



IBM Software Group

Troubleshooting and Resolving Issues with the Intelligent Management Plugin

Robert Outlaw (routlaw@us.ibm.com) , Charlie Wiese (wiese@us.ibm.com)
IBM WebSphere Application Server L2 Support Engineers
5 May 2014



WebSphere® Support Technical Exchange



Agenda

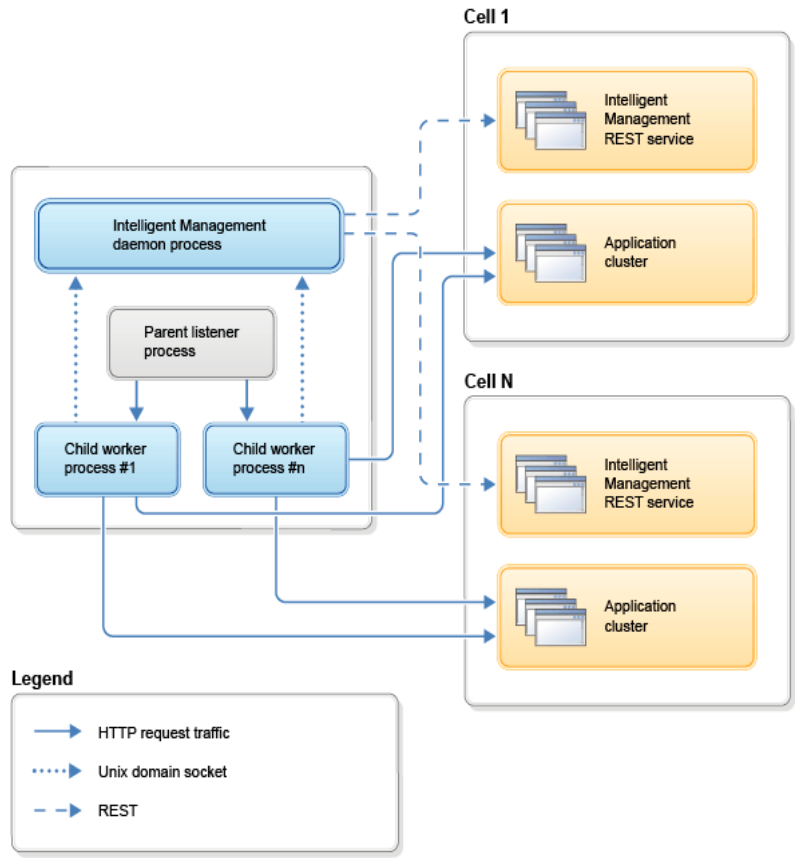
- Intelligent Management Plugin Architecture Discussion
 - ▶ Brief overview-identifying the points of interaction of the system that drive common problems.
- Basic Troubleshooting and Diagnostics
 - ▶ Using the IBM HTTP Server's server-status module and reviewing the http_plugin.log file.
- Enabling and Reviewing Detailed Tracing
 - ▶ Setting request tracing and following requests in the traces to determine cause of encountered errors.



Intelligent Management Plugin

- Provide the features of the On Demand Router without the need for an additional ODR tier in IM environments.
- Leverages existing familiarity and deployment practices with the IBM HTTP Server plugin.
- Biggest differences/challenges were how to deliver up to date state information about the WebSphere environment to the plugin so routing information would be updated.
- Implementation is still not fully complete, some lesser-used features of the ODR still missing from the IM plugin.

Intelligent Management Plugin



Intelligent Management Plugin

- The IM plugin's dynamic consumption of data from the REST service on the Application Server environment is one key potential problem point.
- The expanded (from the standard behavior) logic that is used to determine where requests are sent to from the plugin is the other key departure from the regular WebSphere Plugin implementation.
- Approaching problems with the IM plugin with these two points of interaction in mind is very helpful.

Basic Troubleshooting and Diagnostics

- Make sure you understand the problem you're trying to solve first. Collect the following and place it in a convenient place for reference during troubleshooting.
 - ▶ The name of the application, and the URI returning errors. If the cluster and/or servers where that application is running is known, mark that information as well.
 - ▶ The HTTP response code being received, or the symptom observed by the clients attempting to access the resource.
 - ▶ The type and nature of the error:
 - Does it occur consistently, is it limited only to certain times, or is it seemingly random or spurious?
 - Is it an error that's returning error codes, or (less typical) a load balancing or other type of problem?



Basic Troubleshooting and Diagnostics

- Configure the Apache/IHS server status module and retrieve its data using
`http://<webserverhost>/server-status`
This will provide formatted output from the server showing its current state, including a section specific to Intelligent Management
- Easy and handy to use:
 - ▶ Simple browser interface
 - ▶ Key, important information

Basic Troubleshooting and Diagnostics

- Output from server-status has 3 important sections in the “Intelligent Management Status” section
 - ▶ Listing of all applications in the environment (1 for each application, whether the application is active or not)
 - Application state is not part of this listing
 - ▶ Cluster definitions including a list of all apps in the cluster, their current state, clone IDs, server waits, server runtime state, maintenance mode, etc.
 - ▶ Connectors definitions – DMGRs or nodeagents where the rest service is running
 - Connector group for each Cell, and list of connectors for each cell

Sample server-status output

```
"/cell/routlawCell01/application/IVT Application": {  
  "editions": {  
    "": {  
      "webModules": {  
        "/cell/routlawCell01/application/IVT  
Application/webModule/ivt_app.war": {  
          "contextRoot": "/ivt" }}}}
```

- ▶ Manifest of applications available in the IM environment. Does not include any binding information for these applications-this information is mapped on the server definitions themselves.
- ▶ Application state is not part of this definition-there is no difference between the data representation of stopped and started applications in these fields. Again, this runtime data is stored in the application/cluster definitions.

Sample server-status output

Clusters

```
"/cell/routlawCell01/cluster/routlawNode01:server1": {  
  "servers": {  
    "/cell/routlawCell01/node/routlawNode01/server/server1": {  
      "state": "STARTED",  
      "weight": 1,  
      .....  
      "applications":  
        "IVT Application": {  
          "state": "STARTED",  
          "outstandingRequests": 0  
        }  
    }  
  }  
}
```

- Cluster definitions contain affinity information, application runtime state, server runtime state, weighting information for dynamic workload management, and a manifest of started and stopped applications defined for this server.

Sample server-status output

- Connectors

- Define the list of potential REST gateways the IM daemon can connect to.

```
"connectorGroups": {  
  "routlawCell01": {  
    "state": "STARTED",  
    "failures": 0,  
    "connectors": {  
      "http://routlaw:7063": {  
        "state": "STARTED",  
        "failures": 0
```

- Note that state being STOPPED does not designate a failure condition. It simply means that the plugin does not have a currently outbound request to that endpoint. For each cell definition above there should be at least one STARTED state connector. Note that multiple STARTED states is also not an error condition.

Plugin Logging - http_plugin.log

- This log contains terse debug information describing failed requests

```
[Mon Mar 8 03:24:42 2014] 00002be0 00001360 ERROR: ws_common: webspHEREWriteRequestReadResponse: Failed to find a server
```

```
[Mon Mar 8 03:24:42 2014] 00002be0 00001360 ERROR: ESI: getResponse: failed to get response: rc=256
```

- Look for network connectivity problems, gskit errors (indicating possible SSL issues), other problems at or near the time of the problem occurrence.

Enabling and Reviewing Trace

- Enabling the trace on the plugin is done through the WebSphere Application Server administrative console, by modifying the trace setting for the Web Server object that was configured when the IM plugin was set up.
- The string `http.request:DEBUG` is an excellent catch-all trace string that captures detailed information about a request failure without becoming too verbose.

Enabling and Reviewing Trace

Middleware servers > webserv1 > Intelligent Management > Trace specification

Specify trace to run on the web server plug-in.

Configuration

General Properties

Trace

Set the trace specification and specify a log detail level. Expand the trace specification tree and click a trace specification to select a log detail level.

http.request:DEBUG

- http
- control

Conditional Request Trace

Edit condition

Collect trace when:

[[Subexpression builder](#)] [[Syntax help](#)]

Set the conditional request trace specification and specify a log detail level. Expand the trace specification tree and click a trace specification to select a log detail level.

- http

Apply OK Reset Cancel

Enabling and Reviewing Trace

- [Mon Mar 10 03:32:48 2014] 00002084 0000224c - ODR:DEBUG: performRequestMapping: enter
- [Mon Mar 10 03:32:48 2014] 00002084 0000224c - ODR:DEBUG: rmMapRequest enter (rmaddr=0219B110)
- [Mon Mar 10 03:32:48 2014] 00002084 0000224c - ODR:DEBUG: matchLongestURI: routlaw:8800///ivt/ivtDate.jsp, vhostNum=3
 - ▶ A sequence in the trace like above should be used to identify the problematic request by correlating time issued w/ URI pattern. Then you can follow the thread ID, which is the 2nd hexadecimal string after the timestamp (in this case, 0000224c)
- Follow the thread up to the point where the error status code is issued. In this case, I was recreating a 503, so I see this:
 - ▶ [Mon Mar 10 03:32:48 2014] 00002084 0000224c - ODR:DEBUG: tsSelect: no server was found; Error Status Code=503
 - ▶ [Mon Mar 10 03:32:48 2014] 00002084 0000224c - ODR:DEBUG: odrHttpContextGetTargetEndpoint: exit - tep=null
 - ▶ [Mon Mar 10 03:32:48 2014] 00002084 0000224c - ERROR: ws_common: websphereWriteRequestReadResponse: Failed to find a server

Enabling and Reviewing Trace

- Each error code will have trace information and context around and before it that is largely layman readable. In this case, we could not find a server running the application in question to route the request to. The next step would be to verify in the server-status output that at least one server is running the application matched in the thread.
- On very busy servers it is exceptionally important to filter the requests by URI pattern in order to keep logging from impacting servers.

Summary

- The IM Plugin has two key points of interaction with the backend WebSphere environment-the REST interfaces and the webcontainer ports on the Application Servers themselves.
- The IM Plugin has some very simple and straightforward tools available to collect data about the inputs for its decisions without enabling trace.
- Tracing can be used to selectively isolate and follow step by step a request as the IM plugin processes it.

Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at:
http://www.ibm.com/software/websphere/support/supp_tech.html
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:
<http://www.ibm.com/developerworks/websphere/community/>
- Join the Global WebSphere Community:
<http://www.websphereusergroup.org>
- Access key product show-me demos and tutorials by visiting IBM Education Assistant:
<http://www.ibm.com/software/info/education/assistant>
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically:
<http://www.ibm.com/software/websphere/support/d2w.html>
- Sign up to receive weekly technical My Notifications emails:
<http://www.ibm.com/software/support/einfo.html>

Connect with us!

1. Get notified on upcoming webcasts

Send an e-mail to wsehelp@us.ibm.com with subject line “wste subscribe” to get a list of mailing lists and to subscribe

2. Tell us what you want to learn

Send us suggestions for future topics or improvements about our webcasts to wsehelp@us.ibm.com

Questions and Answers