Razorback Requirements Document

**Q4 Output**

Version 0.3

Revision 1

10/28/2011

## 

Table of Contents

[Introduction 3](#_Toc307575252)

[Customer Focus Statement 3](#_Toc307575253)

[Requirements 3](#_Toc307575254)

[Messages 3](#_Toc307575255)

[Subscription 3](#_Toc307575256)

[Implementation 4](#_Toc307575257)

[Message Format 4](#_Toc307575258)

[Message Data Types 4](#_Toc307575259)

[Object – UUID 5](#_Toc307575260)

[Object – Nugget 5](#_Toc307575261)

[Object – Event ID 6](#_Toc307575262)

[Object – Hash 6](#_Toc307575263)

[Metrics 7](#_Toc307575264)

[Impact 7](#_Toc307575265)

[Future Work 7](#_Toc307575266)

[Enhancements 7](#_Toc307575267)

## Introduction

The output system is for use by systems that wish to integrate with Razorback.

## Customer Focus Statement

Razorback will provide streams of messages when certain events take place so that other systems can take actions based on Razorbacks data and analysis.

## Requirements

### Messages

* Must be presented in some language and machine independent format.
* Primary Alert – Generated when an inspector sends an alert for a block.
* Child Alert – Generated when a child block has transitioned into the bad state and caused this block to be marked bad.
* Seen Alert – Generated when an event is submitted for a block that has previously been marked bad.
* Event – Generated every time a block is seen by the system.
* Log – Generated when a nugget sends a log to be recorded in the database.

### Subscription

* A consumer of messages must be able to subscribe to the minimum set of messages that it requires.

## Implementation

### Message Format

Messages will be placed in the MQ in JSON objects, this is slightly more efficient than using XML and just as well supported by a range of programming languages.

### Message Data Types

JSON supports the following data types:

* Integer
* Boolean
* String
* Array
* Object
* Unions

In order to represent all of the razorback data we are going to have to define a number of objects that will be used in messages:

* UUID
* Nugget
* Event ID
* Hash
* Block ID
* Metadata Item
* Event
* Block

### Object – UUID

{

“name”:”UUID”,

“type”:”object”,

“properties”:

{

"id":

{

“type”:”string”,

“description”:”UUID in string form”,

“required”:true

},

“name”:

{

“type”:”string”,

“description”:”Name of the UUID”,

“required”:true

},

“description”:

{

“type”:”string”,

“description”:”Description of the UUID”,

“required”:false

}

}

}

### Object – Nugget

{

“name”:”Nugget”,

“type”:”object”,

“properties”:

{

"id":

{

“type”:”string”,

“description”:”UUID of the nugget”,

“required”:true

},

“app\_type”:

{

“type”:”object”,

“extends”:

{

“$ref”:”UUID”

},

“required”:”true

},

“nugget\_type”:

{

“type”:”object”,

“extends”:

{

“$ref”:”UUID”

},

“required”:true

}

}

}

### Object – Event ID

{

“name”:”Event\_ID”,

“type”:”object”,

“properties”:

{

"nugget":

{

“type”:”object”,

“extends”:

{

“ref”:”Nugget”

},

“required”:true

},

"seconds":

{

"type":"number",

"description":"Seconds portion of event time stamp".

"required":true

},

"nanoseconds":

{

"type":"number",

"description":"Nano econds portion of event time stamp".

"required":true

}

}

}

### Object – Hash

{

“name”:”Hase”,

“type”:”object”,

“properties”:

{

"type":

{

“type”:”string”,

“enum”:

[

“MD5”,

“SHA1”,

“SHA128”,

“SHA256”,

“SHA512”

],

“description”:”The type of hash in the value field”,

“required”:true

},

"value":

{

"type":"string",

"description":"Hex encoded hash value".

"required":true

},

}

}

### Object – Block ID

{

“name”:”Block\_ID”,

“type”:”object”,

“properties”:

{

"hash":

{

“type”:”object”,

“extends”:

{

“ref”:”Hash”

},

“required”:true

},

"size":

{

"type":"number",

"description":"Size of the block".

"required":true

},

"data\_type":

{

“type”:”object”,

“extends”:

{

“ref”:”UUID”

},

“required”:true

}

}

}

### Object – Metadata Item

{

“name”:”Metadata\_Item”,

“type”:”object”,

“properties”:

{

"name":

{

“type”:”object”,

“extends”:

{

“ref”:”UUID”

},

“required”:true

},

"type":

{

“type”:”object”,

“extends”:

{

“ref”:”UUID”

},

“required”:true

}

“value”:

{

“type”:”string”,

“description”:”The items data in string form”,

“required”:true

}

}

}

### Object – Block

{

“name”:”Block”,

“type”:”object”,

“properties”:

{

"id":

{

“type”:”object”,

“extends”:

{

“ref”:”Block\_ID”

},

“required”:true

},

"parent\_id":

{

“type”:”object”,

“extends”:

{

“ref”:”Block\_ID”

},

“required”:false

}

"parent":

{

“type”:”object”,

“extends”:

{

“ref”:”#”

},

“required”:false

}

“metadata”:

{

“type”:”array”,

“items”:

{

“type”:”object”,

“extends”:

{

“ref”:”Metadata\_Item”

}

},

“required”:false

},

“sf\_flags”:

{

“type”:”number”,

“description”:”Sourcefire Flags Field”,

“required”:false

},

“old\_sf\_flags”:

{

“type”:”number”,

“description”:”Sourcefire Flags Field before they were changed”,

“required”:false

},

“ent\_flags”:

{

“type”:”number”,

“description”:”Enterprise Flags Field”,

“required”:false

},

“old\_ent\_flags”:

{

“type”:”number”,

“description”:”Enterprise Flags Field before they were changed”,

“required”:false

},

}

}

### Object – Event

{

“name”:”Event”,

“type”:”object”,

“properties”:

{

"id":

{

“type”:”object”,

“extends”:

{

“ref”:”Event\_ID”

},

“required”:true

},

"parent\_id":

{

“type”:”object”,

“extends”:

{

“ref”:”Event\_ID”

},

“required”:false

}

"parent":

{

“type”:”object”,

“extends”:

{

“ref”:”#”

},

“required”:false

}

“metadata”:

{

“type”:”array”,

“items”:

{

“type”:”object”,

“extends”:

{

“ref”:”Metadata\_Item”

}

},

“required”:true

}

}

}

### Message – Event

The event message is just an Event object with the following populated:

* ID
  + Time Seconds
  + Time Nano Seconds
  + Nugget ID
* Metadata
* Parent ID
  + Time Seconds
  + Time Nano Seconds
  + Nugget ID
* Block
  + ID
    - Hash
    - Size
    - Data Type
  + SF\_Flags
  + Ent\_Flags

Headers attached to the message if values are present:

* Source\_IPv4
* Dest\_IPv4
* Source\_IPv6
* Dest\_IPv6
* Source\_Port
* Dest\_Port
* Protocol

### Message – Log

Log message has the following format, no additional headers are to be provided.

{

“name”:”Log”,

“type”:”object”,

“properties”:

{

"message":

{

“type”:”string”,

“description”:”Log Message”,

“required”:true

},

"nugget":

{

“type”:”object”,

“extends”:

{

“ref”:”Nugget”

},

“required”:true

}

"timestamp":

{

“type”:”integer”,

“description”:”Time in seconds past the epoc”,

“required”:true

},

“priority”:

{

“type”:”integer”,

“description”:”Message priority”,

“required”:true

},

“event”:

{

“type”:”object”,

“extends”:

{

“ref”:”Event”

},

“required”:false

}

}

}

### Message – Alert

The alert message has the following structure:

{

“name”:”Alert”,

“type”:”object”,

“properties”:

{

"type":

{

“type”:”string”,

“description”:”Log Message”,

“enum”:

[

“Primary”,

“Child”,

],

“required”:true

},

"block":

{

“type”:”object”,

“extends”:

{

“ref”:”Block”

},

“required”:true

},

"trigger\_block":

{

“type”:”object”,

“extends”:

{

“ref”:”Block”

},

“required”:false

},

"event":

{

“type”:”object”,

“extends”:

{

“ref”:”Event”

},

“required”:true

},

"timestamp":

{

“type”:”integer”,

“description”:”Time in seconds past the epoc”,

“required”:true

},

“priority”:

{

“type”:”integer”,

“description”:”Message priority”,

“required”:true

},

“gid”:

{

“type”:”integer”,

“description”:”GID”,

“required”:true

},

“sid”:

{

“type”:”integer”,

“description”:”SID”,

“required”:true

},

“metadata”:

{

“type”:”array”,

“items”:

{

“type”:”object”,

“extends”:

{

“ref”:”Metadata\_Item”

}

},

“required”:true

},

“event\_count”:

{

“type”:”integer”,

“description”:”Number of events associated with the block”,

“required”:false

},

“parent\_count”:

{

“type”:”integer”,

“description”:”Number of parent blocks for this child”,

“required”:false

}

}

}

Headers attached to the message if values are present:

* Source\_IPv4
* Dest\_IPv4
* Source\_IPv6
* Dest\_IPv6
* Source\_Port
* Dest\_Port
* Protocol

A primary alert message will have the following fields populated:

* Message
* Priority
* GID
* SID
* Timestamp
* Metadata
* Event
  + ID
  + Metadata
  + Block
    - ID
* Block
  + ID
  + Metadata
  + SF\_Flags
  + Old\_SF\_Flags
  + Ent\_Flags
  + Old\_Ent\_Flags
  + Parent – (recursive based on event recursion)

A child alert message will have the following fields populated:

* Message
* Priority
* GID
* SID
* Timestamp
* Event Count
* Parent Count
* Block
  + ID
  + Metadata
  + SF\_Flags
  + Old\_SF\_Flags
  + Ent\_Flags
  + Old\_Ent\_Flags
* Trigger Block
  + ID
  + Metadata
  + SF\_Flags
  + Ent\_Flags

### Topics

Alerts:

<Priority>.<Data Type>.<App Type>.<Nugget ID>

Events:

<Data Type>.<App Type>.<Nugget ID>

Logs:

<Priority>.<App Type>.<Nugget ID>

## Metrics

* Message size
* Message count

## Impact

* Requires changes to the dispatcher:
  + Event submission
  + Judgment processing
  + Log processing
  + Flags coping thread

## Future Work

### Enhancements

* TBD