



Ravi Gopalakrishnan
Keith Smith
Christopher Vignola

Improve Application Resiliency with IBM WebSphere Network Deployment V8.5

IBM® WebSphere® Application Server Network Deployment V8.5 (referred to as *Network Deployment* from this point on) improves application server resiliency and delivers enterprise quality service for batch workloads. These capabilities lower total cost of ownership and result in higher availability and agility of the application server environment. This IBM Redpaper™ publication describes several implementation scenarios that Network Deployment clients can review to quickly illustrate the value of these enhanced resiliency and batch capabilities.

Preferred practices summary

This paper focuses on configuration steps that clients must perform to realize the benefits in each scenario and describes the following preferred practices:

1. Use nondisruptive application edition management with an easy rollback option.
2. Protect business critical applications within your application portfolio by prioritizing applications and dynamically scaling resources to handle unanticipated spikes in application load. Avoid incurring huge capital outlays for backup resources, provide load balancing and failover in a multi-cluster environment, and auto-learn application state using intelligent routing.
3. Sense and respond to common problems before user outages occur.
4. Eliminate single points of failure for administrative functions in a Network Deployment cell with the high availability (HA) deployment manager function.
5. Use the online transaction processing (OLTP) infrastructure for new batch workloads.
6. Improve batch elapsed-time by using system managed parallel batches.
7. Extend the reach of enterprise workload scheduling to WebSphere batches.

Terminology

This section introduces you to concepts and terminology that are new in Network Deployment pertaining to the capabilities described in this paper.

Application placement controller

The application placement controller (APC) in Network Deployment decides how many members of each dynamic cluster can be running to most effectively handle the current amount of traffic. It determines the available memory, processor capability, and current use of each node, including resources that are already in use by other processes. It uses this information to determine the optimal placement of each application to best meet the defined performance goals as specified by service policies. The dynamic cluster's isolation policy requirements are also considered when the application placement controller determines where to place instances of a dynamic cluster. APC runs as a highly available single service in the deployment manager or in one of the node agents. If multiple cells share the same physical resources, Network Deployment can be configured in a star topology with one cell designated as the center cell and all other cells designated as points of the star. In the star topology, APC runs in the center cell but not in the point cells.

Autonomic request flow manager

The autonomic request flow manager (ARFM) controller is responsible for controlling the rate at which HTTP requests or Session Initiation Protocol (SIP) messages flow through the on demand router (ODR). It dynamically adjusts the rates at which requests flow to application servers to ensure that the response time goals are met and no application servers are sent more requests than they can handle. ARFM runs in the deployment manager or a node agent as a single service within the cell.

Dynamic cluster

A dynamic cluster in Network Deployment can be resized dynamically at any time. Rather than statically allocating hardware resources to each cluster, hardware can be treated as a single virtual pool of resources. This virtual pool is called a *dynamic cluster*.

On demand router

The on demand router (ODR) is an intelligent HTTP reverse proxy and stateless SIP proxy server that is provided with Network Deployment. It manages request prioritization, flow control, and the dynamic routing of HTTP requests or SIP messages to your application servers.

Parallel job manager

The parallel job manager is a feature of the batch run time that supports parallel batch processing. This feature allows batch jobs to easily partition workload and process concurrently to reduce elapsed job time and help meet service level agreements. The workload can be dynamically spread across multiple Java virtual machines (JVMs) to provide more CPU and memory to complete the work. The parallel job manager handles all details of managing the separate units of work, provides a single job image, and simplifies operational and problem determination concerns.

Service policies

A service policy is an operational policy in Network Deployment that is used to categorize work. It defines a level of importance and a response time goal and allows the administrator to describe how requests are treated in the environment.

Work classes

A work class is associated with a single application and allows an administrator to apply routing policy and service policy to inbound HTTP or SIP work. Each work class contains a default action and an optional set of rules that define conditions and non-default actions.

Workload scheduler connector

The workload scheduler connector (*WSGrid*) is a batch utility that works under the control of all major workload schedulers, such as IBM Tivoli® Workload Scheduler, BMC Control-M, and CA Workload Automation AE. It enables the inclusion of WebSphere batch jobs in the overall enterprise workload scheduling plan. This allows WebSphere batch jobs to be treated just like any other scheduled workload, using existing planning and operational skills.

Scenario 1: Nondisruptive application edition management with roll-back option

Network Deployment allows you to install and manage multiple editions of the same enterprise application within a cell. An application edition refers to a specific instance of a Java Platform, Enterprise Edition EAR file. You can use the edition control center to

seamlessly manage the rollout of new editions to some or all of your servers. Network Deployment also provides a validation mode that gives you the ability to verify functionality of the new edition in your production environment without making it available to all of your users. After validation of the new edition is complete, the application edition can be rolled out into regular production mode without service interruption for your users. Figure 1 illustrates how edition validation works.

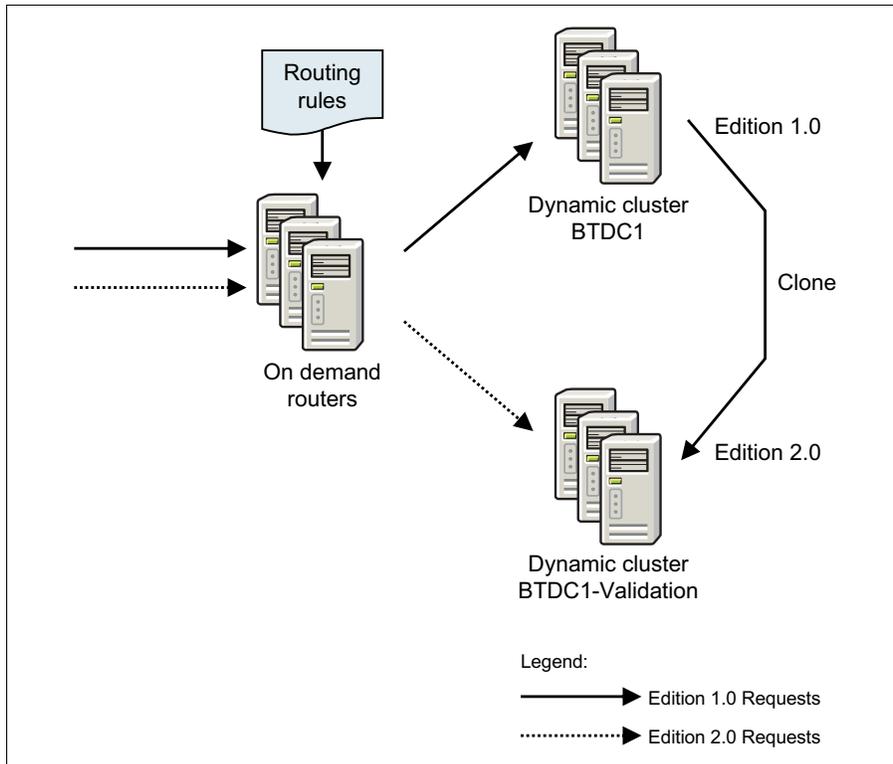


Figure 1 Edition validation

Network Deployment provides the ability to roll back to previous editions with a single click. It is also possible to concurrently run multiple editions of a single application and to selectively route different segments of your application's users to a specific edition.

Figure 2 illustrates how the ODR dynamically routes requests to different editions based on routing rules that are defined by the administrator.

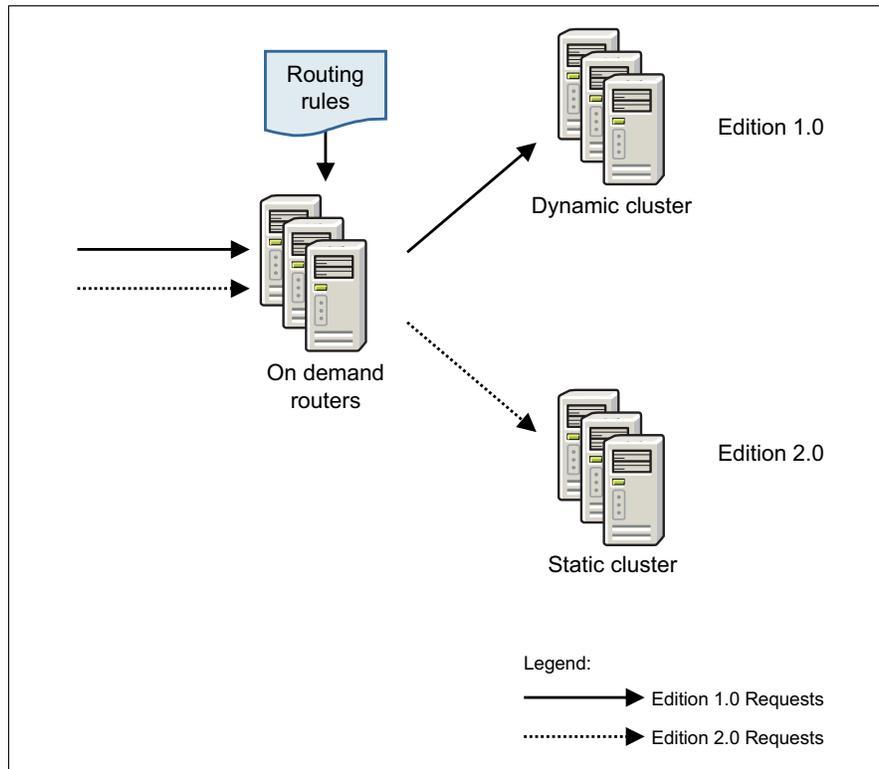


Figure 2 ODR routing

Table 1 contains examples of the problems that are solved by the application edition management feature.

Table 1 Application edition management solutions

| | |
|--|---|
| Running concurrent editions | Activate concurrent application editions to: <ul style="list-style-type: none"> ▶ Run two editions of a single application concurrently and control the switch-over rate from one edition to the other. ▶ Run multiple editions of a single application concurrently and be able to control which client uses which edition. |
| Efficient and effective testing in pre-production environments | Install application editions in pre-production environments to: <ul style="list-style-type: none"> ▶ Reduce the administrative and infrastructure costs of maintaining multiple pre-production environments . ▶ Eliminate production outages that are caused by differences between your production and pre-production environments. |
| Achieve transparency for your customers using validation mode | Install application editions to: <ul style="list-style-type: none"> ▶ Validate a new edition of a WebSphere Java Platform, Enterprise Edition application (that is, an EAR file) in your production environment without disrupting service to your clients. Perform a rollout of an edition to: <ul style="list-style-type: none"> ▶ Roll out a new edition of an application and make the rollout completely transparent to customers who are using the application. |
| Easy rollback option | Perform a rollback on an edition to: <ul style="list-style-type: none"> ▶ Quickly and easily roll back to an earlier edition of an application. |

To enable the application edition management with rollback capability, the administrator does the following steps:

1. Configure the ODR.
2. Install, activate, and set up routing policies for application editions.

For more information: For details about this topic, refer to the product documentation topics under *Deploying and managing application editions without loss of service* at the following website:

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.wve.doc/ae/twve_appedtop.html

Additional related features are node maintenance mode and server maintenance mode, which allow you to easily control the flow of work on a per node or a per application server basis. The following examples show problems that are solved by these maintenance mode features:

- ▶ You want to route traffic away from all application servers on a node so that you can perform application, operating system, or middleware maintenance.
- ▶ You want to route traffic away from a specific application server to perform problem determination on that server.

For more information: More information about these maintenance mode features can be found at the following website:

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp?topic=/com.ibm.websphere.wve.doc/ae/twve_mwmaint.html

Consider the following preferred practices when implementing application edition management in your environment.

- ▶ Define application routing rules to send a configurable percentage of traffic to a new application edition. Application routing rules can be defined to slowly but automatically migrate users to the new edition. The routing rules can even specify start and end dates to automatically ramp up and eventually route all traffic to the new edition at the end date.
- ▶ Use the group rollout option for no queuing and more control by the group size parameter. Use larger groups during low loads for quick rollout. Use smaller groups during heavier load for slow rollout with less potential for performance issues.
- ▶ Use the soft reset rollout option when multiple applications are deployed to a single cluster. This results in a less disruptive user experience. A hard reset is required to reload native code.
- ▶ The default drainage interval is set at 30 seconds. For applications with session state, increase the drainage interval to allow time for as many sessions to expire as possible. If all sessions expire before the drainage interval is completed, the quiesce operation completes earlier and the rollout continues.
- ▶ In situations where you have administrative control over the installation process for a group of applications and you cannot specify an edition name for applications that are deployed in the same cluster, use the **appEditionRename.py** script to give a single edition name to all applications that are currently deployed to the cluster.

Scenario 2: Prioritize, load balance, auto-learn application state, and dynamically scale resources to protect business critical applications

Network Deployment increases server resiliency by providing request prioritization, flow control, and the ability to distribute requests to application servers based on application priority, as illustrated in Figure 3. Network Deployment can queue requests for less important applications to allow requests for more important applications to be handled quicker. This prioritization is achieved by defining application service levels during deployment by using configurable service policies. Each service policy is a user-defined business goal and provides a mechanism to manage application performance and quality of service.

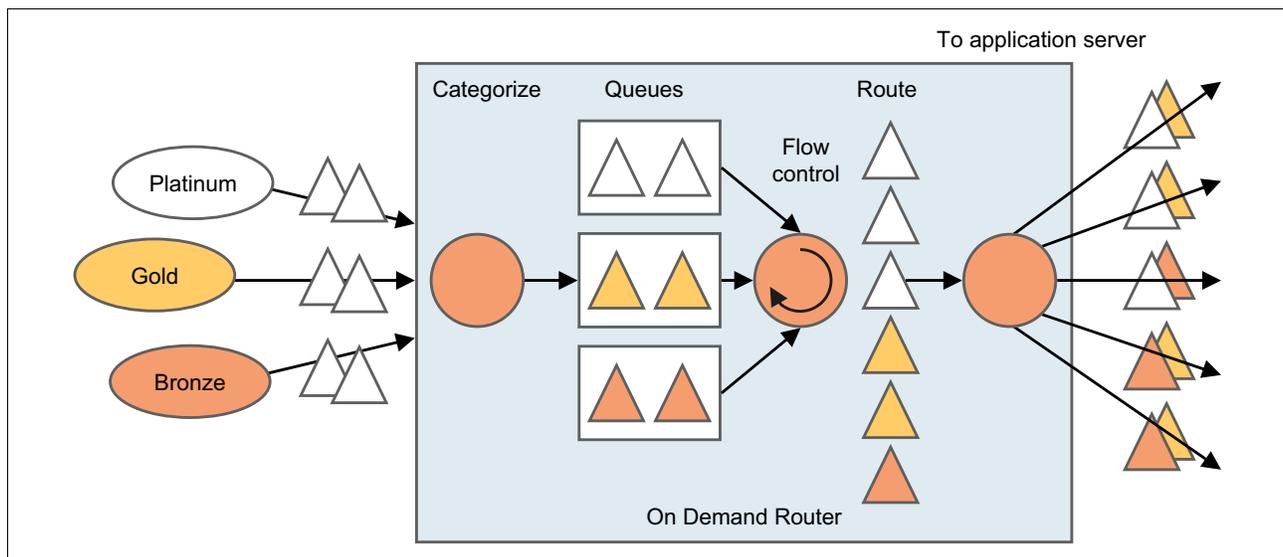


Figure 3 ODR request handling

This request handling feature is illustrated in Figure 3, where an equal amount of requests flow into the ODR with platinum, gold, and bronze depicting a descending order of business priority for applications. After the work is categorized, prioritized, and queued, a higher volume of more important work (platinum) is processed while a lower volume of less important (bronze) work waits to get queued.

Consider the following preferred practices when implementing service policies:

- ▶ To determine the best values for service policy response time goals, enable the visualization data service to keep track of the average response time on a per application basis. After a time period (for example, one month), use the administrative console to graph the data to determine the best average response time. Set the average response time goal to 150% of this value. Configuration information for the visualization service can be found at the following website:

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp?topic=/com.ibm.websphere.wve.doc/ae/twve_xadmvisualizationdata.html

- ▶ If you are unable to find the best average response time goals in your environment, you can set the custom property `disableResponseTimeGoals` to `true`. This allows the APC to use the observed average response times. When using this option, the ARFM queuing must *not* be disabled.

Network Deployment gives you the ability to build a dynamic, virtualized, goal-oriented environment for workload management. Network Deployment dynamically manages workload by starting and stopping more application server instances to accommodate changes in load and balancing processing power among your applications to best meet your defined performance goals. Resources can be treated as a single virtual pool, which is known as a *dynamic cluster*, and can be dynamically allocated, resulting in a highly efficient usage of hardware resources taking advantage of different peak times in application load. As demand for applications running on a dynamic cluster increases or decreases, instances of the dynamic cluster can be started or stopped on nodes within the cluster to accommodate the changes in load. Each node in the set of defined cluster members has a configured instance of the dynamic cluster that is ready to be started dynamically when needed. Figure 4 illustrates this feature.

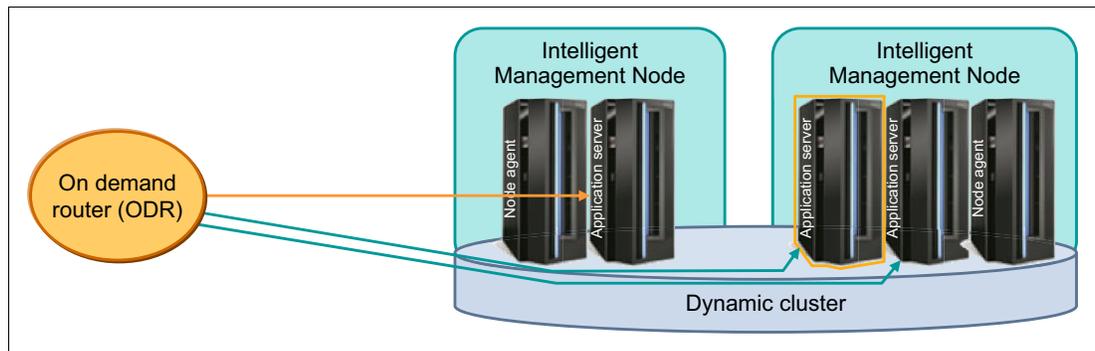


Figure 4 Dynamic cluster configuration

Consider the following dynamic cluster preferred practices:

- ▶ To limit the amount of concurrent traffic to a dynamic cluster, set the custom property `arfmMode` to a value of `Manual`.
You can then set the dynamic cluster custom property to a value like `*=50` which will limit the number of concurrent HTTP requests to this cluster to 50, or an expression like `Gold=*,Silver=15,*=5`. In this example, Gold service policy traffic is not limited at all, Silver is limited to 15 concurrent requests, and all other traffic is limited to five concurrent requests.
- ▶ You can enable the lazy start option for a dynamic cluster to allow low volume applications to be available without consuming resources. To enable this, select the **Stop all instances started during periods of inactivity** option on the configuration page for the dynamic cluster and specify a numeric value in minutes.

The ODR is self-configuring and it automatically learns the state of applications and application servers to which it routes in the same or different cells, thus reducing administrative overhead. It also has a sophisticated load balancing algorithm, which allows it to quickly route around application servers that are slow to respond or do not respond at all. In addition, the ODR is able to load balance or failover on the same application running in two different clusters, which are in the same or different cells.

Consider the following ODR preferred practices:

- ▶ The JVM heap size setting is the single most critical factor for efficient ODR performance. Use the following rule of thumb to decide on the optimal JVM heap size. If MR represents the maximum requests per second that the ODR is expected to handle, the JVM heap size (in MB) would be set to $90 + MR$.
- ▶ Use persistent keep alive connection settings for all inbound (web server to ODR) and outbound (ODR to application server) traffic. These connections will be long lived so that

the ODR can reuse the connections for as long as possible to handle a barrage of requests.

- ▶ Disable ODR caching when not in use. See the *Tuning the ODR* Information Center topic for more details:

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.wve.doc/ae/twve_odrtuneodr.html

- ▶ The web server's plug-in can mark the ODR as being down if the application or application server responds slowly. To prevent this, set the plug-in's ServerIOTimeout setting to:

ServerIOTimeout >= ODR's Outbound Request Read Timeout + Outbound Request Write Timeout + Outbound Request Connect Timeout” + 5 seconds

- ▶ The ODR supports custom logging similar to the Apache mod_log but can also conditionally create custom log entries, increase performance, and decrease the amount of log data to store, manage, and inspect. Use the **manageODR.py** script to configure custom logging. See the following Information Center topic for a list of custom log parameters:

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.wve.doc/ae/cwve_xdcustomlog.html

- ▶ ODR routing rules can be used to route traffic to specific servers within a cluster. For example, you can set up a routing rule to first attempt to route traffic to server names that match a pattern with a failover rule to route the traffic to other servers in the cluster.

Several autonomic managers are provided (see Table 2) which help monitor performance metrics, analyze the monitored data, offer a plan for running actions, and can start these actions in response to the flow of work.

Table 2 Preferred locations for the Autonomic Controllers

| Controller | Generic JVM property | Process |
|--|---|---|
| APC | HAManagedItemPreferred_apc=true | <ul style="list-style-type: none"> ▶ dmgr ▶ node agent |
| ARFM | HAManagedItemPreferred_arfm=true | <ul style="list-style-type: none"> ▶ dmgr ▶ node agent ▶ ODR |
| Health Controller - Health Management Module (HMM) | HAManagedItemPreferred_hmm=true | <ul style="list-style-type: none"> ▶ dmgr ▶ node agent |
| VM Activity Publisher | HAManagedItemPreferred_vmshimactivitypublisher=true | <ul style="list-style-type: none"> ▶ dmgr ▶ node agent |
| HA Plugin-Cfg Generation Service | HAManagedItemPreferred_HAplugincfg=true | <ul style="list-style-type: none"> ▶ dmgr ▶ node agent |

Follow the directions in technote, SWG21425281, if you want to obtain information about how to configure a controller to always start on a particular node agent or deployment manager:

<http://www-01.ibm.com/support/docview.wss?uid=swg21425281>

By default, the APC is not in elasticity mode. By turning on this mode, APC minimizes the number of nodes that are used to meet service policy goals, it reduces resource usage resulting in power savings, and you can set up custom scripts, which are invoked when the APC adds and removes nodes. The custom script would for example, remotely start and stop the node agent.

Also, by default the APC starts only one application server concurrently per node. You can set the `apcConcurrentStartSize` custom property on the APC to a numerical value indicating the maximum number of application servers that can be started concurrently per node. This is particularly useful to minimize the startup time of your cell after a cold restart.

The following examples show problems that are solved by the new performance management features.

Table 3 Problems and solutions

| | |
|---|---|
| Control costs through infrastructure consolidation | <ul style="list-style-type: none"> ▶ Configure multi-cell performance management to reduce your administrative and infrastructure costs by consolidating multiple WebSphere cells into a single cell and still provide different qualities of service to various applications ▶ Configure elasticity mode to reduce your infrastructure costs by achieving a greater degree of server consolidation. This feature reduces memory and CPU costs by minimizing the number of virtual machines in use when application demand decreases ▶ Configure the ARFM to protect your infrastructure against overload caused by traffic spikes |
| Achieve application and application server elasticity | <p>Configure elasticity mode to:</p> <ul style="list-style-type: none"> ▶ Make applications available when needed; but, when they are not needed, these applications are not using any resources ▶ Automatically start and stop application servers as needed based upon CPU utilization, memory, and heap utilization ▶ Automatically provision new virtual machines to handle increases in application demand ▶ Have WebSphere manage the performance of your applications for you automatically based on service policies |
| Visualize performance metrics | <p>Use the Intelligent Manager monitoring tools to:</p> <ul style="list-style-type: none"> ▶ Appropriately compute the charge for the users of your shared infrastructure ▶ Visualize the performance of your applications using graphs or charts over a configurable period of time |

To enable these performance management features, the administrator performs the following actions:

1. Create the dynamic clusters
2. Configure the ODR
3. Configure the APC
4. Configure the ARFM
5. Deploy applications
6. Define service policies and classify application requests with service policy work classes

For more information, see these resources:

For details about this topic, refer to the product documentation for an overview of request flow prioritization at the following website:

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp?topic=/com.ibm.websphere.wve.doc/ae/cwve_odoerequestflow.html

For information about configuring the features in Table 3 on page 10, see the topics under *Configuring the autonomic managers* in the information center, starting at the following website:

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.wve.doc/ae/twve_odautmgr.html

For information about monitoring operations, see the following website:

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.wve.doc/ae/twve_xadm.html

Scenario 3: Sense and respond to common problems before user outages

Network Deployment provides health monitoring features for your environment that can sense and respond to several common software health conditions. Some conditions such as memory leaks, hung threads, slow response times, or an excessive percentage of request time-outs indicate a current problem in an application server. Other conditions, such as age or work performed, can be used to prevent problems in a server hosting an application with known minor errors. The conditions to monitor and the actions to take when they are encountered are defined in customizable and configurable health policies. The health controller autonomic manager controls the health monitoring subsystem and acts on your health policies.

Tip: You can decrease the Control Cycle Length global health control parameter to respond more quickly to health conditions if needed. Similarly, you can increase the Restart Timeout global health control parameter if servers take longer than the default five minutes in your environment.

Figure 5 provides a snapshot of the first step in the health policy setup wizard and the list of predefined health policies that can be monitored.

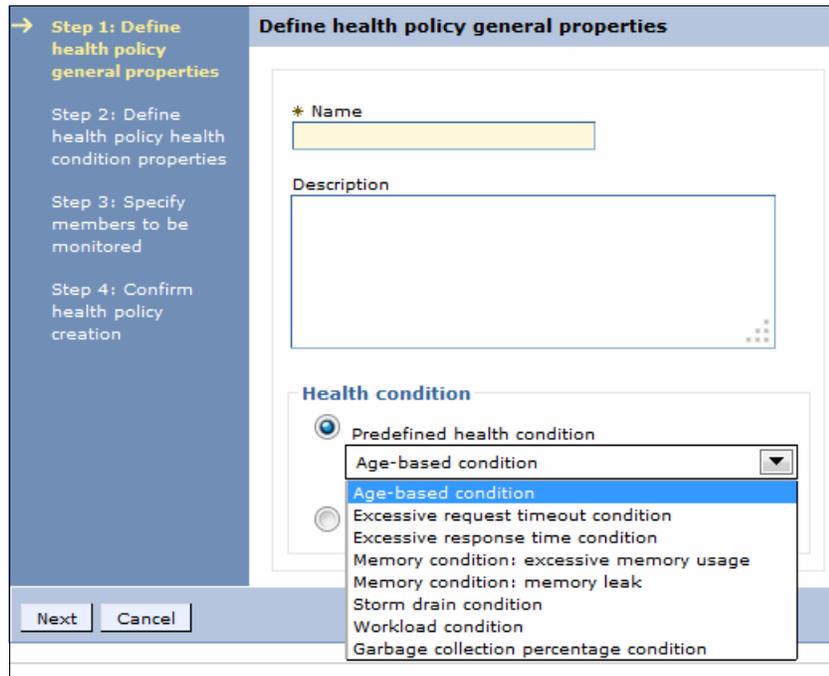


Figure 5 Health policy setup

The predefined health conditions that you can choose from are shown in Table 4.

Table 4 Pre-defined health conditions

| Health condition | Based on: |
|--|--|
| Age-based | Amount of time server has been running |
| Excessive request timeout | % of timed out requests |
| Excessive response time | average response time |
| Memory condition: excessive memory usage | % of maximum JVM heap size |
| Memory condition: memory leak | JVM heap size after garbage collection |
| Storm drain condition | significant drop in response time |
| Workload condition | total number of requests |
| Garbage collection percentage condition | % of time spent in GCs |

Administrators should turn on at least the following predefined health policies:

- ▶ Excessive garbage collection
- ▶ Excessive memory
- ▶ Memory leak
- ▶ Excessive request timeout
- ▶ Excessive response time

All health policies must be in supervised mode first so that you can tune any settings and verify that they are working as expected for your application. After you have performed this tuning and verification, place the health policies into automatic mode.

Network Deployment also allows you to define custom health policies when the predefined health conditions do not fit your needs, providing you the ability to create expressions to define what unhealthy means in your environment. Figure 6 provides a snapshot in the administrative console on how to create a custom health policy.

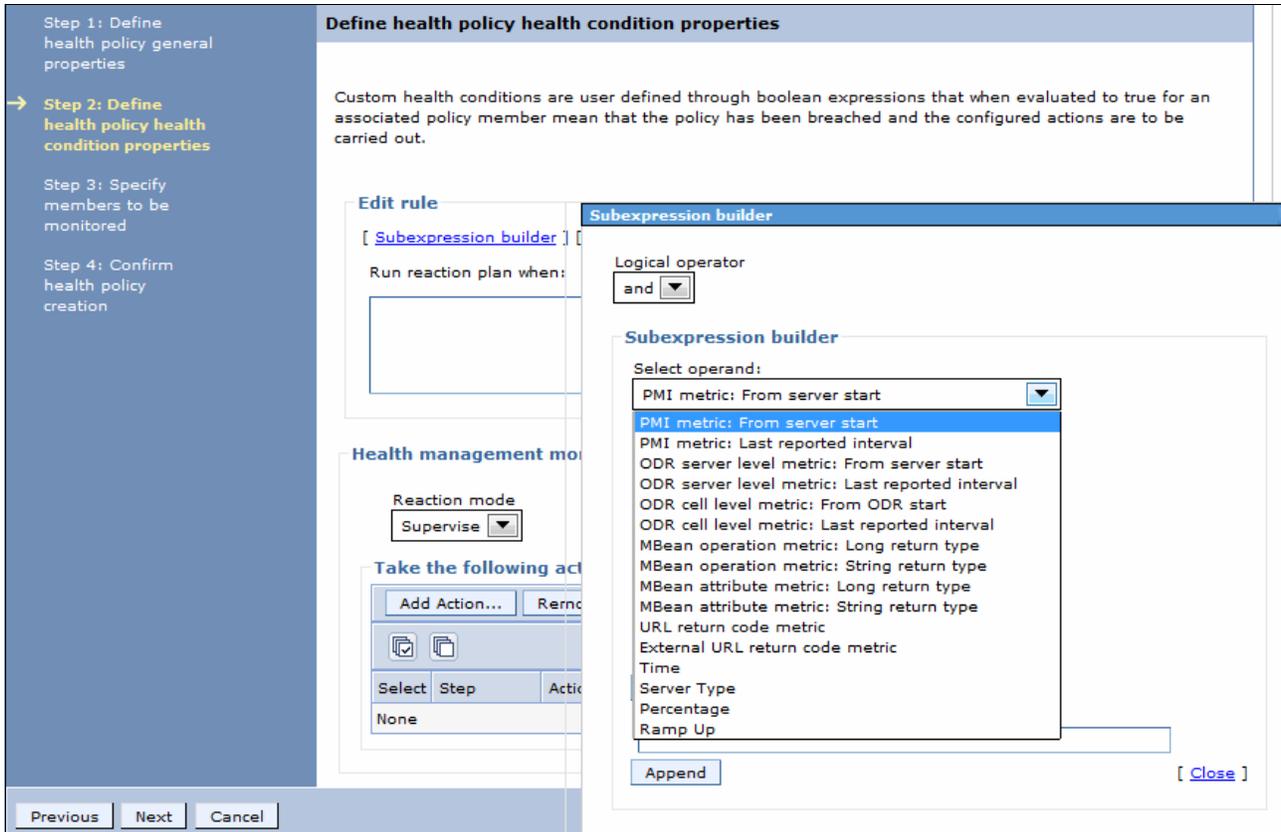


Figure 6 Define a custom health policy

Administrators should turn on the following custom health policies (see Table 5).

Table 5 Health policies

| PMI module | PMI module | PMI module | Preferred actions |
|------------------------|---------------------------|--|---|
| Thread pool module | Concurrently hung threads | PMIMetric_FromLastInterval \$threadPoolModule \$concurrentlyHungThreads > 3L | <ul style="list-style-type: none"> ▶ Take thread dump ▶ Restart server |
| Process module | Process total memory (kb) | PMIMetric_FromLastInterval \$xdProcessModule \$processTotalMemory > 2048L | <ul style="list-style-type: none"> ▶ Restart server |
| Connection pool module | Average wait time (ms) | PMIMetric_FromLastInterval \$connectionPoolModule \$avgWaitTime > 5000L | <ul style="list-style-type: none"> ▶ Custom action to switch to another connection pool ▶ Notify administrator of database issues |

Similarly, you have the ability to define custom actions, which can be any executable file in the target environment, if the predefined health actions do not meet your needs. Network

Deployment provides flexible reaction modes when a health policy breach is detected. The reaction can be configured to occur automatically or you can request to be notified first with the ability to approve or deny the action.

Table 6 illustrates examples of the problems that can be solved by the health management feature.

Table 6 Problems that are solved by the health management feature

| | |
|--|--|
| <p>Auto-detect application server health conditions</p> | <p>Configure health management to:</p> <ul style="list-style-type: none"> ▶ Automatically detect when the average response time from a particular application server stays above a threshold for a certain length of time and take corrective action ▶ Automatically detect when an application server is taking more than a configurable percentage of time performing garbage collection as opposed to doing real work. When this occurs, you want to generate a heap dump, restart the application server without incurring any outage, and notify an administrator ▶ Automatically detect when an application server stops responding and automatically restart the application without incurring an application outage |
| <p>Auto-detect application and connection pool health conditions</p> | <p>Configure health management to:</p> <ul style="list-style-type: none"> ▶ Automatically detect when an application has a memory leak and take corrective action including taking a heap dump, restarting the server without incurring an outage, and notifying an administrator ▶ Automatically detect when a thread is hung, generate a javacore for debugging, restart the application server, and notify an administrator while incurring no application outage <p>Generate Simple Network Management Protocol (SNMP) traps to:</p> <ul style="list-style-type: none"> ▶ Automatically detect when a connection pool to a database slows down, switch to another connection pool, send an email, and generate an SNMP trap |
| <p>Custom health management features</p> | <p>Create health policies to:</p> <ul style="list-style-type: none"> ▶ Require administrator approval before taking corrective action ▶ Trigger a health condition based on a Performance Monitoring Infrastructure (PMI) metric or any MBean call or any combination of them and take corrective action <p>Create health custom policy actions to:</p> <ul style="list-style-type: none"> ▶ Have the flexibility of adding an arbitrary condition or an arbitrary action to the health management system |

For more information: For details about this topic, refer to the product documentation on configuring the Health Management Feature at the following website:

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp?topic=/com.ibm.websphere.wve.doc/ae/twve_odhealth.html

Scenario 4: Eliminate single point of failure for administrative functions by using the high availability deployment manager function

Network Deployment supports redundant deployment managers, allowing multiple deployment manager processes to be concurrently active in a cell by using an active-standby configuration. Configuring multiple deployment managers within a cell provides highly available administrative access. All of the deployment managers work from a single master repository, which is stored on a shared file system. ODRs route administrative requests to the currently active deployment manager, and if the original process fails, the ODR routes the request to one of the other deployment managers. The standby deployment manager is fully initialized and ready to work. However, it cannot be used for administration and will reject any login and Java Management Extensions (JMX) requests. If the active deployment manager is stopped or fails, the highly available deployment manager component recognizes the loss and dynamically switches the standby into active mode so it can take over administrative responsibility. The active and standby deployment managers share workspaces so when a deployment manager takeover occurs, work is not lost. The ODR automatically recognizes the election of the new active deployment manager and reroutes administrative requests to the new active deployment manager.

Failover to the new active deployment manager is depicted in Figure 7.

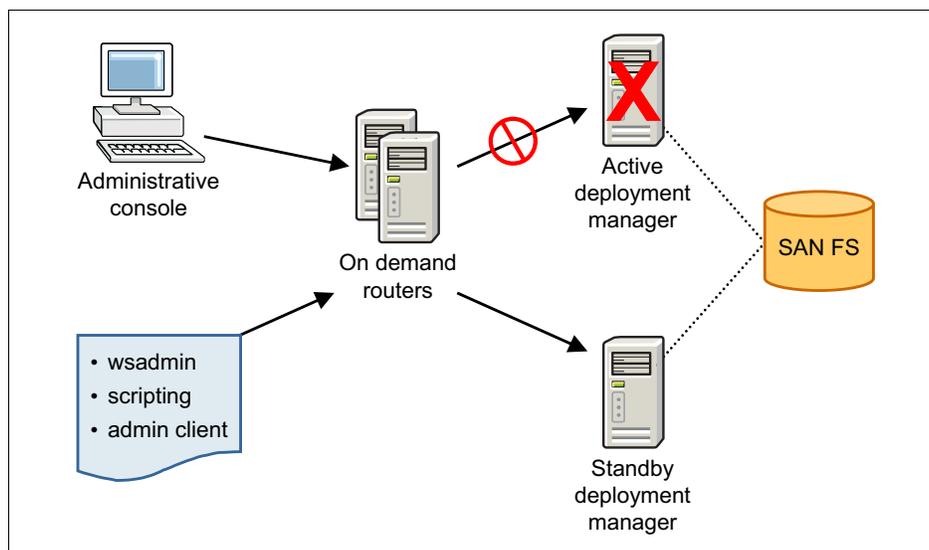


Figure 7 Failover to a new deployment

For more information: For details about this topic, refer to the product documentation on how to configure a high availability deployment manager environment, at the following website:

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp?topic=/com.ibm.websphere.wve.doc/ae/twve_xdsconfig.html

Consider the following preferred practices when implementing a high availability deployment manager topology in your environment:

- ▶ To configure the HADMGR, the prerequisite requirement is a fast locking-capable distributed file system like NFS version 4 or IBM SAN FS, which is used for DMGR configuration repository.
- ▶ The machines that you plan to install the deployment managers on need to have access to the shared network file system and have the file system mounted. The dmgr profiles will be stored out in the network file system, so the profiles are mutually accessible to the primary and secondary dmgr processes.
- ▶ Always check to ensure that the shared file system is mounted on deployment manager machines, because after the machines reboot, previous mounted file systems might not be available.

The next scenarios: The next three scenarios pertain to Network Deployment capabilities for batch workloads. For details, refer to the product documentation for the Compute Grid feature of WebSphere Extended Deployment at:

<http://pic.dhe.ibm.com/infocenter/wasinfo/cgwas80/index.jsp>

Scenario 5: Leverage OLTP infrastructure for new batch workloads

Batch is a workload pattern that delivers high efficiency for bulk processing tasks and enables IT efficiency by using under-utilized capacity. Network Deployment provides the capability to use your OLTP infrastructure for new batch workloads. WebSphere batch workloads gain efficiency when reusing business services through collocation. Container-based checkpoint and restart capability delivers improved resiliency for long running batches. Further efficiency is possible by using system-managed batch parallelization.

To add batch capability to an existing Network Deployment cell, perform the following tasks:

1. Define a batch scheduler cluster
2. Configure the batch scheduler
3. Define the batch execution cluster
4. Implement and install batch applications

Scenario 6: Improve batch elapsed-time by using system managed parallel batch

Use the parallel job manager to convert eligible WebSphere batch workloads to run as parallel subjobs to improve elapsed time.

To use this capability, perform the following tasks:

1. Implement a partition rule for selected batch jobs. See the Parameterizer API topic at the following website:

http://www14.software.ibm.com/webapp/wsbroker/redirect?version=compgrid8&product=was-nd-mp&topic=rgrid_cgpmjspi

2. Update the job xJCL to specify multiple instances. See the run xJCL option at the following website:

http://www14.software.ibm.com/webapp/wsbroker/redirect?version=compgrid8&product=was-nd-mp&topic=cgrid_xdbatchjcl

Scenario 7: Extend reach of enterprise workload scheduling to WebSphere batch

Include WebSphere batch jobs in your enterprise workload scheduler plan by using the WSGrid utility. *WSGrid* is a common connector for use on all platforms and an optional, optimized, native connector for use on IBM z/OS®.

To use this capability, perform the following tasks:

1. Configure WSGrid messaging communication
2. Use WSGrid in your workload scheduler plan
3. Use WSGrid.sh on distributed or z/OS UNIX System Services workload scheduler environments. Use the WSGrid job step program on z/OS for integration with JES/JCL

Authors

This paper was produced by a team of specialists from around the world working at the IBM International Technical Support Organization, Rochester Center.

Ravi Gopalakrishnan is a Product Manager in WebSphere with over 17 years experience in the software industry. Before this, he was a development manager in the IBM Rational® Division within IBM. He has an MBA from Duke University and also has degrees in MS Computer Science and BS Computer Engineering.

Keith Smith is the Chief Architect for WebSphere Virtual Enterprise and Lead Architect for IBM DataPower® Edge solutions. Keith leads the integration of WebSphere Virtual Enterprise technology into products such as DataPower Web Application Gateway, IBM Workload Deployer, PureSystems™, and various stack products and is heavily involved in cloud-based solutions.

Christopher Vignola is the Lead Architect for WebSphere Application Server Systems Management and the Spec Lead for JSR 352 Batch Applications for the Java Platform.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Obtain more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new IBM Redbooks® publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

© Copyright International Business Machines Corporation 2013. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This document REDP-4990-00 was created or updated on May 10, 2013.

Send us your comments in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:
ibm.com/redbooks
- ▶ Send your comments in an email to:
redbooks@us.ibm.com
- ▶ Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400 U.S.A.



Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

DataPower®
IBM®
PureSystems™
Rational®

Redbooks®
Redpaper™
Redbooks (logo) ®
Tivoli®

WebSphere®
z/OS®

The following terms are trademarks of other companies:

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.