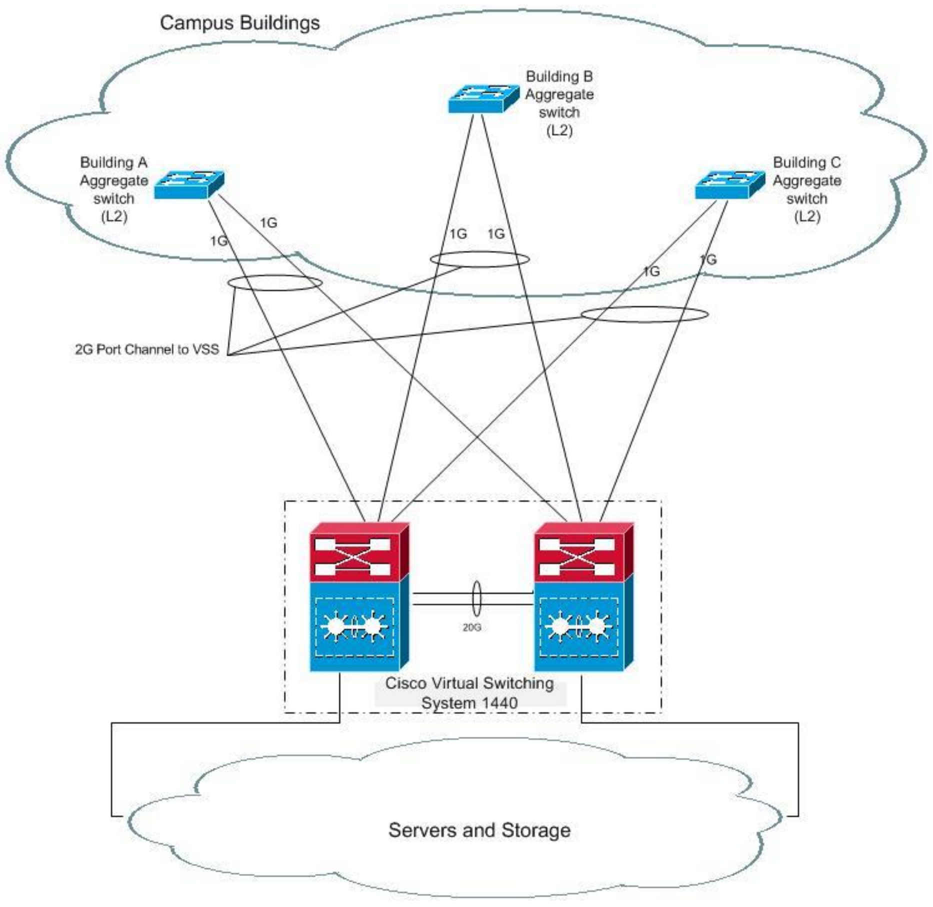
Using Custom Properties to Simplify & Improve Alerting

In any modern medium to large scale network, the sheer volume of information available via monitoring solutions such as Orion Network Performance Monitor (NPM) can make it difficult for network administrators to receive alerts that are of interest to them whilst keeping unwanted alerts to a minimum. In addition, multiple levels of redundancy mean the failure of an individual component may go unnoticed as the service that the component supports remains “up” as far as ICMP and SNMP monitoring is concerned.

This document is intended as a guide to using the Custom Properties available in NPM to clearly identify (or “tag”) components of interest and then configure alerts based on these properties. We use this method extensively to alert on various aspects of our infrastructure; however for this document I will use the following example:

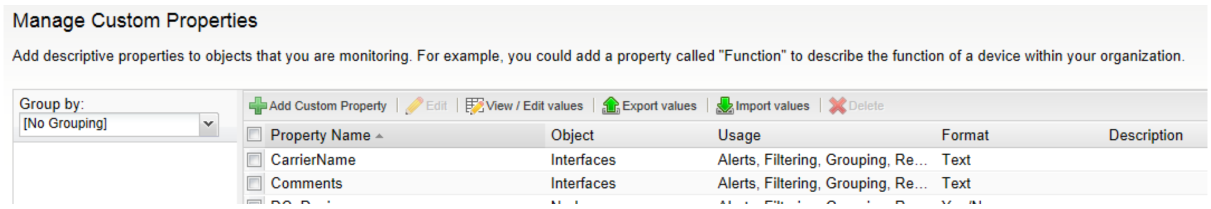

Campus Buildings
Building B Aggregate switch (L2)
Building A Aggregate switch (L2)
Building C Aggregate switch (L2)
1G
1G
1G
1G
1G
1G
2G Port Channel to VSS
20G
Cisco Virtual Switching System 1440
Servers and Storage

In the diagram, we have Campus building aggregate switches running L2 functionality linked to a Cisco VSS 1440 system acting as the Layer 3 core of the network (the VSS 1440 systems are effectively 2x Catalyst 6509 units configured as a single logical unit if you are not familiar with them). Each aggregate switch has a 1Gb link to each of the 6509 chassis, and these are configured as a Port Channel giving a 2Gb link from building to core. This is a great topology for resilience: should any of the individual 1Gb links fail, traffic will automatically pass via the other member of the Port Channel interface with no disruption to service (and no nasty spanning tree elections either ☺)

However, how do we alert our network team when one of these 1Gb links fails? Remember that the aggregate switch will remain up and operational as far as NPM is concerned because our Port Channel interface is still up, and no spanning tree election means that our end users will not notice (and hence complain!). We could set an alert when every 1Gb interface changes status, but if our access switches use 1Gb ports then we will be drowned in emails/texts/pages. Custom Properties is the answer, and here's how to do it.

Step 1: Creating a Custom Property

It's important to remember that a Custom Property in NPM is merely a description that you can apply to an interface, volume or node. Creating them is easy, simply go to **Settings** in your NPM web console and select **Manage Custom Properties** from the **Node & Group Management** section:

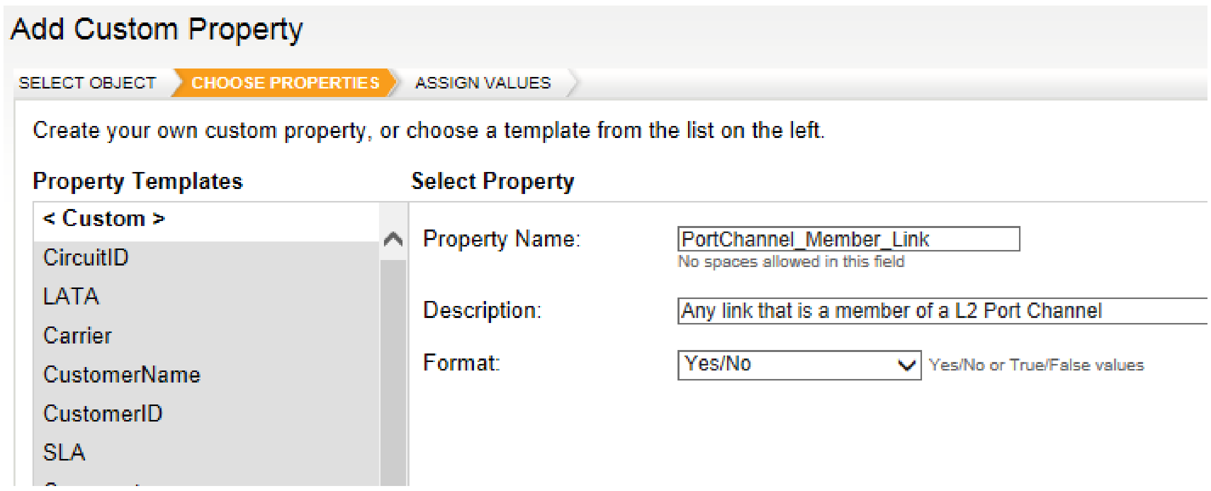

Manage Custom Properties

Add descriptive properties to objects that you are monitoring. For example, you could add a property called "Function" to describe the function of a device within your organization.

Group by: [No Grouping]

Property Name	Object	Usage	Format	Description
CarrierName	Interfaces	Alerts, Filtering, Grouping, Re...	Text	
Comments	Interfaces	Alerts, Filtering, Grouping, Re...	Text	

Select **Add Custom Property** and you will be asked what object type your custom property should be based on (and hence applied to). For this example we'll need to select **Interfaces**. Click on Next and you will be asked to enter some details. As this will be a simple yes/no property (i.e. is the interface a member of a port channel or not) we'll enter the details like this:


Add Custom Property

Create your own custom property, or choose a template from the list on the left.

Property Templates

Select Property

Property Name: PortChannel_Member_Link
No spaces allowed in this field

Description: Any link that is a member of a L2 Port Channel

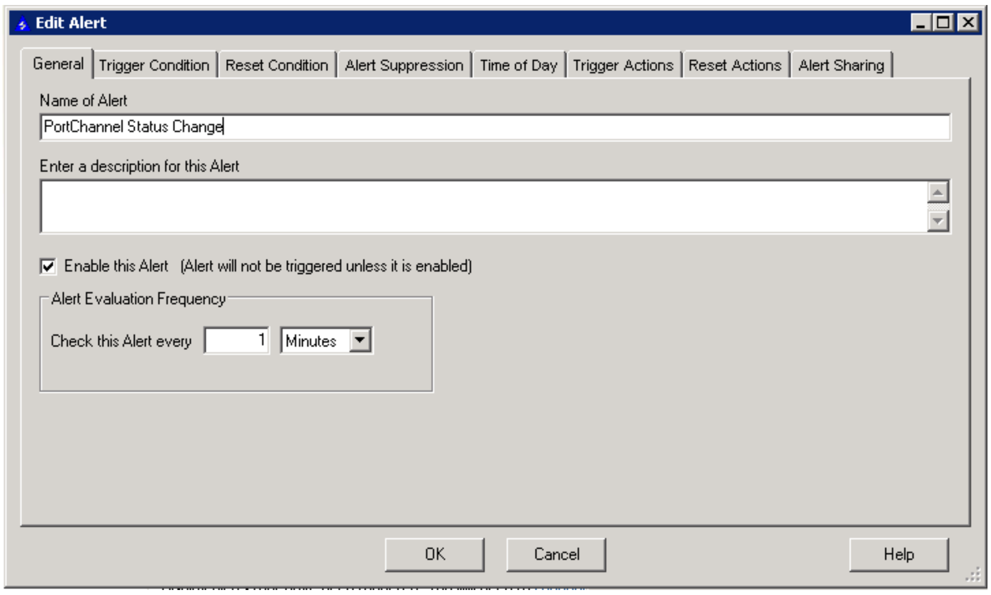
Format: Yes/No — Yes/No or True/False values

Click on **Next** and you'll be asked if you want to add the custom property that you've created to some interfaces. We'll skip this step for now, as we need to configure an alert for this property and then find a suitable victim to test it on. Simply click on **Submit** and the new property will appear in the list.

Step 2: Create an Advanced Alert for the Custom Property

For this step we'll need to be on the Orion Polling Server. Click on **Start > All Programs > SolarWinds Orion > Alerting, Reporting and Mapping > Advanced Alert Manager**:

Click on **Configure Alerts** then click on the **New** button in the Manage Alerts dialog box. Give the alert a meaningful name:



Name of Alert: PortChannel Status Change

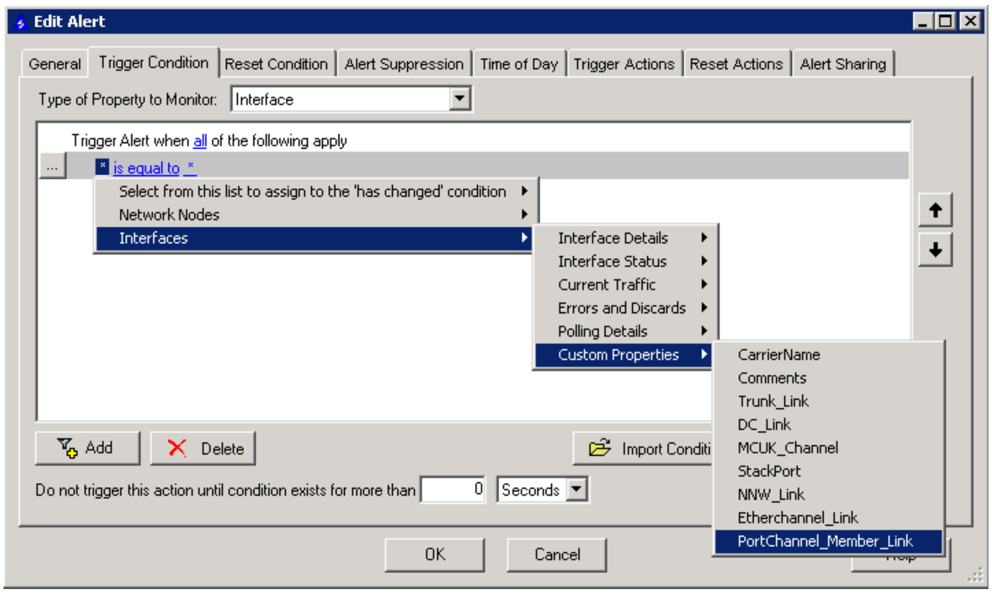
Enter a description for this Alert

☑ Enable this Alert (Alert will not be triggered unless it is enabled)

Alert Evaluation Frequency

Check this Alert every 1 Minutes

In the **Trigger Condition** tab, change the **Type of Property to Monitor:** to **Interface** and then add a simple condition. Select **Interfaces | Custom Properties | PortChannel_Member_Link** (or whatever you named your custom property):



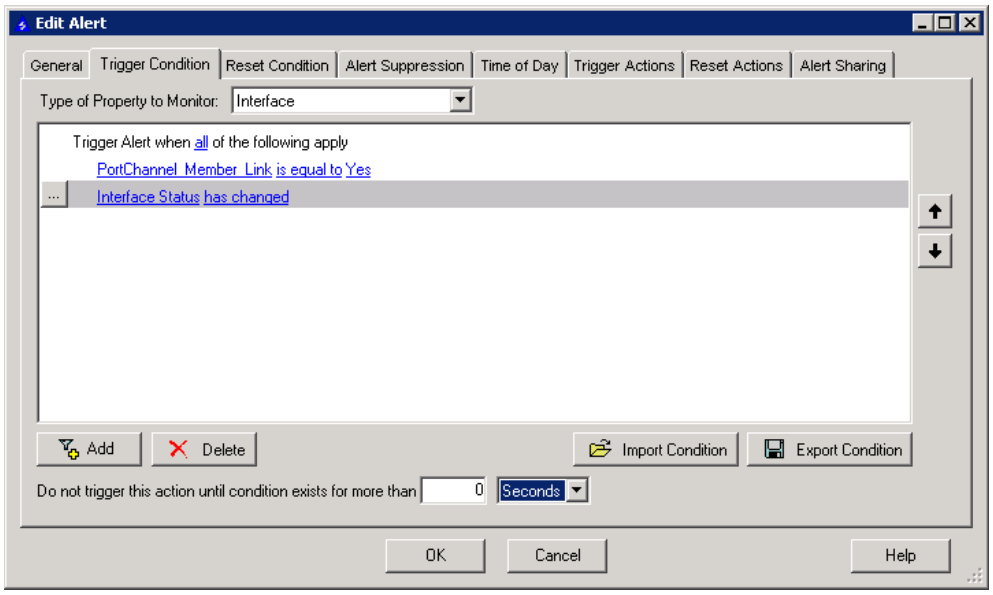
Type of Property to Monitor: Interface

Trigger Alert when all of the following apply

... is equal to *

Do not trigger this action until condition exists for more than 0 Seconds

And then set is equal to Yes. Add another simple condition to choose the event you want to report on. For this example I want to be alerted whenever a port channel member link changes status, so I've selected **Interface Status Has Changed:**

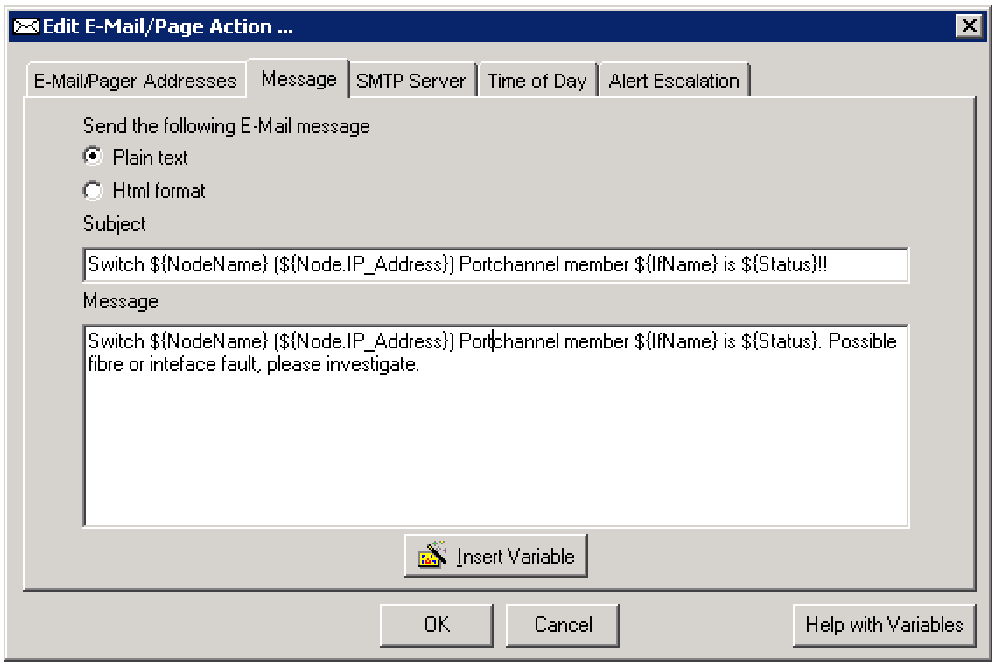


Type of Property to Monitor: Interface

Trigger Alert when all of the following apply
PortChannel_Member_Link is equal to Yes
Interface Status has changed

Do not trigger this action until condition exists for more than 0 Seconds

For the example we'll leave the **Reset Condition**, **Alert Suppression** and **Time of Day** options as the default (we can tune them later if needs be). In the Trigger Actions tab, we'll create an email action that will alert the network operations team:



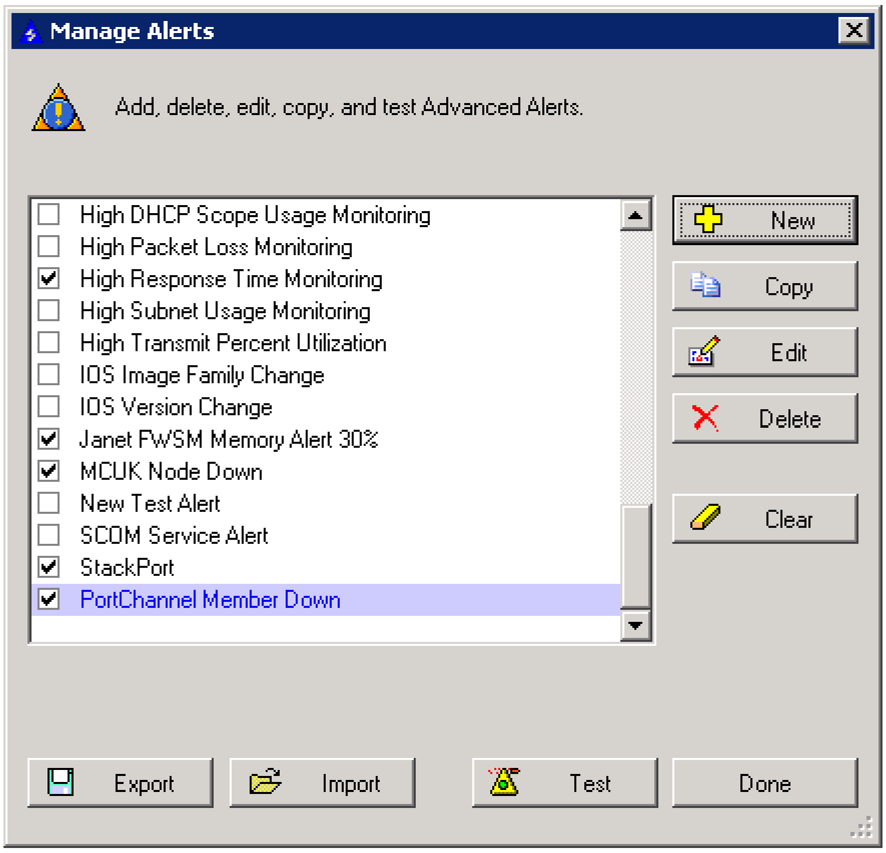
Send the following E-Mail message
[✓ Plain text]
Html format

Subject
Switch ${NodeName} (${Node.IP_Address}) Portchannel member ${IfName} is ${Status}!!

Message
Switch ${NodeName} (${Node.IP_Address}) Portchannel member ${IfName} is ${Status}. Possible fibre or inteface fault, please investigate.

This email will show the Switch Name, the IP address and the Interface name of our Portchannel member and ask the team to investigate. You may also wish to add a NetperfMon event action as well.

Finish configuring your SMTP settings etc and click on OK until you are returned to the Manage Alerts screen, and make sure that your alert is enabled:

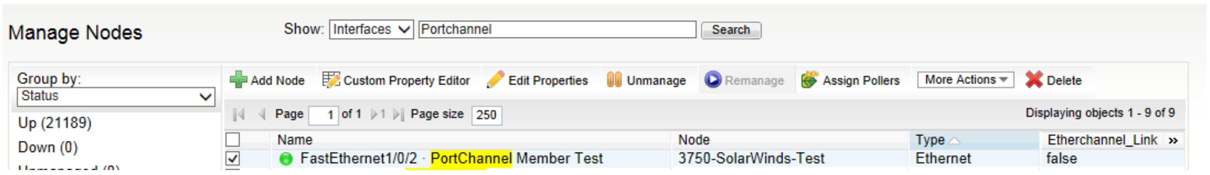


- [ ] High DHCP Scope Usage Monitoring
- [ ] High Packet Loss Monitoring
- [x] High Response Time Monitoring
- [ ] High Subnet Usage Monitoring
- [ ] High Transmit Percent Utilization
- [ ] IOS Image Family Change
- [ ] IOS Version Change
- [x] Janet FWSM Memory Alert 30%
- [x] MCUK Node Down
- [ ] New Test Alert
- [ ] SCOM Service Alert
- [x] StackPort
- [x] PortChannel Member Down

Step 3: Assigning the Custom Property to an Interface and Testing

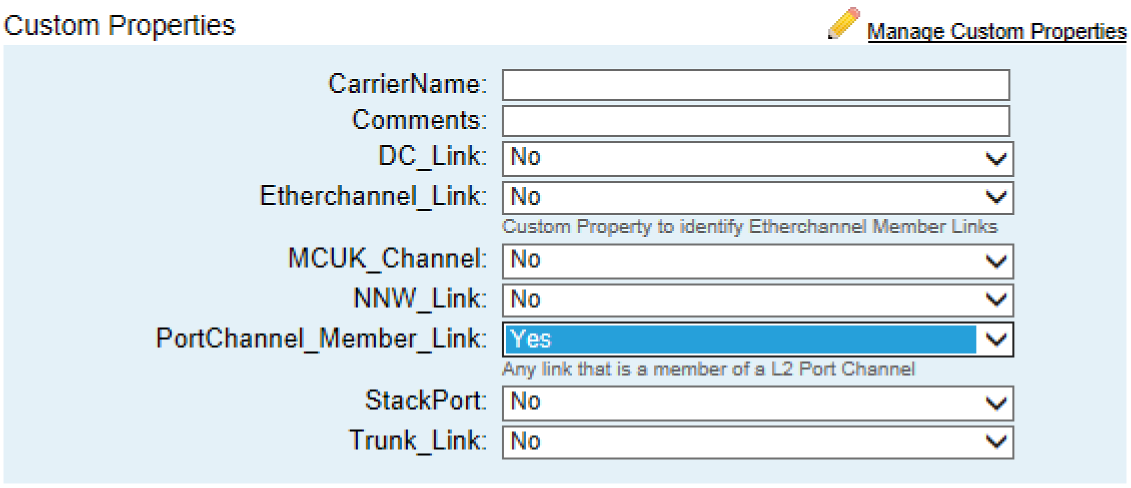
Now that we've created our custom property and an alert based on it, we will need to assign the property to an interface and test that the alert triggers successfully (I am aware that Alerts can be tested from the Manage Alerts box shown in step two, but I'm a firm believer in actually pulling physical links to test things ☺)

From the Manage Nodes page of your NPM web console, identify a suitable candidate interface for testing. I won't discuss how to identify the relevant interfaces here, but if you've given suitable descriptions to the "important" interfaces you wish to monitor carefully (you have haven't you?!?) it should be fairly easy to find. Select the interface you wish to test:



	Name	Node	Type	Etherchannel_Link
✓	FastEthernet1/0/2 · PortChannel Member Test	3750-SolarWinds-Test	Ethernet	false

And click on **Edit Properties**. You'll see that you now have an option to identify this interface as a Port Channel member in the **Custom Properties** section:

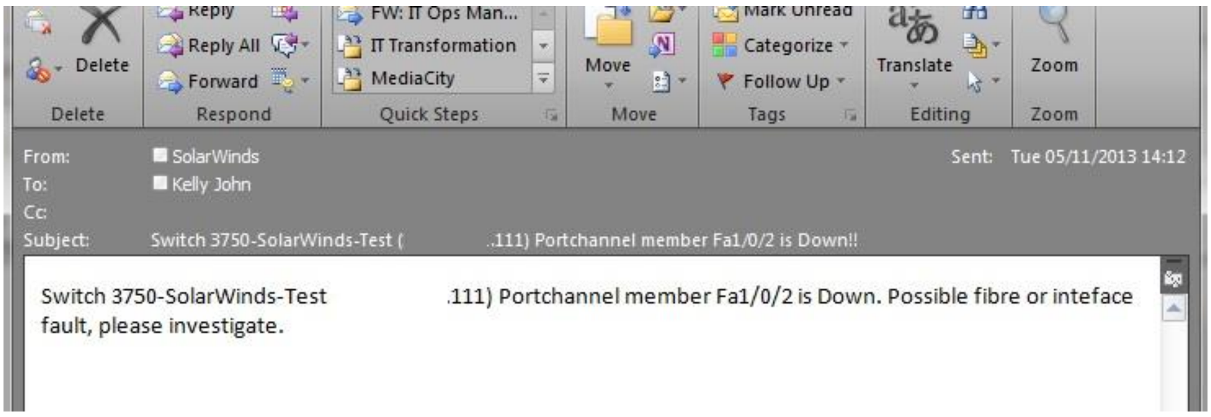

Custom Properties

Manage Custom Properties

CarrierName:	
Comments:	
DC_Link:	No
Etherchannel_Link:	No Custom Property to identify Etherchannel Member Links
MCUK_Channel:	No
NNW_Link:	No
PortChannel_Member_Link:	Yes Any link that is a member of a L2 Port Channel
StackPort:	No
Trunk_Link:	No

Change this to **Yes**, and then click on **Submit**.

To test the alert, simply pull the plug on your Interface ☺ If everything is setup correctly, you should receive an email (and see an alert in NPM if you also set this up as a trigger action):



From: SolarWinds
To: Kelly John
Cc:
Subject: Switch 3750-SolarWinds-Test (.111) Portchannel member Fa1/0/2 is Down!!
Sent: Tue 05/11/2013 14:12

Switch 3750-SolarWinds-Test .111) Portchannel member Fa1/0/2 is Down. Possible fibre or inteface fault, please investigate.

Once you are happy after testing that your alert is working successfully (and going to the correct recipients) all you need to do is apply the Custom Property to the interfaces you want to monitor. For our example, we'd apply it to the two 1Gb interfaces on the building aggregate switches as well as the corresponding interfaces on the VSS1440 chassis; remember that there is no need to add this property to the Port Channel interfaces themselves (i.e. *interface Po1* in Cisco IOS), just those physical interfaces that are members of the port channel.

Other Uses of This Method

This method of alerting on Custom Properties can be applied to almost any situation whereby you wish to be alerted specifically when a particular group of interfaces or nodes change status or does something of interest. To give you some ideas, we use it for the following:

- **Monitoring member switches of a Cisco 3750 Stack:** as multiple switches are grouped into a single logical switch (with a single management IP address) it can be difficult to know when one of the member switches fails or reloads as the management IP address remains up and available to poll. Applying a custom property to the Stack-Port interfaces of these switches allows the network team to be alerted when a member switch fails.

- **Specifying interfaces for threshold alerts:** for some specific interfaces, it can desirable to alert when the utilisation reaches a certain percentage that is lower than you would have set for the rest of your network.

- **Sending alerts for a group of nodes/interfaces to other interested parties:** For example alerting the physical security team whenever an interface connecting a security camera goes down.