

Advanced Email Notification Variables

Document ID: Q000068

Last Revised On: February 20, 2008

This article applies to the following:

Product Version: IssueNet 4.0 and later

Component(s): Administrator

Solutions(s): All

If you have reviewed the default notification actions that ship with IssueNet solutions, you have seen the use of basic variables to specify notification recipients. Default issue notification actions routinely use variables such as: `$(CurrentObject.PrimaryContactID)` or `$(CurrentObject.SubmittedByID)` to notify contacts specified on the issue form. In addition to these basic variables, the IssueNet Platform supports a variety of other variable forms for sending notifications to different groupings of contacts. For example, notification variables can be used to notify contacts based on the group membership of the users they are linked to, based on a link to an issue or folder, or based on an attribute such as contact type or company name.

Selecting Notification Recipients Based on Contact Relationships

The key to using advanced notification variables is to remember that any variable which returns contacts can be used to specify notification recipients. If a variable returns a collection of contacts, the IssueNet platform will send notifications to all of them. As a result, you can use variables which get contacts based on relationships - in addition to contacts which are specified as field values.

For example, if you want to send a notification to any contact linked to a current object such as an issue, you can get the contacts by specifying the relationship name in the variable:

`$(CurrentObject.Contacts)`

The variable above will notify all of the contacts linked to the current object. However, if you want to notify specific contacts, you can use the “[]” operator to select specific contact objects:

`$(CurrentObject.Contacts[Company='Elsinore'])`

`$(CurrentObject.Contacts[@Reason='Send Notification'])`

The first variable will notify contacts based on the value of the Company property of the contact class. The second variable will notify contacts based on the value of the reason code for the link. The reason code is the optional value that allows you to indicate why two items are linked. By using variables like the ones above, you can provide your users with a simple means to specify ad hoc notification recipients by linking contacts to an issue.

Similarly, you can use a variable to look up contacts based on an objectID. For example, some organizations want the ability to notify contacts related to the parent folder of an issue. This method would allow different sets of notification recipients to be specified for each issue folder. The following variable notifies any contacts linked to the parent folder of an issue:

`$(CurrentObject.ParentID->Contacts)`

In this variable we have used the '->' operator to get the parent folder based on the parent ID property and then selected the contacts from the folder. Please note that this variable will not inherit to Child Projects. This variable can also be combined with other operators to select specific contacts. The following variable will notify contacts with the company value "Elsinore" on the parent folder of an issue.

`$(CurrentObject.ParentID->Contacts[Company='Elsinore'])`

Selecting Notification Recipients Using Query Results

Variables also provide a way to select contacts that are not related to the current object through a property or relationship. The variable syntax allows you to specify contacts using the results of a query. For example, if you wanted to notify any contact in the database where the company value is "Elsinore" you would use a variable like the following:

`$(QueryResults[Name='Name of Query That Return Contacts with the Company Named Elsinore'])`

Some organizations want to notify contacts based on the group membership of the users they are linked to. In that case you can use a query in a variable to get the required groups and then use the standard variable syntax to get the contacts based on relationships between groups, user and contacts. The following is an example of a variable that notifies contacts based on user's group membership.

`$(QueryResults[Name='Name of Query which Fetches Required Groups'].Users.Contact)`

Additional Information

The example variables in this tech tip reference properties of the Current Object. However, these same techniques can be used when using workflow variables such as WorkflowIssue and WorkflowTask which are accessible when executing workflow transitions.

Additional Resources Related to Notifications

- [IssueNet Notification Rules \(Tech Tip\)](#)
- [Using the Pre Tag in Notification Templates \(Tech Tip\)](#)
- [IssueNet Notification Configuration \(Tech Tip\)](#)