time they are created. At creation time, an email address needs to be entered in the Delegated Access self-service page (Share My Information – Details page). To send the email notification to the proxies, custom logic is used to retrieve the proxy's email address from that specific page. If instead you want to send the notifications to a specific email address, a *static address* can be entered. This option should mostly be used when you are testing the notification templates.

Pages Used to Set Up Notification Framework

Page Name	Definition Name	Navigation	Usage
Generic Template Definition	WL_TEMPLATE_GEN	PeopleTools, Workflow, Notifications, Generic Templates	Define the templates to send email notifications. Define the Message Text and the variables to include. **Note**: The list of variables contained in each of the messages is delivered with your system and require extended coding effort to be modified.
Notification Setup	SCC_NTF_SETUP	Set Up SACR, System Administration, Utilities, Notifications, Notification Setup	Set up the recipients for the notifications. In this page a notification template is created by associating to a generic template. The recipient's configuration is also detailed here.
Notification Consumer Setup	SCC_NTF_CON_CFG	Set Up SACR, System Administration, Utilities, Notifications, Notification Consumer Setup	Specify the generic templates and application classes that the system will use for sending notifications related to a specific consumer.

Step 1 - Configuring the Generic Templates for Notifications Framework

Access the Generic Template Definition page (PeopleTools, Workflow, Notifications, Generic Templates), which defines the basic message template along with any variables to be inserted into the message. The data for these variables is populated by the Notifications Framework consumer in its code, and the bind variables are replaced with the consumer-populated data at runtime to generate the notification's content.

Generic Template Definition | Blackberry Email Responses

Template: DA_DELEGATOR_GRANT

***Description:** Delegator Email (on Grant Acc)

Instructional Text: Send the Delegator a courtesy email notification after delegating access to a Proxy.

Priority:

***Sender:** User **Email ID:**

Subject: Access to some of your school data has been shared with %2

Message Text: <PRE>Dear %1, access to your data has been granted to %2 (%3) to perform or review the following transactions:
%4

If you did not authorize this, please log-in to your account and make the proper adjustments inside the <I>%5</I> page.

You are solely responsible for the actions taken on your behalf by the

Below is the list of available variables for this template.

You can use template variables within your subject or message text.

The following variables can also be used:

%Date, %DateTime, %Time, %ServerTimeZone, %EmailAddress, %NotificationPriority, %NotificationToList, %NotificationCCList

Template Variables

*Value	*Description
%1	Delegator Name
%2	Proxy Name
%3	Delegator Relation with Proxy
%4	Transaction Access Details
%5	Information Access Page

Save | Return to Search | Previous in List | Next in List | Notify | Add | Update/Display

Generic Template Definition page

Template	Name of the generic template that will later be associated to the Template Name in the Notification Setup page.
Description	Description of the template.
Instructional Text	Notifications Framework does not use this field. Enter informational text about the generic template.
Priority	Notifications Framework does not use this field. For information purposes only.

Sender

Select how you want the system to derive and display the From email address value in the email notification. Select one of the following values: *User, System, System-Blackberry,* or *Other*:

User: The system uses the sender's email address defined in its user profile to populate the From field of the email message. This email address is defined under *PeopleTools, Security, User Profiles, User Profiles, General*. Click the Edit Email Address link. This value is derived from the %EmailAddress people code system variable.

When no email address is populated in the sender's user profile, the system uses a default value. The default value is set up in the SCC_COMMON:NOTIFICATION:NotificationConstants application class. Delivered with your system the default value is set as *peoplesoft@oracle.com*. Modify this default value accordingly.

Note: If you want to keep the sender anonymous or if the sender's email address is not to be displayed to the recipients, do not select this value. For example, when a notification is sent as part of the Delegated Access functionality, you may not want to display the delegator's email address (student email address) to the proxy (a third party). Displaying a student's email address might infringe your privacy regulations.

See *PeopleBooks PeopleTools: PeopleCode Language Reference*, System Variables for more information on the %EmailAddress system variable.

System: The system uses the email address value defined in the configuration file (.cfg) through the respective system variables (%SMTPSender). For example, you can maintain the privacy of delegator (not showing email address) by specifying Sender as "System" or "Other".

When no email address is found, the system uses a default value. The default value is set up in the SCC_COMMON:NOTIFICATION:NotificationConstants application class. Delivered with your system the default value is set as *peoplesoft@oracle.com*. Modify this default value accordingly.

Note: The email address you configure in the configuration file must be a valid email address, otherwise the email notifications will not be sent.

See *PeopleBooks PeopleTools: PeopleCode Language Reference*, System Variables for more information on the %SMTPSender system variable.

	System-Blackberry: The system uses the email address value defined in the configuration file (.cfg) through the respective system variables (%SMTPBlackberryReplyTo). When no email address is found, the system uses a default value. The default value is set up in the SCC_COMMON:NOTIFICATION:NotificationConstants application class. Delivered with your system the default value is set as *peoplesoft@oracle.com*. Modify this default value accordingly. **Note**: The email address you configure in the configuration file must be a valid email address, otherwise the email notifications will not be sent. **See** *PeopleBooks PeopleTools: PeopleCode Language Reference*, System Variables for more information on the %SMTPBlackberryReplyTo system variable. *Other*: When selected, the Email ID field is required. Enter the email address of your choice to be displayed in the From field of the email message. Use this option to protect the privacy of the sender. **Note**: The email address you enter in the Email ID field does not need to be valid. This allows you to enter a meaningful email address such as: YourInstitutionName@edu.com.
Email ID	The system uses the email address entered in this field only if you have selected *Other* in the Sender field.
Subject	Enter the subject that will later appear on the email message. The subject can contain variables.
Message Text	Enter the message text contained in the email message. The message text can contain variables.

Template Variables

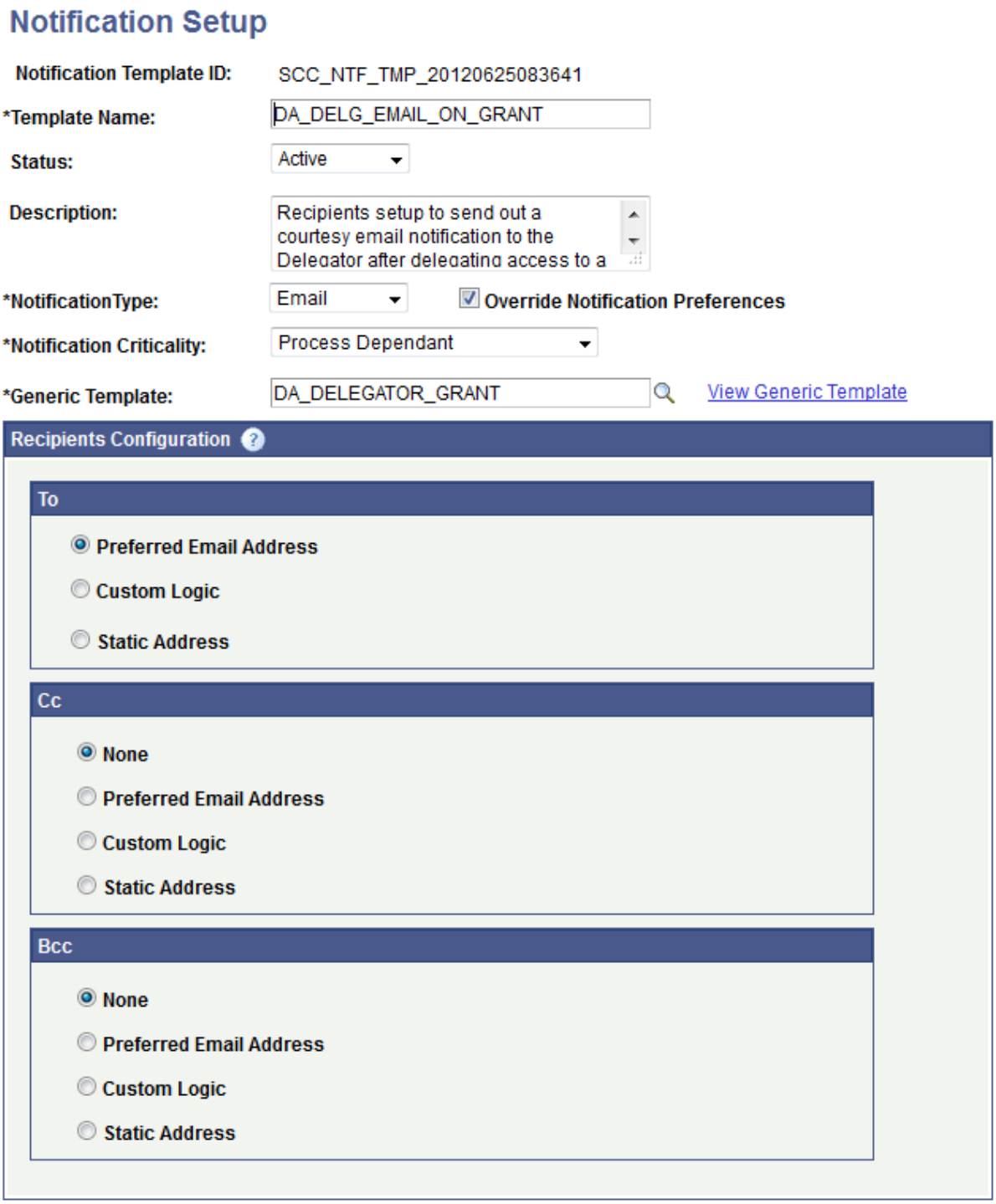
If bind variables are used in the Subject or Message Text, the same must be defined in the Template Variables grid. The consumer populates the list of all values that can be used in a StringHashMap having a (Key, Value) pair, with bind variable number as the key. These values are called as Notification Context Parameter values. The required values are then used in the generic template.

Template variables can be used in any order and in any quantity. You may use only 3 or 4 variables from the entire list or maybe none. These can be any values from the list; for example, %5, %13 and %17 if we are using three variables. These values would be populated at runtime and the sent mail would have these values replacing the bind variables.

Warning! When any template variables are used, their population logic also needs to be coded in the consumer code where the notifications are triggered. For example, Delegated Access provides a list of template variables and the logic to populate them is also delivered in the Delegated application class code. Therefore, addition of any new template variables requires substantial programming efforts.

Step 2 - Configuring Notification Setup for the Generic Templates

Access the Notification Setup page (Set Up SACR, System Administration, Utilities, Notifications, Notification Setup). This page links the generic template discussed previously into the Notifications Framework, sets the channel, criticality, and how the recipients are determined.



Notification Setup page (email channel selected)

Notification Template ID	Displays the auto-generated unique ID. When you add a new notification template record, this field displays the value *NOID* until you save the record. When you save the new record, the field displays the ID that the system assigned to the template.
Template Name	Informative name for the Notification Template ID created above. A template created here needs to be associated to a generic template name. You can give the template the same name of the generic template name or give a more informative name.
Status	Activate or inactivate the notification template. The status must be “Active” for a notification to be generated from the notification template. If the status of a template is “Inactive”, an error message is logged for a notification criticality of “Process Dependant” template and information message logged for an “Informative” template.
Notification Type	Type of notification to send. Valid values are *Alerts*, *Email*, *SMS* and *Worklist*. This is the channel to be used by the notification.
Override Notification Preferences	Only available if the Notification Type selected is Email or SMS. If selected, this causes the recipient’s notification preferences (set up in Self Service > Campus Personal Information > Notification Preferences) to be ignored. The system would then use the values set for the Preferred Email Address or Preferred Phone Number as set via Campus Community > Personal Information > Biographical > Addresses/Phones > Electronic Addresses (for Email Address) / Phones (for Phone Number). For most situations it would be left unchecked, but in circumstances where a recipient must get an email, regardless of the preferences (if any), it should be checked. The primary reason for this field is that some notifications must be sent to a recipient even if they have decided to disable Email or SMS Notifications. An example would be emails sent from the system to enable a user to retrieve a password, in the case of a forgotten password.

Notification Criticality	Indicates how critical the template is from the business process perspective, and may be set to either *Informative* or *Process Dependant*. If it is critical to the business, we advise that you select *Process Dependant*, otherwise select *Informative*. This information is later used to categorize the errors that may have happened when triggering the email template and sending the email notification. If there is an *error* with a notification template that has been set as *Informative*, then the system *continues* with the processing. This error is considered as a *Normal Error.* If, however, there is an error with a notification template set as *Process Dependant*, the system stops processing, as the business process would be impacted by this error. This is classified as a *Reconcile Error*. Normal errors are logged in the Notification Error logging tables. Reconcile errors are logged in the system's App Server Logging (same as *Exceptions*). These would only be logged when tracing is enabled. For example, within Delegated Access framework, an email notification is sent to the proxy to indicate a transaction has been delegated and inform them of the steps to take to make use of this privilege. The email notification sent to the proxy is critical for the completion of the delegation process. In this case the email template set for the proxy can be marked as *Process Dependant*. A second email is also sent to the person who delegated access as a courtesy to inform him or her that access to her or his data was delegated to somebody else. The template for this second email is mostly informational (*Informative)*. The log tables holding the error and warning information in case of informational (*Informative)* templates are: • SCC_NTF_LOG • SCC_NTF_LOGPARM The app server logs hold the error messages for *Process Dependant* templates, and can be found as a trace file in the App Server log folder. Consequently, a trace needs to be enabled to log these messages.
Notification Life (in days)	This option becomes available if the channel (Notification type) selected is Worklist. This option is used to timeout the worklist item notification based on the number of days set in this field.
Generic Template	Select the generic template for which you want to define the recipients.

Recipients Configuration

Use this group box to indicate how the system should set the To, CC, and BCC values in the notification messages. Note that the sections (To, CC, and BCC) available in this group box vary depending on the Notification Type (channel) selected:

1. Emails - To, CC, and BCC
2. Alerts – To
3. SMS – To
4. Worklists – To

None

Available only for CC and BCC.

Select if you do not want to include an email address in the CC or BCC section of the email.

Preferred Email Address

Select this option if you want to use the recipient's email address that is marked as preferred in the email address record (EMAIL_ADDRESSES).

Choose this option mostly if you know the recipients have an ID in your system (the email address record is keyed by EMPLID).

Note that if the recipient has set their notification preferences (Self Service > Campus Personal Information > Notification Preferences) to another email address, then it is used for the email rather than the preferred email address, unless the Notification Setup page's 'Override Notification Preferences' is checked (Set Up SACR > System Administration > Utilities > Notifications > Notification Setup)

Custom Logic

Select if you want an application class to determine how the email address of the recipient should be set. You have to create this PeopleCode application class and specify the class's name in the Application Class field.

When you select the Custom Logic option, the Application Class field becomes available.

For an example of an application class containing custom logic to determine an email address to use, see the Delegated Access application class called SCC_DA:NOTIFICATION:DACustomLogicProvider.

Static Address

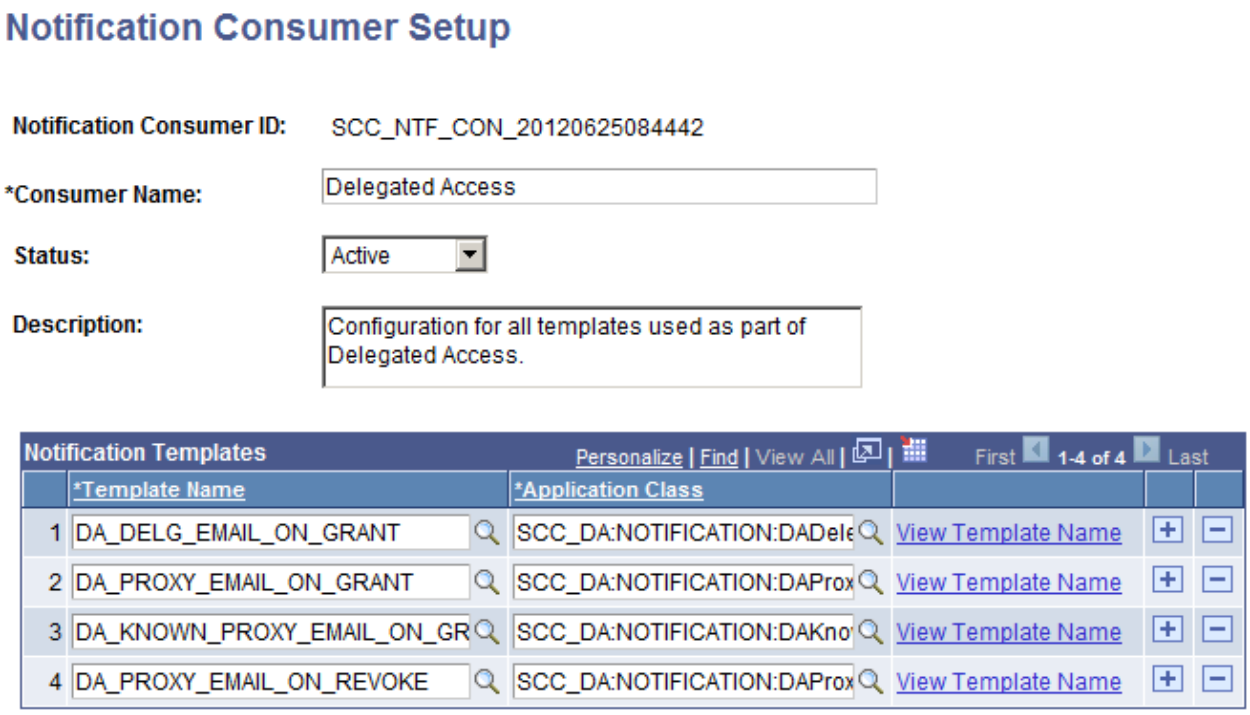
Select if you want all the email messages to be sent to a particular email address. Enter this email address in the Email Address field.

When you select the Static Address option, the Email Address field becomes available.

Use this option mostly when you are testing the notification templates to verify the content of the email messages generated and the required configuration is properly done.

Step 3 - Set Up a Notification Consumer

Access the Notification Consumer Setup page (Set Up SACR, System Administration, Utilities, Notifications, Notification Consumer Setup). This is the point where, for a consumer, the template and notification set up are associated with the consumer code (the application class) that will populate the template fields and send the notification via the chosen channel to the recipient.



Notification Consumer Setup page

Notification Consumer ID	Displays the auto-generated unique ID. When you add a new notification consumer record, this field displays the value *NOID* until you save the record. When you save the new record, the field displays the ID that the system assigned to the consumer.
Consumer Name	Enter the name of the consumer that will consume the notification framework.
Status	Activate or inactivate the notification consumer. If the consumer is inactive, the system will not send any notifications.

Notification Templates

Use this group box to specify the templates that you have configured for the notification consumer on the Notification Setup page. To process notifications, the framework needs to know which templates are valid to use for sending out emails with respect to the consumer's business process or scenario. For each of the templates, specify the application class logic which the Notifications Framework will use to identify the valid template for triggering the notification. For example, within Delegated Access, two email notifications are sent at the time the user delegates new access to somebody. After making the data change and saving the component, one email notification is sent to the user and one to the person to whom access has been delegated. These two notifications use two different templates and are generated under a specific logic. The logic is defined in an application class. For examples of application classes containing logic to trigger a notification, look at any of the delivered Delegated Access application classes:

- SCC_DA:NOTIFICATION:DADelegGrantTemplateProvider
- SCC_DA:NOTIFICATION:DAProxyGrantTemplateProvider
- SCC_DA:NOTIFICATION:DAKnownProxyGrantTmpltProvider
- SCC_DA:NOTIFICATION:DAProxyrevokeTemplateProvider

The Application Class, configured against each consumer Template Name, would be invoked (returns *true* or *false*) to determine if that particular template is to be used for sending notifications or not.

Step 4 - Trigger the Notifications Framework

This step describes how functionality would initiate a notification and depends on the Consumer functionality involved. An example would be the Delegated Access - Share My Information page, from where the Notifications Framework is triggered.

As displayed below, the Contact Name, Relationship and Email Address are defined in the page. The transactions for which access is to be granted/revoked are then selected/cleared. On selecting the Save button, the Notifications Framework is triggered via the Application Class code associated with it.

Figure 4 Delegate Access "Share My Information" Page

Users can refer to the Delegate Access Application Class SCC_DA:NOTIFY as a reference for how to trigger the Notifications Framework as a consumer.

Methods in Notifications Framework to be used by Consumer

Note: As of Bundle 29, consumers no longer use the NotificationManager application class; they now need to extend the **AbstractNotification class**.

On instantiating the AbstractNotification class, a Notification Header object is instantiated. On calling the createNotificationDetail() method of AbstractNotification class, Notification Detail object/s are created. Next, the consumer needs to use the createConsumerNotificationItems() method of AbstractNotification to create the item/s associated with the transaction. Finally, the send() method of the AbstractNotification must be invoked. This in turn calls the Header send(), followed by Detail send() and Item send(). Item send() invokes the NotificationRouter send(), which creates the appropriate channel object, in this case Email (EML), and calls the send() of the EmailChannel App Class. This in turn calls the send() method of the PeopleTools created PT_WF_NOTIFICATION.Notification which delivers the Email notification.

The methods that need to be used by the consumer to consume the Notifications Framework are listed below:

1. Write an AppClass that extends **AbstractNotification** class. The Consumer ID needs to be passed to the Super Class constructor here, as shown below.

   Sample Code:

   ```
   %Super = create SCC_COMMON:NOTIFICATION:AbstractNotification(&Consumer_ID);
   ```

2. Implement the abstract method **createNtfContext().** This will create the context for the consumer as already explained in HDD section above.

   Sample Code:

   ```
   method createNtfContext
      /+ Returns SCC_COMMON:NOTIFICATION:INotificationContext +/
      /+ Extends/implements SCC_COMMON:NOTIFICATION:AbstractNotification.createNtfContext
   +/

      Return create SCC_DA:NOTIFICATION:DANotificationContext();

   end-method;
   ```

3. Implement the abstract method **populateNtfContext**(). This method populates the context create above with the necessary values as will be required during the processing.

   Sample Code:

   ```
   /* Downcast */
      Local SCC_DA:NOTIFICATION:DANotificationContext &myNtfCntxt = &p_ntfContext As
   SCC_DA:NOTIFICATION:DANotificationContext;

      Rem Populate DANotification Context values;
      &myNtfCntxt.EMPLID = &Emplid;
      &myNtfCntxt.SCC_DA_PRXY_ID = &proxyId;
   ```

4. Invoke the **send()** method

   %This.send();

5. Override the OnSuccess() and OnError() for custom behavior.
   Sample Code:

```
method onSuccess

   SQLExec("Update PS_SCC_DA_PROXY set SCC_DA_NTF_REQ_ID=:1 where  EMPLID=:2 and
SCC_DA_PRXY_ID=:3", %This.ntfReqID, &Emplid, &proxyId)

end-method;


method onError

SQLExec("Update PS_OUTPUT set RESULT= :1", &Error);

end-method;
```

Step 5 - Set Up SMS Notification Page

This setup is only needed for consumers using the SMS channel to send notifications (Set Up SACR, System Administration, Utilities, Notifications, SMS Notification Setup).

Figure 5 SMS Notification Setup Page

This page stores information about the account details of the SMS Gateway that sends the SMS notification beyond the system. As well as the usual security details like user name, password and sender name, it also has information about the URL of the SMS Gateway, which is used to create the final URL with parameters for sending the SMS notification. These details are saved to the record SCC_NTF_SMS.

Username The username for the SMS Gateway account used for sending SMS.

Password The password for the SMS Gateway account used for sending SMS.

Sender	The Sender name for the SMS Gateway account used for sending SMS.
SMS Method	SMS can be sent either by HTTP URL posting or SOAP method using HTTP protocol.
SMS Gateway URL	The URL of the SMS Gateway provider.
Application Class Path	The Customer-implemented Application Class which extends the BaseSMSChannel. This class is used primarily to create the final SOAP Message/URL which is to be sent for triggering the SMS. It needs to be updated after modifications have been made to the page.

Step 6 - Integration Broker Set Up for the SMS Channel

Note: Currently, Integration Broker set up is only needed for the SMS channel.

Nodes

Create a new External Node or use an existing Node (PeopleTools, Integration Broker, Integration Setup, Nodes, this node has been delivered as part of April 2013 Bundle changes to the Notification Framework). Set up the properties of the Node as shown in the screenshots below.

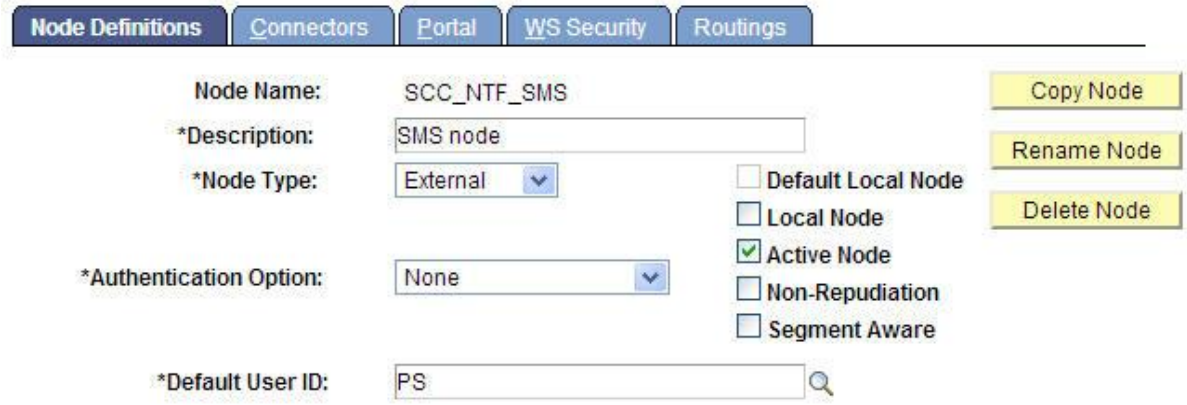


Figure 6 Node Definition Page

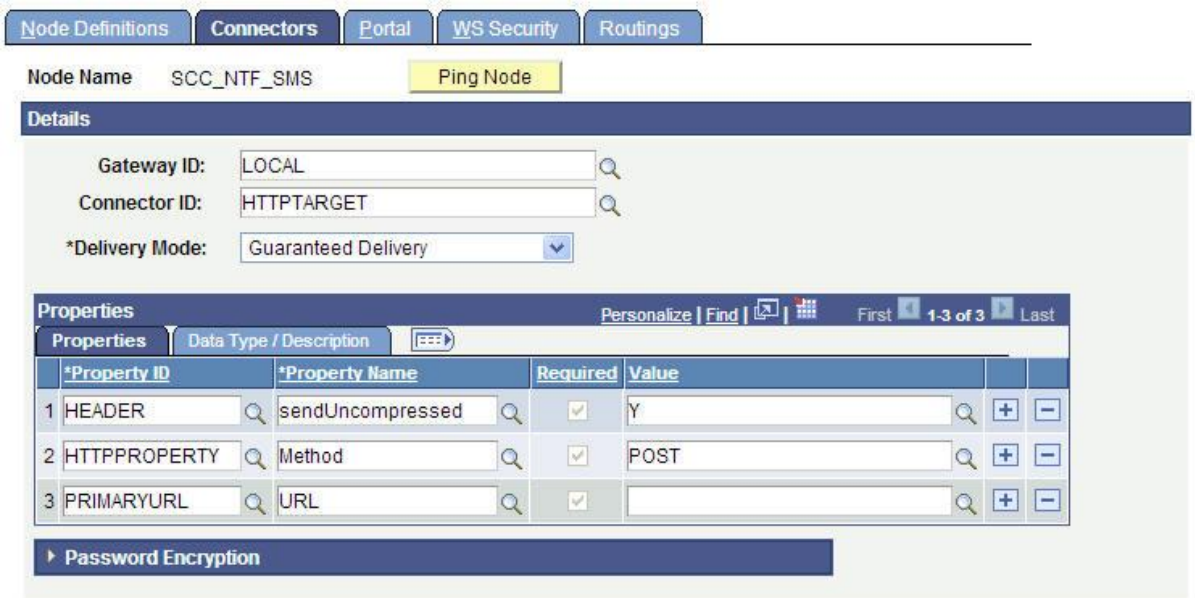


Figure 7 Node - Connectors Page

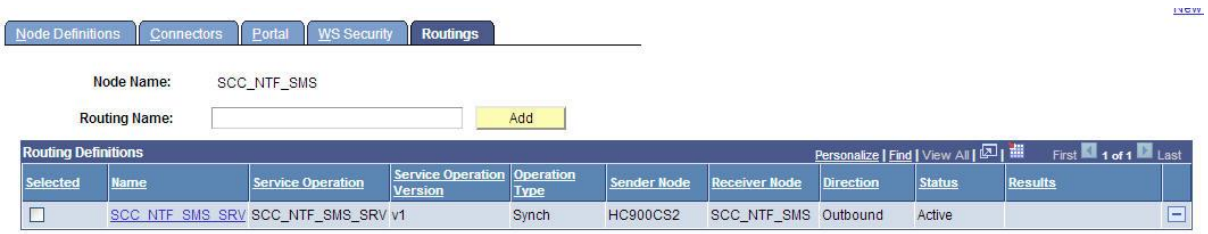


Figure 8 Node - Routings Page

Ensure that there is an Active Outbound Routing from the local node to the external node. If not create/generate an active routing.

Message

Create a new Non-Rowset based message as shown below.

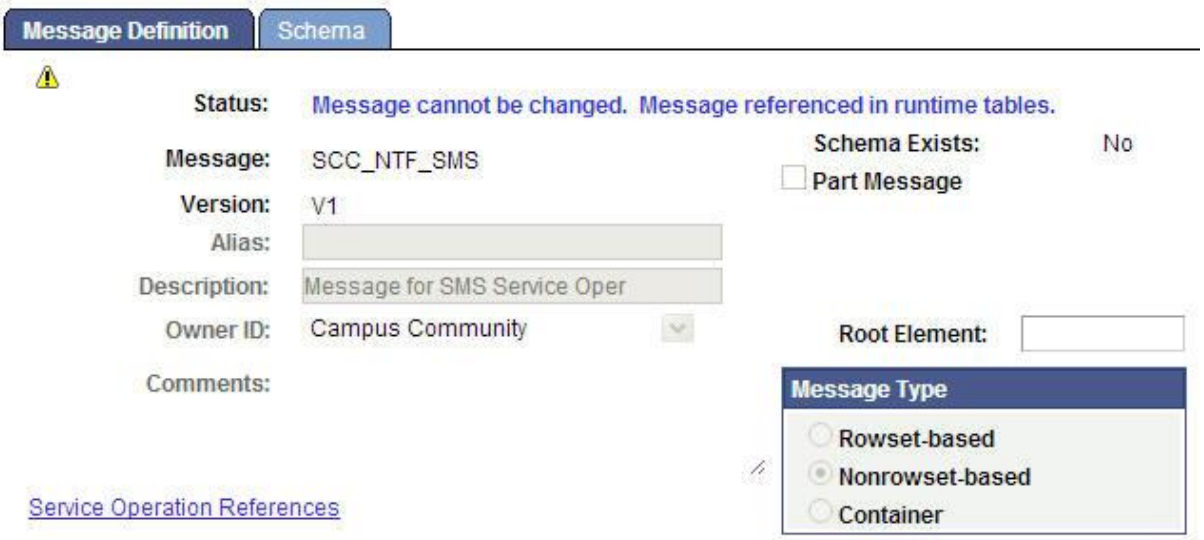


Figure 9 Message Definition Page

Service Operation

Create a new Service Operation, as shown below. This should use the message created above:

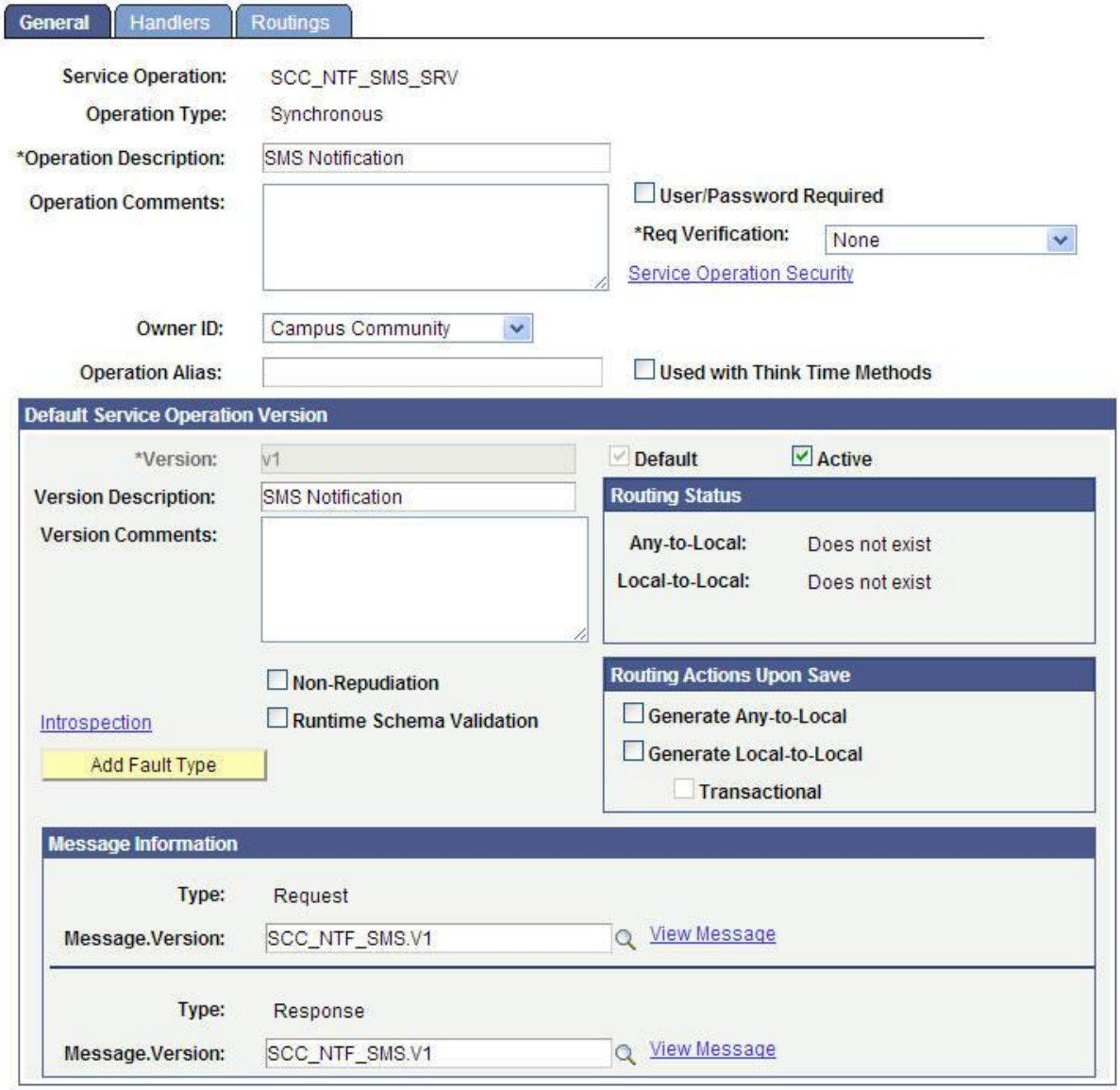


Figure 10 Service Operation - General Page

On the Routings tab, see that there is an active outbound Routing from the Local Node to the External node created above, as shown below:

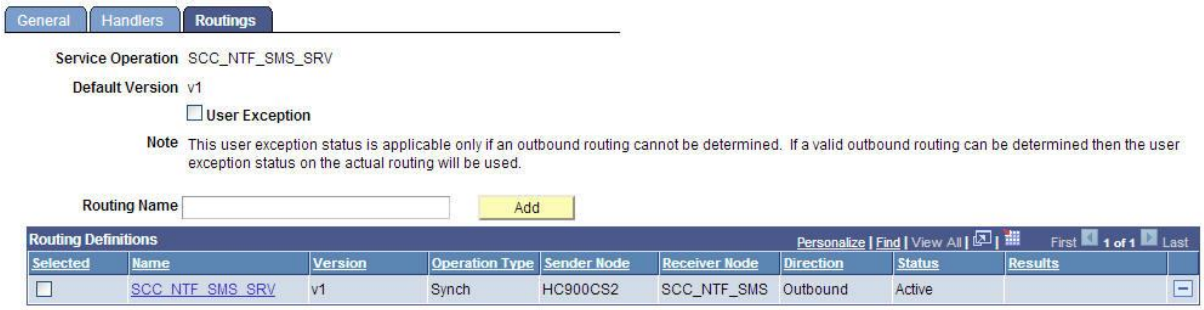


Figure 11 Service Operation - Routings page

Service

Create a new Service, and add the Service Operation to the Service, as shown below:

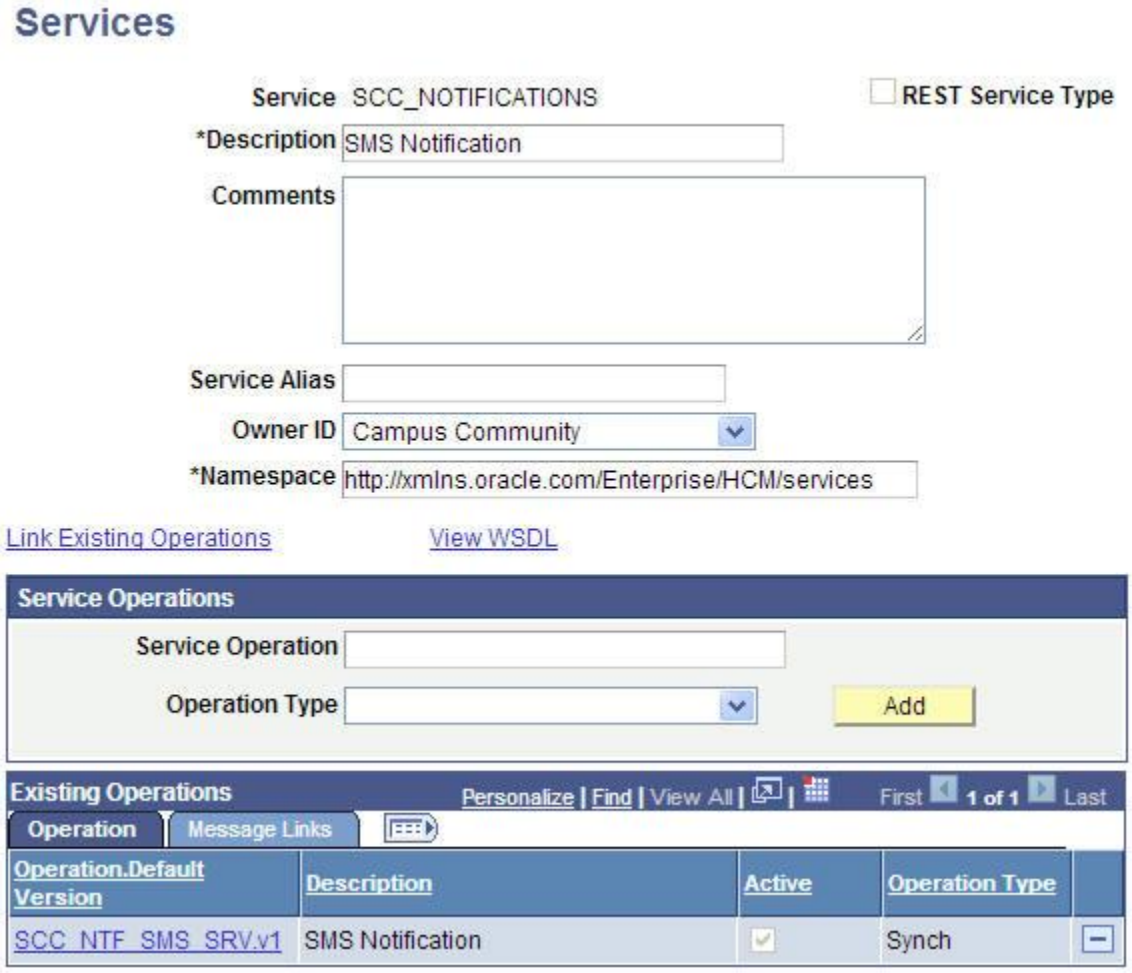


Figure 12 Service Page

Please note that the above service needs to be called by PeopleCode in the consumer application.

Batch Notifications Run Control Page

The batch notifications run control page (Set Up SACR, System Administration, Utilities, Notifications, Batch Notifications) can be used to send notifications in batch mode. This run control uses the Population Selection feature to select the EMPLIDs of the recipients who will receive the notifications. Currently it can be used to send Email, SMS and Alert notifications. It cannot send Worklist notifications, nor can it send notifications if the template recipient is configured to “Custom Logic”.

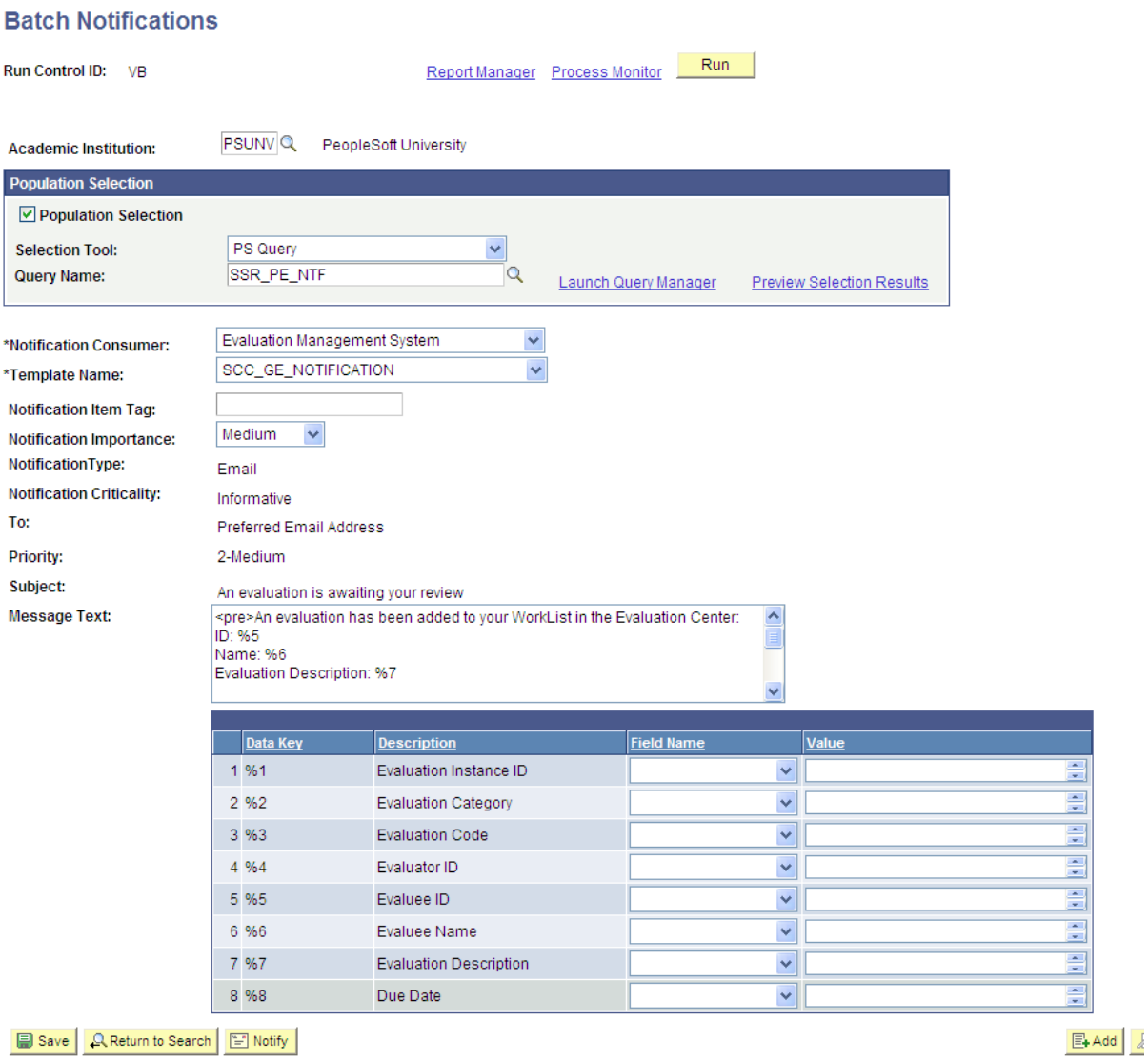


Figure 13 Batch Notifications Run Control Page

Following are the details of each field on the page:

Academic Institution	The academic institution to be used in the PS Query criteria.
Population selection	When this check box is selected, the selection tool becomes available. (this run control does not support manual selection)
Selection Tool	Select the Population Selection tool available for this component.
Notification Consumer	Enables selection of a consumer from the dropdown list of all Notifications Framework consumers.
Template Name	Enables selection of a template from the dropdown list of all active templates for the selected consumer. Currently, only templates of types Email, SMS & Alert and not set to use “Custom Logic” in recipient configuration are available.
Notification Item Tag	Enables the entry of ad-hoc text to tag the notification. The text entered in this field is used to populate the notification item

	SCC_NTFREQ_ITM_TAG column.
Notification Importance	By default this importance is populated with the generic template priority field. A user can override this value.
Notification Type	This is a read-only field populated from the selected notification template.
Notification Criticality	This is a read-only field populated from the selected notification template.
To	This is a read-only field populated from the selected notification template.
Priority	This is a read-only field populated from the selected notification template.
Subject	This is a read-only field populated from the selected notification template.
Message Text	This is a read-only field populated from the selected notification template.
Bind variables	This grid is populated based on the template name selected and the population query selected. • Data Key: Bind variable key from the generic template. • Description: Bind variable description from the generic template. • Field Name: This dropdown is populated with the list of fields selected in the PS Query. The user can map the field from the PS Query to the bind variable, so that the value from the query can be used as bind value for the bind variable in the template. • Value: This is a free form text. A user can enter a value in this field so that this static value can be used as bind value for the bind variable in the template.

Notifications Preferences Setup For Users

Users and administrators can set and amend their preferences for how they will receive their notifications using the following pages (and are recommended to do so). Please note that these preferences will be ignored if the Notification Setup page option of Override Notification Preferences has been selected). This capability is offered for situations where a recipient must be sent a notification even if they have not set their preferences yet, and particularly relates to Delegate Access/New User Registration.

Self Service Users

For individual users, the Self Service Notification Preferences page enables them to specify which phone or email type should receive notifications (Self Service, Campus Personal Information, Notification Preferences). A major reason for this feature is so that the user can set their phone type to one which can handle an SMS message, unlike the traditional fixed line phones.

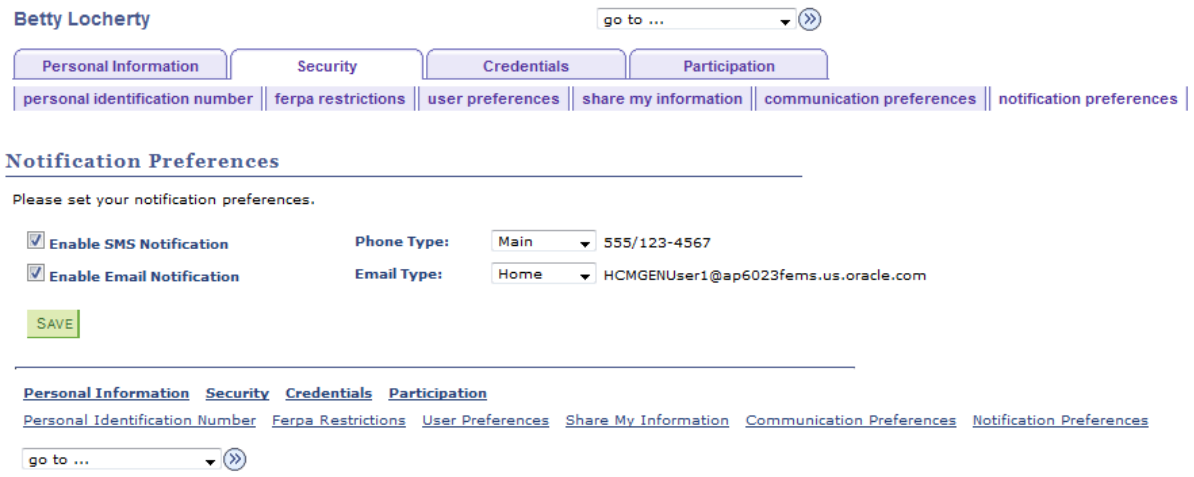


Figure 14 Self Service Notification Preferences

Administrative Users

Administrative users can use the Notification Preferences page to update individual user preferences (Campus Community, Personal Information, Biographical, Personal Attributes, Notification Preferences):

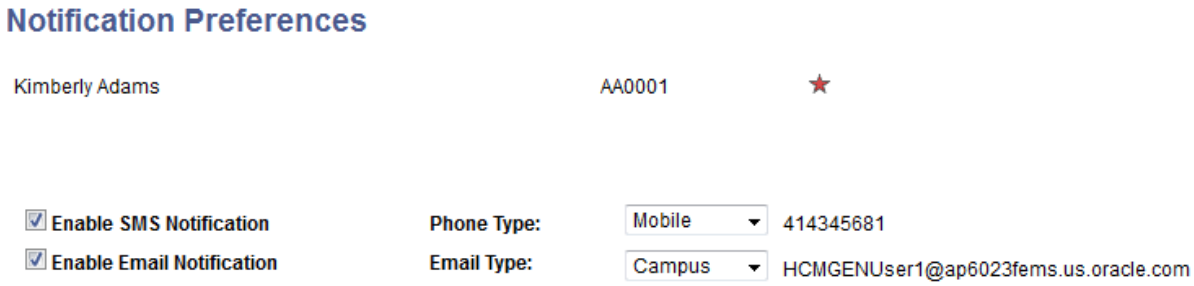


Figure 15 Admin Notification Preferences Page

Users must have at least one email or SMS type to be able to set a preference on these pages. If a phone or email address does not exist, the user can use the “Define Emails” or “Define Phone Numbers” hyperlink that will automatically appear to set them, as illustrated below:

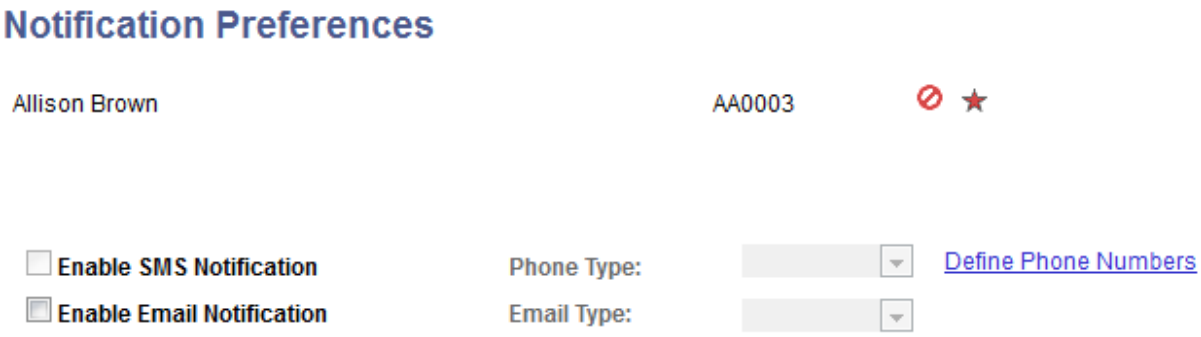


Figure 16 Admin Notification Preferences Page with define phone number hyperlink

See: *PeopleSoft Campus Community 9.0 Fundamentals PeopleBook*, Managing Biographical Information, Managing Personal Attributes Information

See: *PeopleSoft Campus Self Service 9.0 PeopleBook*, Using Self-Service Campus Personal Information, Using Self-Service Personal Attributes Data

--oo0oo—