



Email Integration



- LightSpeed PM – A Certified Quest Partner

Quest™

Updated 11-19-2019

Foglight Integrations – Current OOTB Status

- Foglight allows email notification, but is very restrictive with regard to controlling what is sent, how it is sent, and what it looks like
 - Email action per rule and per severity.
 - Limited to registry variables being scoped on individual objects.
 - Not possible to easily assign multiple email targets based on service.
 - Email management difficult and many use cases require lots of groovy scripting to implement.
 - Event Driven Rule
 - Limited to forwarding emails based on custom implemented logic.
 - Very cumbersome to define and enforce total and precise control.
 - Lots of groovy scripting required.
 - Hard to implement using one rule.
 - No support for Email escalation

Email Integration – Highlights

- Our Custom Email Integration simplifies management of Email dispatching within Foglight:
 - Customizable Subject, message and recipients per severity
 - Support for text and HTML emails
 - Time based escalation per severity.
 - Email dispatching to multiple dynamic targets per severity.
- Advanced Rich UI for all functionalities
- **Remote Administration** of solution across servers
- Decision engine for total and precise control
 - 20 Levels of granularity
 - Severity Level control

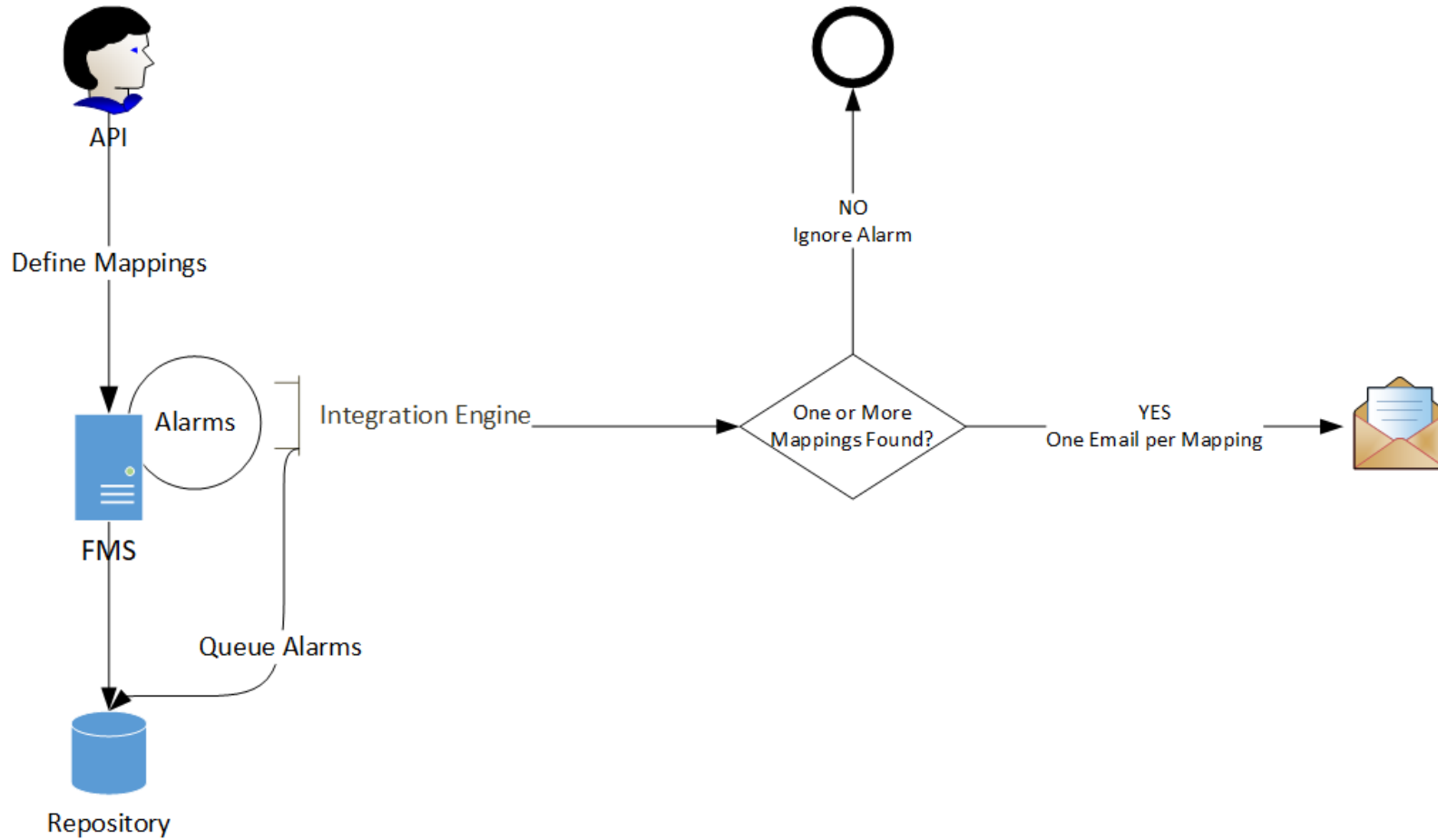
Email Integration – Highlights

- Pattern editor to easily access many of the information related to the alarm. (No groovy required)
 - Access to dozens of fields to easily provide values to parameters
 - A dozen common operators available to easily avoid any coding
- Use of groovy scripts for power users
 - Use of groovy scripting with complete injected flow context
 - Ability to reject/delay ticket generation
- Persisted Queue so no events are lost
 - Guaranteed delivery
 - Keep history for rejected, failed and successful operations for total control
 - Pause or delay control
 - Consumer Thread settings/control as to manage concurrency and volume.

Email Integration – Highlights

- Alarm Transition Support
 - Can be turned ON or OFF
 - Update tickets when alarm is transitioning from one severity to another.
 - Can be either to higher severity only or any transition
 - Close ticket only when the last alarm in the transition has been cleared
 - All transition information is accessible from the Pattern Editor
- Alarm Flood Prevention
- Auto Acknowledge Alarms

Email Integration –Flow



Email Integration – Decision Engine

- The decision engine is based on various mappings defined within the solution. A very rich UI is provided to create the mappings.

- Low Priority

	Service	Host	Agent Type	Agent	Topology Object	Rule
0						
1		x				
2			x			
3				x		
4						x
5					x	
6		x				x
7			x			x
8				x		x
9					x	x
10	x					
11	x	x				
12	x		x			
13	x			x		
14	x					x
15	x				x	
16	x	x				x
17	x		x			x
18	x			x		x
19	x				x	x

No Service

With Service

- High Priority

Email Integration – Changing Settings

Integration Pack Registry Settings - Local FMS

Navigation: Expert View

Bookmarks: There are no bookmarks

Homes: Local FMS

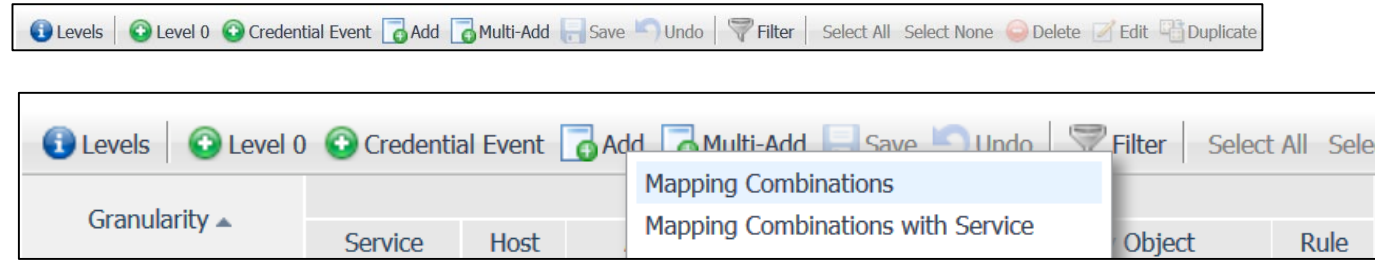
Dashboards: Foglight Servers

Host Name: There Is No Data To Display

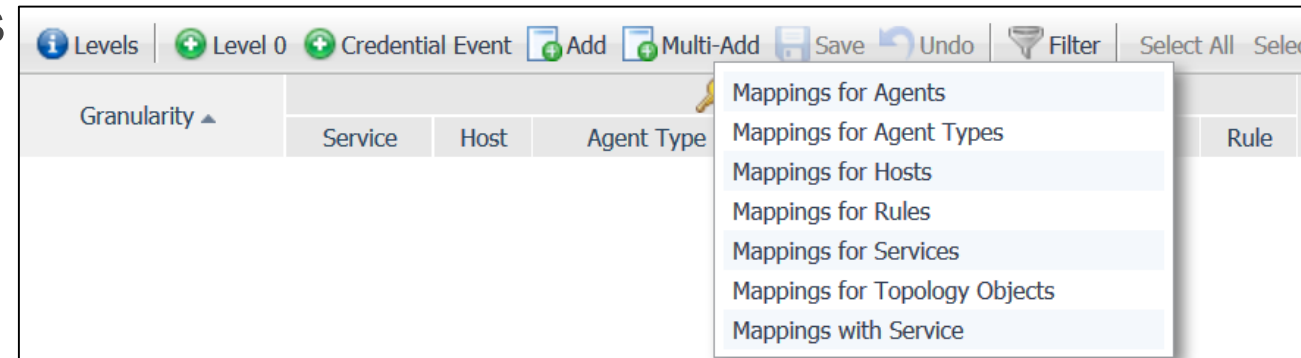
Name	Value	Scoping	
Email Integration			
Open Ticket			
PSO.EmailIntegration.OpenTicket.NoEmptyValue	true		If true any tag that returns a null will be replaced by a space. If false null values are simply not outputted.
PSO.EmailIntegration.OpenTicket.TimeZone	GMT		Defines the time zone to use when outputting date and time in the Command.
PSO.EmailIntegration.OpenTicket.DateTimeFormat	dd/MM/yyyyHH:mm:ss		Defines the time zone to use when outputting date and time in the Command.
Queue			
Ignore			
PSO.EmailIntegration.Queue.Ignore.Normal	true		Should cleared alarms be ignored by the queuing engine? Ignored alarms are not queued thus reducing the overall load on the system.
PSO.EmailIntegration.Queue.Ignore.Warning	false		Should Warning alarms be ignored by the queuing engine? Ignored alarms are not queued thus reducing the overall load on the system.
PSO.EmailIntegration.Queue.Ignore.Critical	false		Should Critical alarms be ignored by the queuing engine? Ignored alarms are not queued thus reducing the overall load on the system.
PSO.EmailIntegration.Queue.Ignore.Fatal	false		Should Fatal alarms be ignored by the queuing engine? Ignored alarms are not queued thus reducing the overall load on the system.
PSO.EmailIntegration.Queue.Ignore.Rules			A list of comma separated rule names that the queue should ignore. Alarms generated by ignored rules are not queued thus reducing the overall load on the system.
Delay			
PSO.EmailIntegration.Queue.Alarm.DelayPeriod.Warning	0		Time in minutes a warning alarm should be delayed in the queue before being forwarded to the target system if still active (not cleared).
PSO.EmailIntegration.Queue.Alarm.DelayPeriod.Critical	0		Time in minutes a critical alarm should be delayed in the queue before being forwarded to the target system if still active (not cleared).
PSO.EmailIntegration.Queue.Alarm.DelayPeriod.Fatal	0		Time in minutes a fatal alarm should be delayed in the queue before being forwarded to the target system if still active (not cleared).
PSO.EmailIntegration.Queue.AlarmTransition.Enable	false		Should alarm transition be supported. If enabled the integration will update a ticket vs creating a new one when an alarm goes from one state to another.
PSO.EmailIntegration.Queue.Alarm.AcknowledgeUponSuccess	false		Should the alarm be acknowledged when the alarm has been successfully processed.
PSO.EmailIntegration.Queue.Alarm.EnforceServiceFilter	false		When an alarm is being processed enforce the filters applied to the service. If true a service is not considered to be a hit (mapping).
PSO.EmailIntegration.Queue.PausePeriod	0		Time in minutes no alarms should be forwarded to the target system. This is useful when the target system is unavailable due to maintenance.
Carbon Copy			
PSO.EmailIntegration.CarbonCopy.Fatal			Comma separated list of recipients to CC the email to when a mapping is found and the alarm's severity is fatal.
PSO.EmailIntegration.CarbonCopy.Critical			Comma separated list of recipients to CC the email to when a mapping is found and the alarm's severity is critical.
PSO.EmailIntegration.CarbonCopy.Normal			Comma separated list of recipients to send the email to when a mapping is found and the alarm's severity is normal.
PSO.EmailIntegration.CarbonCopy.Warning			Comma separated list of recipients to CC the email to when a mapping is found and the alarm's severity is warning.
Escalation			
PSO.EmailIntegration.Escalation.CarbonCopy.Fatal			Comma separated list of recipients to CC the email to when a mapping is found, the alarm's severity is fatal and the alarm has been acknowledged.
PSO.EmailIntegration.Escalation.Recipient.Fatal			Comma separated list of recipients to send the email to when a mapping is found, the alarm's severity is fatal and the alarm has been acknowledged.
Recipient			
PSO.EmailIntegration.Recipient.Fatal			Comma separated list of recipients to send the email to when a mapping is found and the alarm's severity is fatal.
PSO.EmailIntegration.Recipient.Critical			Comma separated list of recipients to send the email to when a mapping is found and the alarm's severity is critical.
PSO.EmailIntegration.Recipient.Warning			Comma separated list of recipients to send the email to when a mapping is found and the alarm's severity is warning.
PSO.EmailIntegration.Recipient.Normal			Comma separated list of recipients to send the email to when a mapping is found and the alarm's severity is normal.
Multiple Mappings			
PSO.EmailIntegration.MultipleMappings.Enable	false		Should the solution send emails to all the alarm mappings (true) that are a hit for an alarm or should it only send it to the alarm mapping that triggered the alarm.

Email Integration – Mappings Editor

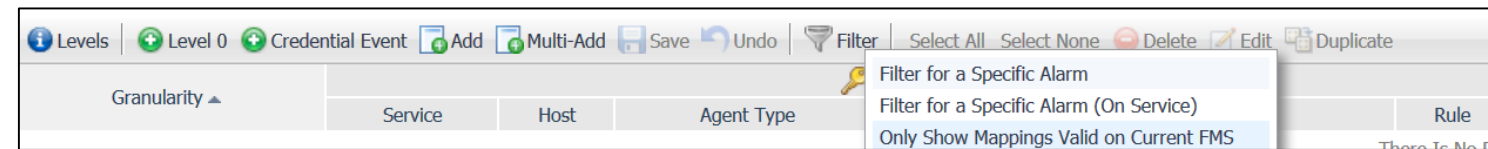
- Combinations



- Multiple Mappings




































- Filtering



- . . .

Email Integration – Field Editor

Subject	: @EM_ESCALATION_SEVERITY?[Escalation @EM_ESCALATION...			
Message	: @ALARM_MESSAGE			
Escalation Severity	: def alarm = args [1] def mapping = args [2] ...			
Escalation Severity Name	: def tagToValueMap = args [3] switch (tagToValueM...			
Recipient	: def alarm = args [1] def mapping = ...			
CC	: def alarm = args [1] def mapping = ...			
Content Type	: text/plain			
Escalation Time Warning 				
Critical 				
Fatal 				

Email Integration – Pattern Editor

The screenshot shows the 'Alarm Severity Name Elements' window. On the left, a 'Pattern Element' list contains '@ALARM_SEVERITY_NAME'. The main editor area contains the text '@ALARM_SEVERITY_NAME'. Below the editor is an 'Apply' button. At the bottom, there are two panels: 'TAGS' and 'OPERATORS'.

Remember to click Apply if you manually edit the pattern.
Spaces are automatically trimmed, please use Edit Complete to insert spaces and have them preserved.

@ALARM_SEVERITY_NAME

Apply

TAGS

Tag	Caption
AGENT_TYPE	Agent Type
AGENT_UID	Agent UID
ALARM_ACKNOWLEDGED_BY	Alarm Acknowledged By
ALARM_ACKNOWLEDGED_TIME	Alarm Acknowledged Time
ALARM_CLEARED_BY	Alarm Cleared By
ALARM_CLEARED_TIME	Alarm Cleared Time
ALARM_CREATED_TIME	Alarm Created Time
ALARM_ID	Alarm ID
ALARM_IS_ACKNOWLEDGED	Is Alarm Acknowledged
ALARM_IS_CLEARED	Is Alarm Cleared
ALARM_LINK	Alarm Link

OPERATORS

Operator	Short Description
?[]	If tag has a value.
[]	If tag does not have a value.
?[]:[]	If tag has a value else.
^{}{}	Replace string with another.
=0	Equality
=0?[]:[]	If equal else
![[]]	Groovy Script
^U	Uppercase
^L	Lowercase
^T	Trim
^N	New Line Output
^R	Return

Email Integration – System Requirements

- **Minimum required FMS version**

5.9.3

- **Supported Databases**

Microsoft SQL

Oracle

MySQL

PostgreSQL

Minimum Version

2008 (version 10.0.1600 or later)

9i R2

5.1.45

9.4.0