

**Solution Brief**

2012

D-Man Distributed monitoring for IBM Websphere

Your business can only be as flexible as the IT systems that support it. So, to survive and thrive in an environment of market volatility and intense competition, you need an IT infrastructure that can cope with constant change. With WebSphere® Application Server, you have a foundation for your J2EE and SOA architecture that is flexible and secure enough to cope with any challenge.

Based on years of experience in large e-Business projects, SecurIT has developed its revolutionary D-Man™ monitoring solution. D-Man provides a flexible, light-weight framework for monitoring complex distributed WebSphere environments.

This paper contains a high-level overview of the architecture of D-Man and how it can be used for monitoring WebSphere infrastructures.

For a better understanding of this document, it is recommended for the user to be familiar with IBM WebSphere Application Server and to have read the "SecurIT D-Man. Distributed monitoring for e-Business environments" Product Description.

Solution Brief

About IBM WebSphere Application Server

IBM WebSphere Application Server provides a scalable infrastructure for hosting SOA and J2EE applications. WebSphere can be deployed in a single server, non-federated mode for easy deployments or in a scalable, highly-available federated mode. The latter one reflects a typical deployment within large enterprises. With D-Man for WebSphere, SecurIT mainly focuses on federated architectures.

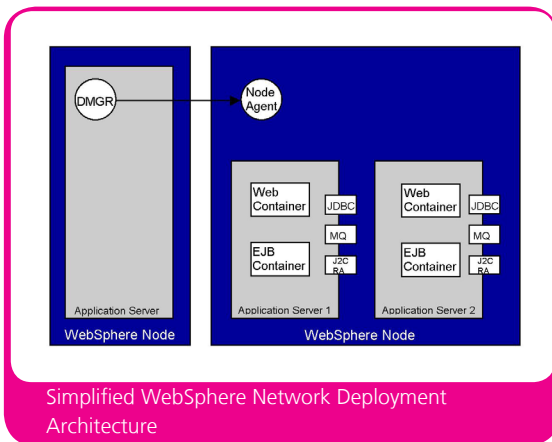
IBM WebSphere Application Server contains two very crucial modules that are essential for managing complex distributed WebSphere environments: WebSphere Application Server Network Deployment and Java Management Extensions.

While WebSphere Application Server Network Deployment is used to manage the WebSphere infrastructure itself, the Java Management Extensions provide the instrumentation to manage WebSphere applications and resources. Jointly, these two modules provide the basis for checking the health of the overall WebSphere infrastructure and the hosted applications and resources. As such these two modules provide the most logical integration point between WebSphere and any monitoring solution, like SecurIT D-Man.

To fully grasp the impact of such integration, below we have provided a short introduction to WebSphere Application Server Network Deployment and Java Management Extensions.

WebSphere Application Server Network Deployment

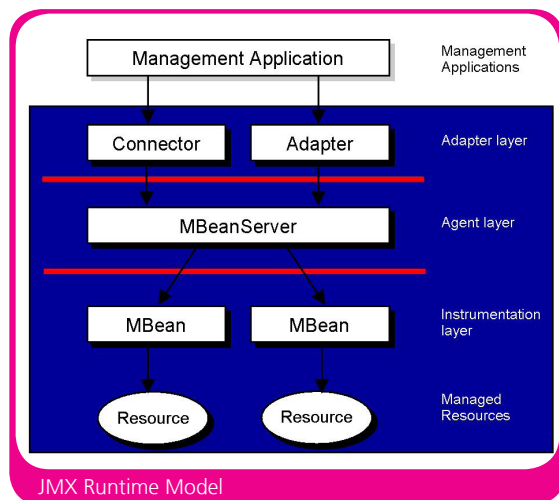
The following diagram illustrates the components within a WebSphere Application Server Network Deployment Infrastructure.



A **Network Deployment Manager** (DMGR) is the central administration instance of the system. A **WebSphere Node** is a physical construct hosting a series of Application Servers for scalability and availability purposes. A **Node Agent** is situated on each WebSphere Node. A Node Agent is responsible for synchronizing application configuration- and availability data between a WebSphere Node and the Deployment Manager.

The availability and consistence of the synchronization events between Node Agents and Deployment Manager is crucial for the health of a WebSphere Infrastructure and as such this should be closely watched.

WebSphere introduces a Management Runtime based on the **Java Management eXtensions (JMX)** specification. JMX is a standard framework for managing resources for Java Applications – the following diagram provides an overview of JMX.



An application which offers manageable resources needs to be instrumented (cfr. managed). The application and its resources get connected to a **MBean** (Managed Bean). The MBean provides an application level resource management Interface.

An MBean itself interacts with a **MBeanServer** which serves as the backend for distributed Management Applications.

Because the JMX specification is transport protocol independent, a vendor can map the JMX management protocol to HTTP, IIOP or JMS. A Management Application can either connect directly to a MBeanServer using a **Connector**, or if it chooses to use another protocol, through an **Adapter**.

JMX events provide crucial information about the availability and health of WebSphere applications and resources and as such they should be closely watched.

About SecurIT D-Man Distributed Monitoring

Traditional monitoring systems don't fit very well e-Business environments. Because they were initially designed to monitor internal processes and applications mainly from a technical point of view, they don't pay too much attention to infrastructural security and the user's perspective.

The first aspect is reflected by the heavy frameworks on which many of these systems rely. These frameworks tend to use protocols that are not supported within a Demilitarized Zone (DMZ) or across firewalls. The latter aspect might result in e-Business environments of which the isolated tiers are in perfect health, but where the service as a whole does not behave as expected.

SecurIT has designed and built a monitoring solution for Distributed e-Business Environments, from the ground up, with exactly these aspects in mind. It addresses both, helping IT-Administrators isolating and solving IT problems appropriately as well as end-to-end Monitoring of business applications to identify service outage before it effects end-users.

This solution was released by SecurIT under the name SecurIT D-Man in 2004. The D-Man monitoring framework consists of two main components:

- D-Man Application Manager (DAM)
- D-Man Data Manager (DDM)

The D-Man Application Manager (DAM) represents a traditional monitoring agent. It is used to gather information from any component in the infrastructure that needs to be monitored. Typically there will be one DAM residing on every single server in the e-Business environment. A DAM is responsible for monitoring any activity that is relevant to the server on which it resides. This could be local as well as remote activity.

The following example illustrates some aspects of an e-Business environment that can be monitored by a DAM residing on a server within the DMZ:

- Availability of the process running a Secure Reverse Proxy
- Response time of a remote Directory Server (e.g. LDAP) when validating a username/password authentication by the Secure Reverse Proxy
- Resource consumption of a Web Application Server

The D-Man Data Manager (DDM) sits at the core of the D-Man monitoring framework. On one site it is used for the administration of the D-Man environment while on the other site it is responsible for collecting and analyzing the data that is gathered by all DAMs in the infrastructure.

Please refer to the "SecurIT D-Man. Distributed monitoring for e-Business environments" Product Description for additional information on the Core D-Man Infrastructure.

The D-Man for IBM WebSphere Application Server monitoring architecture

The main advantage of D-Man for WebSphere is that it builds upon the well known D-Man framework. This means that it can leverage an existing D-Man infrastructure and the available e-Business monitors and that it allows enterprises to easily extend this infrastructure into their WebSphere Application Server environment. With D-Man for WebSphere enterprises can concentrate on the monitoring of their WebSphere systems and the applications running on top of it while integrating the monitoring results within their overall e-Business status reporting.

The D-Man for WebSphere monitoring architecture is very light-weight. It provides the monitors that an enterprise needs out-of-the-box. It leverages the monitors to such an extent that they provide reliable information based on the knowledge of how WebSphere infrastructure components interact with each other.

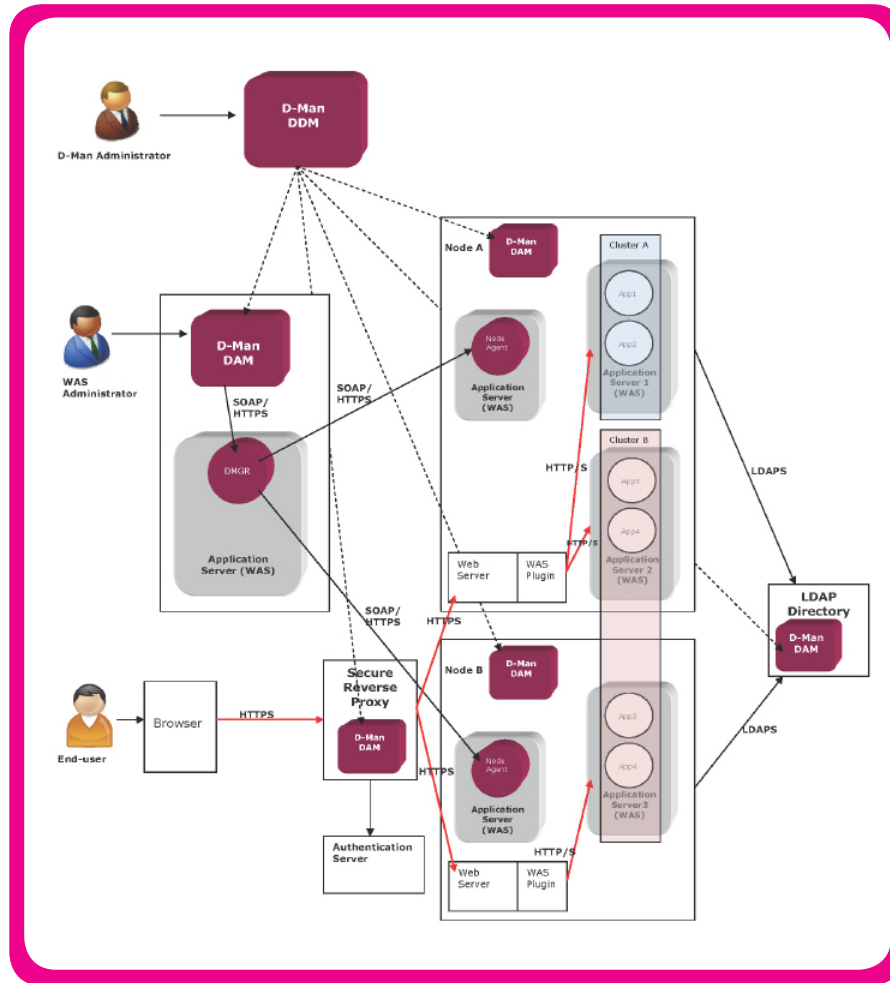
Based on experience gained in large-scale WebSphere projects, the monitors were designed to go far beyond the basic functionality of traditional monitors. By deciding for D-Man for WebSphere, enterprises take advantage of smart "pathfinders" which detect weaknesses of a WebSphere system before they can cause larger damage and create bad end-user experience.

The diagram on the next page provides an overview of how D-Man for WebSphere can be integrated into a WebSphere reference architecture. It also shows how it can leverage existing D-Man components to provide a complete end-to-end monitoring solution.

E.g. it contains a D-Man DAM on a reversed proxy, that is used to provide access to the WebSphere based applications, and it contains a D-Man DAM on an LDAP server, which is one of the base components on which WebSphere security relies.

End-users, situated on a client tier, connect to the Enterprise using a Browser-based HTTPS connection which gets intercepted by a Secure Reverse Proxy that is responsible to act as Web Application Firewall. It authenticates users and provides coarse-grained access control on URLs. The Reverse Proxy passes the authenticated request to Web Server situated on a WebSphere Node.

Solution Brief



The Web Server passes the request to the Business Application that is hosted by the WebSphere Application Server. The application can be potentially served by a Cluster of Application Servers. The WebSphere Infrastructure is secured using an LDAP registry. Once the business requests hits the application server the user gets authenticated and the business request gets validated by the WebSphere Application Server Security Runtime.

This scenario shows that several components need to be working properly in order for the end-user request to be services successfully. Therefore all these components should be monitored to make sure that the Business Application is fully operational.

D-Man for IBM WebSphere monitoring concept

As shown by the previous figure IBM WebSphere is usually part of a larger e-Business system also consisting of components like a Secure Reverse Proxy, Authentication Server, LDAP Server and other servers.

To monitor an IBM WebSphere based environment, it is important that also all these other components stay in the picture and that monitored events are correlated

between them. Finally, also the health of the systems on which these components run, are part of the equation.

Therefore D-Man for IBM WebSphere is based on three important pillars:

- Traditional D-Man Monitors
- Generic D-Man e-Business Monitors
- D-Man for WebSphere Monitors

These different pillars are described in more detail below.

Traditional D-Man Monitors are used to monitor the status of the different WebSphere components and components on which WebSphere relies.

Some examples are:

- Availability, memory and CPU usage of a WebSphere Node
- Availability, memory and CPU usage of a WebSphere Deployment Manager
- Availability, memory and CPU usage of the LDAP process
- Availability, memory and CPU usage of the front-end HTTP server
- Logfile monitors for WebSphere, LDAP and HTTP logs

Solution Brief

These already provide a good indication of the fact that the WebSphere environment is ready to operate. It does however not yet provide evidence that it is really providing any service or running any applications.

In order to have an indication of the WebSphere operational status the list of monitors can be extended with the **Generic D-Man e-Business Monitors**. These monitors show whether the WebSphere environment can rely on the required surrounding components that are crucial for its serviceability.

Some examples are:

- Validity of X.509 Certificates
- Integrity and response times of HTTP Reversed Proxy and front-end HTTP server
- Responsiveness (HTTP) of a WebSphere based application
- LDAP Search Response time
- Serviceability of an external security server (e.g. TAM Policy Server)

In WebSphere environments that are secured using Tivoli Access Manager, end-to-end monitoring can be achieved by using the D-Man for Tivoli Access Manager monitoring suite (see also: SecurIT D-Man. Distributed monitoring for Tivoli Access Manager (Product Description).

Finally the set of monitors is completed with the **D-Man for WebSphere monitors**. These monitors are aimed at the correct and optimal operation of WebSphere and the applications hosted by it. These monitors are described in more detail in the following paragraphs.

D-Man for WebSphere monitoring dimensions

D-Man for WebSphere monitors are classified according to a two dimensional model distinguishing them both in Category and Platform.

D-Man for WebSphere monitoring category

We will first look at the categories. Monitor categories refer to the technology on which the monitors are built.

D-Man for WebSphere monitors mainly fall into three categories

- Event based monitors
- Query based monitors
- Logfile based monitors

While the used technology might not have a direct impact on what kind of information can be provided by the monitor (in many cases the same functionality can be provided within all these categories), it does influence the efficiency of the monitor according to

the monitored infrastructure and the SLAs of that infrastructure.

Problem Statement

Event based monitors provide an interrupt kind of notification. "As soon as something changes, we'll let you know, otherwise we remain silent". Obviously this is the optimal approach. However, such monitors rely on a runtime system that is part of the monitored environment (e.g. WebSphere JMX). If the WebSphere runtime fails, the event based monitors might also fail.

An example

An Event based monitor might register for a JMX event to track the status of an application. During a short maintenance interrupt the monitoring system might miss an event that states that the application stopped. Once the monitor is restarted it will assume the application is still running and won't report a problem. With a Query based monitor, the problem will be revealed at the next monitor cycle.

Categories

Event based monitors rely on the JMX subsystem of WebSphere. They are very efficient because they will only provide alerts when something changes. However, not all issues are mapped onto events and events might not show up when the runtime on which it relies is not available.

Query based monitors are built around the APIs that are provided by WebSphere. Most of them use the PMI (Performance Monitoring Interface) extension. Because they are not event driven, they should be scheduled to run at pre-defined intervals. The advantage is however that they are less dependent of the WebSphere runtime and will provide useful information even if the WebSphere runtime would fail.

Logfile based monitors rely on the native D-Man logfile adapter functionality. Logfile monitors look for the occurrence, or absence, of configurable messages in logfiles produced by the monitored system. Logfile monitors have the advantage that they do not rely at all on the runtime and provide useful information even if the monitored system completely fails. On the downside, these monitors can however be CPU intensive and as such should be used carefully.

D-Man for WebSphere monitoring platform

The next dimension is related to the platform on which the monitors are operated. With platforms we refer to the server on which the WebSphere instance is running.

We make a distinction between the following two platforms:

- WebSphere Deployment Manager
- WebSphere Node

Basically the monitoring category is independent of the platform. This means that, with the exception of

Solution Brief

a few examples, most monitors can run both on the WebSphere Deployment Manager as on any WebSphere Node.

The best platform on which to operate a monitor again depends on the monitored infrastructure and the SLAs of that infrastructure.

Problem Statement

By running the monitors on the WebSphere Deployment Manager we can largely reduce the complexity of the monitoring infrastructure. Because of the WebSphere sub-system, we are now still able to check the status of the different WebSphere Nodes, Application Servers and applications running on them.

However, if for some reason this WebSphere sub-system would fail, we cannot be sure anymore about the quality of the monitored information. For this reason, it is recommended to run the monitors as close to the source as possible (preferably on the WebSphere Nodes), at least for those nodes that operate very crucial applications.

An example

An application monitor running on the WebSphere Deployment Manager might state that an application running on a particular WebSphere node is available. That node however is experiencing a problem reaching the LDAP server because of an invalid Certificate. As a result of this the application is not serving any requests but this goes unnoticed for the application monitor on the WebSphere Deployment Manager.

While running monitors only on the WebSphere Deployment Manager can seriously reduce the complexity of the monitoring environment, it might fail to notice service problems until the system breaks down completely.

Platforms

Monitors running on the WebSphere Deployment Manager basically are restricted to monitoring the health and status of the local Application Server directly and of the remote Application Servers (on WebSphere Nodes) and their applications by means of the WebSphere runtime (e.g. JMX, PMI, ...). In theory they should be able to keep an eye on the overall WebSphere cluster and its applications and as such reduce the overhead of a monitoring system to an absolute minimum.

However, because they rely on a service provided by the subsystem they are monitoring (WebSphere runtime) it is obvious that these monitors alone are not sufficient.

Monitors running on the WebSphere Nodes basically only have access to the local Application Servers and their applications. As such these monitors should be deployed on any WebSphere Node within the infrastructure (or cluster). While creating a larger overhead than monitors running on the WebSphere Deployment Manager, they provide more accurate and up-to-date information.

D-Man for WebSphere monitoring model

It is clear that while monitors belonging to these different categories and platforms do provide overlapping functionality, there isn't a single best monitoring model for all WebSphere environments.

The best model is one that aims at finding a compromise between efficiency, availability and usability and is driven by the WebSphere architecture and the WebSphere application SLAs. Such a model correlates between the results produced by a collection of monitors from all categories and platforms.

Apart from providing the monitors itself, SecurIT D-Man for WebSphere provides guidelines that allow customers to tailor the monitoring model to best suite their environment and expectations. In other words it guides the monitoring operator in his process of selecting the most appropriate D-Man for WebSphere monitors and model.

D-Man for WebSphere monitors

D-Man for WebSphere provides the following WebSphere monitors out-of-the-box. Please note that list is not exhaustive as new monitors are added on a regular basis.

D-Man for WebSphere Status Monitors

These monitors report on the status of a particular WebSphere component. It makes a distinction between the following states:

- Running
- Starting
- Undefined
- Stopping
- Stopped

These monitors are able to report this state for the following WebSphere components or modules:

- WebSphere Application Server
- WebSphere Cluster
- WebSphere Deployment Manager
- WebSphere Node
- WebSphere Application

The monitors are available in the following categories and on the following platforms:

- Event based monitor
- Query based monitor
- Logfile based monitor
- WebSphere Deployment Manager
- WebSphere Node

Solution Brief

D-Man for WebSphere Performance Monitors

An important indicator for a healthy WebSphere system is its resource consumption in terms of memory, CPU and I/O activity.

The help identifying performance problems and to help tuning an environment, WebSphere collects performance data and provides interfaces that allow external applications to monitor this performance data.

The D-Man for WebSphere Performance Monitors leverage these interfaces. They provide access to the following (non-exhaustive list of) data:

Java Virtual Machine

- Uptime
- Free Memory
- Used Memory
- Heap Size
- Garbage Collection time
- Garbage Collection count
- Garbage Collection interval

Web Application

- Average response time
- Total number of requests
- Reload Count

Thread Pools

- Active time
- Active count
- Pool size

Session Manager

- Active Count
- Live Count

System Data

- CPU usage since last restart
- CPU usage since last measurement
- Free memory

The monitors are available in the following categories and on the following platforms:

- Event based monitor
- Query based monitor
- WebSphere Deployment Manager
- WebSphere Node

D-Man for WebSphere Synchronisation monitor

This monitor keeps an eye on the synchronisation process between the WebSphere Deployment Manager and the WebSphere Nodes. As this synchronisation is crucial within a distributed (clustered) WebSphere environment to allow the Deployment Manager to have full control over the environment, this monitor is of utmost importance.

The monitor is available in the following categories and on the following platforms:

- Event based monitor
- Logfile based monitor

- WebSphere Deployment Manager
- WebSphere Node

D-Man for WebSphere LDAP monitor

The most WebSphere configurations are using LDAP to secure a WebSphere environment. Hence an unavailability of the LDAP Server can cause an outage of the entire WebSphere System.

Following issues may occur upon unavailability of LDAP:

- WebSphere will not be able to authenticate a user
- The WebSphere Nodes will not synchronize anymore

The D-Man for WebSphere LDAP Monitor will provide event information using standard LDAP mechanisms to acquire information on its current state and whether it is reachable from a certain WebSphere Node. If required, the communication with LDAP can be secured using SSL.

The monitor is available in the following categories and on the following platforms:

- Query based monitor
- Logfile based monitor

- WebSphere Deployment Manager
- WebSphere Node

D-Man for WebSphere Version Monitor

The functionality of WebSphere is very sensible on version numbers. In many environments it is happening that a new WebSphere Node is getting installed on a new set of hardware, and the install procedure uses a different Minor version or fixpack. Though WebSphere interoperates quite well between different versions on a cross cell level, IBM strongly recommends synchronizing the WebSphere versions within a Cell. It is highly recommended to run at exactly the same major, minor and fixpack level on all of the nodes within a Cell, including the WebSphere Deployment Manager.

This monitor reports on deviations of a certain Version Number.

The monitor is available in the following categories and on the following platforms:

- Query based monitor
- WebSphere Deployment Manager
- WebSphere Node

Solution Brief

D-Man for WebSphere Trace Service Monitor

A trace file written by WebSphere can affect performance dramatically. Sometimes in an operational environment an operator sets the trace specification to a very high level and forgets to set it back.

This monitor keeps an eye on the trace level of any WebSphere Node and generates an alert if the level increases a certain threshold too long.

The monitor is available in the following categories and on the following platforms:

- Query based monitor
- WebSphere Deployment Manager
- WebSphere Node

Name Service Monitor

If a Name Service does not respond to naming lookup requests, then this effects the functionality of the whole WebSphere Cell – applications would not be able to lookup referenced objects such as EJBs and Servers would not be able to startup the applications correctly anymore.

WebSphere provides various ways to lookup an object within a name service: using a local call, using a node agent level remote call or using a cell level remote call. The Cell level Name Service is the most critical one.

Naming Lookups will be processed by this Monitor on each Node. In addition the JMX Monitor will monitor the Name Service from the Deployment Manager's point of view.

The monitor is available in the following categories and on the following platforms:

- Query based monitor
- Event based monitor
- WebSphere Deployment Manager
- WebSphere Node

JMS Monitor

The primary concern of a JMS server is bridging the transport of asynchronous messages to JMS providers such as MQ. In large scale environments WebSphere MQ is the commonly used provider for asynchronous messaging.

A JMS Server must be available on every Node. It is the communication endpoint for each server situated on a WebSphere Node.

A JMS server which is not responding causes failure to all JMS Clients and services (such as Message Driven Beans) relying on asynchronous messaging on a node.

This Monitor verifies the availability of any JMS Server within the WebSphere environment. The monitor is

available in the following categories and on the following platforms:

- Query based monitor
- WebSphere Node

WebSphere Administration Event Monitor

This Monitor provides an operational tool to help identifying system changes or other system events which get registered and audited by the Deployment Manager. A typical use case is that where a server has been shutdown accidentally or a service failure on a backend connector has been audited.

D-Man for WebSphere acts as single, unified interface to such administrative events.

The monitor is available in the following categories and on the following platforms:

- Query based monitor
- WebSphere Node

Session Manager Monitor

The WebSphere Web Container Session Manager manages the Application Session Lifecycle of user sessions. For performance reasons application sessions can be set to time out after a certain inactivity interval has been reached. In this case the sessions are cleared from the session manager cache. When this cache reaches however its maximum amount of entries, users might loose active sessions.

While this information is made available through the WebSphere Performance interface, it relies on the availability of PMI. This Session Manager Monitor does not require PMI.

The Session Manager Monitor will generate an event when the number of entries in the cache reaches a critical level.

The monitor is available in the following categories and on the following platforms:

- Query based monitor
- WebSphere Node

WebSphere Portal Server Monitors

WebSphere Portal Server provides an implementation of a Portlet Container. Portlets are special purpose servlets that focus on a better user experience by improving and unifying the look and feel of application interface.

The WebSphere Portal Server is using a similar security model as the WebSphere Application Server, though it relies on its own interaction with a User Registry (e.g. LDAP) to authenticate users and a Portal Database

Solution Brief

to personalise views. The unavailability of these repositories would result in a failure of the WebSphere Portal Server environment.

The WebSphere Portal Server monitors keep an eye on the health of these repositories.

The monitor is available in the following categories and on the following platforms:

- Query based monitor
- Logfile based monitor
- WebSphere Node

JDBC Monitor

Many WebSphere based applications rely on business data located within a relational database accessed by a JDBC Connection which is typically managed by a JDBC Connection Pool Manager. Applications ask the JDBC Connection Pool Manager for a Managed Connection which represents a Session to the Database. There are various Use Cases which cause an application client to fail its database lookup:

- The Connection Manager has received its maximum session limit
- The JDBC Driver used for the connection is incompatible with the Database
- The JDBC Configuration is incorrect
- The credentials used by the application are incorrect
- Network problems between the Application Server and the Database
- Inappropriate configured firewall between the Application Server and the Database
- Unavailability of the Database System
- Unavailability of the Server hosting the Database System

A failing database connection is usually causing the application to fail serving the business requests and causes bad user experience.

This monitor provides various mitigations against the risk of database and hence business outages by watching for the most common configurations within the application server. It frequently tries to connect to the database backend system the same way the application tries to use it.

The monitor is available in the following categories and on the following platforms:

- Query based monitor
- WebSphere Deployment Manager
- WebSphere Node

Other Monitors

Other Monitors that are currently under development and that will become available soon are:

- EJB Monitors
- JTS Monitors
- Web Container Monitors
- WebSphere Security Monitors

D-Man for WebSphere Key Features

Platform independent. D-Man is written in Java and as such monitoring agents are supported on any platform running a JRE. Among the tested platforms are: AIX, HP-UX, RedHat, Solaris, SuSE and Windows.

Straightforward installation. All D-Man components are provided as native platform packages. Alternatively, all components can also be provided as single TAR files that allow straightforward fully-automated installation and configuration. Furthermore, there are no external pre-requisites.

Embedded Security. All internal D-Man transactions run over HTTPS using mutually authenticated SSL. Furthermore, connections are only established from secure to less-secure zones. This makes D-Man entirely compliant with firewall and DMZ policies, allowing it to even scale beyond the company boundaries. And last, but not least, monitor scripts are never written to the file system. As such there is no risk of these scripts being overwritten by malicious users or programs.

Self Monitoring. D-Man provides extensive self-monitoring. By means of embedded monitoring of the monitoring agents and a D-Man heartbeat, any internal problem can be detected and reported (e.g. to TEC or HP Openview) immediately.

Based on Open Standards. D-Man is entirely based on open standard. For its internal communication it relies on HTTP and SSL. For rendering and exporting events it uses HTML, XML and XSL. Integration with commonly used monitoring systems (TEC, HP Openview, etc.) can be accomplished using SNMP, SMTP, CLI or any native API plug-in. And finally it also supports JMX for registering to event based systems.

Central Configuration and Administration. Once installed, all configuration and administration of D-Man is done through a central D-Man component using only a browser. As such, there is no need to access the monitoring agents directly, not even when new monitoring targets are being deployed.

User-centric and Technical Monitoring. D-Man provides a balance between user-centric and technical monitoring. Where problems in any e-Business environment usually have a technical cause, because of the inherent redundancy of such environments, the impact of any technical problem is best measured from the end-user's perspective. D-Man correlates end-user's experience back to technical causes.

Solution Brief

Event Correlation. Most monitoring solutions provide features to correlate event information. These features are however so hard to configure, and require expert knowledge, that they are hardly used. D-Man provides a very straightforward point-and-click correlation engine which makes it easy to cross-analyse event information. Furthermore, the D-Man suites come out-of-the-box with pre-defined correlation rules.

Focus on WebSphere expertise. Leveraging the framework of SecurIT D-Man, the focus of D-Man for IBM WebSphere Application Server is situated around the WebSphere expert knowledge that is embedded within the D-Man for WebSphere monitors. D-Man for IBM WebSphere Application Server represents a virtual WebSphere expert that will be of unmatched value to an operator trying to analyse and solve any WebSphere production issue.

Embedded WebSphere interfaces. A lot of interesting status information can be retrieved directly from WebSphere through the exposed interfaces (e.g. JMX, PMI). These interfaces require however to establish a context with the WebSphere Application Server. Being only available in Java, the interfaces are hard to access from traditional monitoring solutions. Because each D-Man DAM has the WebSphere interfaces embedded, the contexts are provided transparently and can be leveraged by all WebSphere monitors.

Pro-active WebSphere monitoring. Within critical e-Business environments, pro-active monitoring is of utmost importance. The idea is to capture problems before their effects hit the end-user. D-Man for WebSphere provides a number of pro-active monitors that control WebSphere's network usage and thread handling. These monitors will warn operators when chances are high that problems are about to occur.

LDAP - Light-weight Directory Access Protocol

DAM - D-Man Application Manager

DDM - D-Man Data Manger

DMGR - Deployment Manager

J2EE - Java 2 Enterprise Edition

JDBC - Java Database Connectivity

JMX - Java Management eXtensions

IIOp - Internet inter-ORB protocol

* * *

For more information www.securit.biz

Glossary

JTS - Java Transaction Service

JDBC - Java Database Connectivity

JNDI - Java Naming and Directory Interface

HTTP - Hypertext Transfer Protocol

PMI - Performance Measurement Infrastructure

Copyright © 2012 SecurIT bvba. All Rights Reserved

No part of this document may be copied, translated, or reduced to any electronic medium or machine readable form, in whole or in part, without the prior written consent of SecurIT bvba. SecurIT has made every effort to ensure that the information in this document is accurate and complete; however, SecurIT assumes no responsibility for any errors or omissions. Information in this document is subject to change without notice.

Other product names mentioned in this paper may be trademarks or registered trademarks of their representative companies and are hereby acknowledged.