Working with the Notifications Framework

Understanding the Notifications Framework

The Notifications Framework provides a consistent, extensible communication mechanism that Campus Solutions product areas (consumers) can use to enable communications from their areas and a generic extensible framework which meets the real-time notifications needs of current features.

Customers can also craft their own solutions to use the framework without reinventing the lower level coding that enables notifications to be sent and tracked.

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):

- Email
- SMS
- Alerts (for example, an informational message that appears on a portal homepage)
- Worklist Items (an actionable hyperlink that appears on a portal homepage)
- Push Notifications (for mobile apps on iOS/Android)
- Announcements

The Notifications Framework:

- Delivers an extensible, light-weight PeopleCode framework that can adapt to future needs.
- Provides options to co-exist with, consume or be utilized by the existing COMMGEN processing.
- Supports both real-time and batch notifications.
- Provides a generic configurable notification solution that is not bound to any particular consuming application or to any particular UI technology.
- Provides a Notification code level API that is flexible enough to be useful to all consuming applications; for example, Application Classes, Components, Application Engine programs, web services and so on.
- Supports Timeout processing for worklist items.
- Provides the ability to generate Reminder notifications for existing notifications.

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

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Channel. As new notification types are identified, a new channel that conforms to the standard Channel interface will need to be created and registered with the framework.

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 Delegated Access, New User Registration, Campus Mobile), 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 (such as email, SMS, alerts or worklists) based on some transaction performed by a consuming application (the consumer). For example, if an employee wants to share some data with a person via email and hence grants access to some of his/her data (as is the case with Delegated Access functionality), then the consumer can use the Notifications Framework to send email notifications to the person.

The content to be sent to the concerned parties must be created as a Generic template, which is then used to create the notification. The 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.

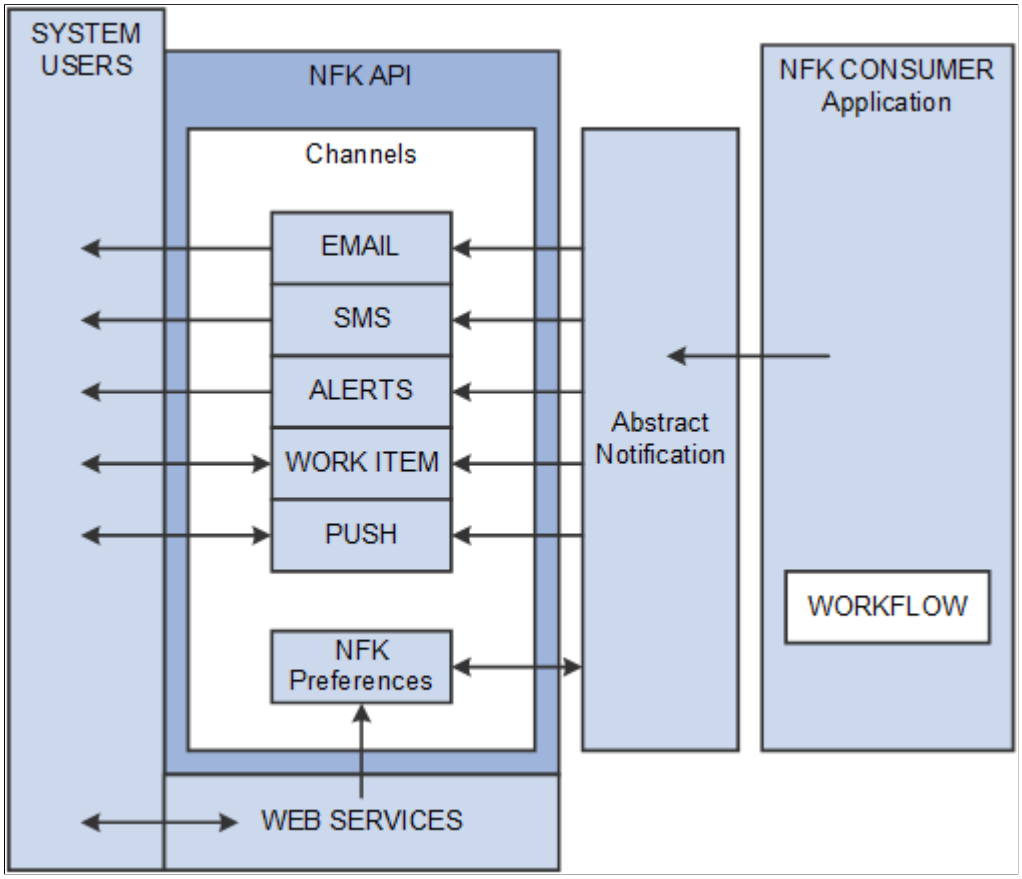
Finally, the items are checked for any errors or warnings, and if the items are found to be eligible for sending, then they are sent using the People Tools Notification Classes provided

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

(PT_WF_NOTIFICATION.Notification App Class). The following diagram illustrates the relationships between the components of the framework:

Image: Notifications Framework High Level Architecture

This example illustrates the Notifications Framework High Level Architecture.



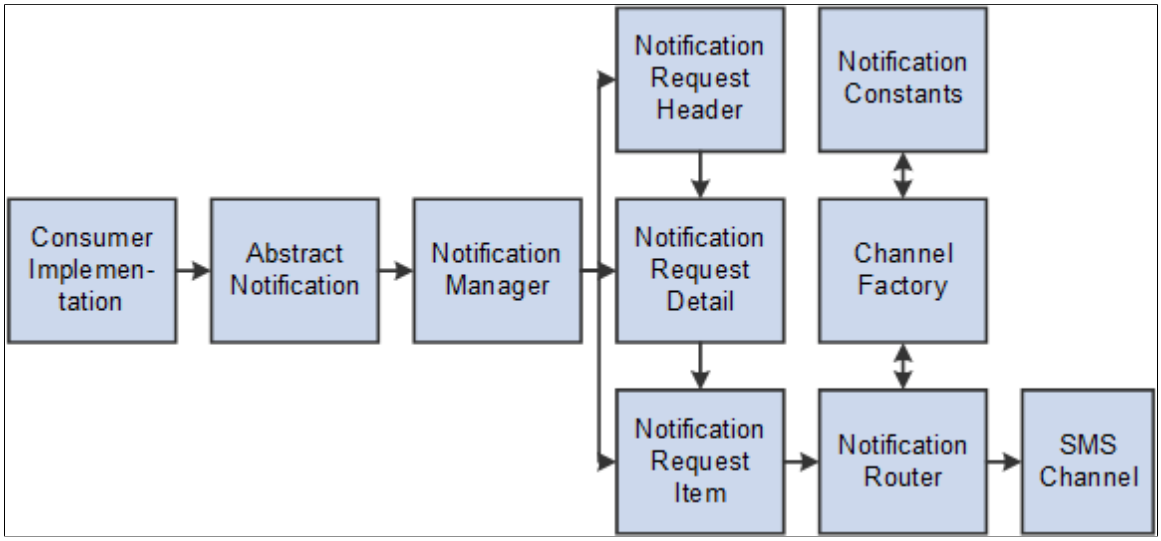
In the diagram above, the AbstractNotification class acts as a layer of encapsulation over the NotificationManager, which is the gateway for all consumers using the Notifications Framework. This is the Abstract Base class that consumers of the Notifications Framework should extend and implement in order to minimize consumer effort in sending notifications. After all processing is completed the

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

NotificationRouter calls the applicable Notification Channel to deliver the notification type item/items created.

Image: Notifications Framework Communications Diagram

This example illustrates Notifications Framework Communications.



Notifications Framework Entities:

Entity	*Record*	*Notes*
Notification Request Header	SCC_NTFREQ_ HDR	N/A
Notification Request Detail	SCC_NTFREQ_DTL	N/A

NotificationRequestDetailER	SCC_NTF_DTL_ER	The NotificationRequestDetailER Entity is used to support multiple recipients like TO, CC, BCC for Email. This entity allows the framework to:
		1. Store the multiple Email TO, CC, BCC recipients.
		2. Link to the NotificationRequestDetail entity.
		When an Email notification is to be sent to multiple recipients (TO, CC, BCC), the consumer needs to set the notification context property SCC _NTF_AUDCE to 'ER', EMPLID to null, and populate all the below properties as appropriate:
		`/* Multiple Recipients for Email (To,CC,BCC) */`
		`property array of string arrEMPLID_TO;`
		`property array of string arrEMPLID_CC;`
		`property array of string arrEMPLID_BCC;`

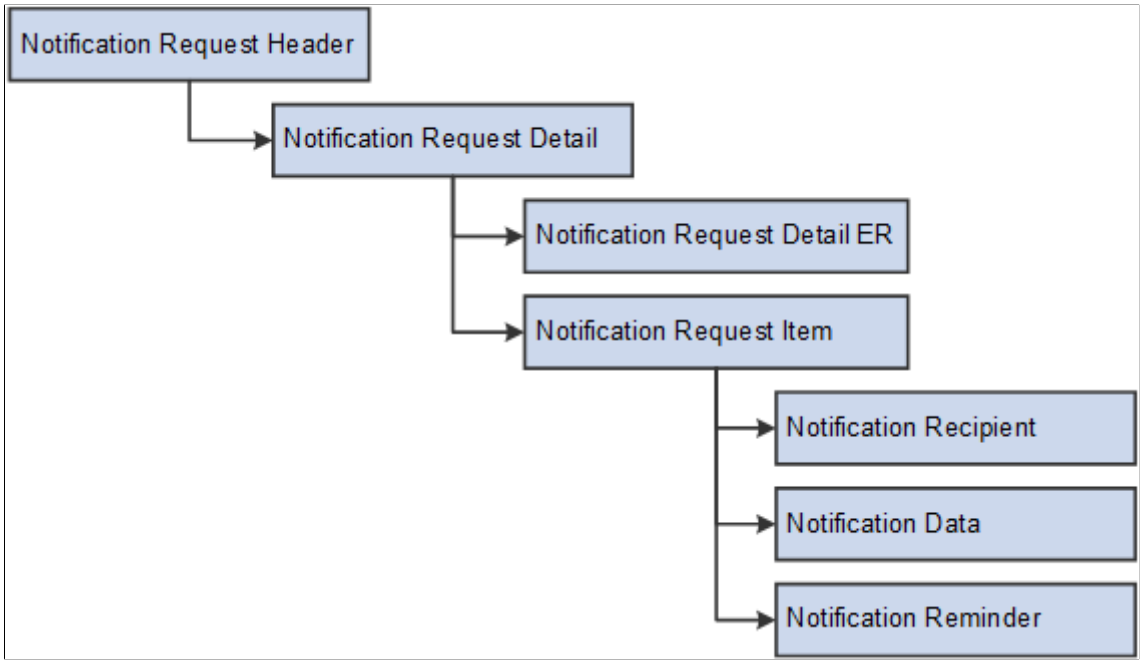
Notification Request Item	SCC_NTFREQ_ITM	N/A
Notification Recipient	SCC_NTFREQ_ RECP	N/A

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Entity	*Record*	*Notes*
Notification Data	SCC_NTFREQ_ DATA	N/A
Notification Reminder	SCC_NTFREQ_ RMDR	The Notification Reminder Entity is only used when a reminder notification is created for a previously generated notification. The Notification Reminder entity allows the framework to: 1. Store the date(s) when Reminder(s) were created for a notification. 2. Link the original notification with the reminder notification.

Image: Entity Hierarchy for Notifications

This example illustrates the Entity Hierarchy for Notifications.



Setting Up and Consuming the Notifications Framework

This section discusses how to set up the Notifications Framework for sending e-mail notifications using the e-mail channel. Where other notification channels differ from the e-mail 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 *PeopleTools: Workflow Technology*, “Using Notification Templates,” for a more comprehensive understanding.

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

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

If you know the recipients have an EMPLID in the system, you can select to use the e-mail address that you have on file which is marked as Preferred (subject to the user's notification preferences, if any – see below). If instead the e-mail address should be obtained from a specific place, then custom logic can be used. For example, within Delegated Access, an e-mail notification is sent to the proxies at the time they are created. At creation time, an e-mail address must be entered in the Delegated Access self-service page (Share My Information – Details page). To send the e-mail notification to the proxies, custom logic is used to retrieve the proxy's e-mail address from that specific page. If instead you want to send the notifications to a specific e-mail address, a static address can be entered. This option should mostly be used when you are testing the notification templates.

Complete the following steps to set up and consume the Notifications Framework:

1. Configure the Generic Templates for Notifications Framework.
2. Configure Notification Setup for the Generic Templates.
3. Set Up a Notification Consumer.
4. Trigger the Notifications Framework.
5. Set Up the SMS Notification page.
6. Integration Broker set up for the SMS channel.
7. Set Up Push Notifications page.
8. Configure IB to use Socket Connector for Push channel delivered services.
9. Install SSL Certificates and configure platform specific server keys.
10. Activate all the IB objects delivered for Push channel.
11. Schedule APNS Feedback job.

Pages Used to Set Up the Notifications Framework

Page Name	*Definition Name*	*Navigation*	*Usage*
Generic Template Definition	WL_TEMPLATE_GEN	PeopleTools, Workflow, Notifications, Generic Templates	Define the templates to send e-mail 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.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Page Name	*Definition Name*	*Navigation*	*Usage*
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 uses for sending notifications related to a specific consumer.
SMS Notification Setup	SCC_NTF_SMS	Set Up SACR, System Administration, Utilities, Notifications, SMS Notification Setup	Set up SMS notifications.
Push Notifications Setup	SCC_NTF_PUSH_SETUP	Set Up SACR, System Administration, Utilities, Notifications, Push Notifications Setup	Set up Push notifications.

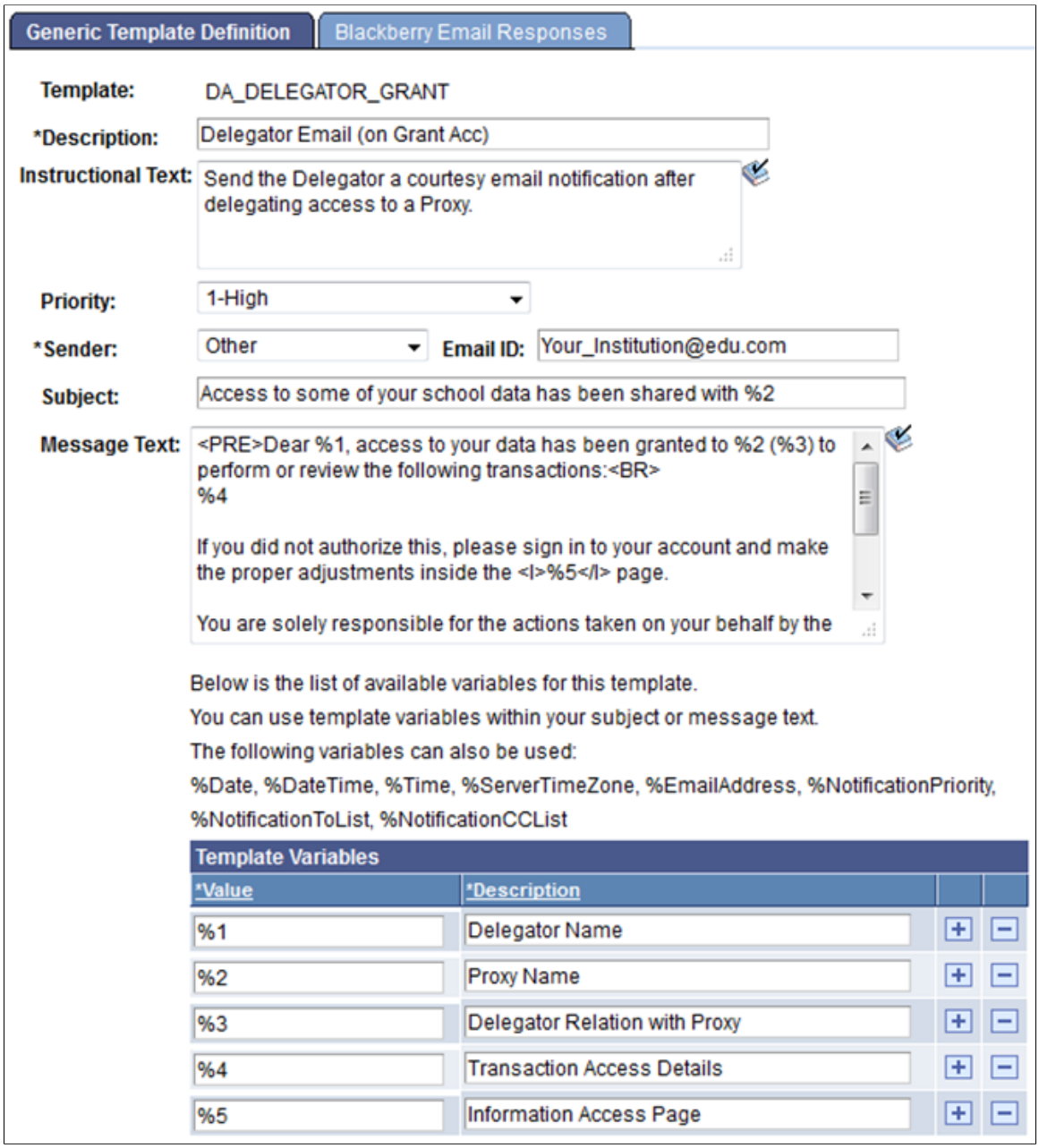
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Configuring the Generic Templates for Notifications Framework

Access the Generic Template Definition page (PeopleTools, Workflow, Notifications, Generic Templates).

Image: Generic Template Definition Page

This example illustrates the fields and controls on the Generic Template Definition Page. You can find definitions for the fields and controls later on this page.



This page is used to define 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.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Template Displays the name of the generic template to be associated to the Template Name in the Notification Setup page.

Description Enter a 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 e-mail address value in the e-mail notification:

- Other: When selected, the E-mail ID field is required. Enter the e-mail address of your choice to be displayed in the From field of the e-mail message. Use this option to protect the privacy of the sender.

  Note: The e-mail address you enter in the E-mail ID field does not need to be valid. This allows you to enter a meaningful e-mail address such as: YourInstitutionName@edu.com.

- System: The system uses the e-mail 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 e-mail address) by specifying Sender as “System” or “Other”. When no e-mail 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 e-mail address you configure in the configuration file must be a valid e-mail address, otherwise the e-mail notifications are not sent. See *PeopleTools: PeopleCode Language Reference* “System Variables” for more information on the %SMTPSender system variable.

- System-Blackberry: The system uses the e-mail address value defined in the configuration file (.cfg) through the respective system variables (%SMTPBlackberryReplyTo). When no e-mail 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.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Note: The e-mail address you configure in the configuration file must be a valid e-mail address, otherwise the e-mail notifications are not sent. See *PeopleTools: PeopleCode Language Reference* "System Variables" for more information on the %SMTPBlackberryReplyTo system variable.

- User: The system uses the sender's e-mail address defined in its user profile to populate the From field of the e-mail message. This e-mail address is defined under PeopleTools, Security, User Profiles, User Profiles, General. Click the Edit E-mail Address link. This value is derived from the %EmailAddress people code system variable. When no e-mail 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 e-mail 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 e-mail address (student e-mail address) to the proxy (a third party). Displaying a student's e-mail address might infringe your privacy regulations. See *PeopleTools: PeopleCode Language Reference* "System Variables" for more information on the %EmailAddress system variable.

E-mail ID	Enter an e-mail address to be used only if you have selected Other in the Sender field.
Subject	Enter the subject for the e-mail message. The subject can contain variables.
Message Text	Enter the message text for the e-mail 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

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

we are using three variables. These values are populated at runtime and the sent mail would contain these values replacing the bind variables.

Warning! When any template variables are used, their population logic also must 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.

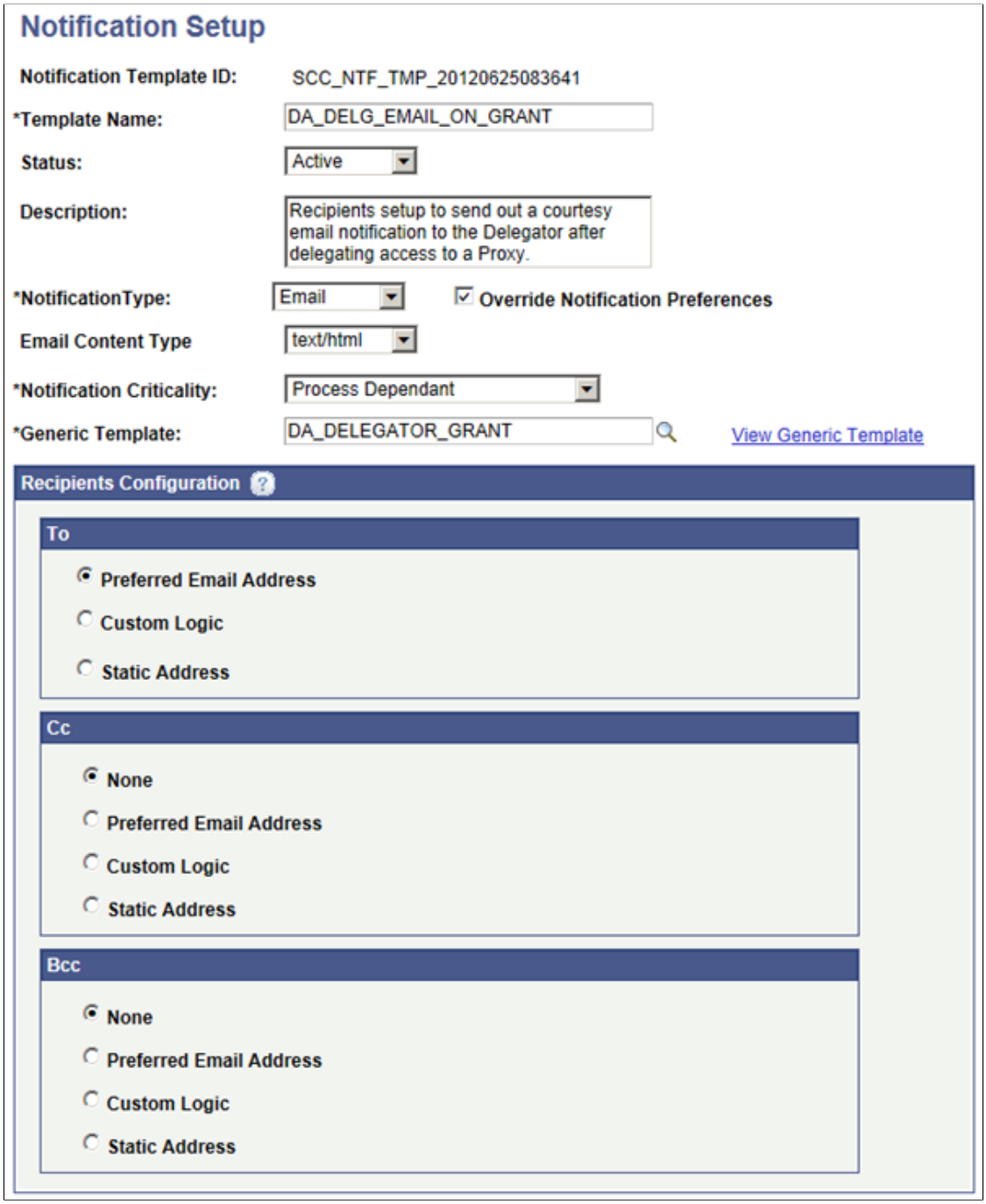
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Configuring Notification Setup for the Generic Templates

Access the Notification Setup page (Set Up SACR, System Administration, Utilities, Notifications, Notification Setup).

Image: Notification Setup Page, Email Channel Selected 1 of 2

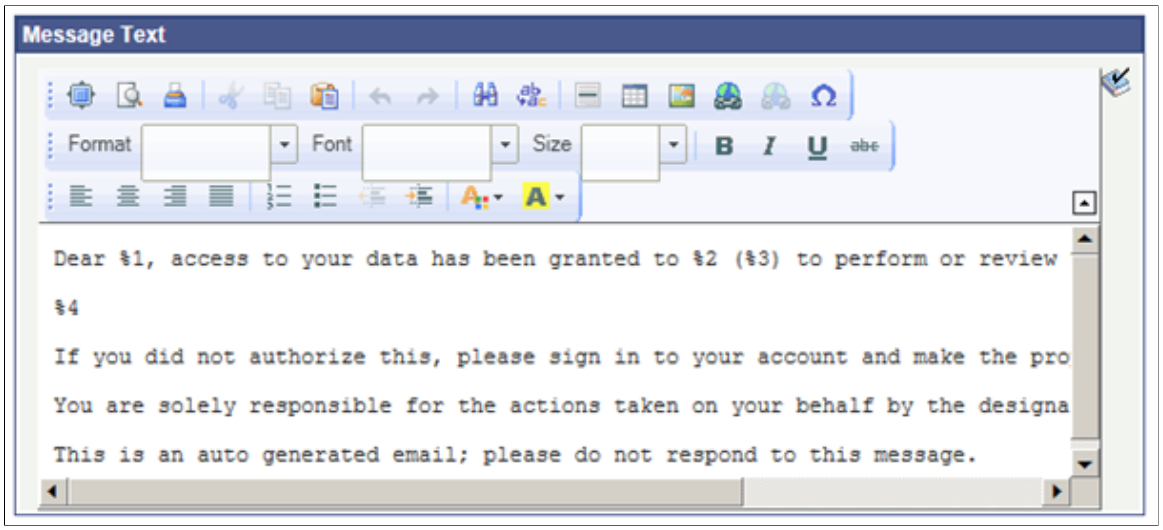
This example illustrates the fields and controls on the Notification Setup, Email Channel Selected, page 1 of 2. You can find definitions for the fields and controls later on this page.



 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Image: Notification Setup Page, Email Channel Selected 2 of 2

This example illustrates the fields and controls on the Notification Setup, Email Channel Selected, page 2 of 2. You can find definitions for the fields and controls later on this page.



This page links the generic template discussed previously into the Notifications Framework, sets the channel, criticality, and how the recipients are determined.

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 must be associated to a generic template name. You can give the template the same name as the generic template name or 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, E-mail, SMS, Worklist and Push. This is the channel to be used by the notification.

Override Notification Preferences

Only available if the Notification Type selected is E-mail 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 E-mail Address or Preferred Phone Number as set via Campus Community > Personal Information > Biographical >

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Addresses/Phones > Electronic Addresses (for E-mail Address) / Phones (for Phone Number). For most situations, the check box would be cleared, unchecked, but in circumstances where a recipient must get an e-mail, regardless of the preferences (if any), it should be selected. The primary reason for this field is that some notifications must be sent to a recipient even if they have decided to disable E-mail or SMS Notifications. An example would be e-mails sent from the system to enable a user to retrieve a password, in the case of a forgotten password.

E-mail Content Type

Available when the e-mail notification type is selected – options are text/html or text/plain. If html is selected, any HTML tags in the message are respected.

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 e-mail template and sending the e-mail 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 be logged only when tracing is enabled.

For example, within Delegated Access framework, an e-mail notification is sent to the proxy to indicate a transaction has been delegated and to inform them of the steps to take to make use of this privilege. The e-mail notification sent to the proxy is critical for the completion of the delegation process. In this case the e-mail template set for the proxy can be marked as Process Dependant. A second e-mail is also sent to the person who delegated access as a courtesy to inform the person that access to her or his data was delegated to somebody else. The template for this second e-mail 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

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

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 or Push.

For Worklist, this option is used to time-out the worklist item notification based on the number of days set in this field.

For Push, this option is used as expiry for the push message in APNS/GCM. If not set, the default value is 7 days.

Callback Application Class

This option becomes available if the channel (Notification type) selected is Worklist. Allows consumers to carry out consumer specific actions when a worklist event occurs.

Consumers create the callback application class by extending the base callback event handler (SCC_COMMON:NOTIFICATION:CALLBACK:BaseNotificationEventHandler).

Currently the only events supported are Timeout and Reminder.

Used by the Worklist Timeout and Reminder Batch process (SCC_NTFWKLTO).

Application Name

This option becomes available if the channel (Notification type) selected is Push. Select the mobile Application Name (configured in Push Notifications Setup page) for which you want to use the template.

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:

- E-mails (To, CC, BCC)
- Alerts (To)
- SMS (To)
- Worklists (To)

None

Select if you do not want to include an e-mail address in the CC or BCC section of the e-mail. Available only for CC and BCC.

Custom Logic

Select if you want an application class to determine how the e-mail 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

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

	becomes available. For an example of an application class containing custom logic to determine an e-mail address to use, see the Delegated Access application class called SCC_DA:NOTIFICATION:DACustomLogicProvider.
Preferred E-mail Address	Select if you want to use the recipient's e-mail address that is marked as preferred in the e-mail address record (EMAIL_ADDRESSES). Choose this option mostly if you know the recipients have an ID in your system (the e-mail 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 e-mail address, then it is used for the e-mail rather than the preferred e-mail address, unless the Override Notification Preferences check box is selected on the Notification Setup page (Set Up SACR > System Administration > Utilities > Notifications > Notification Setup).
Static Address	Select if you want all the e-mail messages to be sent to a particular e-mail address. Enter this e-mail address in the E-mail Address field. When you select the Static Address option, the E-mail Address field becomes available. Use this option mostly when you are testing the notification templates to verify the content of the e-mail messages generated and the required configuration is properly done.

Message Text

This group box appears only if the Notification Type is Email with an Email Content Type of text/html. The formatted text is saved differently depending on the browser used. Set the Format field to *Normal* before creating or updating the message text.

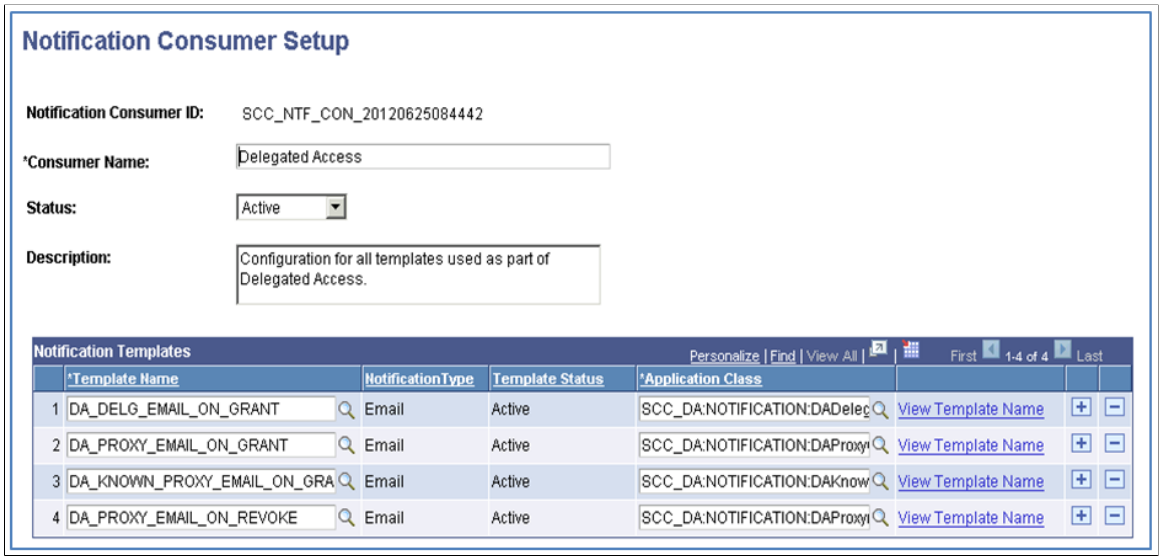
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Setting Up a Notification Consumer

Access the Notification Consumer Setup page (Set Up SACR, System Administration, Utilities, Notifications, Notification Consumer Setup).

Image: Notification Consumer Setup Page

This example illustrates the fields and controls on the Notification Consumer Setup page. You can find definitions for the fields and controls later on this page.



This is the point where, for a consumer, the template and notification setup are associated with the consumer code (the application class) that populate the template fields and send the notification via the chosen channel to the recipient.

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 consumes the notification framework.

Status Activate or inactivate the notification consumer. If the consumer is inactive, the system does not send any notifications.

Notification Templates

Use this grid 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 uses to identify the valid template for triggering the notification. For example, within Delegated Access, two email notifications are sent at the time that 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.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

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, is invoked (returns true or false) to determine if that particular template is to be used for sending notifications or not.

In the same AppClass created above, include logic to populate the EMPLID property of the Notification Context with the recipient's EMPLID. This value may be left blank if the notification recipient does not have an EMPLID (for example within Delegated Access, email notifications are sent to the proxies which may or may not have an EMPLID identified).

When the Notifications Framework is used to dynamically determine the template(s) to trigger multiple notifications, the recipient(s) may be different for each notification. Therefore, on selecting a template to trigger the notification, BaseNotificationContext.EMPLID must be populated dynamically with the respective recipient EMPLID. This must be taken care of by the NFK Consumers in their respective Template Provider App Class code (an Application Class configured for each Template Name in Notification Consumer Setup page).

During NFK processing, the provided EMPLID Context value is stored in SCC_NTFREQ_DTL.EMPLID record field. Later, this stored recipient's EMPLID information is used as part of the Notifications Center page. The page displays the notifications sent to that specific EMPLID.

Warning! Failing to properly populate the EMPLID Context field with the proper recipient's EMPLID may result in displaying in the Notifications Center page notifications sent to someone else.

Triggering the Notifications Framework

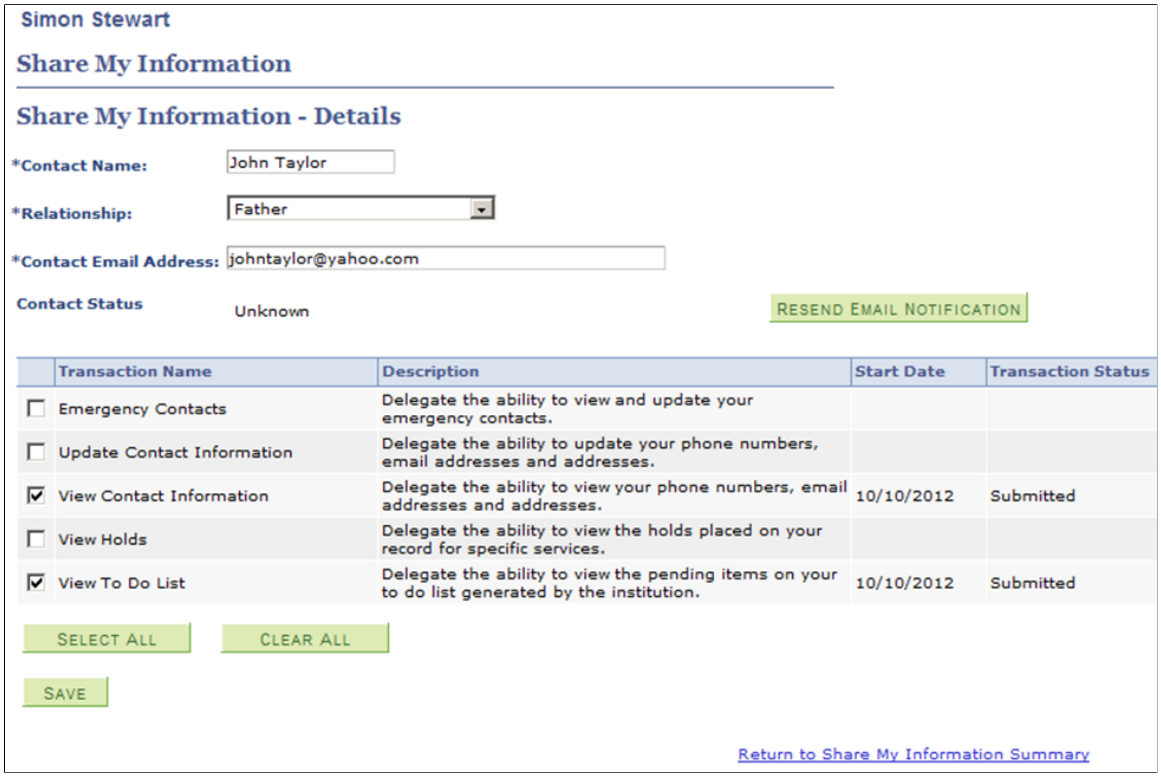
This section describes how functionality initiates a notification and depends on the Consumer functionality involved. An example is the Delegated Access - Share My Information self-service

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

page, from where the Notifications Framework is triggered. When the page is saved, the Notifications Framework is triggered via the Application Class code associated with it.

Image: Delegated Access Share My Information Example Page

This example illustrates the fields and controls on the Delegated Access Share My Information example.



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

See Developer Reference for Setting Up Delegated Access

Methods in Notifications Framework to be used by Consumer

Before invoking Notifications Framework to process or send notifications, consumers can make sure that the consumer and template set are to active. The following methods are examples of error handling:

- Method is ConsumerActive(&pSccNtfConld As string) returns Boolean;
- Method is TemplateActive(&pSccNtfreqTmpltid As string) returns Boolean;

This helps to avoid unnecessary processing by consumers/framework.

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

1. Write an AppClass that extends AbstractNotification class. The Consumer ID must be passed to the Super Class constructor here, as shown in the sample code below:

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

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

2. Implement the abstract method createNtfContext(). This creates the context for the consumer as already explained in the previous section.

   Sample Code:

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

   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 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;
```

Note: If a consumer chooses to send a single notification then it can be done by directly populating the &myNtfCntxt.SCC_NTFREQ_TMPLTID with the Template ID value. If the Notification Context property: SCC_NTFREQ_TMPLTID is not populated then the Notifications Framework uses createConsumerNotificationDetailsItems() method to dynamically create single or multiple notification items based on consumer templates setup.

---

4. Invoke the send(), remind(), or resend() method.

   - %This.send();

     During this processing, the framework creates objects such as Notifications Header, Notification Detail & Notification Item records using the Notification Context data populated by the consumer. Finally, the *send()* method for each notification item is invoked using the appropriate channel (like Email, SMS, Alert or Worklist) as per the Notification Template Setup.

   - %This.remind(Notification ID, Detail Sequence Number, Item Sequence Number)

     This method is used to create a “reminder” notification for a previously generated notification. The parameters to this method are the Notification ID, Detail Sequence Number, and Item Sequence Number of an existing notification.

     During this processing, the framework invokes the send() method to create a new notification. Then the framework stores the date/time and the keys of the new notification in the original notification.

   - %This.resend(Notification ID)

     This method is used to resend a previously generated notification. The only parameter to this method is the Notification Request ID of the notification to be resent. The notification data related to that particular Notification Request ID is retrieved, and the notification is sent again. This is applicable for only previously sent notifications. A ‘isResend’ flag decides whether the

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

current transaction is a send or resend. As with send, the resend also invokes the save() method of message log and logs the info/warning messages generated as part of the resend.

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 a⇒
nd SCC_DA_PRXY_ID=:3", %This.ntfReqID, &Emplid, &proxyId)

end-method;

method onError
SQLExec("Update PS_OUTPUT set RESULT= :1", &p_excep.toString());
end-method;
```

The onError method in AbstractNotification, if not overridden in the implementing class, logs an error in the App Server logs – "Notification cannot be sent for Notification Consumer XXXXXXXX with Request ID YYYYYY due to following error: ACTUAL ERROR". This behavior can be overridden in the implementing class onError method.

Setting Up SMS Notifications

Access the SMS Notification Setup page (Set Up SACR, System Administration, Utilities, Notifications, SMS Notification Setup).

Image: SMS Notification Setup Page

This example illustrates the fields and controls on the SMS Notification Setup page. You can find definitions for the fields and controls later on this page.

This setup is only needed for consumers using the SMS channel to send notifications.

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 such as 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.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Username, Password, and Sender	Enter the Username, Password, and Sender for the SMS Gateway account used for sending SMS.
SMS Method	Select the SMS method. SMS can be sent either by HTTP URL posting or SOAP method using HTTP protocol.
SMS Gateway URL	Enter the URL of the SMS Gateway provider.
Application Class Path	Enter 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.

Sample of Customized Application Class Path Code

Bundle 46. Documentation-only update.

You can either use a URL or SOAP method to trigger an SMS notification. Depending on your setup, the SCC_COMMON:NOTIFICATION:SMSChannel invokes either a *getURLParams()* or *getSOAPMessage()* method. You can code either of the two:

- *getURLParams()* to return the required string URL
- *getSOAPMessage()* to construct an XML node as required

Note: You must hard code the API ID provided by your third party service provider (Clickatell, in this example) in the application class code. The other parameter values are picked up from the SMS Notifications Setup page.

A sample of customized application class code:

```
import SCC_COMMON:NOTIFICATION:BaseSMSChannel;

class Clickatell extends SCC_COMMON:NOTIFICATION:BaseSMSChannel
method Clickatell();
method getURLParams(&sendTo As string, &messageText As string, &userName As string,⇒
 &password As string, &sender As string, &gatewayURL As string) Returns string;

method getSOAPMessage(&sendTo As string, &messageText As string, &userName As strin⇒
g, &password As string, &sender As string, &XmlNod As XmlNode out);
end-class;

method Clickatell
  %Super = create SCC_COMMON:NOTIFICATION:BaseSMSChannel();
end-method;

method getURLParams
/+ &sendTo as String, +/
/+ &messageText as String, +/
/+ &userName as String, +/
/+ &password as String, +/
/+ &sender as String, +/
/+ &gatewayURL as String +/
/+ Returns String +/
/+ Extends/implements SCC_COMMON:NOTIFICATION:BaseSMSChannel.getURLParams +/

  Local string &URL = &gatewayURL | "?user=" | &userName | "&password=" | &password⇒
 | "&api_id=" | "9999999" | "&to=" | &sendTo | "&text=" | &messageText;
```

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
  Return &URL;

end-method;

method getSOAPMessage

   /+ &sendTo as String, +/
   /+ &messageText as String, +/
   /+ &userName as String, +/
   /+ &password as String, +/
   /+ &sender as String, +/
   /+ &XmlNod as XmlNode out +/
   /+ Extends/implements SCC_COMMON:NOTIFICATION:BaseSMSChannel.getSOAPMessage +/

   Local string &api_id = "9999999";
   Local XmlNode &tempNode1, &tempNode2;
   &tempNode1 = &XmlNod.AddElement("api_id").AddText(&api_id);
   &tempNode1 = &XmlNod.AddElement("user").AddText(&userName);
   &tempNode1 = &XmlNod.AddElement("password").AddText(&password);
   &tempNode1 = &XmlNod.AddElement("text").AddText(&messageText);
   &tempNode1 = &XmlNod.AddElement("to");
   &tempNode2 = &tempNode1.AddElement("item").AddText(&sendTo);

end-method;
```

Setting Up Integration Broker for the SMS Channel

Note: Currently, Integration Broker setup is needed for the SMS and Push channels.

Nodes

Create a new External Node or use an existing Node. Oracles delivers the SCC-NTF_SMS node. Set up the properties of the Node as shown in the screenshots below.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Access Node Definitions PeopleTools, Integration Broker, Integration Setup, Nodes

Image: Node Definitions Page SMS Notifications

This example illustrates the fields and controls on the Node Definitions page for SMS Notifications.

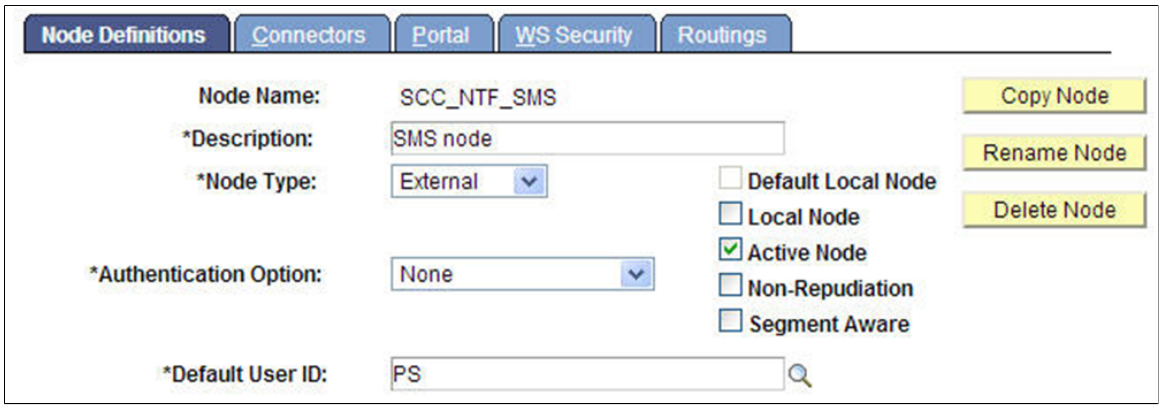


Image: Node Connectors Page SMS Notifications

This example illustrates the fields and controls on the Node Connectors page for SMS Notifications.

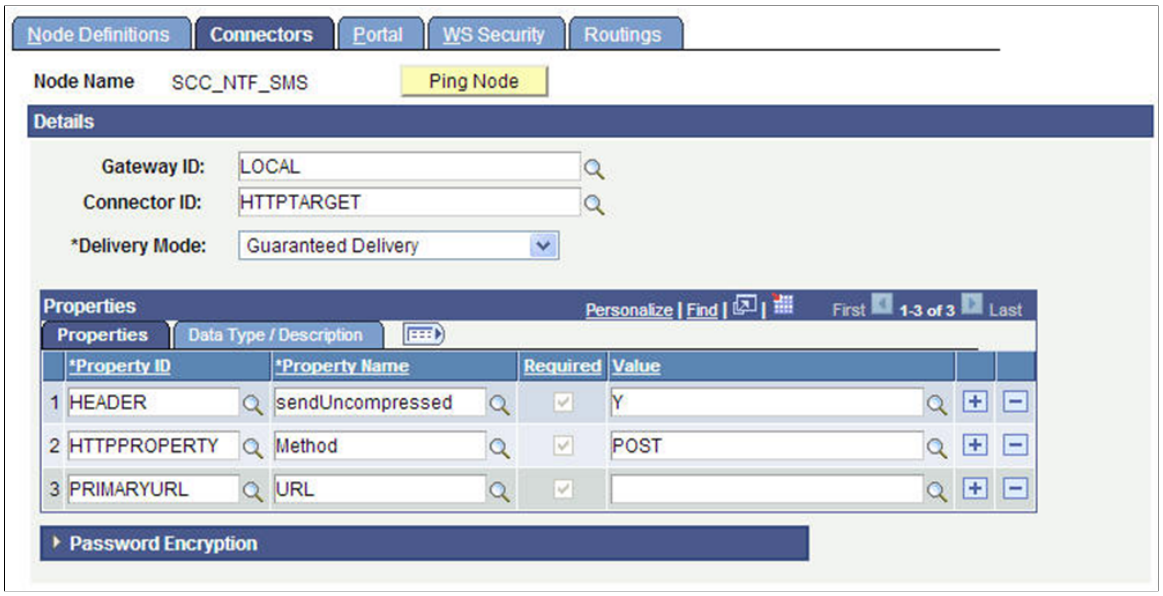
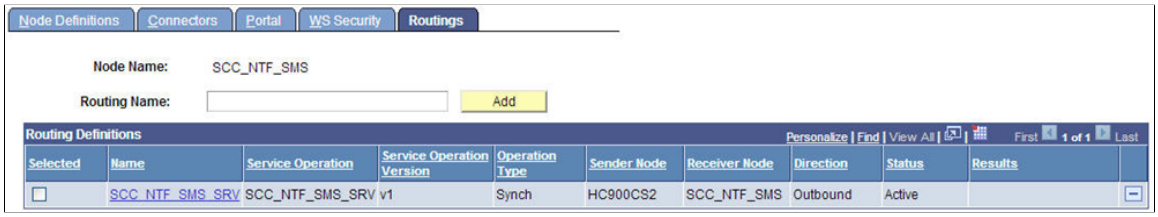


Image: Node Routings Page SMS Notifications

This example illustrates the fields and controls on the Node Routings page for SMS Notifications.



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

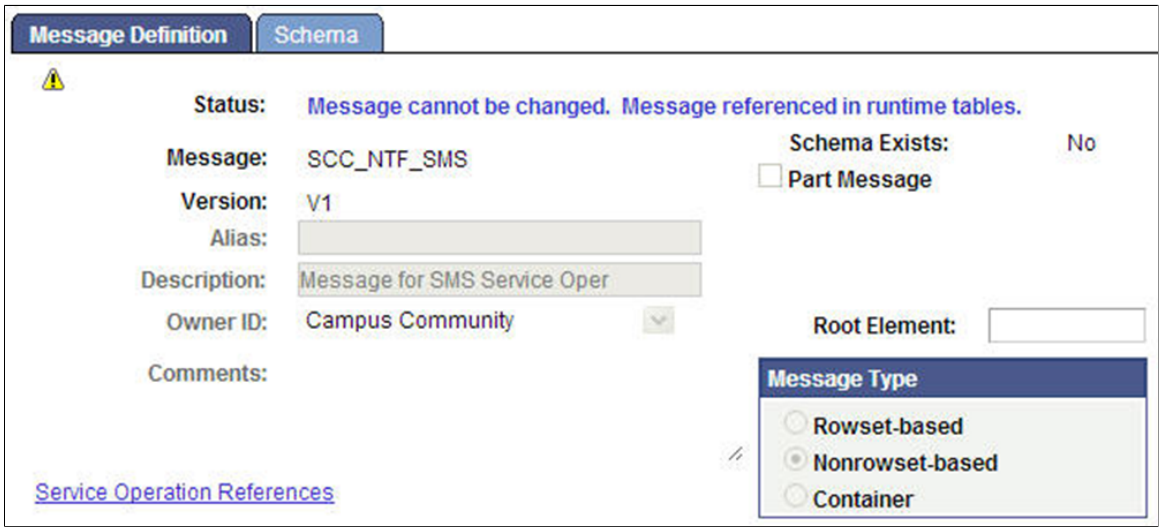
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Message

Create a new Non-Rowset based message as shown below. Access the Message Definition page (PeopleTools, Integration Broker, Integration Setup, Messages).

Image: Message Definition Page SMS Notifications

This example illustrates the fields and controls on the Message Definition page for SMS Notifications.



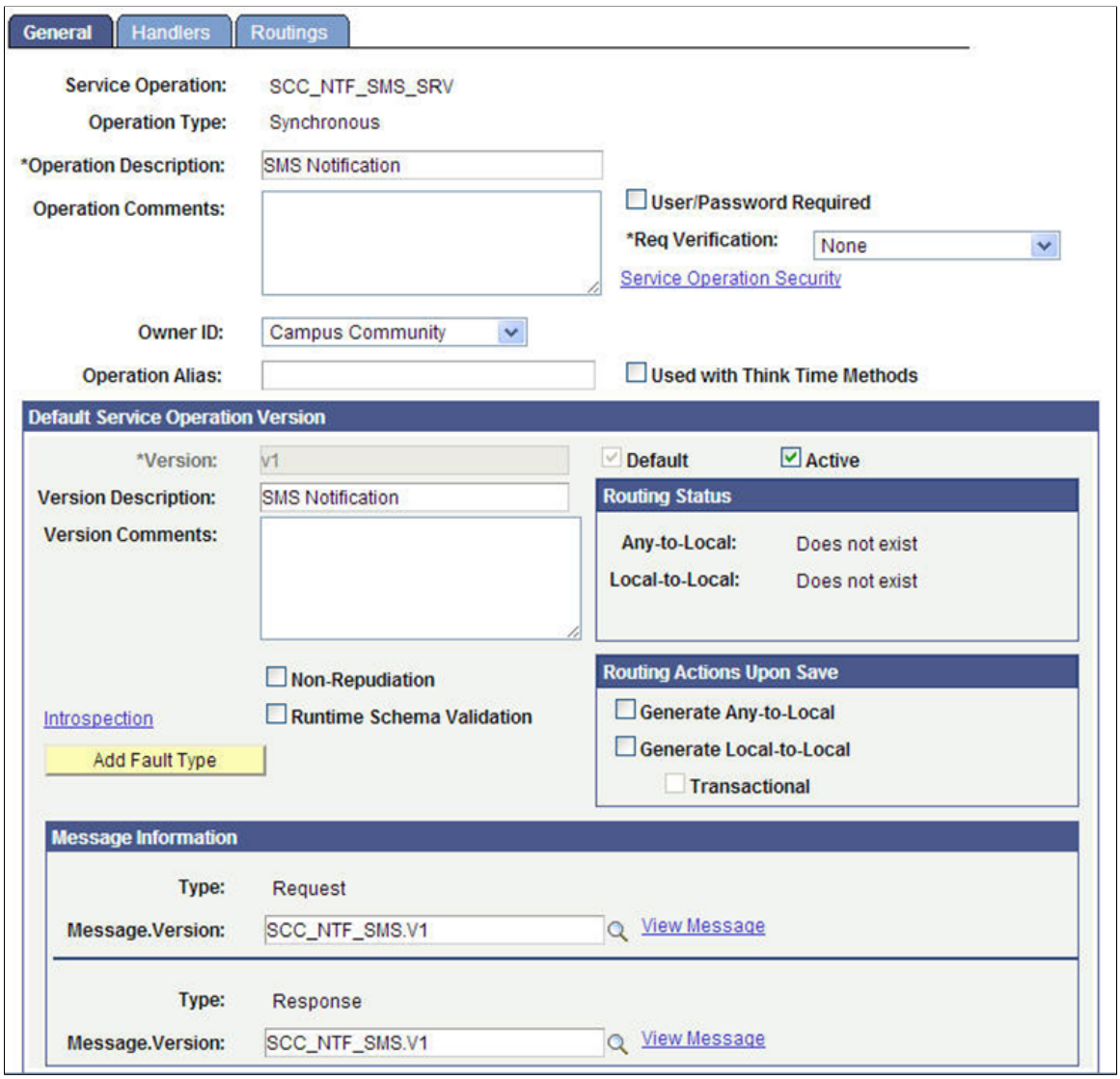
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Service Operation

Create a new Service Operation, as shown below, using the message created above. Access the Service Operation General page (PeopleTools, Integration Broker, Integration Setup, Service Operations).

Image: Service Operation General Page SMS Notifications

This example illustrates the fields and controls on the Service Operation General page for SMS Notifications.

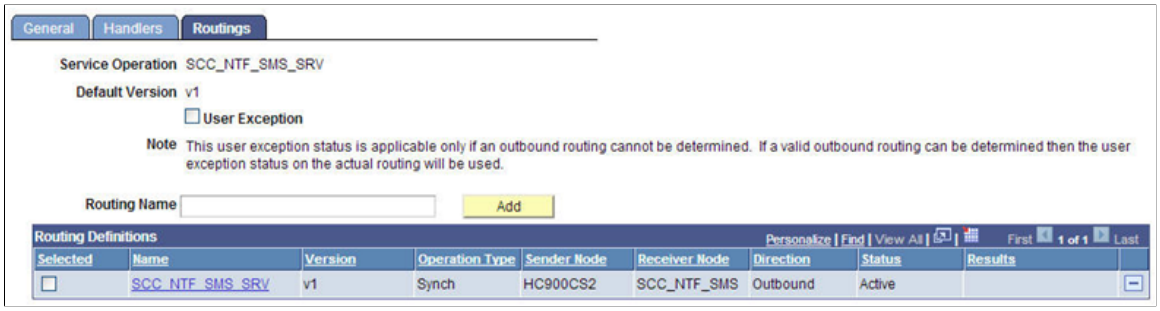


Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

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:

Image: Service Operation Routings Page SMS Notifications

This example illustrates the fields and controls on the Service Operation Routings page for SMS Notifications.



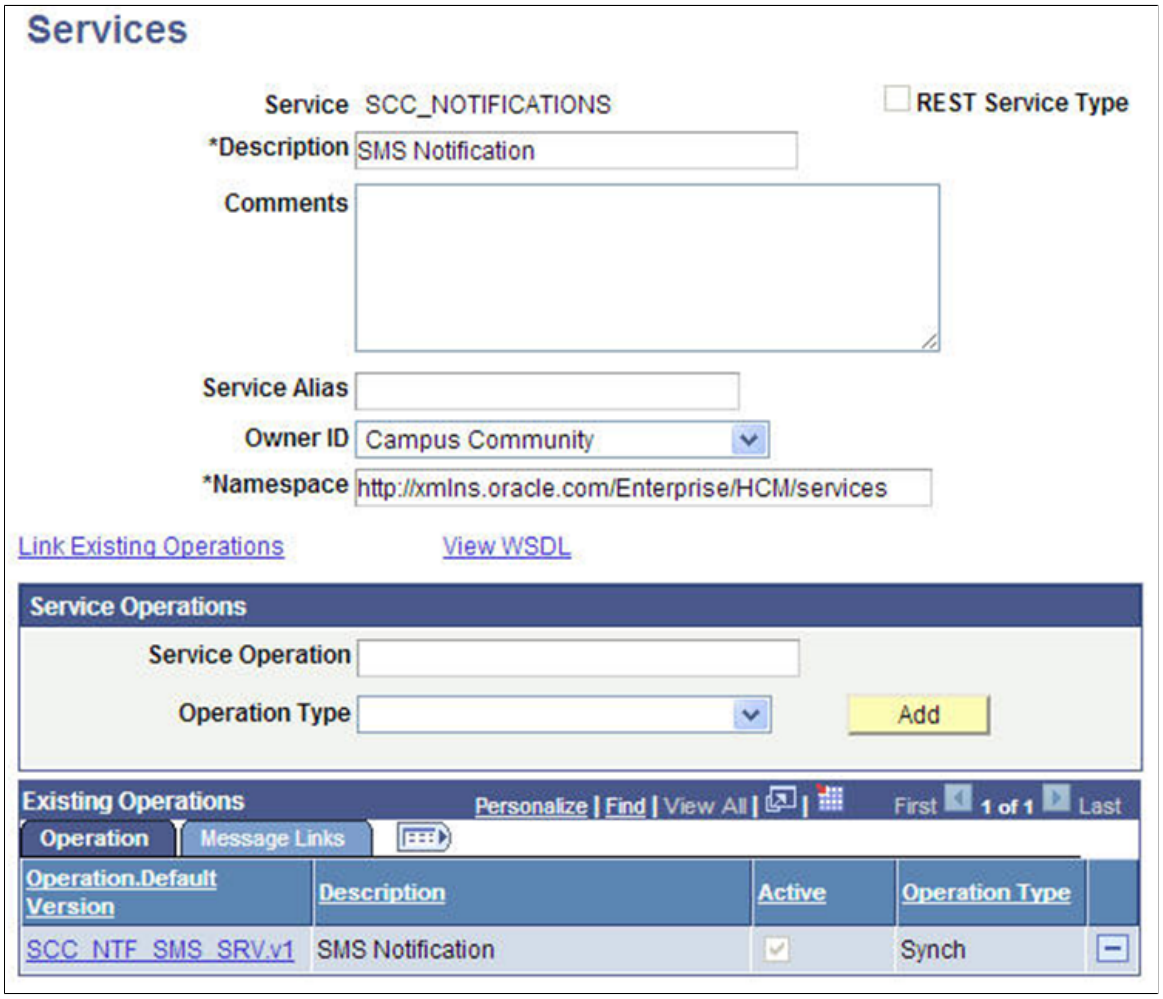
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Service

Create a new Service and add the Service Operation to the Service, as shown below. Access the Services page (PeopleTools, Integration Broker, Integration Setup, Services).

Image: Services Page SMS Notifications

This example illustrates the fields and controls on the Services page for SMS Notifications.



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

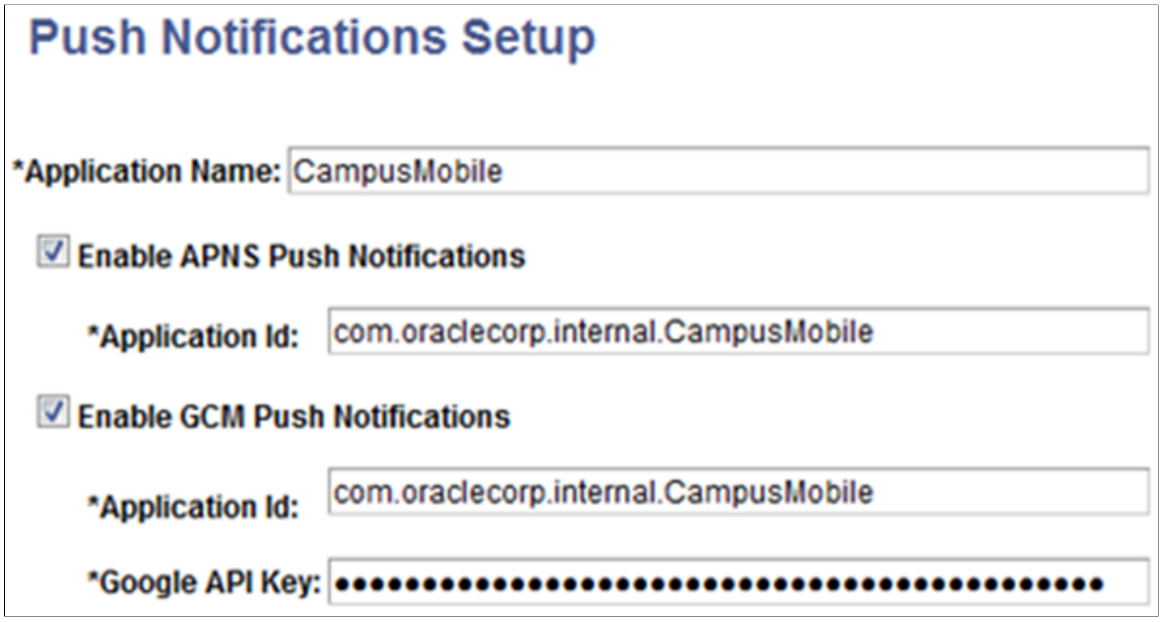
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Setting Up Push Notifications

Access the Push Notifications Setup page (Set Up SACR, System Administration, Utilities, Notifications, Push Notifications Setup).

Image: Push Notifications Setup Page

This example illustrates the fields and controls on the Push Notifications Setup page. You can find definitions for the fields and controls later on this page.



Push notifications are notifications sent from an external source, such as a server, to an application on a mobile device. They may appear as messages in the form of an alert, or as a banner, depending on the state of the application and user settings. When users are notified, they can launch the application, or they can ignore the notification, in which case the application is not activated. Notifications can also accompany an alert message with a brief, distinctive sound.

On iOS, the notification service is Apple Push Notification Service (APNS). Google Cloud Messaging (GCM) for Android provides the notification service for applications installed on Android-powered devices. A set of templates is available through the Campus Solutions Notifications Framework that enables Campus Solutions event handling through a push channel.

Application Name	Enter a unique value across Campus Solutions that identifies the application or module configured to send push notifications.
Enable APNS Push Notifications	Select this check box to enable Apple smartphone users to receive push notifications from Campus Mobile and then enter an Application ID.
Application ID	Enter a globally unique fully qualified Application ID, such as edu.myuniversity.CM. The Application ID can be any string; however it must be globally unique for all applications accessing the APNS. For more information on using the Apple Push Notification service, refer to the iOS Developer Library section “Apple Push Notification Service.”

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Note: For GCM push notifications, the Application ID can be the same or different from the one used for the APNS.

Enable GCM Push Notifications Select this check box to enable Android smartphone users to receive push notifications from Campus Mobile and then enter an Application ID and a Google API Key.

Google API Key Enter a generated key that gives the application server authorized access to Google services. You must obtain this key, typically using the Google API console, in order to use it for Google-based web service interaction.

For information about the Google API process, see: http://developer.android.com/google/gcm/gs.html#create-proj

Configuring Integration Broker to Use Socket Connector for Push Channel Delivered Services

Some Campus Solutions functionality, such as Campus Mobile, uses an SSL Socket connector for Apple Push Notifications. This connector is delivered with Campus Solutions and must be loaded and configured.

To load gateway connectors:

1. Copy the SSLSocketTargetConnector.class and SSLSocketTargetConnector$1.class files from <PS_HOME>\class (for Windows) or <PS_HOME>/appserv/classes (for UNIX) to the following location: <PIA_HOME>\webserv\<DOMAIN>\applications\peoplesoft\PSIGW.war\WEB-INF \classes\com \peoplesoft\pt\integrationgateway\targetconnector.
2. Restart the web server.
3. Access the Gateways page (PeopleTools, Integration Broker, Configuration, Gateways).
4. Enter the URL and click the Load Gateway Connectors button.
5. Verify that Connector ID SSLSOCKETTARGET is loaded and save the page.

To configure gateway properties:

1. Access the Gateways page (PeopleTools, Integration Broker, Configuration, Gateways).
2. Click Gateway Setup Properties.
3. Provide User ID and Password in the Gateway Properties page.
4. Click OK.
5. Click the Advanced Properties Page link on the PeopleSoft Node Configuration page.
6. On the Gateway Properties page, provide values for secureFileKeystorePath and secureFileKeystorePasswd.
7. Click OK.

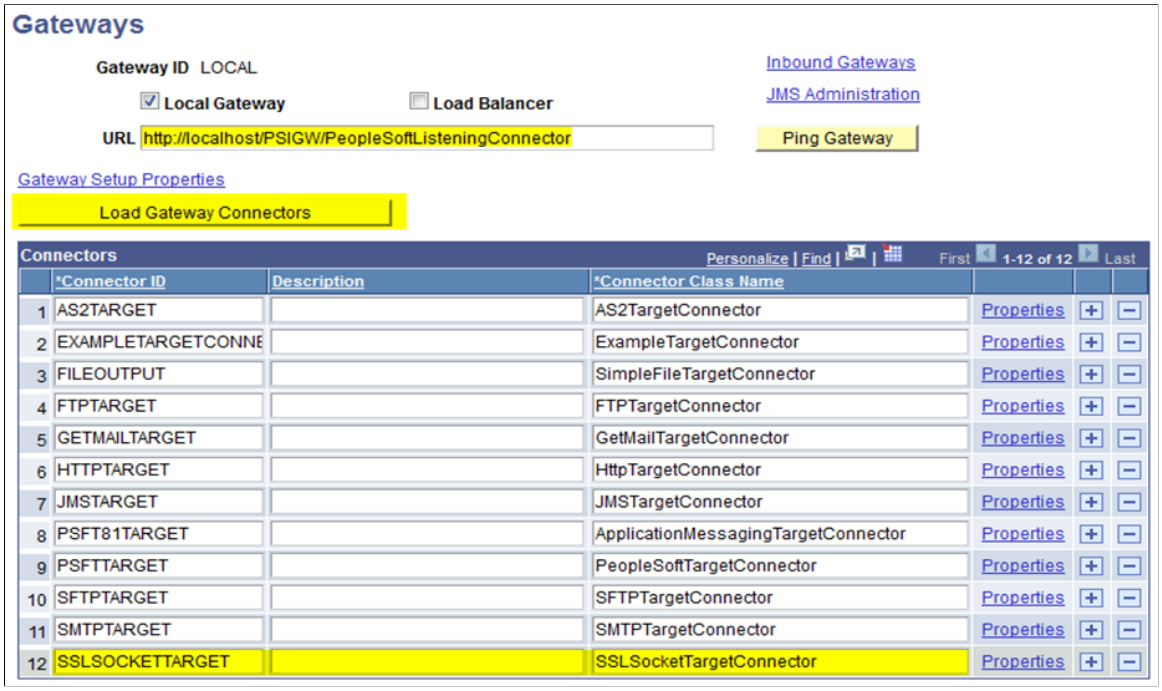
 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

8. If the web server is behind a firewall, then provide values for ig.proxyHost and ig.proxyPort on the Gateway Properties page.

9. Click OK.

Image: Loading Custom Gateway Connector for Push Notifications

This example illustrates the fields and controls on the Loading Custom Gateway Connector for Push Notifications example.



Installing SSL Certificates and Configuring Platform Specific Server Keys

This setup is only needed for consumers using the PUSH channel. When making secured connections to and from the Apple and Google push notification servers, SSL certificates and platform specific server keys need to be in place and configured. Follow the steps below in order to configure SSL Certificates and server keys on Apple Push Notification Server (APNS) and Google Cloud Messaging (GCM) for Android.

For APNS

Install the SSL distribution certificate and private cryptographic key obtained on the server computer, on which the provider code runs and from which it connects with the development or production versions of APNs. To do so, complete the following steps:

1. Open Keychain Access utility and click the My Certificates category in the left pane.

2. Find the certificate that should be installed and disclose its contents. Both a certificate and a private key are displayed.

3. Select both the certificate and key, choose File > Export Items, and export them as a Personal Information Exchange (.p12) file.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

4. Import this .p12 into pskey keystore on the webserver.

See https://developer.apple.com/library/mac/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ProvisioningDevelopment.html#//apple_ref/doc/uid/TP40008194-CH104-SW1.

For GCM

Import root ca certificate accepted by Google into pskey keystore on peoplesoft web server.

Download GCM Certificate (ex: Geo Trust Global CA) from the browser with URL https://android.googleapis.com/.

Activating All Integration Broker Objects Delivered for Push Channel

This setup is needed only for consumers using the PUSH channel. Several IB objects delivered with the Notifications Framework need to be activated when enabling applications to use the Push Channel. The following sections list out the specific service operations, routings, queues and notification handlers which need to be enabled.

Service Operations

When configuring applications which use the Push Channel capability, ensure the following service operations and routings are activated:

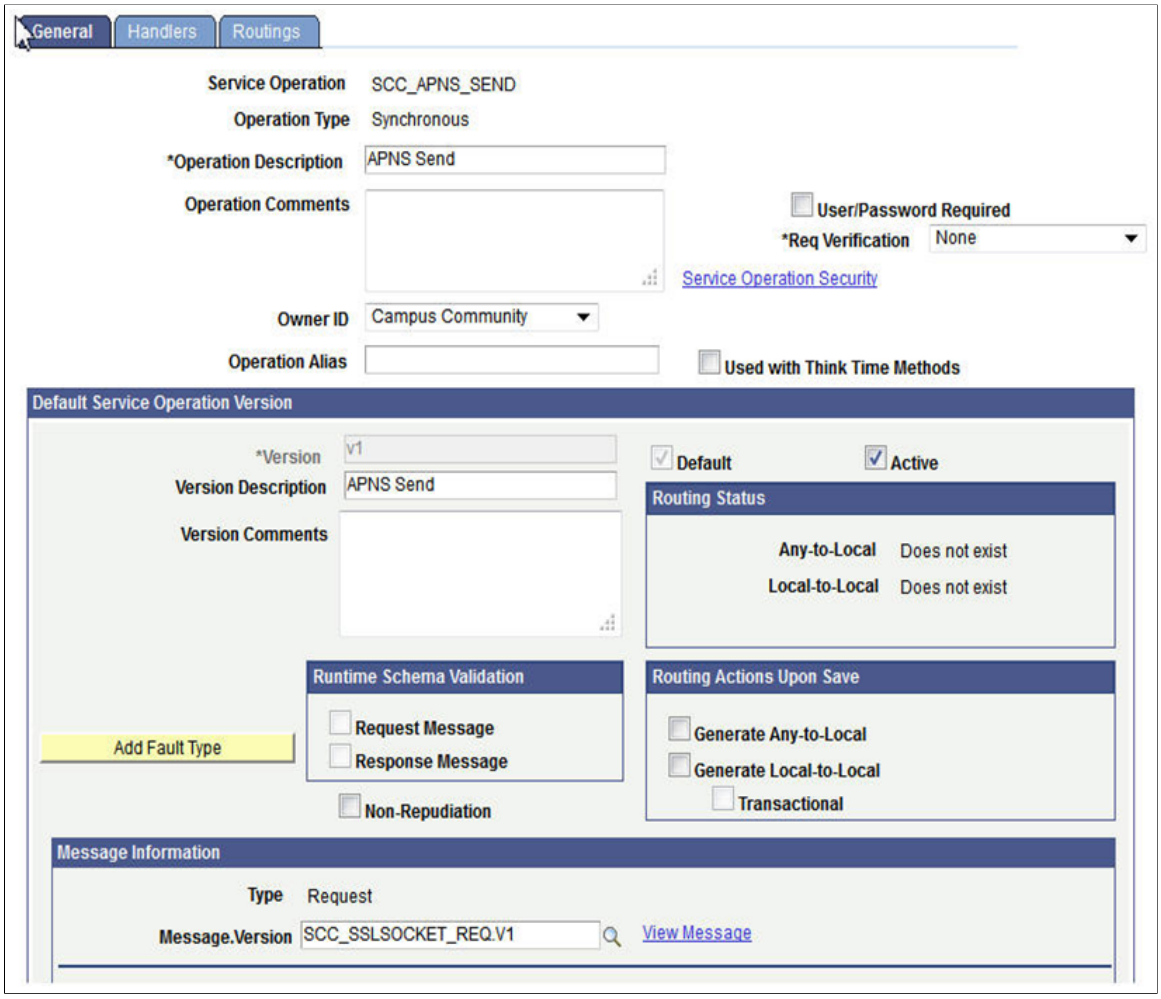
Service	***Service Operation***	***Type***	***Routing***
SCC_NOTIFICATIONS	SCC_PUSH_NOTIFICATIONS	Asynchronous	-
SCC_NOTIFICATIONS	SCC_APNS_FEEDBACK	Synchronous	SCC_APNS_FEEDBACK
SCC_NOTIFICATIONS	SCC_APNS_SEND	Synchronous	SCC_APNS_SEND
SCC_NOTIFICATIONS_CONSUMER	SCC_GCM_SEND_POST	Synchronous	SCC_GCM_SEND_POST
SCC_NOTIFICATIONS_R	SCC_DEVICE_REGISTER_POST	Synchronous	SCC_DEVICE_REGISTER_POST

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

The following screenshot is an example of the SCC_NOTIFICATIONS->SCC_APNS_SEND service operation activation. Access the Service Operation General page (PeopleTools, Integration Broker, Integration Setup, Service Operations).

Image: Service Operation General Page Push Channel

This example illustrates the fields and controls on the Service Operation General page for Push Channel.



To activate the service operation, ensure the Active check box is selected.

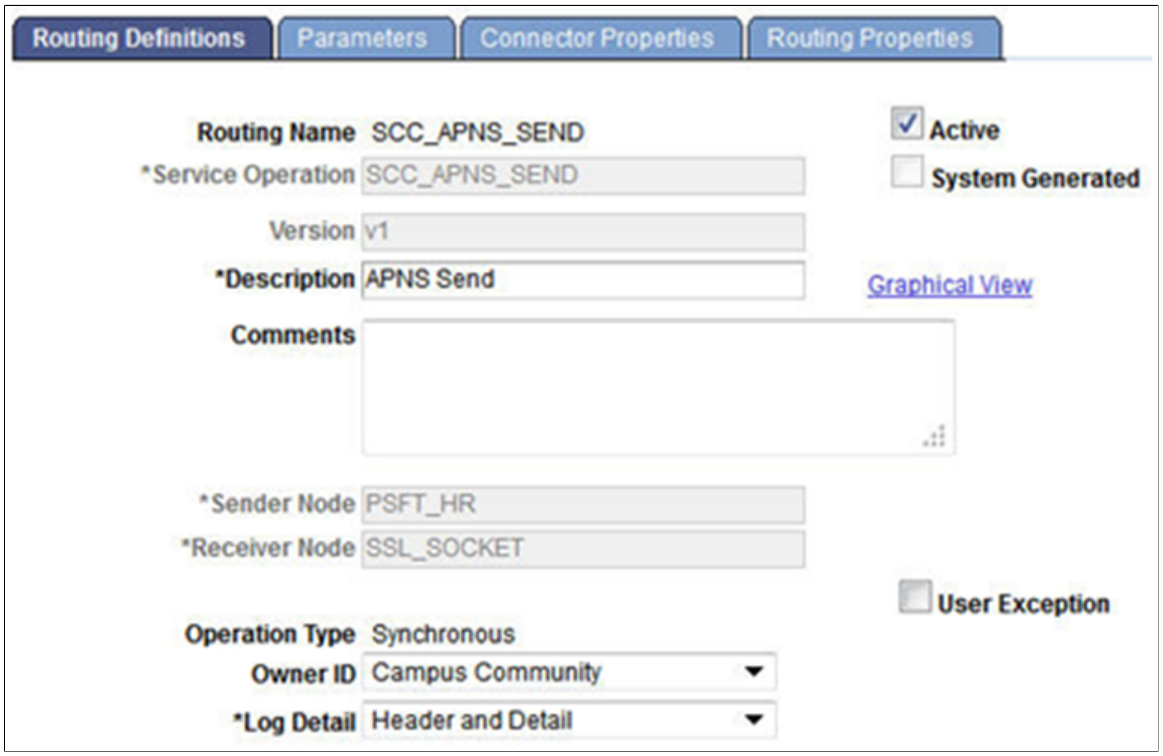
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Routings

The following screenshot is an example of the routings activation. Access the Routing Definitions page (PeopleTools, Integration Broker, Integration Setup, Routings).

Image: Routing Definitions Page

This example illustrates the fields and controls on the Routing Definitions page.



To activate the routing, select the Active check box, save, and continue.

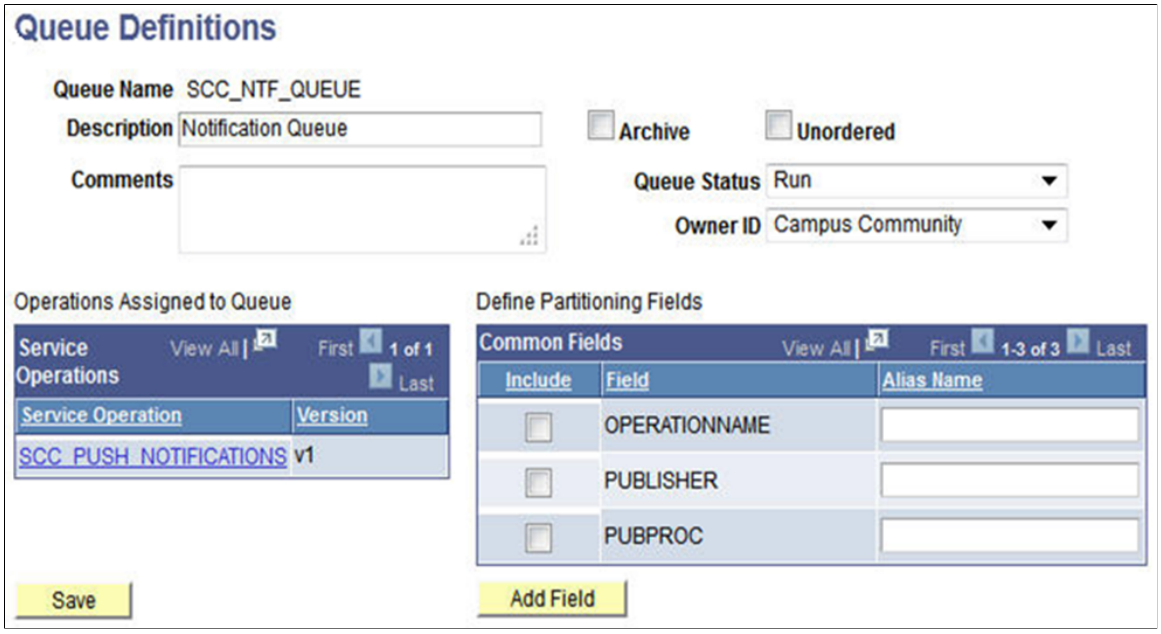
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Queue: SCC_NTF_QUEUE

Access the Routing Definitions page (PeopleTools, Integration Broker, Integration Setup, Queues).

Image: Queue Definitions Page

This example illustrates the fields and controls on the Queue Definitions page.



Ensure the Queue Status is set to Run, save, and continue.

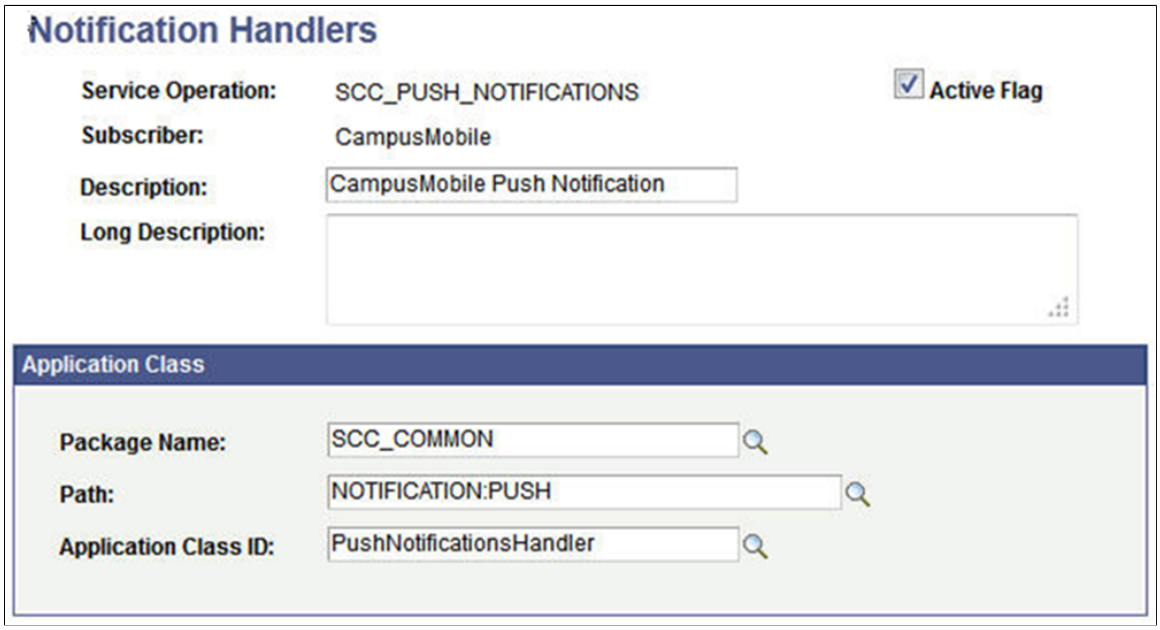
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Notification Handlers

Activate the Notification Handler for SCC_PUSH_NOTIFICATIONS in SOA registry Set Up SACR, System Administration, Integrations, Notification Handlers.

Image: Notification Handlers Page

This example illustrates the fields and controls on the Notification Handlers page.



Scheduling the APNS Feedback Job

This setup is only needed for consumers using the PUSH channel for APNS. APNS has a feedback service that maintains a list of devices for which there were failed-delivery attempts (that is, APNS was unable to deliver a push notification to an application on a device). The Campus Solutions Notifications Framework provides a job, APNSAE, which can be scheduled with a recurrence to periodically connect with the feedback service to see which devices have persistent failures so that it can refrain from sending push notifications to them. It is recommended to schedule the APNSAE job with a weekly recurrence.

Setting Up and Using the Events Framework

Use the Events Framework to track events and send notifications if required. The Events Framework provides the flexibility to track events and to connect to the Notifications Framework to send notifications.

The Events Framework can be used with or without the Notifications Framework.

The Batch Events Framework provides the flexibility to track batch events alone or with batch notifications for multiple consumers with multiple templates.

This topic discusses how to:

1. Configure an event for the Events Framework.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

2. Trigger the Events Framework.

Page Used to Set Up the Events Framework

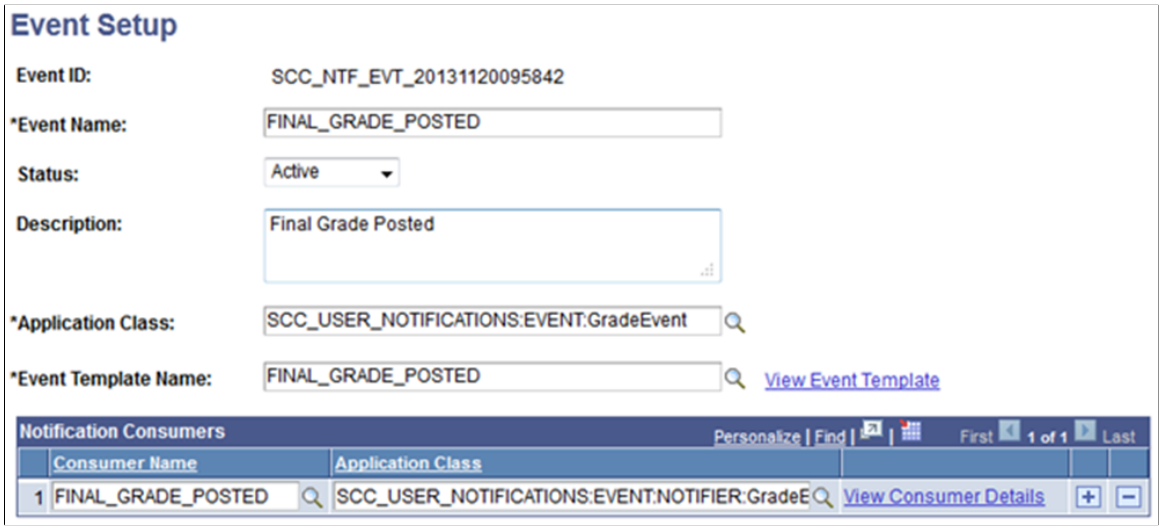
Page Name	*Definition Name*	*Navigation*	*Usage*
Event Setup	SCC_NTFEVT_CFG	Set Up SACR, System Administration, Utilities, Notifications, Event Setup	Configure the event and specify the notification consumers related to the event to send notifications when the event is tracked.

Configuring an Event for the Events Framework

Access the Event Setup page (Set Up SACR, System Administration, Utilities, Notifications, Event Setup).

Image: Event Setup Page

This example illustrates the fields and controls on the Event Setup page. You can find definitions for the fields and controls later on this page.



Event ID Displays a read-only unique ID field which is generated automatically after successful save of component.

Event Name Enter the event name.

Status Select the event status. Only active events can be raised.

Description Enter the event description.

Application Class Select the application class which is extending AbstractEvent class (SCC_COMMON:NOTIFICATION:EVENT:AbstractEvent) which handles the event notifications.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

	Select the SCC_ COMMON:NOTIFICATION:EVENT:BatchEvent app class for batch events.
Event Template Name	Select a Generic template which is configured for event message text and event context data.
View Event Template	Select this link to open the Generic Template Definition page for the selected template.
Consumer Name	Select the Notification consumer for this event to connect the Notifications Framework into the Events Framework. The Notifications Framework is invoked for the selected consumers when the event is raised.
Application Class	Select the application which is extending AbstractEventNotifier (SCC_ COMMON:NOTIFICATION:EVENT:AbstractEventNotifier) application class which handles the connection with the Notifications Framework. Select SCC_ COMMON:NOTIFICATION:EVENT:BatchEventNotifier app class for batch events.
View Consumer Details	Select this link to open the Notification Consumer Setup page for selected consumer.

Triggering the Events Framework

These methods must be used by the Events Framework to raise the event and to invoke the Notifications Framework:

1. Write an Event context App class that extends BaseEventContext.

   Sample Code:

   ```
   class GradeEventContext extends SCC_COMMON:NOTIFICATION:EVENT:BaseEventContext
   ```

2. Write an AppClass that extends AbstractEvent class. The Event ID must be passed to the Super Class constructor here, as shown below.

   Sample Code:

   ```
   %Super = create SCC_COMMON:NOTIFICATION:EVENT:AbstractEvent(&pEvtId);
   ```

   a. Implement the abstract method populateEventContext(). This method populates the event context with necessary information for further processing.

      Sample Code:

      ```
      method populateEventContext
         /+ &p_eventContext as SCC_COMMON:NOTIFICATION:EVENT:IEventContext +/
         /+ Extends/implements SCC_COMMON:NOTIFICATION:EVENT:AbstractEvent.popu⇒
      lateEventContext +/
         Local SCC_USER_NOTIFICATIONS:EVENT:CONTEXT:GradeEventContext &grdConte⇒
      xt = &p_eventContext As SCC_USER_NOTIFICATIONS:EVENT:CONTEXT:GradeEventCo⇒
      ```

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
ntext;
   Local Record &recStdEnrl = &grdContext.recPayload;
   Local SCC_USER_NOTIFICATIONS:UTIL:UserNotificationUtil &ntfUtil = crea⇒
te SCC_USER_NOTIFICATIONS:UTIL:UserNotificationUtil();

   &grdContext.getEventParameterMap().Put("%1", &ntfUtil.getClassDescript⇒
or(&recStdEnrl));

   &grdContext.getEventDataMap().Put("EVENT_TYPE", "FINAL_GRADE_POSTED");
   &grdContext.getEventDataMap().Put("INST", String(&recStdEnrl.INSTITUTI⇒
ON.Value));
   &grdContext.getEventDataMap().Put("CAREER", String(&recStdEnrl.ACAD_CA⇒
REER.Value));
   &grdContext.getEventDataMap().Put("TERM", String(&recStdEnrl.STRM.Valu⇒
e));
   &grdContext.getEventDataMap().Put("GOTO", "Grades");

end-method;
```

b. Override the onSucess(), onError() for custom behavior.

3. Raise the event by invoking the EventHandler.raiseEvent() by passing the event ID and event context or by passing the asynchronous service operation and rowset based on need.

   Sample Code:

```
Local SCC_COMMON:NOTIFICATION:EVENT:EventHandler &evtHndlr = create SCC_COMMON⇒
:NOTIFICATION:EVENT:EventHandler();

&evtHndlr.raiseEvent("SCC_NTF_EVT_20131120095842", &grdCntxt);

Or

&evtHndlr.publishMessage(OPERATION.SCC_NTF_PUBLISH_STDNT_ENRL, &rs_stdnt_enrl,⇒
 "");
```

4. Write an Event Notifier App class that extends AbstractEventNotifier for each notification consumer for an event if notifications should be sent along with event notifications. The Consumer ID must be passed to the Super Class constructor here, as shown below.

   Sample Code:

```
%Super = create SCC_COMMON:NOTIFICATION:EVENT:AbstractEventNotifier(&pNtfConsI⇒
D);
```

   Implement the raiseEventNotification() abstract method to send consumer notifications by invoking by existing notification consumer app class which extends AbstractNotification.

   Sample Code:

```
method raiseEventNotification
   /+ &pEventContext as SCC_COMMON:NOTIFICATION:EVENT:IEventContext +/
   /+ Returns Number +/
   /+ Extends/implements SCC_COMMON:NOTIFICATION:EVENT:AbstractEventNotifier.r⇒
aiseEventNotification +/
   Local SCC_USER_NOTIFICATIONS:EVENT:CONTEXT:GradeEventContext &grdContext = ⇒
&pEventContext As SCC_USER_NOTIFICATIONS:EVENT:CONTEXT:GradeEventContext;
   Local Record &p_EnrlRec = &grdContext.recPayload;
   Local string &ntfMode = &grdContext.ntfMode;

   Local SCC_USER_NOTIFICATIONS:NOTIFICATION:GradeNotifier &grdNtf = create SC⇒
C_USER_NOTIFICATIONS:NOTIFICATION:GradeNotifier(%This.ntfConsID);
   &grdNtf.sendGrade(&p_EnrlRec.EMPLID.Value, &p_EnrlRec, &ntfMode);
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
   Return %This.getNotificationReqID(&grdNtf);
end-method;
```

Sending Batch Notifications and Events

The Batch Notifications run control page can be used to send notifications, events, or both in batch mode. This run control uses the Population Selection feature to select the EMPLIDs of the recipients who receive the notifications. Currently it can be used to send either events with or without batch notifications or batch notifications only with appropriate configuration. When the Enable Event Tracking check box is selected, it can be used to send events or events plus notifications to the consumers configured from the selected event. When Enable Event Tracking is not selected it can be used to send batch notifications for multiple consumers. Currently it can be used to send Email, SMS, Alert and Push notifications. It cannot send Worklist notifications, nor can it send notifications if the template recipient is configured to “Custom Logic”.

This topic discusses how to:

1. Manage batch notifications and events.
2. Use the Worklist Batch Process.

Page Used to Send Batch Notifications and Events

Page Name	*Definition Name*	*Navigation*	*Usage*
Batch Notifications	SCC_BAT_NTF_RC	Set Up SACR, System Administration, Utilities, Notifications, Batch Notifications	Send notifications, events, or both in batch mode.

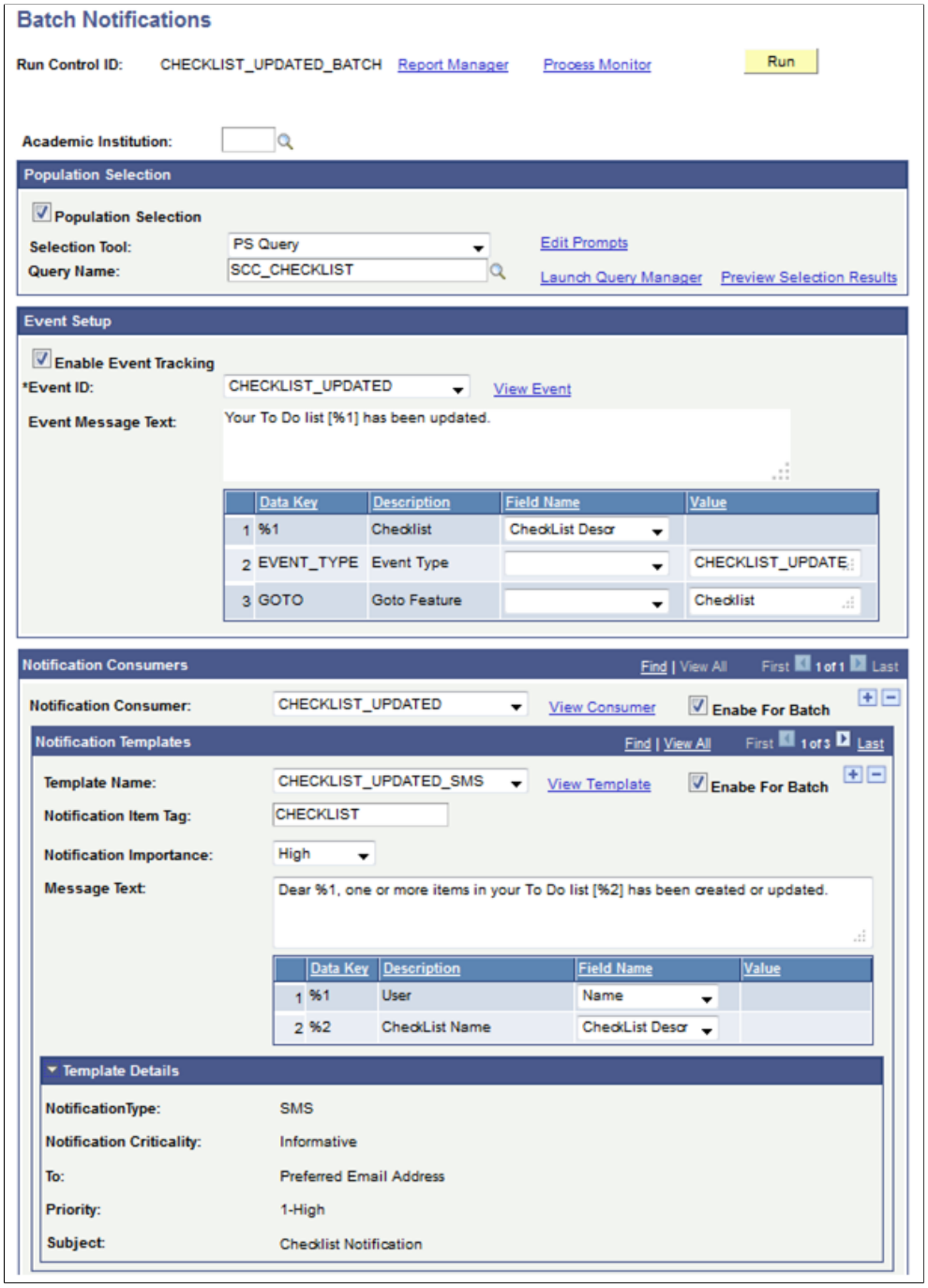
 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Managing Batch Notifications and Events

Access the Batch Notifications page (Set Up SACR, System Administration, Utilities, Notifications, Batch Notifications).

Image: Batch Notifications Page

This example illustrates the fields and controls on the Batch Notifications page. You can find definitions for the fields and controls later on this page.



Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Population Selection

Population Selection Select this check box to enable the Selection Tool. This run control does not support manual selection.

Selection Tool Select the Population Selection tool available for this component.

Event Setup

Enable Event Tracking Select this check box when event tracking is needed. If it is selected, event configuration setup is available to configure it and send events. If it is not selected, only notification consumer and templates configuration are available to send notifications.

Event ID Select an Event ID from the list of the Notifications Framework events.

View Event Select this link to access the Event Setup page for selected event ID.

Event Message Text Displays a read-only field populated from the selected event template configured in event setup.

Event Bind variables This grid is populated based on the event selected and the population query selected.

- Data Key: Displays the bind variable key from the generic template.
- Description: Displays the bind variable description from the generic template.
- Field Name: This is populated with the list of fields selected in the PS Query. Select 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: Enter a static value in this field for use as a bind value for the bind variable in the template.

Notification Consumers

Notification Consumer Select a consumer from the list of all active Notifications Framework consumers/Event framework consumers based on ‘Enable Event Tracking’ check box selection. If the ‘Enable Event Tracking’ check box is selected the list is populated with active consumers of selected event; otherwise the list is populated with all active notification framework consumers.

View Consumer Click this link to access the Notification Consumer Setup page.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Enable for Batch Select this check box to enable the flexibility to mark the consumer as enable/disable for the current batch process rather than deleting and configuring again.

Notification Templates

Template Name Select a template from the list of all active templates for the selected consumer. Currently, only templates of types Email, SMS, Alert & Push and those not set to use “Custom Logic” in recipient configuration are available.

View Template Click this link to access the Notification Template setup page.

Enable for Batch Select this check box to enable the flexibility to mark the template as enable/disable for current batch process rather than deleting and configuring again.

Notification Item Tag Enter 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 Select Notification Importance. By default this importance is populated with the generic template priority field.You can override this value.

Message Text This is a read-only field populated from the selected notification template.

This grid is populated based on the template name selected and the population query selected.

- Data Key: Displays the bind variable key from the generic template.
- Description: Displays the bind variable description from the generic template.
- Field Name: This drop-down list box is populated with the list of fields selected in the PS Query. Select 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: Enter a static value in this field for use as a bind value for the bind variable in the template.

Template Details

These fields are read-only fields populated from the selected notification template.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

PS Query Creation Steps for Batch Notifications

PS Query should be created using the following steps to support dynamic queries in terms of selected fields and consumers can create their own queries.

Population Selection Context definition Name: Batch Notifications

Results record: SCC_BAT_NTF_TGT

Image: SCC_BAT_NTF_TGT Record Structure

This example illustrates the SCC_BAT_NTF_TGT Record Structure.

Record Fields | Record Type

Nu	Field Name	Type	Len	Format	Short Name	Long Name
1	PROCESS_INSTANCE	Nbr	10		Instance	Process Instance
2	SCC_SEQ_ID	Nbr	10		ID	Sequence ID
3	EMPLID	Char	11	Upper	ID	Empl ID
4	NUMBER2	Nbr	2		No	Number
5	NUMBER10	Nbr	10		No	Number
6	NUMBER20	Nbr	20		No	Number
7	DATETIME1	DtTm	26	Scnds	DateTime	DateTime
8	DATETIME2	DtTm	26		DateTime	DateTime
9	DATETIME3	DtTm	26	McroS	DateTime	DateTime
10	DATE1	Date	10		Date 1	Date 1
11	DATE2	Date	10		Date 2	Date 2
12	DATE3	Date	10		Date 3	Date 3
13	DESCR5_1	Char	5	Mixed	Descr	Description
14	DESCR5_2	Char	5	Mixed	Descr	Description
15	DESCR5_3	Char	5	Mixed	Descr	Description
16	DESCR10	Char	10	Mixed	Descr	Description
17	DESCR20	Char	20	Mixed	Descr	Description
18	DESCR30	Char	30	Mixed	Description	Description
19	DESCR50	Char	50	Mixed	Descr	Description
20	DESCR100	Char	100	Mixed	Descriptio	Description
21	DESCR35	Char	35	Mixed	Description	Description
22	DESCR35_1	Char	35	Mixed	Description	Description
23	DESCR50_1	Char	50	Mixed	Descr	Description
24	DESCR100_2	Char	100	Upper	Descr	Description
25	DESCRLONG	Long	0		Descr	Description

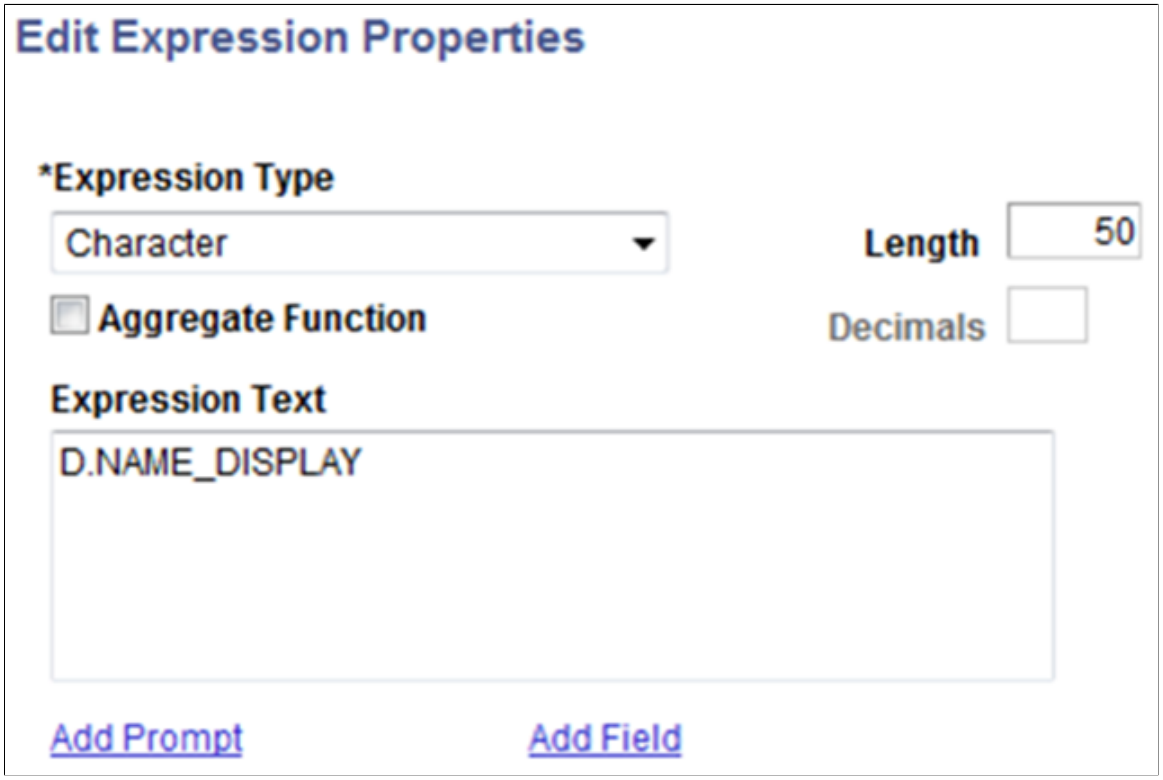
Usually Population Selection works with a predefined set of columns (field) list for selection. This predefined fields list is used in PS Query. To make the PS Query dynamic in terms of column list selection, Batch Notifications Run Control uses an approach to create PS Query with ‘Expression’ and use this expression as Field with a Unique name mapped to one of the appropriate fields in SCC_BAT_NTF_TGT.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

For example, if a field ‘D.NAME_DISPLAY’ needs to be added to the select list (example from a delivered Query ‘SCC_CHECKLIST’), then create an expression as follows:

Image: Edit Expression Properties Page

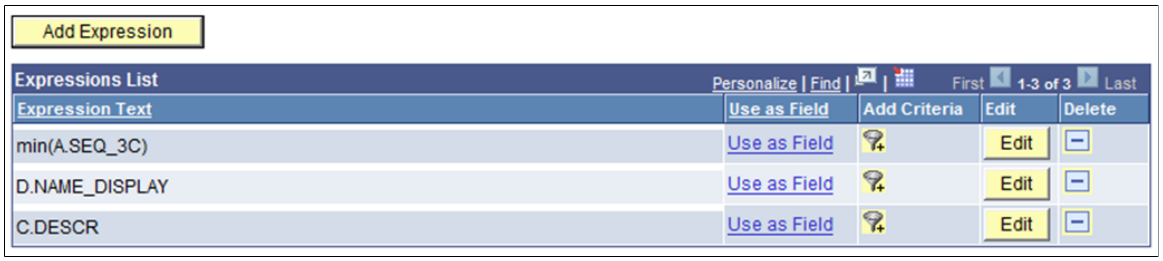
This example illustrates the fields and controls on the Edit Expression Properties page.



Use the above Expression as field by clicking the Use as Fieldlink:

Image: PSQuery Expression Use as Field Link Example

This example illustrates the fields and controls in the PSQuery Expression “Use as Field” Link Example.

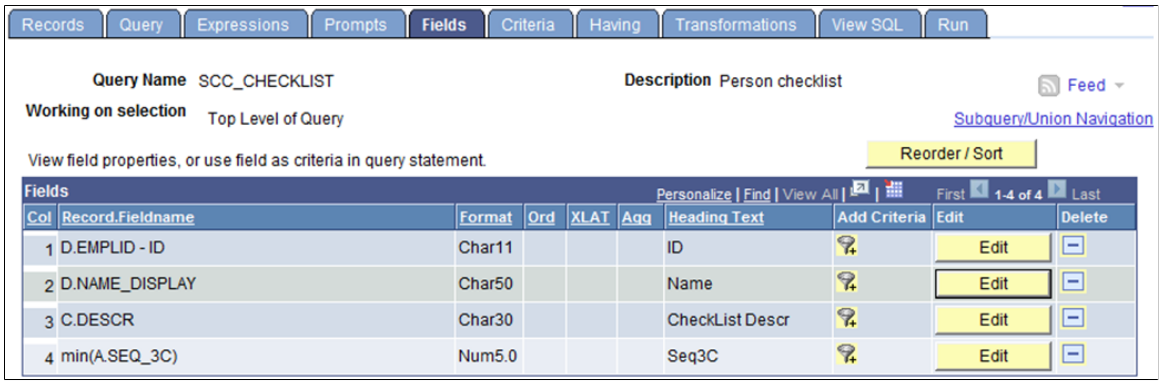


Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Click the Edit button for the field D.NAME_DISPLAY in the Fields tab of the Query manager:

Image: PSQuery Fields Page

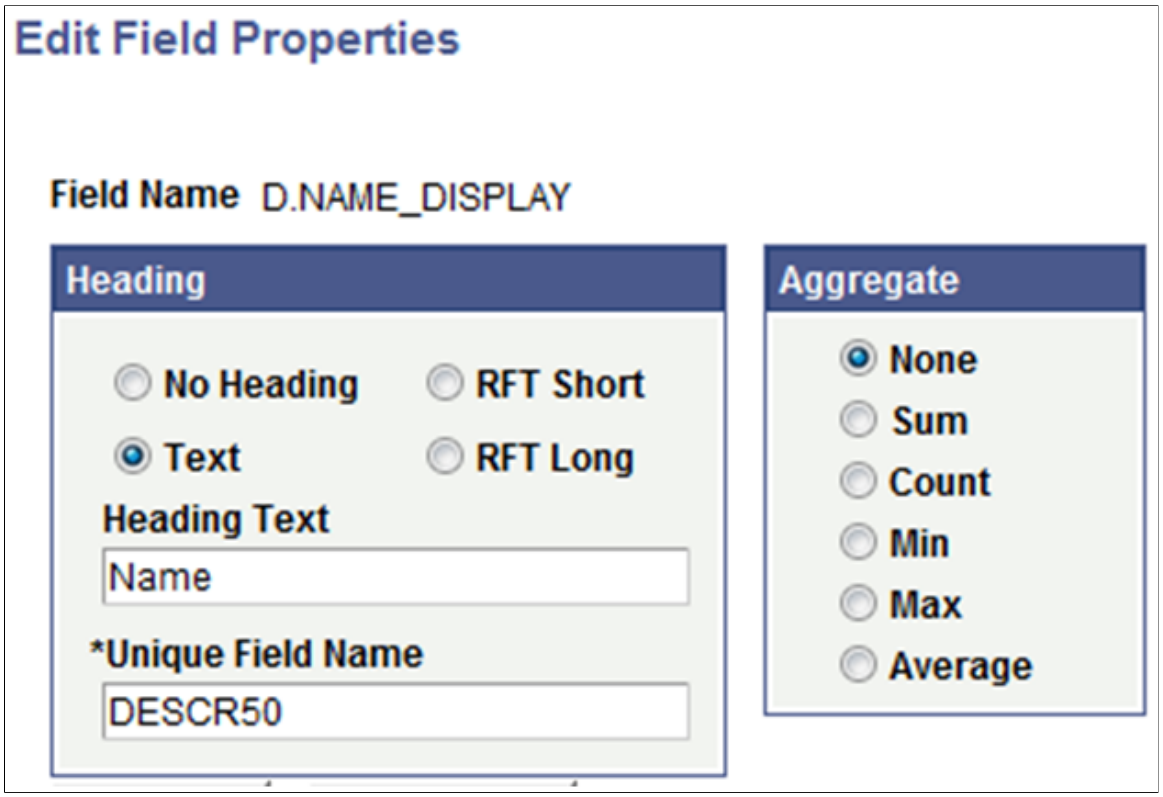
This example illustrates the fields and controls on the PSQuery Fields page.



Enter the appropriate Heading Text and map the Unique Field Name with one of the appropriate available fields in the results record (SCC_BAT_NTF_TGT) - in this example - DESCR50.

Image: PSQuery Edit Field Properties Page

This example illustrates the fields and controls on the PSQuery Edit Field Properties page.



While executing such PS Queries using Population Selection, the result data is inserted in to the results record with the fields mapping. In this example, D.NAME_DISPLAY is inserted in DESCR50 and it is also used in the Batch Notifications process.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Using the Worklist Batch Process

The Worklist Batch Process (SCC_NTFWKLTO) handles Timeout and Reminder processing for worklist notifications. The process selects worklist items and then invokes the callback application class defined in the notification template setup.

The process consists of two steps:

1. Selects Worklist items that have timed out – not worked/completed and the due date has passed. For each item selected, the status is set to Timeout.
2. Selects Worklist items that may need a reminder notification to be sent – not worked/completed and the due Date has not passed.

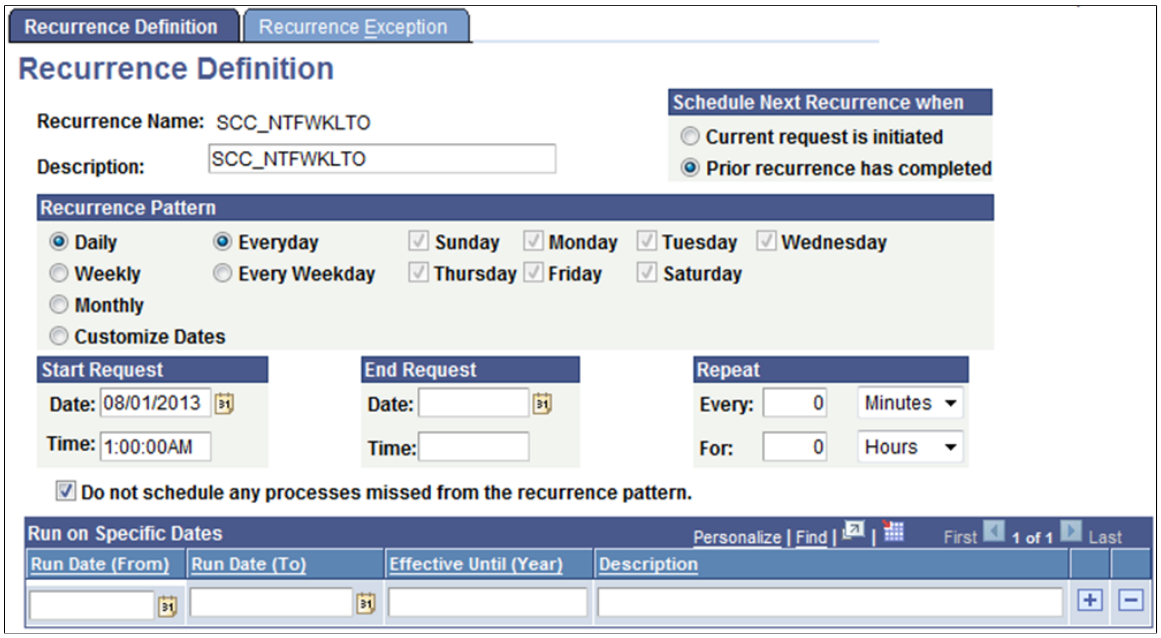
Setting up a Recurrence for the Worklist Batch Process

The process does not have a Run Control component. The process should be run after midnight to avoid any timing issues (date changes). A recurrence (PeopleTools, Process Scheduler, Recurrences) must be created and then assigned to the Process Definition for SCC_NTFWKLTO.

Be sure to select the check box for “Do not schedule any processes missed from the recurrence pattern”.

Image: Recurrence Definition Page

This example illustrates the fields and controls on the Recurrence Definition page.

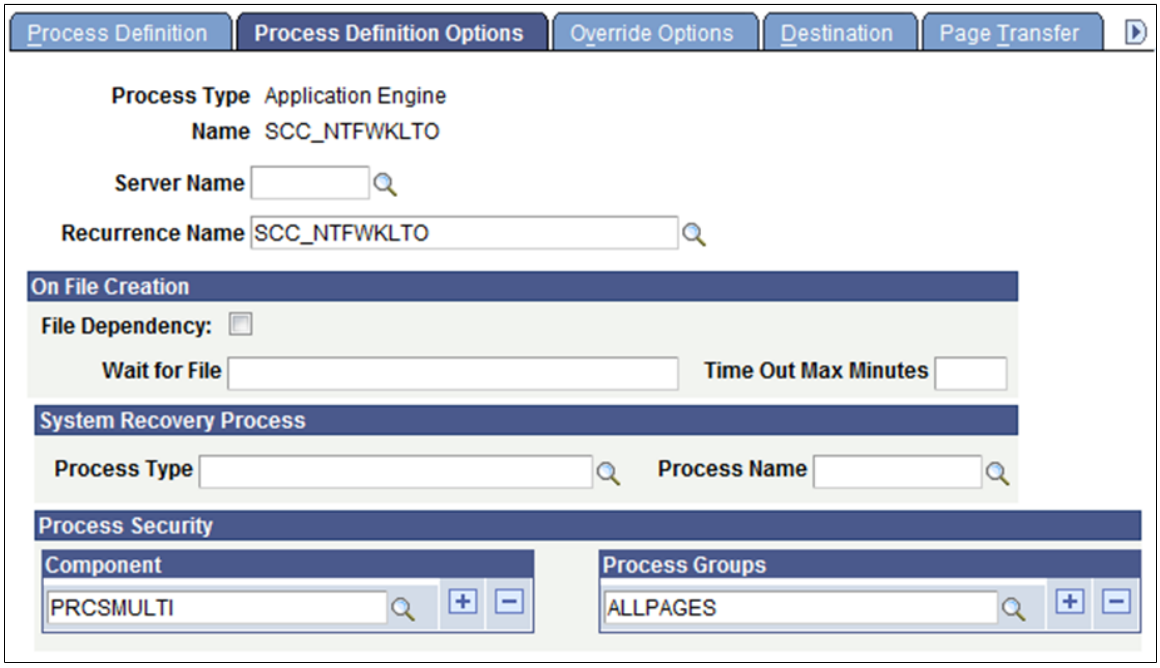


Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Once the Recurrence has been created, it can be assigned to the SCC_NTFWKLTO process definition (PeopleTools, Process Scheduler, Processes).

Image: Process Definitions Options Page

This example illustrates the fields and controls on the Process Definitions Options page.



Managing Notification Preferences

Self-service users can set and update their preferences for how they receive notifications using the Notifications Preferences page in Self Service.

See "Managing Notification Preferences in Self Service" (PeopleSoft Campus Solutions 9.2: Self Service).

Administrators can also manage notification preferences for users

Note: These preferences are ignored if the Override Notification Preferences check box has been selected on the Notification Setup page. This capability is offered for situations where a recipient must be sent a notification even if they have not set their preferences, and particularly relates to Delegated Access/New User Registration.

See Configuring Notification Setup for the Generic Templates.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Page Used to Manage Notification Preferences

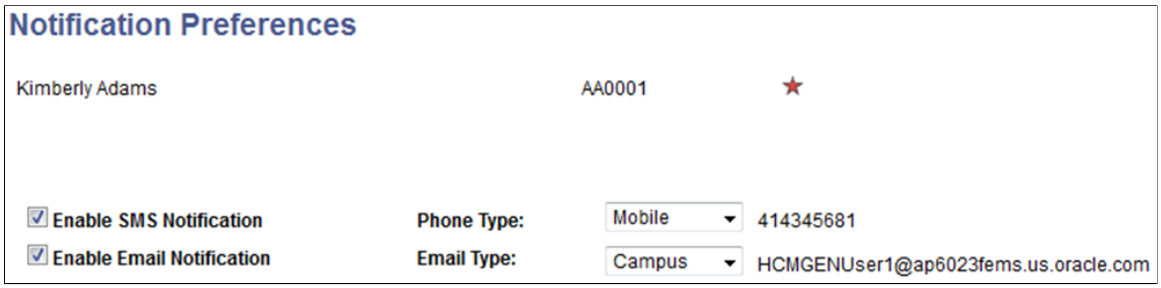
Page Name	*Definition Name*	*Navigation*	*Usage*
Notification Preferences	SCC_NTF_PREF	Campus Community, Personal Information, Biographical, Personal Attributes, Notification Preferences	Set and update users' notification preferences.

Setting and Updating Notification Preferences

Access the Notification Preferences page (Campus Community, Personal Information, Biographical, Personal Attributes, Notification Preferences).

Image: Notification Preferences Page Administrative

This example illustrates the fields and controls on the Notification Preferences Page (Administrative).



Administrators can use this page to set and update users' notification preferences. Users must have at least one email or SMS type for a preference to be set.

See Managing Personal Attributes Information.

See "Using Self-Service Personal Attributes Data" (PeopleSoft Campus Solutions 9.2: Self Service).

Setting Up Security for Admin Notifications

The Admin Notification pages are similar to the self-service Notifications Center pages.

See "Using the Self-Service Notifications Center" (PeopleSoft Campus Solutions 9.2: Self Service).

However, unlike the self-service pages, the admin pages include the following features:

1. They show notifications for many notification recipients.
2. Subject to security settings, announcements can be reviewed, generated and deleted from the Announcements tab.
3. Subject to security settings, Worklist items can be updated, created and deleted.

While the self-service pages are secured by user ID, the admin notifications pages are secured at the notification consumer, then user ID levels.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Before an administrator can see data in the admin pages, the administrator must have particular permission in addition to the usual page permissions. This security operates at two levels:

- Level 1: Notification Consumer security Before notifications belonging to a particular consumer can be seen in PIA, the consumer must be included and assigned an access code in the Campus Community Installation component. This level operates like a “master switch” and enables the consumer to control whether their notifications are visible in the notifications admin pages, and if so whether read only or read/update.
- Level 2: User ID security After a consumer’s notifications have been globally enabled, they can be further refined at the user ID level. For example, a Delegated Access consumer may be set as Read/Update at the installation level, but not all users may be granted that level of access and may only require Read Only access.

This topic discusses how to:

- Set up notification consumer admin access.
- Set up notification consumer security.

Pages Used to Set Up Security for Admin Notifications

Page Name	***Definition Name***	***Navigation***	***Usage***
Notification Consumer Admin	SCC_NTF_INSTCON	Set Up SACR, Install, Campus Community Installation, Notification Consumer Admin	Configure notification consumer admin access.
Notification Consumer Security	SCC_NTF_OPR_CON	Set Up SACR, Security, Secure Student Administration, User ID, Notification Consumer Security	Set access levels to consumers at operator level.

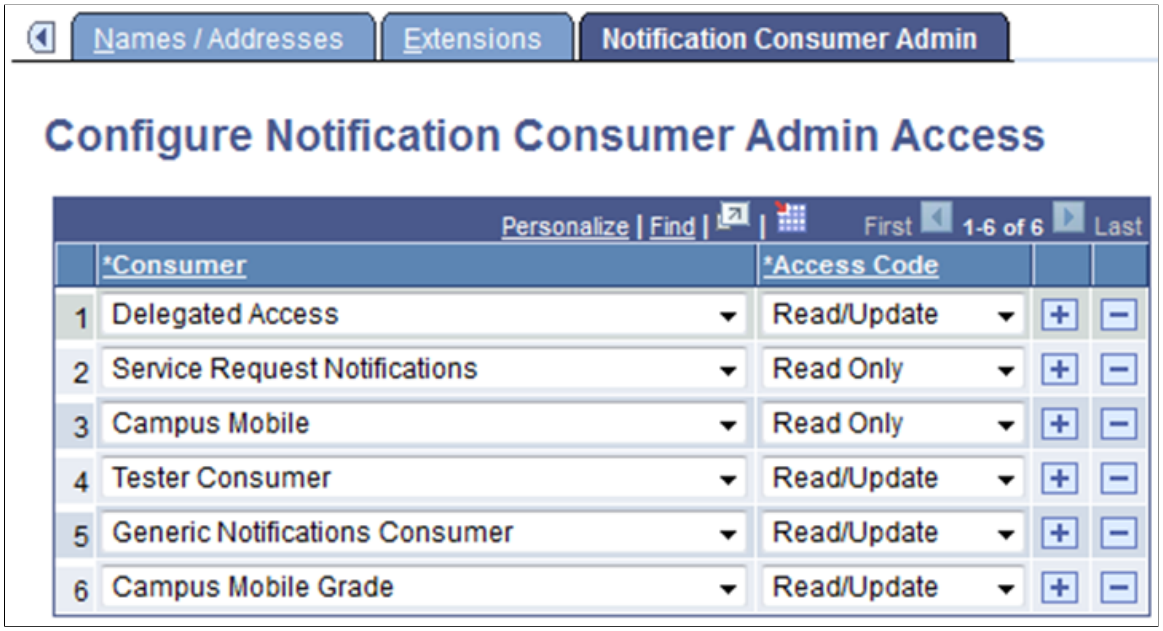
 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Setting Up Notification Consumer Admin Access

Access the Notification Consumer Admin page (Set Up SACR, Install, Campus Community Installation, Notification Consumer Admin).

Image: Notification Consumer Admin Page

This example illustrates the fields and controls on the Notification Consumer Admin page. You can find definitions for the fields and controls later on this page.



This is level 1 security setup.

Consumer

Select a Notifications Framework Consumer with which various notifications templates and their related processing code are associated.

Access Code

Select an Access Code to establish whether a user can only view the notification or can also update it.

Note: The extent to which a notification can be updated is limited,and impacts only Worklist Items (indicating if completed) and Announcements (creation and deletion). For the other notification types, Read/Update is effectively the same as Read Only.

When level 2 security exists for a particular consumer for at least one user, you cannot change the level 1 access setup for that particular consumer. If you try to change, you receive an error message: *Cannot change the setup*.

When level 2 security exists for a particular consumer for at least one user, you cannot delete level 1 access setup for that consumer. If you try to delete, you receive an error message: *Cannot delete this row*.

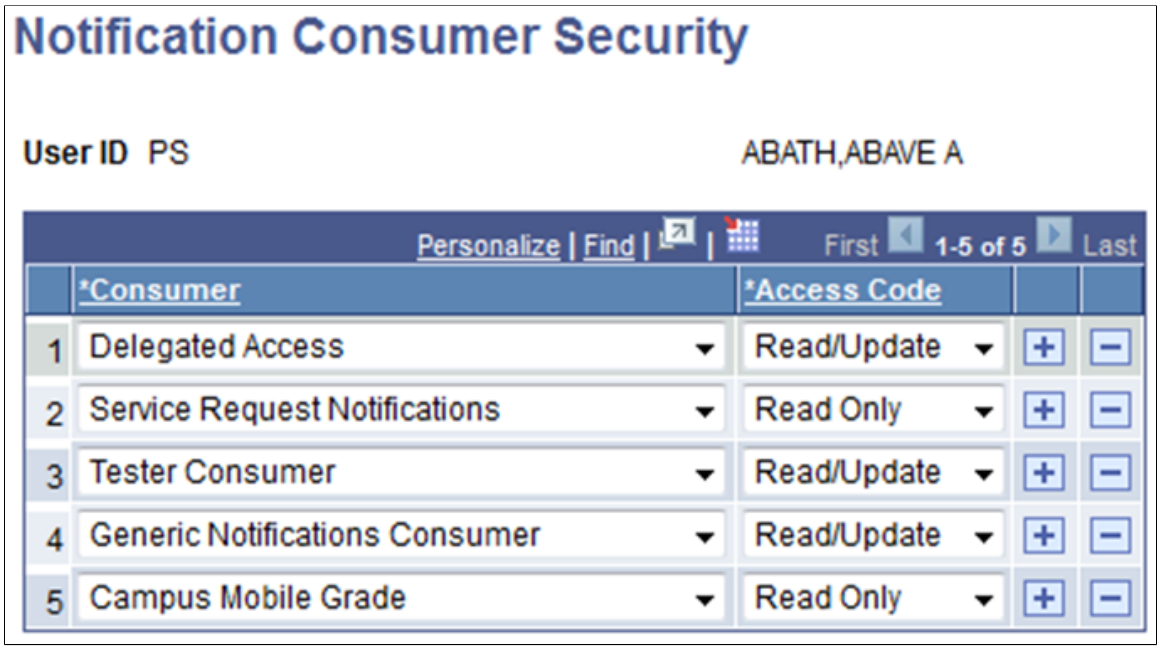
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Setting Up Notification Consumer Security

Access the Notification Consumer Security page (Set Up SACR, Security, Secure Student Administration, User ID, Notification Consumer Security).

Image: Notification Consumer Security Page

This example illustrates the fields and controls on the Notification Consumer Security page.



This is level 2 security setup, which is at the user level.

If level 1 security for any particular consumer is Read Only, then the only Access Code option here is Read Only.

When a logged-in user has read-only access to a particular consumer, then all details of (Email, Alert, Worklist, SMS) notifications that are created using that consumer appear in read-only mode on the Notifications Administration Overview page and no option exists to create announcements. This is also the case if a logged-in user has read-only access to Generic Notifications Consumer.

Using Admin Notifications

While the self-service Notifications pages are intended for a user to check and interact with his or her own notifications, the Admin Notifications pages are intended for users such as a research supervisor or administrative manager who are responsible for notifications sent to particular users.

Because the Admin Notification pages can display notifications for multiple users, a search page is provided so that someone such as an administrator can either:

- Search for specific notification recipients using search criteria in the conventional manner *or*
- Show all notifications that they are permitted to view by selecting the Show All Notifications check box.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

If no notifications are found, or the user lacks the requisite permissions, the Notifications Administration Overview page shows a message No Notification items found for selected criteria. Otherwise the user can view the notifications that appear in an almost identical manner to those shown for the self-service Notifications Center, with following exceptions if the administrator's access mode is set to Read/Update:

1. In the Worklist Details page, the Worklist item can be marked as completed.
2. On the Announcements tab, the trash can (deletion) icon and the Create Announcement button become available.

Page Used to Review and Manage Notifications

Page Name	*Definition Name*	*Navigation*	*Usage*
Notifications Administration Overview	SCC_NTF_ADMN_MAIN	Campus Community, Notifications, Admin Notifications, Notifications Administration Overview	Review and manage notifications.

Reviewing and Managing Notifications

Image: Notifications Administration Overview Page

This example illustrates the fields and controls of the Notifications Administration Overview page.

Features of the Notifications Administration Overview page include:

- Each notification type has its own tab where notifications of that type are shown.
- The notifications are shown in descending date and time order, so that the newest one is at the top, unless the user resorts the list by another column heading.
- For Email, the Subject of the email and Sender information is shown. The emails listed here are those that were sent with this particular ID as the 'To' recipient. The emails that were sent with this ID as 'Cc' or 'Bcc' recipient are not shown.
- The Period default value is 7 days.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

- All details shown on the Email tab are shown on the SMS and Alerts tabs, except Sender information. Because SMS does not have a Subject, the message text is displayed instead.
- The Worklist tab has an extra field Life Left Days that shows how many days remain until the deadline for the worklist item is reached. Similar to SMS and Alerts, the Worklist tab does not include a Sender column.
- The Worklist tab also has a Status field with the values Unactioned and Actioned. If permitted, the user can change the status of the worklist item by selecting the Details link for Worklists and selecting the Read/Actioned check box. The worklist item is then listed in the Actioned list.
- The Announcements tab also includes extra features for the creation and removal of announcements, subject to security, which is discussed in the next section.

Announcements

An administrator can use the Announcements feature to announce information to ALL students, without being restricted to specific EmplDs. The Notifications alert channel was enhanced to provide this functionality. Announcements are essentially alert-type notifications. However, unlike alerts, there is no recipient EMPLID for announcements; instead an audience indicator (SCC_NTF_AUDCE) is populated, indicating the particular alert is an Announcement.

The SCC_NTF_AUDCE field is found in the Notification Detail record (SCC_NTFREQ_DTL) with possible values of NA and ST. This field defaults to NA. The ST value indicates that this Alert type notification is actually an Announcement applicable for all students.

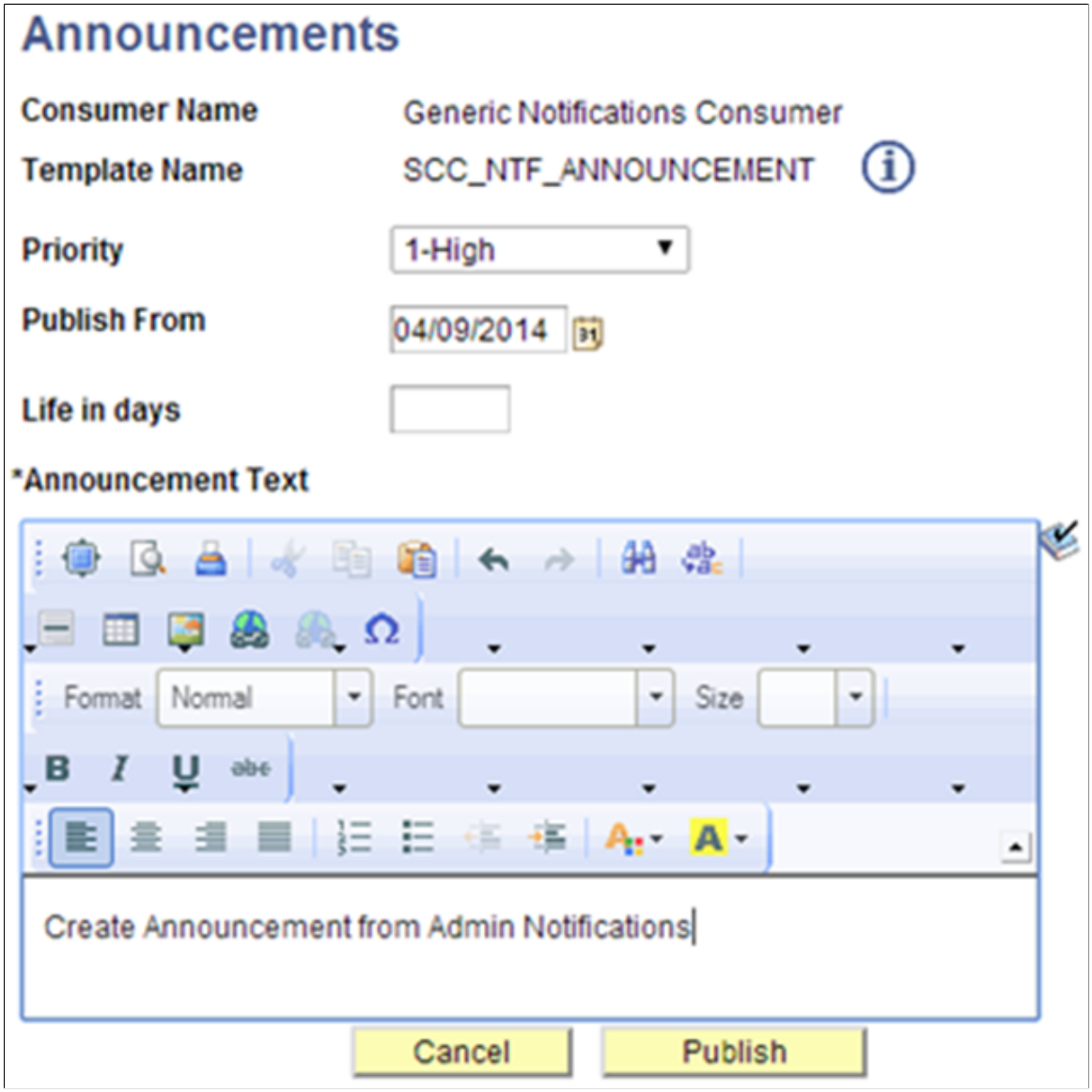
To create an announcement, the user clicks the Create Announcement button on the Announcement tab, which opens an Announcements window (shown below). The user then enters data in the required fields. Details of the template can be viewed by clicking on the information icon next to the template name.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

After the required details are populated, the user clicks the Publish button to send the Announcement to all students.

Image: Announcements Page

This example illustrates the fields and controls of the Announcements page.



Announcements sent from this page can also be viewed in the self-service Notifications Center.

See "Using the Self-Service Notifications Center" (PeopleSoft Campus Solutions 9.2: Self Service).

The recent announcements also appear on the SCE springboard page.

Refer to the Simplified Campus Experience documentation in My Oracle Support (ID 1600569.1).

Features of the Announcments tab include:

- Administrators who have read/update access to Generic Notifications Consumer can create announcements using the Create Announcement button.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

- All announcements including the timed-out announcements are listed on the Announcements tab.
- Depending on the access level setup for logged-in user for that consumer, the Delete icon is visible on the Announcements tab. If a user has read-only access for a consumer, the Delete icon is not visible for that particular row. When an admin user has read/update access the Delete icon is available.
- When creating announcements:
  - A user can set the priority of the announcement as High, Medium and Low.
  - The date entered in the Publish From field is displayed as the Date From field in the Announcement tab. If Publish From is greater than current date, the announcement appears in the grid from that date.
  - The value entered in the Life in days field is added to the date entered in the Publish From field and it becomes the Date To value on the Announcement tab.
  - Announcement Text is a mandatory field. If it is blank, a warning message is displayed.
  - If a user clicks the Publish button, the announcement appears in the Announcements grid.

Using Online Notifications

Online notifications functionality provides the ability for a user to create a notification manually and is implemented in the following Research Tracking components:

- Candidate Management
- Thesis Management
- Candidate Management Override
- Administrator Profile
- Administrator Self-service
- Service Request Management
- Project Management

A consumer ID has been created for each of these components with their respective templates. The bind variables which are particular to these specific components are defined in generic templates. Online notifications support Email, Alert and SMS notification channels.

See "Setting Up Online Notifications" (PeopleSoft Campus Solutions 9.2: Student Records).

For further information about using online notifications, including the Create Notification page, see: "Using Online Notifications" (PeopleSoft Campus Solutions 9.2: Student Records).

Implementing Online Notifications

Two generic pages are created:

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

1. A subpage which has the Send Notification button. (No PeopleCode is present in this. Consumers have to write PeopleCode for this button to populate the notification context, consumer ID and Bind variable data which is specific to this component. A sample PeopleCode is given below.)

2. A secondary page for selecting the values (channel, template, To, Cc, Bcc) and send notifications.

The PeopleCode for populating Consumer ID and the component specific data (that is, values for the template variables) is written at component record field (Send Notification) level in the fieldchange event. This PeopleCode varies from component to component because Consumer ID and template variables differs. Here is the sample code for the (Research Tracking) Service Request Management component. A component variable for notification context is created and consumer ID is populated.

```
import SCC_ADHOC_NTF:AdhocNotificationContext;
Component SCC_ADHOC_NTF:AdhocNotificationContext &adhoc;
&adhoc = create SCC_ADHOC_NTF:AdhocNotificationContext();
&adhoc.SCC_NTF_CONSUMER_ID = "SCC_NTF_CON_20130211040026";

Local Record &recdtl, &recPersonName;
Local string &RecipientEMPLID, &StudentName, &nRequestType, &nRequestSubtype, &nCom⇒
ment, &nStatus, &nReqCat, &URL, &instDesc, &typeDesc, &subtypeDesc, &statusDesc, &s⇒
sStatusDesc, &RecipientName, &catDesc, &ApproverName, &AssignedToName,&candNbr, &ap⇒
prvID, &assgnTo, &reqDt, &actionDt, &effDt;;

&Emplid = SSR_RS_REQSTHDR.EMPLID.Value;
&Institution = SSR_RS_REQSTHDR.INSTITUTION.Value;
&nRequestType = SSR_RS_REQSTHDR.SSR_RS_RQUEST_TYPE.Value;
&nRequestSubtype = SSR_RS_REQSTHDR.SSR_RS_REQ_SUBTYPE.Value;
&ReqID = SSR_RS_REQSTHDR.SSR_RS_REQUEST_ID.Value;
&candNbr = SSR_RS_REQSTHDR.SSR_RS_CANDIT_NBR.Value;
&reqNmbr = SSR_RS_REQSTHDR.SSR_REQ_SEQUENCE.Value;
&reqstdtl = GetLevel0()(1).GetRowset(Scroll.SSR_RS_REQSTDTL);
&recdtl = &reqstdtl.GetRow(CurrentRowNumber()).GetRecord(Record.SSR_RS_REQSTDTL);
&ReqSeq = &recdtl.SSR_RS_REQ_SEQ.Value;
&RecipientEMPLID = &recdtl.SSR_RS_ASSIGNEDTO.Value;
     &nComment = &recdtl.SSR_RS_DESCRLONG.Value;
&nStatus = &recdtl.SSR_RS_STATUS.Value;
&apprvID = &recdtl.SSR_RS_APPRVR_ID.Value;
&assgnTo = &recdtl.SSR_RS_ASSIGNEDTO.Value;
&reqDt = &recdtl.SSR_RS_REQUEST_DT.Value;
&actionDt = &recdtl.SSR_RS_ACTION_DT.Value;
&effDt = &recdtl.EFFDT.Value;
&recPersonName = CreateRecord(Record.PERSON_NAME);
&recPersonName.EMPLID.Value = &Emplid;
If &recPersonName.SelectByKey() Then
   &StudentName = &recPersonName.NAME_DISPLAY.Value;
 End-If;

SQLExec("select name_display from %table(PERSON_NAME) where emplid=:1 ", &Recipient⇒
EMPLID, &RecipientName);
SQLExec("select name_display from %table(PERSON_NAME) where emplid=:1 ", &assgnTo, ⇒
&AssignedToName);
SQLExec("select A.descr100, A.ssr_Rs_request_cat from PS_SSR_RS_REQSTTYP A where in⇒
stitution=:1 and ssr_Rs_rquest_type=:2 and %effdtcheck(SSR_RS_REQSTTYP ef1, A, %cur⇒
rentdatein)  ", &Institution, &nRequestType, &typeDesc, &nReqCat);
SQLExec("select A.descr100 from PS_SSR_RS_REQSTCAT A where A.institution=:1 and A.s⇒
sr_Rs_request_cat=:2 and %effdtcheck(SSR_RS_REQSTCAT ef1, A, %currentdatein)  ", &I⇒
nstitution, &nReqCat, &catDesc);
SQLExec("select A.descr50 from PS_SSR_RS_RQSUBTYP A where A.institution=:1 and A.ss⇒
r_Rs_rquest_type=:2 and A.ssr_Rs_req_subtype=:3 and %effdtcheck(SSR_RS_RQSUBTYP ef1⇒
, A, %currentdatein)  ", &Institution, &nRequestType, &nRequestSubtype, &subtypeDes⇒
c);

&adhoc.getParameterMap().Put("%1", &RecipientName);
&adhoc.getParameterMap().Put("%2", &RecipientEMPLID);
&adhoc.getParameterMap().Put("%3", &Emplid);
&adhoc.getParameterMap().Put("%4", &StudentName);
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

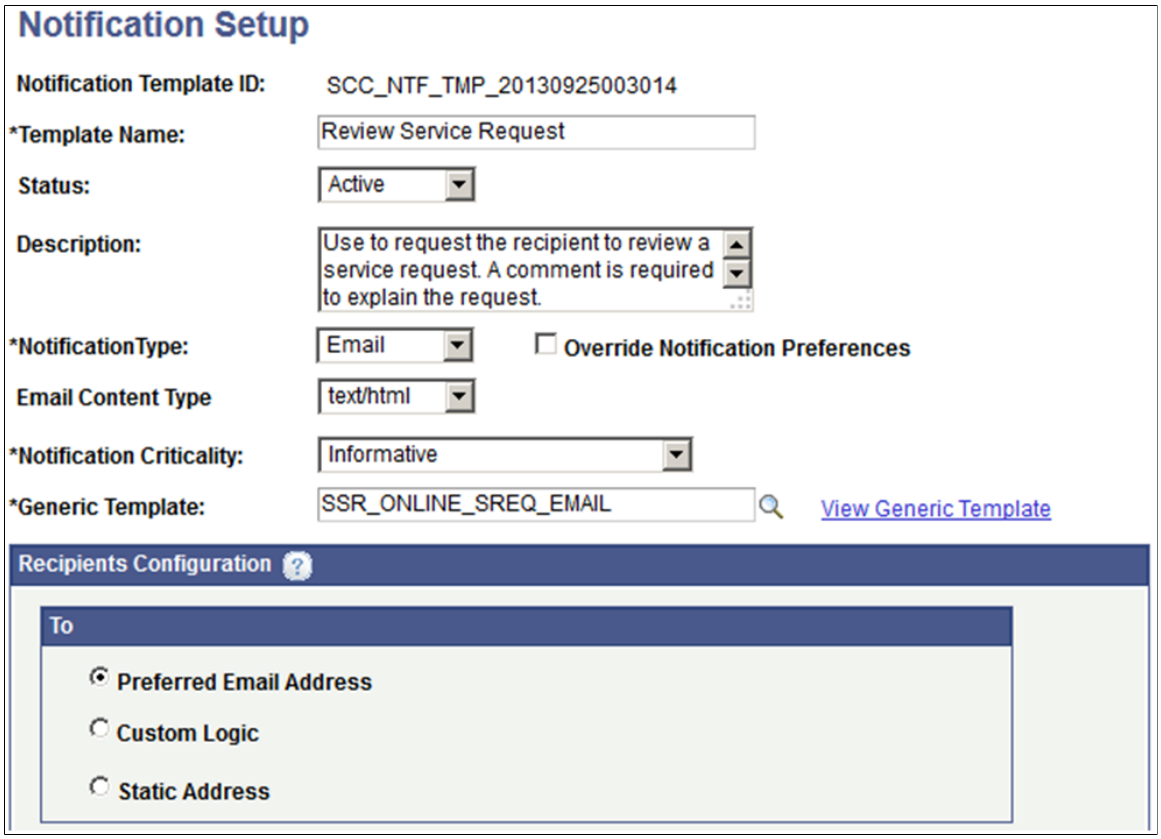
```
&adhoc.getParameterMap().Put("%5", &instDesc);
&adhoc.getParameterMap().Put("%6", "" | &ReqID);
&adhoc.getParameterMap().Put("%7", "" | &ReqSeq);
&adhoc.getParameterMap().Put("%8", &catDesc);
&adhoc.getParameterMap().Put("%9", &typeDesc);
&adhoc.getParameterMap().Put("%10", &subtypeDesc);
&adhoc.getParameterMap().Put("%11", &nComment);
&adhoc.getParameterMap().Put("%12", &statusDesc);
&adhoc.getParameterMap().Put("%13", &URL);
&adhoc.getParameterMap().Put("%14", &candNbr);
&adhoc.getParameterMap().Put("%15", &reqDt);
&adhoc.getParameterMap().Put("%16", &apprvID);
&adhoc.getParameterMap().Put("%17", &ApproverName);
&adhoc.getParameterMap().Put("%18", &assgnTo);
&adhoc.getParameterMap().Put("%19", &AssignedToName);
&adhoc.getParameterMap().Put("%20", &actionDt);
&adhoc.getParameterMap().Put("%21", &effDt);
&adhoc.getParameterMap().Put("%22", "" | &reqNmbr);
DoModalS(Page.SCC_NTF_ADHOC, "", 0, 0);
```

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Following is a sample notification setup for online notifications:

Image: Notification Setup Page

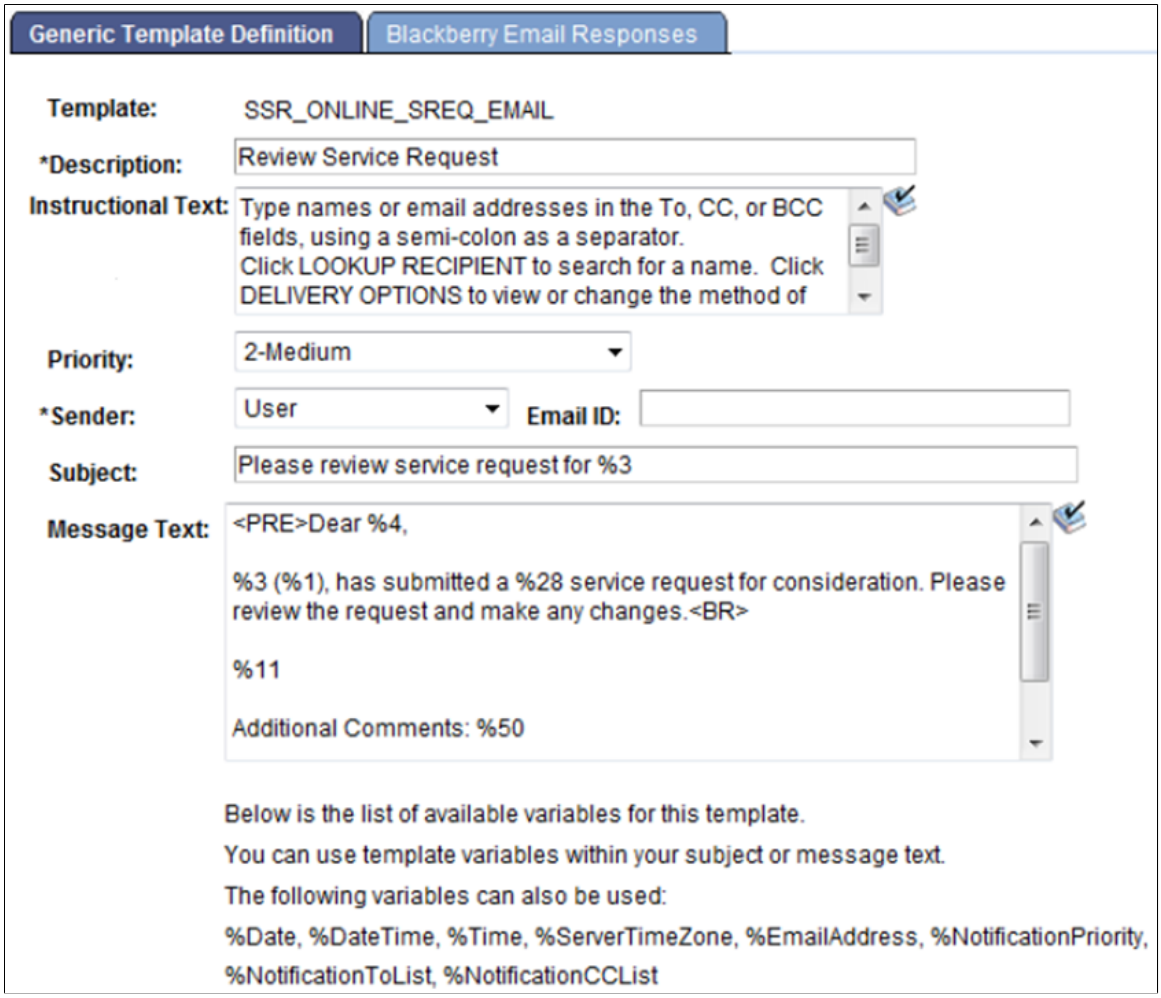
This example illustrates the fields and controls on the Notification Setup page.



Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Image: Generic Template Definition Page for NFK

This example illustrates the fields and controls on the Generic Template Definition Page for NFK



NFK Web Services GET and SEND (SCE)

As part of Simplified Campus Experience (SCE) features, two web services were created for Notifications Framework – GET and SEND.

Refer to the Simplified Campus Experience documentation in My Oracle Support (ID 1600569.1)

- SCC_GET_NOTIFICATIONS (primarily for the retrieval of Alerts and Worklist Items, but can also be used to retrieve Item detail for Email and SMS).
- SCC_SEND_NOTIFICATIONS (primarily for sending Emails and SMS from devices that do not contain native support, but can also be used to send Alerts and Worklist Items).

These service operations have been created under a common existing service SCC_NOTIFICATIONS. The schema for the GET and SEND request messages are shown below.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

GET Request

SCC_GET_NOTIF_REQ

The request message gives flexibility to retrieve from any channel type using generic filtering criteria from the Item or to search using specific Item keys if they are known. All notifications for the logged-in user are fetched by the service. Further details on the tag items are mentioned below.

```
<?xml version="1.0"?>
<SCC_GET_NOTIF_REQ xmlns="http://xmlns.oracle.com/Enterprise/services">
  <EMPLID></EMPLID>
  <SCC_NTFREQ_ID></SCC_NTFREQ_ID>
  <SCC_NTF_CON_ID></SCC_NTF_CON_ID>
  <SCC_NTFREQ_TYPE></SCC_NTFREQ_TYPE>
  <SCC_NTFREQ_ITMTAG1></SCC_NTFREQ_ITMTAG1>
  <SCC_NTFREQ_ITMTAG2></SCC_NTFREQ_ITMTAG2>
  <SCC_NTFREQ_ITMTAG3></SCC_NTFREQ_ITMTAG3>
  <SCC_NTFREQ_ITMTAG4></SCC_NTFREQ_ITMTAG4>
  <SCC_NTFREQ_ITM_STS></SCC_NTFREQ_ITM_STS>
  <SCC_NTFREQ_ITM_TAG></SCC_NTFREQ_ITM_TAG>
  <SCC_NTF_DATE_FROM></SCC_NTF_DATE_FROM>
  <SCC_NTF_DATE_TO></SCC_NTF_DATE_TO>
  <SCC_NTFREQ_SUB_STS></SCC_NTFREQ_SUB_STS>
  <SCC_NTFREQ_DTL_SEQ></SCC_NTFREQ_DTL_SEQ>
  <SCC_NTFREQ_ITM_SEQ></SCC_NTFREQ_ITM_SEQ>
  <SCC_NTF_AUDCE></SCC_NTF_AUDCE>
</SCC_GET_NOTIF_REQ>
```

Tag Item	***Tag Item Details***
EMPLID	The EMPLID is always populated with the logged-in user's EMPLID as it supports only self-service users.
SCC_NTFREQ_ID	When mentioned, the Notification Request ID fetches items having the same value for Request ID as what has been passed. This is essentially a filter.
SCC_NTFREQ_TYPE	Possible values are EML, SMS, WKL and ALT. When supplied, this parameter fetches only notifications of the given type for the particular EMPLID.
SCC_NTFREQ_ITM_TAG ... SCC_NTFREQ_ITMTAG4	This fetches items based on the tags associated with them.
SCC_NTFREQ_ITM_STS, SCC_NTFREQ_SUB_STS	This filters items based on their Status and Sub-Status. Items with a status such as ERROR, or sub-status such as DELETED/TIMEDOUT are not qualified for selection.
SCC_NTFREQ_DTL_SEQ, SCC_NTFREQ_ITM_SEQ	This, when combined with the notification request ID, focuses on a single item, which is then fetched by the get service.
SCC_NTF_DATE_FROM, SCC_NTF_DATE_TO	When values are given for from and to dates, all the SMS/EML notifications created within the given range are retrieved. For ALT/WKL all items which are active in the given date range are retrieved. This also includes all open-ended items.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Tag Item	*Tag Item Details*
SCC_NTF_AUDCE	It can have two possible values currently as mentioned above - ST/NA. When ST value is passed, only the Announcement Notifications is returned by the service. In case no value (is treated as NA) or NA value is passed, all notifications – EML/SMS/WKL/ALT – are fetched along with the announcement notifications.

This is a sample response message:

```
<?xml version="1.0"?>
<SCC_GET_NOTIF_RESP xmlns="http://xmlns.oracle.com/Enterprise/services">
    <NTK_ITEM>
      <SCC_NTFREQ_ID>282</SCC_NTFREQ_ID>
      <SCC_NTFREQ_DTL_SEQ>1</SCC_NTFREQ_DTL_SEQ>
      <SCC_NTFREQ_ITM_SEQ>1</SCC_NTFREQ_ITM_SEQ>
      <EMPLID>SR0448</EMPLID>
      <SCC_NTF_CON_ID>SCC_NTF_CON_20130512074705</SCC_NTF_CON_ID>
      <SCC_NTF_DATE_FROM>2013-07-08</SCC_NTF_DATE_FROM>
      <SCC_NTF_DATE_TO></SCC_NTF_DATE_TO>
      <SCC_NTFREQ_TYPE>ALT</SCC_NTFREQ_TYPE>
      <SCC_NTFREQ_TMPLTID>SCC_NTF_TMP_20130512074210</SCC_NTFREQ_TMPLTID>
      <SCC_NTFREQ_SUBJECT>Test Template for ALERT Notifications</SCC_NTFREQ_SUBJECT⇒
>
      <SCC_NTFREQ_IMPTNCE>2</SCC_NTFREQ_IMPTNCE>
      <SCC_NTFREQ_ITM_STS>S</SCC_NTFREQ_ITM_STS>
      <SCC_NTFREQ_SUB_STS>U</SCC_NTFREQ_SUB_STS>
      <SCC_NTFREQ_ITM_TAG>VISWA</SCC_NTFREQ_ITM_TAG>
      <SCC_NTFREQ_ITMTAG1>TAG1</SCC_NTFREQ_ITMTAG1>
      <SCC_NTFREQ_ITMTAG2>TAG2</SCC_NTFREQ_ITMTAG2>
      <SCC_NTFREQ_ITMTAG3>TAG3</SCC_NTFREQ_ITMTAG3>
      <SCC_NTFREQ_ITMTAG4>TAG4</SCC_NTFREQ_ITMTAG4>
      <SCC_ROW_ADD_DTTM>07/30/2013 17:19:56.000000</SCC_ROW_ADD_DTTM>
      <SCC_NTFREQ_MSGTEXT>This is a test template for ALERT Notifications.

      Value 1 - Test1
      Value 2 - tst2
      Value 5 - tst3
      </SCC_NTFREQ_MSGTEXT>
      <SCC_NTF_AUDCE>NA</SCC_NTF_AUDCE>
    </NTK_ITEM>
</SCC_GET_NOTIF_RESP>
```

Note: If a SOAP service request XSD contains the “languageCd” parameter, then the service is enabled for National Language Support (NLS) and an ISO Locale value needs to be passed as the languageCd variable. Valid values can be found by navigating to PeopleTools, Utilities, International, Languages.

SEND Request

SCC_SEND_NOTIF_REQ

The Send request is comprised of the Item detail, a parameter map of tokens, and optional Base64 attachments (for the email channel only).

```
<?xml version="1.0"?>
<SCC_SEND_NOTIF_REQ xmlns="http://xmlns.oracle.com/Enterprise/services">
    <EMPLID>SR0451</EMPLID>
    <SCC_NTF_CON_ID></SCC_NTF_CON_ID>
    <SCC_NTF_DATE_FROM></SCC_NTF_DATE_FROM>
    <SCC_NTF_DATE_TO></SCC_NTF_DATE_TO>
```

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
    <SCC_NTFREQ_TMPLTID>SCC_NTF_TMP_20130404201255</SCC_NTFREQ_TMPLTID>
    <SCC_NTFREQ_IMPTNCE></SCC_NTFREQ_IMPTNCE>
    <SCC_NTFREQ_ITM_TAG>TEST</SCC_NTFREQ_ITM_TAG>
    <SCC_NTFREQ_ITMTAG1></SCC_NTFREQ_ITMTAG1>
    <SCC_NTFREQ_ITMTAG2></SCC_NTFREQ_ITMTAG2>
    <SCC_NTFREQ_ITMTAG3></SCC_NTFREQ_ITMTAG3>
    <SCC_NTFREQ_ITMTAG4></SCC_NTFREQ_ITMTAG4>
    <PARAMETER_MAP>
       <PARAMETER>
          <SCC_NTFREQ_DAT_KEY>%1</SCC_NTFREQ_DAT_KEY>
          <SCC_NTFREQ_DAT_VAL>This is a Test email.</SCC_NTFREQ_DAT_VAL>
       </PARAMETER>
       <PARAMETER>
          <SCC_NTFREQ_DAT_KEY>%2</SCC_NTFREQ_DAT_KEY>
          <SCC_NTFREQ_DAT_VAL>From the friendly NFK</SCC_NTFREQ_DAT_VAL>
       </PARAMETER>
    </PARAMETER_MAP>
    <ATTACHMENTS>
       <ATTACHMENT>
          <fileName>Scooby.jpg</fileName>
          <Base64Data>/9j/4AAQSkZJRgABAQEAYAB</Base64Data>
       </ATTACHMENT>
    </ATTACHMENTS>
 </SCC_SEND_NOTIF_REQ>
```

The Send request must have the following parameters populated – EMPLID and SCC_NTFREQ_TMPLTID.

Other than the parameters already explained above, the send request also holds the template parameters - SCC_NTFREQ_DAT_KEY and SCC_NTFREQ_DAT_VAL. This holds the value of the bind variables defined in the generic template.

If the notification web service results in a successful send, an empty XML is returned as response.

NFK Web Services for Push Notifications

As part of Push Notifications, the following REST web service operation was created for Notifications Framework: SCC_DEVICE_REGISTER_POST (Used to register a device with Campus Solutions to receive push notifications).

This service operation has been created under a common existing service: SCC_NOTIFICATIONS_R.

This is the schema for the request message, SCC_DEVICE_REGISTER_REQ:

```
<?xml version="1.0"?>
<SCC_DEVICE_REGISTER_REQ isDeleted="" xmlns="http://xmlns.oracle.com/Enterprise/ser⇒
vices">
  <DEVICE_TOKEN isDeleted="">
    <SCC_USERNAME></SCC_USERNAME>
    <SCC_NTF_DEV_TKNID></SCC_NTF_DEV_TKNID>
    <SCC_NTF_TOKEN_TYPE></SCC_NTF_TOKEN_TYPE>
    <SCC_NTF_MOB_APP_ID></SCC_NTF_MOB_APP_ID>
    <SCC_NTF_CREATE_DT></SCC_NTF_CREATE_DT>
  </DEVICE_TOKEN>
</SCC_DEVICE_REGISTER_REQ>
```

Tag Item	***Tag Item Details***
SCC_USERNAME	User Name is populated with logged-in UserName in mobile device.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Tag Item	Tag Item Details
SCC_NTF_DEV_TKNID	Token ID value is populated with tokenId/RegistrationId received from APNS/GCM after successful registration of the device with APNS/GCM to receive Push Notifications.
SCC_NTF_TOKEN_TYPE	Token Type is populated with value APNS/GCM based on mobile device OS. If it is Android device the value is GCM and if it is an Apple Device the value is
APNS. SCC_NTF_MOB_APP_ID	This value is Mobile Application ID to register for push Notifications with Campus Solutions.
SCC_NTF_CREATE_DT	Dummy element

This is a sample response message:

```
<SCC_DEVICE_REGISTER_RESP xmlns="http://xmlns.oracle.com/Enterprise/services">
   <DEVICE_REG_SUCCESS>Y</DEVICE_REG_SUCCESS>
</SCC_DEVICE_REGISTER_RESP>
```

NFK Web Services for Events

As part of Push Notifications, the following rest web service operations were created for the Notifications Framework:

- SCC_NTF_GET_EVENTS_R_POST (primarily for the retrieval of events list along with read/unread/total events count)
- SCC_NTF_UPDATE_EVENTS_R_POST (primarily for updating the event status which returns the latest read/unread/total events count)

These service operations have been created under a common existing service – SCC_NOTIFICATIONS_R.

SCC_NTF_GET_EVENTS_R_POST

SCC_NTF_GET_EVENTS_REQ (request message for SCC_NTF_GET_EVENTS_R_POST)

```
<?xml version="1.0"?>
<SCC_NTF_GET_EVENTS_REQ_R>
 <NUM_PAST_DAYS>7</NUM_PAST_DAYS>
 <INCLUDE_EVENTS>N</INCLUDE_EVENTS>
</SCC_NTF_GET_EVENTS_REQ_R>
```

Tag Item	Tag Item Details
NUM_PAST_DAYS	Past number of days to retrieve events.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Tag Item	Tag Item Details
INCLUDE_EVENTS	• "Y" – includes events in the response along with read/unread/total events count. • "N" – only read/unread/total events count in the response.

SCC_NTF_GET_EVENTS_RESP (response message for SCC_NTF_GET_EVENTS_R_POST)

```
<?xml version="1.0"?>
<SCC_NTF_GET_EVENTS_RESP xmlns="http://xmlns.oracle.com/Enterprise/services">
   <TOTAL_EVENTS_COUNT>12</TOTAL_EVENTS_COUNT>
   <READ_EVENTS_COUNT>8</READ_EVENTS_COUNT>
   <UNREAD_EVENTS_COUNT>4</UNREAD_EVENTS_COUNT>
   <EVENTS>
      <SCC_NTF_EVENT>
         <SCC_NTFEVT_REQ_ID>487</SCC_NTFEVT_REQ_ID>
         <EMPLID>SR12030</EMPLID>
         <SCC_NTFEVT_STATUS>R</SCC_NTFEVT_STATUS>
         <SCC_NTFEVT_STARTDT>03/21/2014 10:04:43.000000</SCC_NTFEVT_STARTDT>
         <SCC_NTFEVT_MESSAGE>Your final grade for SOC 103.2 has been posted.</SCC_N⇒
TFEVT_MESSAGE>
         <DATA>
            <SCC_NTFEVT_DATA>
               <SCC_NTFEVT_DAT_KEY>CAREER</SCC_NTFEVT_DAT_KEY>
               <SCC_NTFEVT_DAT_VAL>UGRD</SCC_NTFEVT_DAT_VAL>
            </SCC_NTFEVT_DATA>
            <SCC_NTFEVT_DATA>
               <SCC_NTFEVT_DAT_KEY>EVENT_TYPE</SCC_NTFEVT_DAT_KEY>
               <SCC_NTFEVT_DAT_VAL>FINAL_GRADE_POSTED</SCC_NTFEVT_DAT_VAL>
            </SCC_NTFEVT_DATA>
            <SCC_NTFEVT_DATA>
               <SCC_NTFEVT_DAT_KEY>GOTO</SCC_NTFEVT_DAT_KEY>
               <SCC_NTFEVT_DAT_VAL>Grades</SCC_NTFEVT_DAT_VAL>
            </SCC_NTFEVT_DATA>
            <SCC_NTFEVT_DATA>
               <SCC_NTFEVT_DAT_KEY>INST</SCC_NTFEVT_DAT_KEY>
               <SCC_NTFEVT_DAT_VAL>PSUNV</SCC_NTFEVT_DAT_VAL>
            </SCC_NTFEVT_DATA>
            <SCC_NTFEVT_DATA>
               <SCC_NTFEVT_DAT_KEY>TERM</SCC_NTFEVT_DAT_KEY>
               <SCC_NTFEVT_DAT_VAL>0710</SCC_NTFEVT_DAT_VAL>
            </SCC_NTFEVT_DATA>
         </DATA>
      </SCC_NTF_EVENT>    </EVENTS>
</SCC_NTF_GET_EVENTS_RESP>
```

Tag Item	Tag Item Details
TOTAL_EVENTS_COUNT	Total number of events from NUM_PAST_DAYS sent in the request.
READ_EVENTS_COUNT	Total number of read events from NUM_PAST_DAYS sent in the request.
UNREAD_EVENTS_COUNT	Total number of unread events from NUM_PAST_DAYS sent in the request.
SCC_NTFEVT_REQ_ID	Event request ID which uniquely identifies an event.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Tag Item	Tag Item Details
EMPLID	Employee ID.
SCC_NTFEVT_STATUS	Status of the event. “R” - Read “U” - Unread .
SCC_NTFEVT_STARTDT	Event occurred datetime.
SCC_NTFEVT_MESSAGE	Event Message
SCC_NTFEVT_DAT_KEY	Key for data map (key-value pairs).
SCC_NTFEVT_DAT_VAL	Value for data map (key-value pairs).

SCC_NTF_UPDATE_EVENTS_R_POST

SCC_NTF_UPDATE_EVENTS_REQ (request message for SCC_NTF_UPDATE_EVENTS_R_POST)

```
<?xml version="1.0"?>
<SCC_NTF_UPDATE_EVENTS_REQ>
 <NUM_PAST_DAYS>7</NUM_PAST_DAYS>
 <EVENTS>
 <SCC_NTF_EVENT>
  <SCC_NTFEVT_REQ_ID>216</SCC_NTFEVT_REQ_ID>
  <SCC_NTFEVT_STATUS>R</SCC_NTFEVT_STATUS>
 </SCC_NTF_EVENT>
 </EVENTS>
</SCC_NTF_UPDATE_EVENTS_REQ>
```

Tag Item	Tag Item Details
NUM_PAST_DAYS	Past number of days to retrieve events.
SCC_NTFEVT_REQ_ID	Event request ID for which status has to be updated.
SCC_NTFEVT_STATUS	Status of the event to be updated. “R” - Read “U” - Unread.

SCC_NTF_UPDATE_EVENTS_RESP (response message for SCC_NTF_UPDATE_EVENTS_R_POST)

```
<?xml version="1.0"?>
<SCC_NTF_UPDATE_EVENTS_RESP xmlns="http://xmlns.oracle.com/Enterprise/services">
   <TOTAL_EVENTS_COUNT>12</TOTAL_EVENTS_COUNT>
   <READ_EVENTS_COUNT>8</READ_EVENTS_COUNT>
   <UNREAD_EVENTS_COUNT>4</UNREAD_EVENTS_COUNT>
   <RESP_DTTM>03/25/2014 12:38:47.000000</RESP_DTTM>
</SCC_NTF_UPDATE_EVENTS_RESP>
```

Tag Item	Tag Item Details
TOTAL_EVENTS_COUNT	Total number of events from NUM_PAST_DAYS sent in the request.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Tag Item	*Tag Item Details*
READ_EVENTS_COUNT	Total number of read events from NUM_PAST_DAYS sent in the request.
UNREAD_EVENTS_COUNT	Total number of unread events from NUM_PAST_DAYS sent in the request.
RESP_DTTM	Timestamp of the response.

Testing the Notifications Framework

Page Used to Test the Notifications Framework

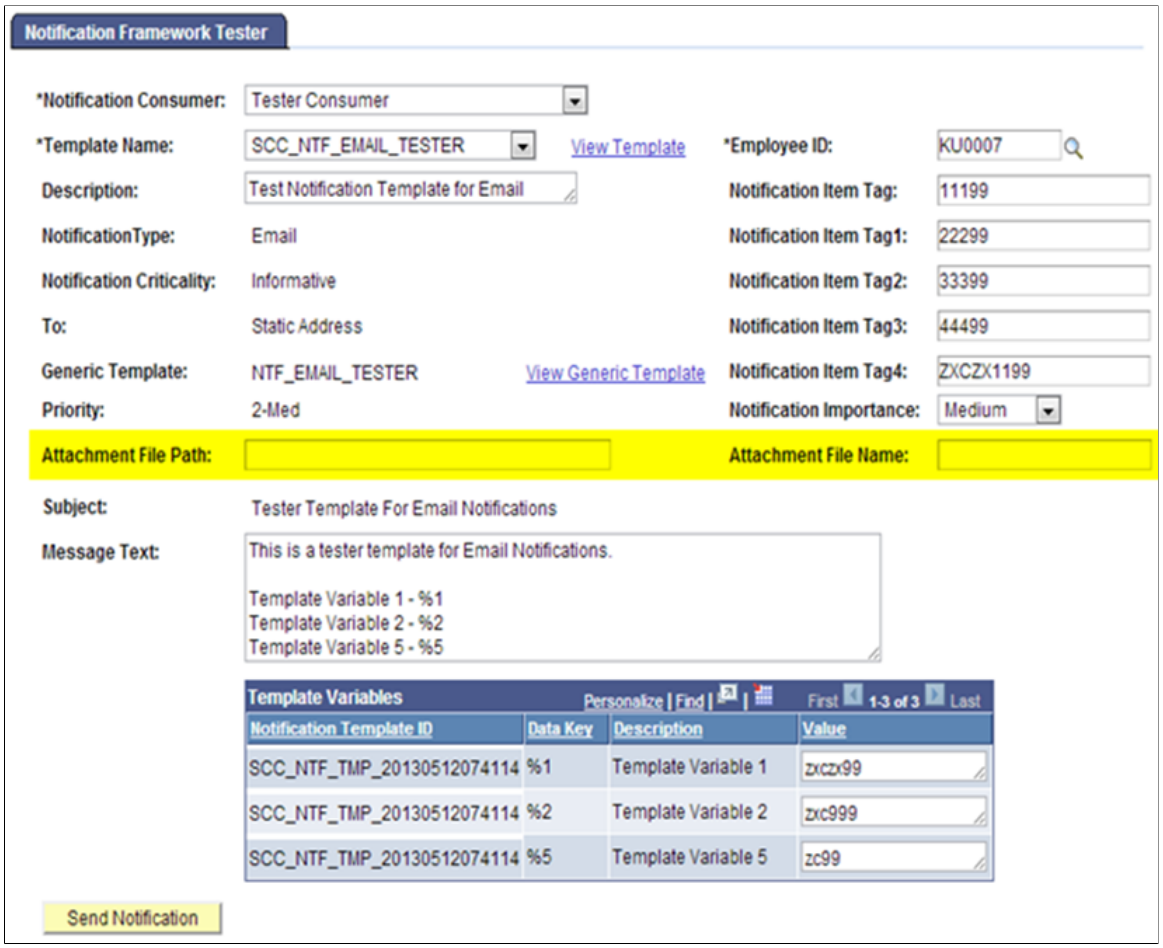
Page Name	*Definition Name*	*Navigation*	*Usage*
Notification Framework Tester	SCC_NTF_TESTER	Set Up SACR, System Administration, Utilities, Notifications, Notification Framework Tester.	Test email, SMS, alert and Worklist notifications.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Access the Notification Framework Tester page (Set Up SACR, System Administration, Utilities, Notifications, Notification Framework Tester).

Image: Notification Framework Tester

This example illustrates the fields and controls on the Notification Framework Tester page. You can find definitions for the fields and controls later on this page.



Notification Consumer Select the consumer for which you want to test notifications. Tester Consumer has been created specifically for this purpose.

Template Name Select the notification template that you want to use for testing the notifications framework. This list is populated based on selection of the Notification Consumer. All templates for the consumer which have a value of Preferred or Static for Email Address/Phone Number/EmplID on the Notification Setup page are available. Templates with a value of Custom Logic are not populated in the list.

View Template Select this link to access the Notification Setup page to view the selected template.

View Generic Template Select this link to access the Generic Template Definition page for the selected template.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Broadcast Audience Flag This field appears only when the Notification Type is Alert. If you select N/A, you must then select an EmplID. If you select Student, the notification is sent to all EmplIDs.

Employee ID The Employee ID field should be populated with the context Employee ID value. The same would reside in the database in SCC_NTFREQ_DTL.EMPLID (Record.Field). Retrieval of Notification items would also be based on the same Employee ID.

Notification Item Tag (1– 4) Enter any value for these fields, which are treated as tags for the Notification items. The same would also be populated in the Notification Context.

Notification Importance If you set Notification Importance here, it overrides the Importance value set in the Generic Template setup.

Attachment File Path This field appears only for Email templates. Here, you must provide the shared location and file name, for example, *\\slc00sun\Share\Roses.jpg*. The file would then appear as an attachment in the email notification as “Roses.jpg”.

Attachment File Name This field appears only for Email templates. Enter the name of the file as you want it to display in the email attachment. If you enter “Lotus” here, the above attachment would appear as “Lotus.jpg” in the email notification. You can also override the file attachment type here. If you enter a filename “Lotus.txt”, the attachment type is changed from .jpg to .txt. If no value is entered here, the file has the original name as in the file path.

The Notification Type, Notification Criticality, To, Priority, Subject, and Message Text fields are populated based on the selected template. The template variables defined for a generic template also appear when the template is selected. Populate the Value field against each template variable (data key) in the grid, Save and then click Send Notification.

See Configuring the Generic Templates for Notifications Framework.

See Configuring Notification Setup for the Generic Templates.

Purging Notifications Framework Records

The Notifications Framework can generate a huge amount of transactional data. For performance improvement and data managing requirements, the process of archive and purge becomes very important.

PeopleSoft Data Archive Manager is used for archiving or purging NFK transaction data.

The end-user enters the date until when the archiving is required. After the date is entered, the process retrieves all data from the NFK transaction records to the corresponding NFK history records which have been created as part of this feature.

PS Query is the selection tool used for filtering the data to be archived/purged. The NFK records have been added under SCC_NTFREQ_HDR in the Query tree. Simultaneously history records were created

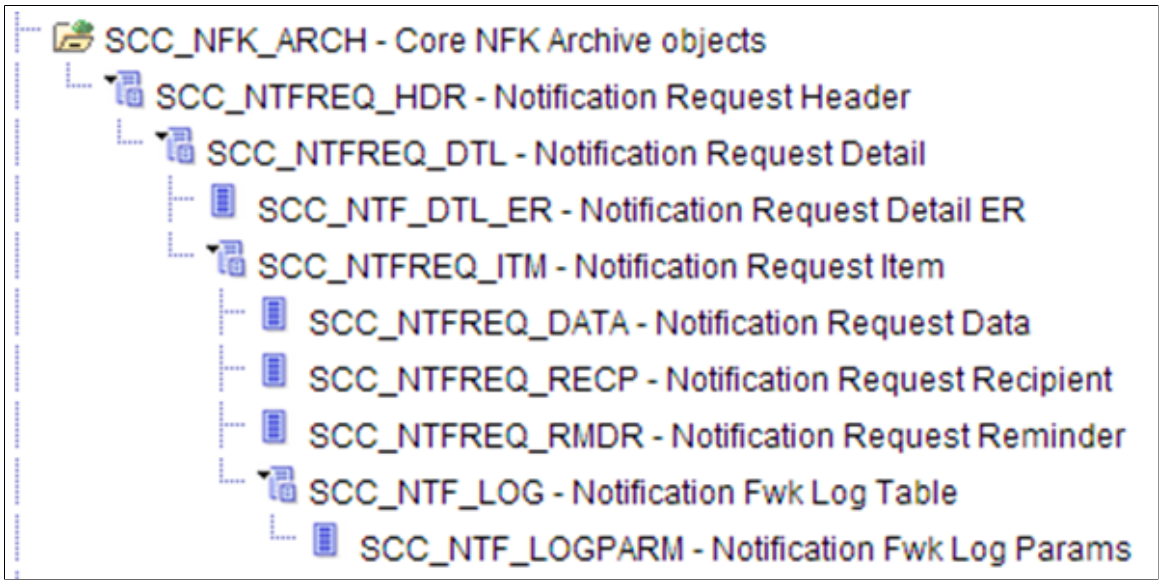
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

for each NFK transaction record. This is where the archived data is available after the archive process is run. Archive object and Archive Templates are also delivered as part of this feature.

Query tree – NFK Archive: A child group has been added as shown below:

Image: NFK Archive Query Tree

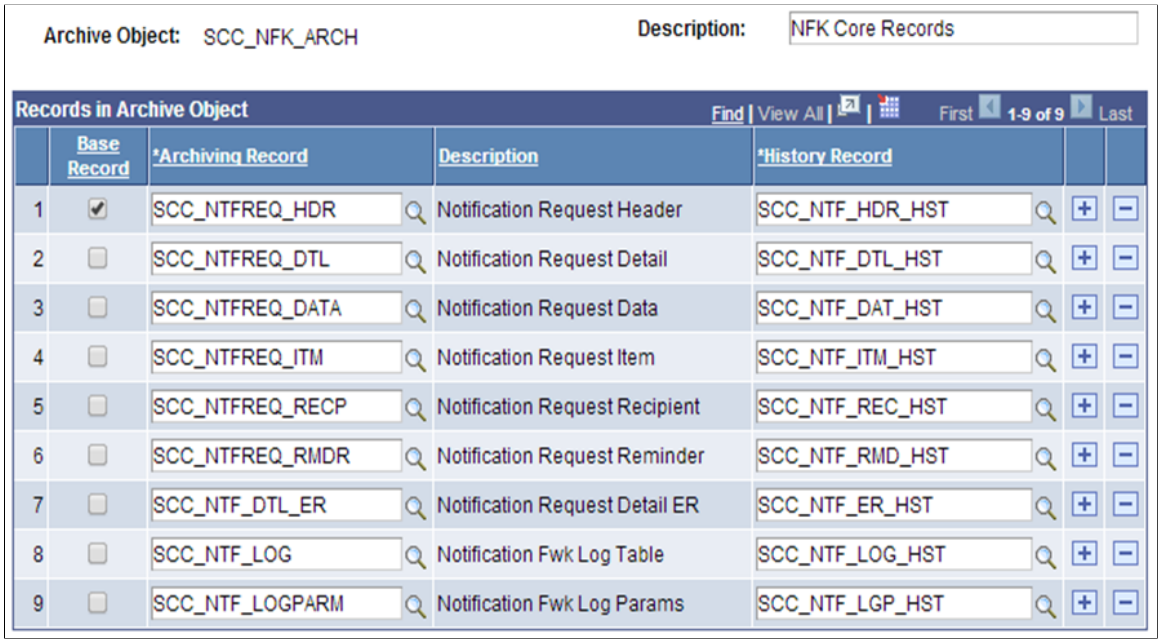
This example illustrates the NFK Archive Query Tree.



The Archive Object definition delivered. Access the Manage Archive Objects page (PeopleTools, Data Archive Manager, Manage Archive Objects).

Image: Manage Archive Object page for NFK Objects

This example illustrates the Manage Archive Object page for NFK Objects.



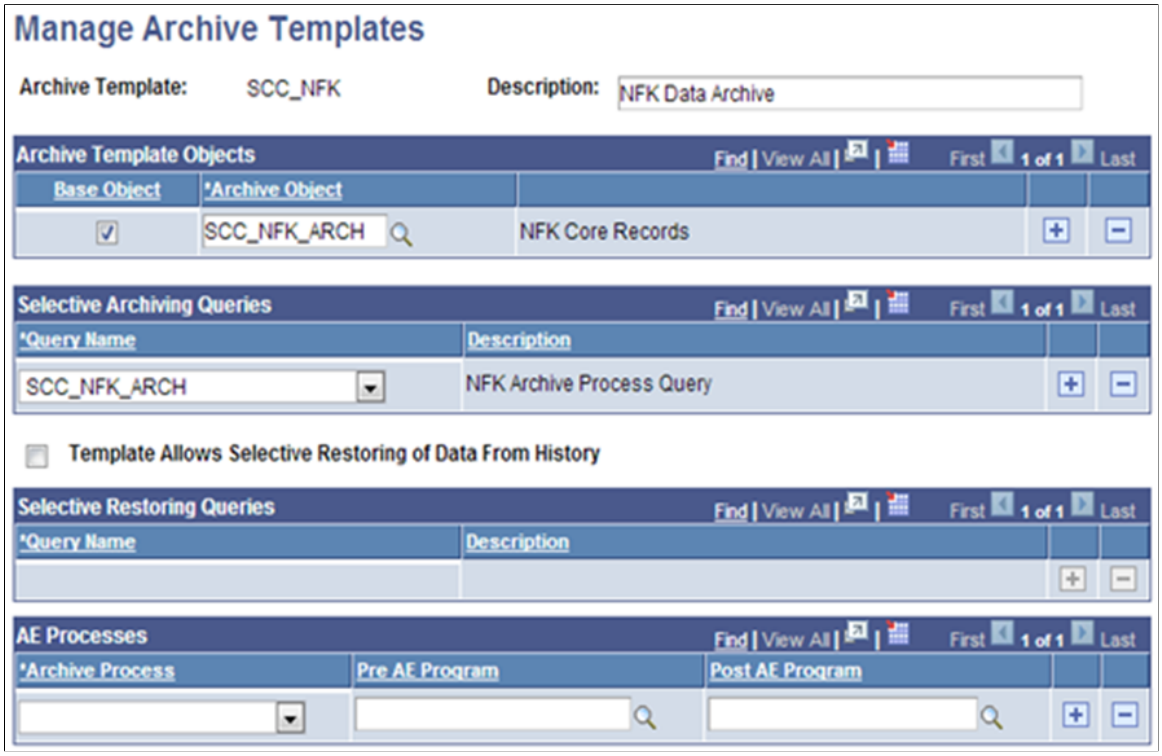
	Base Record	*Archiving Record	Description	*History Record
1	✔	SCC_NTFREQ_HDR	Notification Request Header	SCC_NTF_HDR_HST
2		SCC_NTFREQ_DTL	Notification Request Detail	SCC_NTF_DTL_HST
3		SCC_NTFREQ_DATA	Notification Request Data	SCC_NTF_DAT_HST
4		SCC_NTFREQ_ITM	Notification Request Item	SCC_NTF_ITM_HST
5		SCC_NTFREQ_RECP	Notification Request Recipient	SCC_NTF_REC_HST
6		SCC_NTFREQ_RMDR	Notification Request Reminder	SCC_NTF_RMD_HST
7		SCC_NTF_DTL_ER	Notification Request Detail ER	SCC_NTF_ER_HST
8		SCC_NTF_LOG	Notification Fwk Log Table	SCC_NTF_LOG_HST
9		SCC_NTF_LOGPARM	Notification Fwk Log Params	SCC_NTF_LGP_HST

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

The Archive Template delivered. Access the Manage Archive Templates page (PeopleTools, Data Archive Manager, Manage Archive Templates).

Image: Manage Archive Templates page

This example illustrates the Manage Archive Templates page for NFK Objects.



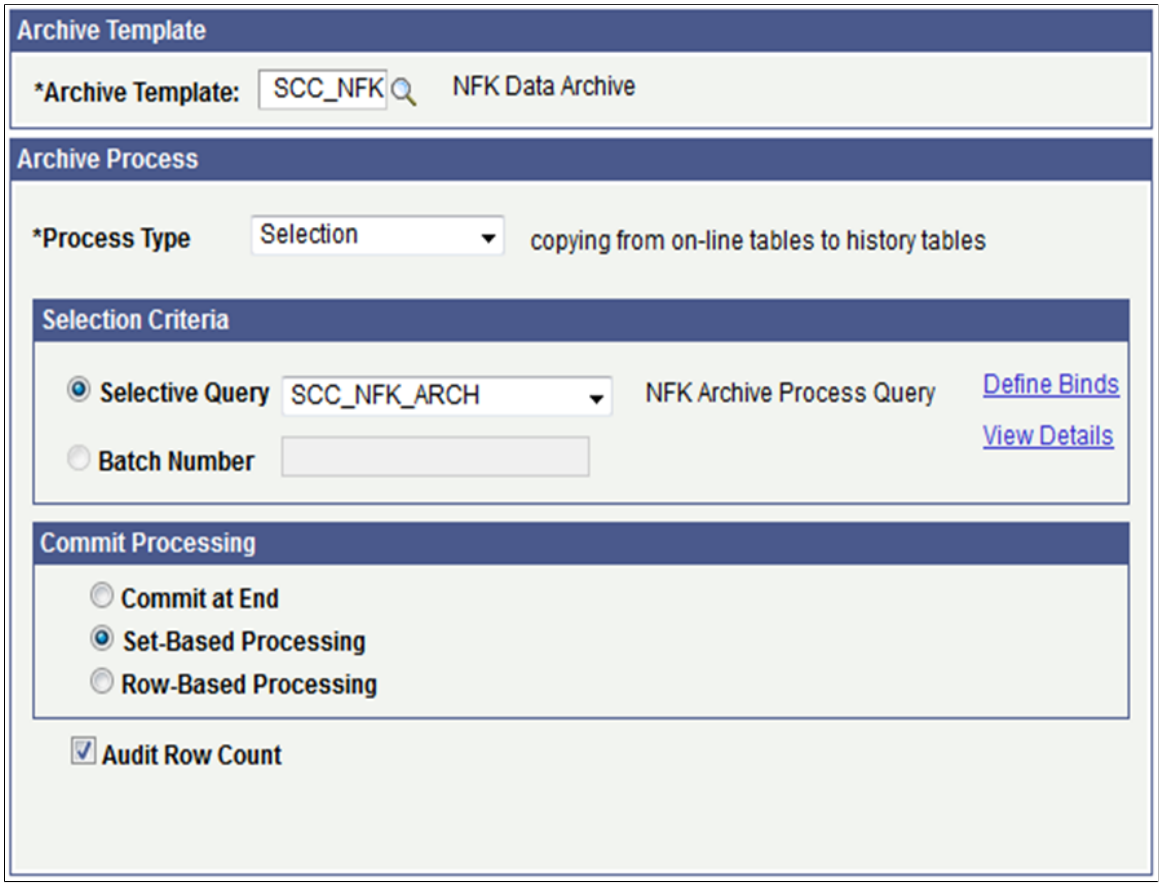
Access the Archive Data To History page (PeopleTools, Data Archive Manager, Archive Data To History).

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

To run the Archive/Purge process, run the SCC_NFK archive template with the following options as shown below:

Image: Archive Data To History page for NFK Objects

This example illustrates the Archive Data To History page for NFK Objects.



Use the Define Binds link to indicate the date until which the Archive is required.

Archiving Events

To archive the Event NFK transaction records the end-user selects a date until when the archiving is required, and check boxes for read and unread events. After those details are entered, the process retrieves all data from the Event NFK transaction records to the corresponding Event NFK history records which have been created as part of this feature.

PS Query is the selection tool used for filtering the data to be archived/purged. We have added the SCC_NTF_EVENT record under existing Query tree. Simultaneously history records were created for each Event NFK transaction record. This is where the archived data is available after the archive process is run. Archive object and Archive Templates are also delivered as part of this feature.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Query tree – Child record for Event NFK Archive:

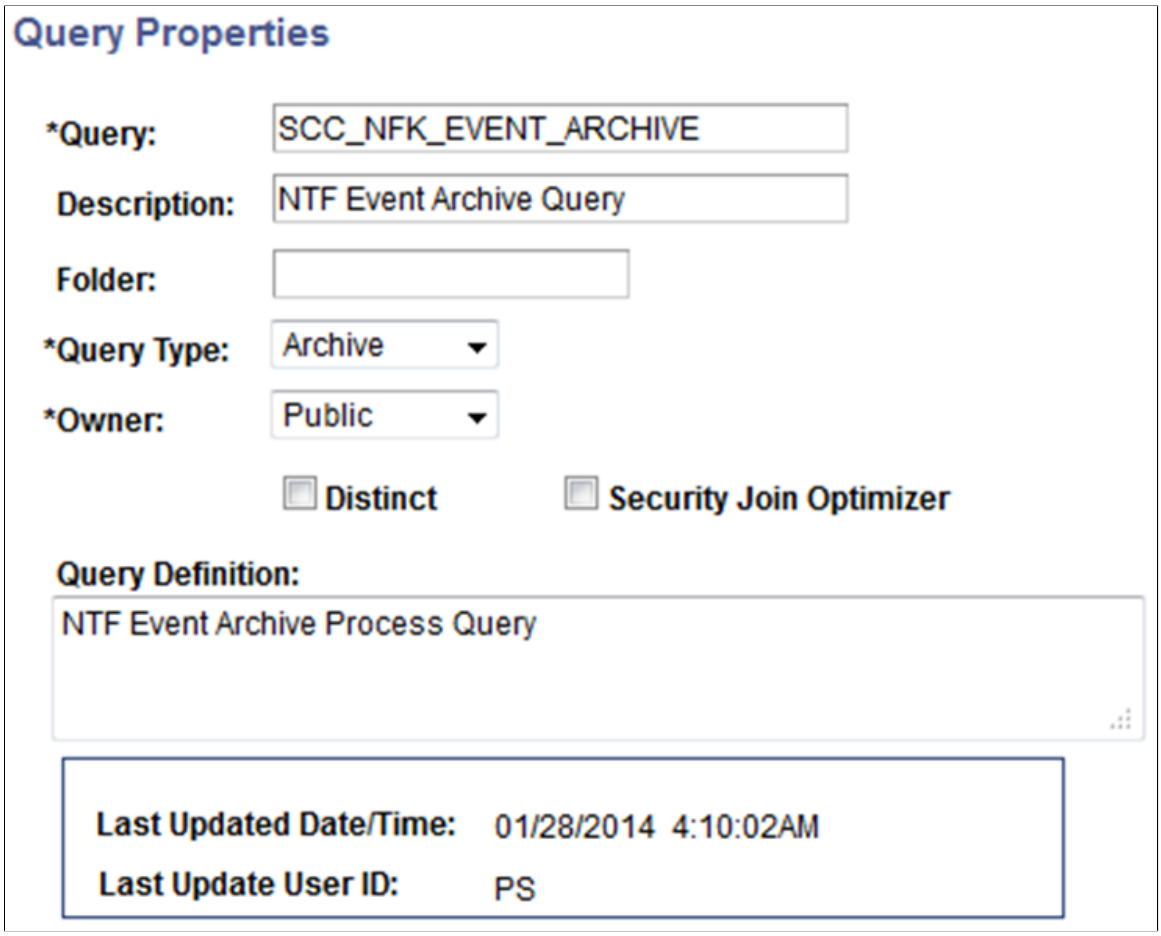
Image: NFK Event Archive Query Tree

This example illustrates the NFK Event Archive Query Tree.

PSQuery:

Image: Query Properties page

This example illustrates the fields and controls on the Query Properties page.

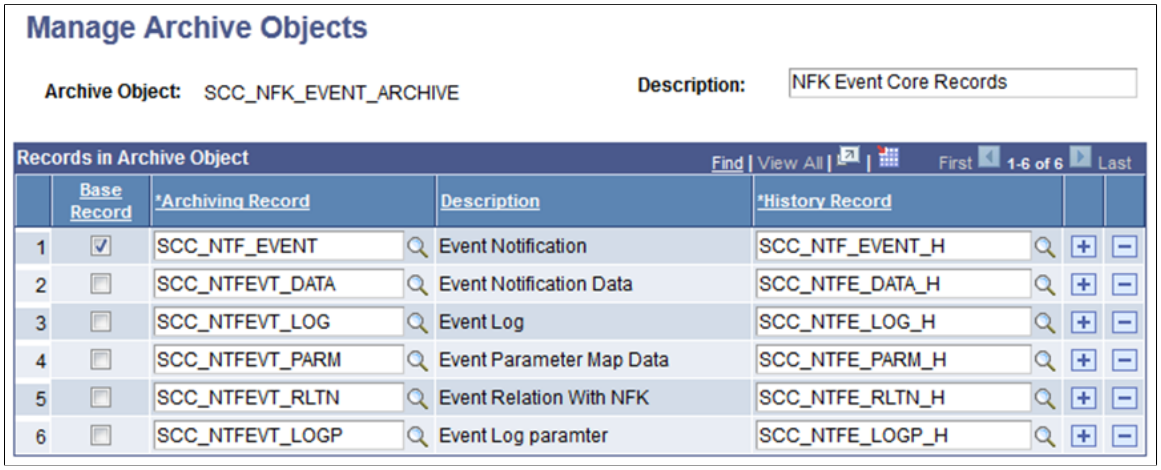


Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Access the Manage Archive Objects page (PeopleTools, Data Archive Manager, Manage Archive Objects).

Image: Manage Archive Object page for NFK Events

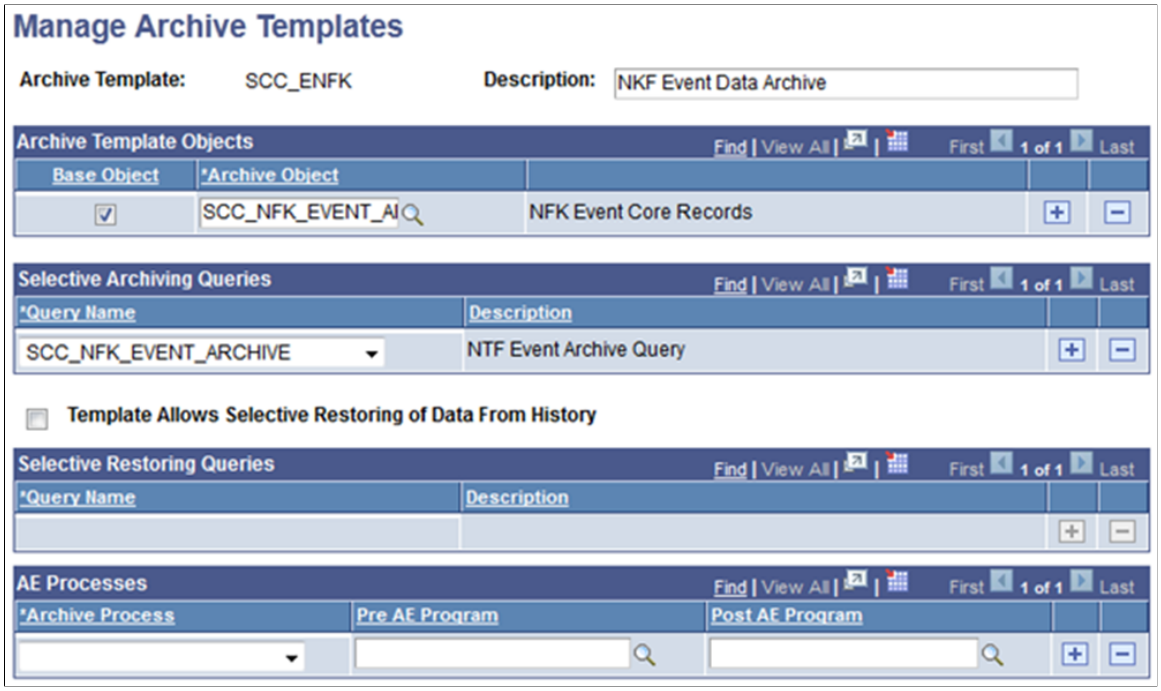
This example illustrates the fields and controls on the Manage Archive Object page for NFK Events.



Access the Manage Archive Templates page (PeopleTools, Data Archive Manager, Manage Archive Templates).

Image: Manage Archive Templates page for NFK Events

This example illustrates the fields and controls on the Manage Archive Templates page for NFK Events.



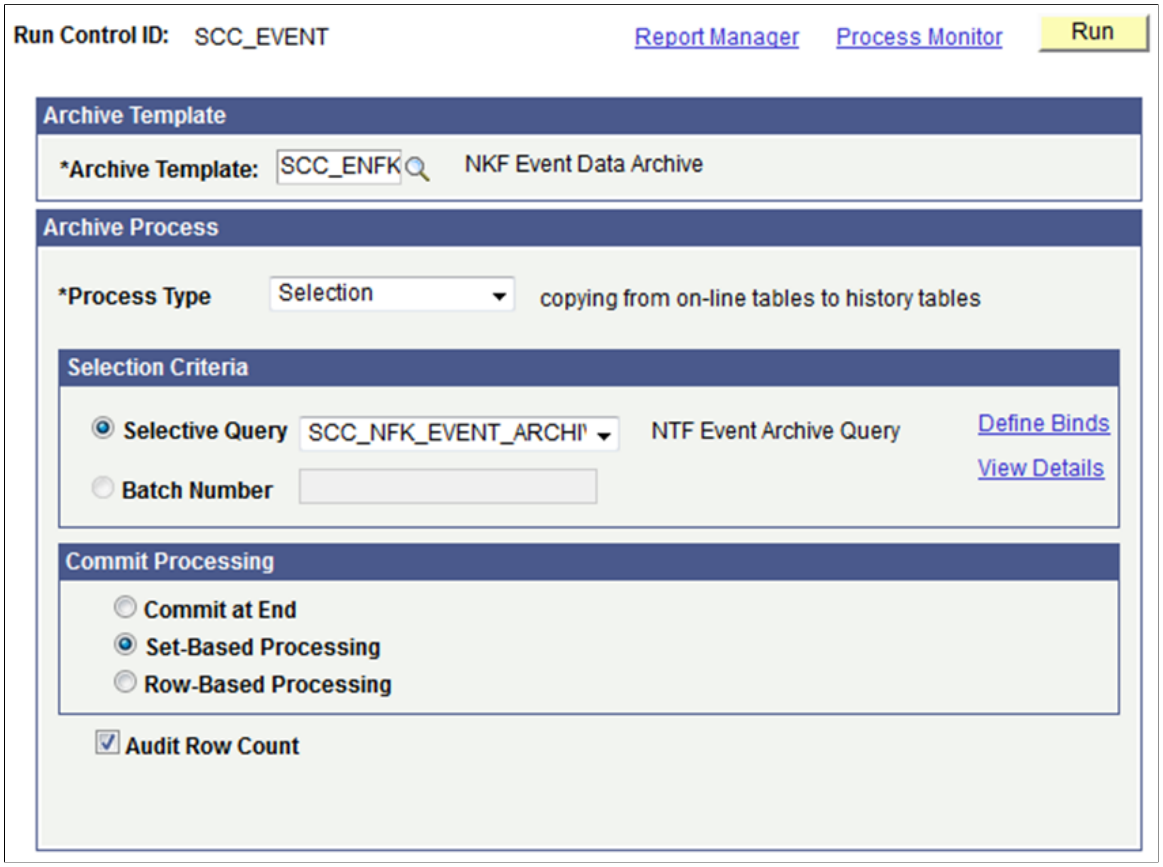
Access the Archive Data To History page (PeopleTools, Data Archive Manager, Archive Data To History).

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

To run the Archive/Purge process, create a run control ID, run the SCC_ENFK archive template with the following options as shown below:

Image: Archive Data To History page for NFK Events

This example illustrates the fields and controls on the Archive Data To History page for NFK Events.



Define the date until when the Archive is required and set check boxes to select read and unread events using the Define Binds link.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.