Ensemble Alerting

Simple alerting in Ensemble does not give us as many options as using Managed alerts. With Simple Alerting we cannot control which errors we should get and what times we should be getting the errors. Think of Managed Alerts like using NotificationCFG.dat in eGate.

Ensemble documentation on [Monitoring Alerts](http://intengtest:57772/csp/docbook/DocBook.UI.Page.cls?KEY=EMONITOR_alerts#EMONITOR_C233140).

# Introduction to Alerts

Alerts provide a way for Ensemble to automatically notify users about a serious problem or condition that requires a quick resolution to ensure that the production continues operating normally. When properly configured Ensemble generates alerts for potentially serious problems and should not generate any alerts caused by normal variations in the production performance.

The other monitoring features described in this document require the user to actively check for a problem. Typically, users only check for a problem after it has had a noticeable impact on the production’s performance. Once you have configured alerts to notify users automatically, alerts may make it possible to resolve issues before they have a serious impact on performance and become critical problems.

Ensemble can automatically send an alert message to users when specified thresholds are exceeded, when specified events occur, or when user code explicitly generates alerts. You can process alerts in a number of ways:

* Alerts are only written to a log file and there is no automatic notification.
* A simple alert notification system, where all alerts get sent to a list of users.
* Routing alert notification where selected alerts are sent to different users depending on the kind of alert and the component that generated it.
* An alert management framework that notifies users about alerts, escalates unresolved alerts, and documents the current state and history of actions taken to resolve the alert.

In configuring any alert notification system, it is important to calibrate the level that triggers the alert and ensure that the users being notified understand the alert and know how to respond. If the trigger level is set too high, problems may already have a significant impact on performance before Ensemble notifies users. But, if the trigger level is set too low, Ensemble sends out many notifications during the normal operation of the production and users tend to ignore these notifications and may not respond to the few critical ones among them.

Alerts are messages of type **Ens.AlertRequest** that can be generated by any business service, process, or operation in a production. Ensemble always stores alerts messages in the log. If there is a production component named Ens.Alert, Ensemble sends all alert messages to it. Ens.Alert typically is an operation, such as **EnsLib.Email.AlertOperation**, a routing process, or the Alert Manager, which has the class **Ens.Alerting.AlertManager**.

There are many components to Managed Alerts within Ensemble.

#### Services

Ens.Alerting.AlertMonitor

#### Processes

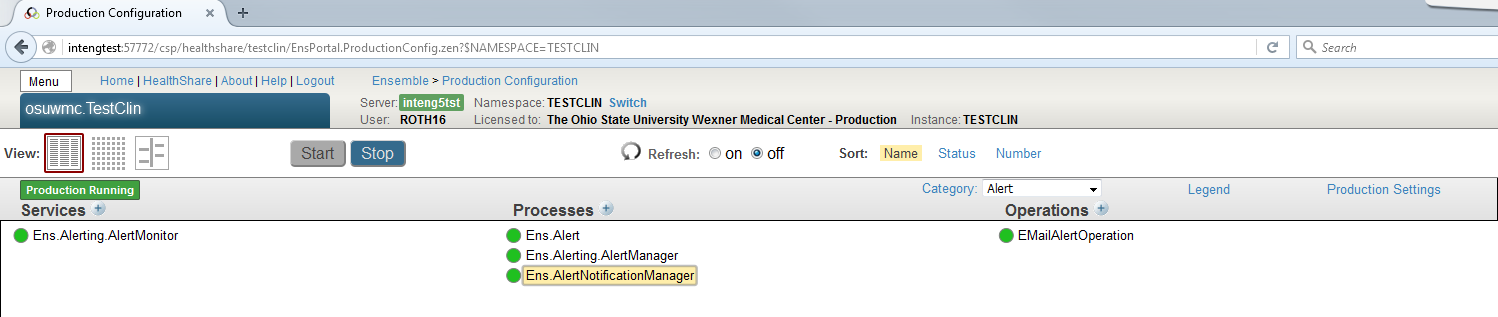
Ens.Alert

Ens.Alerting.AlertManager

Ens.AlertNotificationManager

#### Operations

EMailAlertOperation

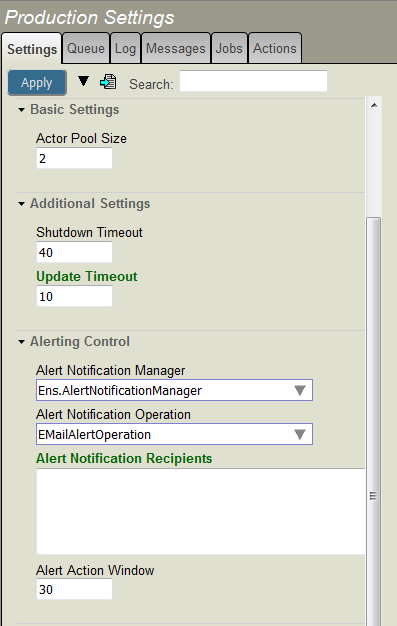


# Settings

There are a couple of settings that need set in order for Alerting to work.

## Production Settings

The following need to be set at the Production Settings level



Alert Notification Manager

* Enter the name that you specified for the Notification Manager.

Alert Notification Operation

* Enter the name you specified for the email alert operation.

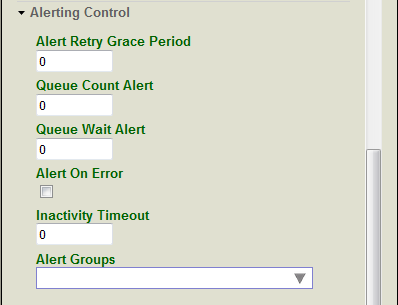
Alert Notification Recipients

* Enter valid email addresses, separated by commas. By default, the alert management sends all alert notifications to this email address list. If you use the Alert Notification Manager rule to specify destinations for the messages, this field is not used. Perhaps we set this to [\_isinterfacegroup\_@osumc.edu](mailto:_isinterfacegroup_@osumc.edu).

### Alert Action Window

* Enter a number of minutes. This is the default number of minutes that a user has to resolve and close an alert before the next reminder message is sent. If you include the Alert Monitor in your production, by default it sends a reminder after the specified number of minutes have expired and resets the notification time for the managed alert by adding the specified number of minutes to the current time.

## Service/Processes/Operations



### Alert On Error

* Check this option if you want the component to generate an alert when it encounters an error. Other production settings control what conditions are considered errors. These settings are described later in this section.
* Do not check Alert On Error on any component that is involved in delivering or processing alerts.

### Alert Grace Retry Period

* This is a time in seconds that the component will retry sending its output before issuing an alert. This setting is most commonly used for business operations. If the component is retrying and eventually succeeds within this time there will be no alert. Setting this to a value such as 60 seconds will suppress alerts on transient issues such as a dropped network connection that fixes itself, but won’t wait too long before alerting you to a real problem.

### Inactivity Timeout

* This is a time in seconds that the component will wait for a request before issuing an alert. This is typically used for business services. If you set it to a value such as 300 seconds for a busy system you will be alerted fairly quickly if you stop receiving messages. For quieter systems you might want a longer interval. This value applies all day, so you might get false positives on off-peak times, when traffic is much lower than normal, but you can filter these in your alert handler if necessary.

### Queue Wait Alert

* This is a time in seconds that a message can stay on a queue before it generates an alert. Setting this to a value such as 300 seconds would be a good starting point. For critical systems, 5 minutes might be too long, but for other less critical systems a longer interval may be appropriate.

### Queue Count

* Specifies the number of items on a queue. When the queue size reaches this number Ensemble generates an alert. Setting this to a value such as 10 causes an alert when a queue starts to build on a critical item indicating some delay. But some queues may get this large during normal operation and if a component receives a nightly batch of work, setting this parameter may cause unnecessary alerts.

### Alert On Bad Message

* In routers that provide validation, such as the HL7 router, this setting specifies that any message that fails the specified validation generates an alert. This alert is in addition to sending the original message to the bad message handler.