

Build Your Own Plugin

Dan Zada

vmware®

© 2014 VMware Inc. All rights reserved.

Agenda

- Why Should We?
- Plugin vs. Solution
- Plugin's Components
- XML vs. Java Plugin
- JMX Monitoring – Live Demo
- Process Monitoring – Live Demo

Why Should We?

- There isn't a plug-in for the resource you'd like to monitor
 - Your own application
- Plug-in exists but does not fully satisfy your requirements
 - For example: a new version of the resource type is out there
 - You should consider using the existing plug-in as a starting point



Plugin vs. Solution

- Plugin
 - Java/XML based software
 - Defines the modeling and monitoring of specific technology
 - Data collection

- Solution
 - Defines set of UI resources such as: dashboards, widgets etc.
 - Defines alerts and symptoms
 - Can contain one or more plugins
 - Data consumption

Plugin's Components

- Discovery
 - Discover resources, usually looking at the process table to detect all instances of the monitored product/technology
 - Detect as much configuration options as possible (e.g. listening port)
- Measurement
 - Retrieve Metrics of monitored resources
 - Only one metric is gathered during each execution
- Control Actions
 - Execute actions (like stop, start, ...) against the monitored resource
 - Will be supported in vROps in future releases

XML vs. Java Plugin

- Every plug-in has an XML-based descriptor
- What are support classes/base plugins?
 - Java classes that can be re-used in certain ‘types’ of plug-ins
 - Examples: Process, SQL, JMX, Perfmon.
- A plug-in might also have associated Java classes
 - When you need to add custom logic that is not already covered by one of the PDK’s support classes

Plan Your Plugin

- How would you detect the relevant resources?
- How would you model the technology?
- Which info would you need in order to collect metrics?
- How would you determine if the resource is 'available'?
- Can you utilize one of the existing base plugins?

Thank You!