



IBM Software Group

IBM WebSphere Application Server V6.1

Diagnostic Providers



@business on demand.

© 2006 IBM Corporation
Converted to video May 15, 2015

This presentation will cover Diagnostic Providers in WebSphere® Application Server V6.1.

Agenda

- Overview
- Using Diagnostic Providers



This presentation will give you an overview of the Diagnostic Provider framework, and will also discuss the ways you can use Diagnostic Providers.

Section

Overview



This section will introduce Diagnostic Providers.

Diagnostic Providers

- Diagnostic Providers are MBeans that provide diagnostic data for individual components
 - ▶ State dump
 - ▶ Configuration dump
 - ▶ Self-diagnostic test routines
- Error messages from supported components contain a diagnostic provider identifier (DPID)
 - ▶ Components can be queried directly by DPID to get more information
 - ▶ Components can also be selected from a tree view



The Diagnostic Provider framework is a new feature in WebSphere Application Server V6.1. Diagnostic Providers are JMX MBeans that provide diagnostic data for individual components within WebSphere Application Server processes. They enable you to ask a selected component to dump its configuration data, current state data, or to run a self-diagnostic test routine. Components that are instrumented with Diagnostic Providers can log their Diagnostic Provider ID, which is a JMX ObjectName to log files when they write error messages. This allows you to directly query the component that generated the error message. You can also query a component at any time by selecting it from a tree view.

Diagnostic Provider support

- New framework in V6.1
- Only implemented by a few components in this release, including:
 - ▶ Web container
 - ▶ Connection manager
 - ▶ Runtime Performance Advisor
 - ▶ Node synchronization
 - ▶ Node agent
 - ▶ Deployment Manager



Since this framework is new in version 6.1, Diagnostic Providers have not been implemented for all components. A few major components, including the Web Container and the Connection Manager have implemented Diagnostic Providers in this release.

Viewing diagnostic data

- Using the administrative console
 - ▶ Navigate to *Troubleshooting > Diagnostic Provider*
- Using wsadmin
 - ▶ DP MBeans can be queried directly from wsadmin
 - ▶ The DiagnosticService MBean provides a simpler interface to DP MBeans
- Using JMX
- Data saved to the file system is stored in `<profile_root>/logs`



To view the data generated by Diagnostic Providers in the administrative console, click the “Diagnostic Provider” link under the “Troubleshooting” menu. You can also query diagnostic provider MBeans directly using wsadmin or a JMX client of your choice. The DiagnosticService MBean is also available, providing a simpler interface to diagnostic data than querying Diagnostic Providers directly. You can save diagnostic data to the file system from the administrative console, and the data will be written to the “logs” directory of the profile that is running your administrative console, which would be the Deployment Manager’s profile in a Network Deployment configuration.

Self-diagnostic tests

- Instructs a component to run a simple diagnostic test to verify functionality
- Examples:
 - ▶ AdvisorDP can test for a memory leak
 - Requires Runtime Performance Advisor to be enabled
 - ▶ NodeAgentDP can ping servers to verify connectivity
 - ▶ DeploymentManagerDP can ping nodes
 - ▶ NodeSyncDP



Some components have implemented self-diagnostic test routines using the Diagnostic Provider framework. These tests are simple tests that verify the basic functionality of a component. A single component can have several such tests. Some of the self-tests included in this release include the ability of the Deployment Manager to “ping” node in a cell to verify connectivity, a test of the node synchronization capability, and the ability of the Runtime Performance Advisor to test for a memory leak.

Configuration data

- Displays current and startup configuration values in use by the selected component
 - ▶ Distinguished as startup-* or current-*
- Convenient troubleshooting aid
 - ▶ Collects configuration information for the component in a single place
- Some components can display hundreds of configuration values
 - ▶ Can be filtered in the administrative console
 - ▶ Can be saved to a text file



When asked for a configuration data dump, an instrumented component will display both its startup configuration values and the configuration values currently in use. The names of the displayed values are distinguished by prepending the words “startup” and “current”. A configuration data dump can be a very useful troubleshooting aid, because it presents all of the configuration values being used by a component in a single place, and you can see the values that are currently in use at a given time. The amount of information displayed can sometimes be daunting, because some components can display hundreds of configuration values. When viewing data in the administrative console, the data can be filtered using the standard console filtering buttons, and configuration data can also be dumped to a text file, which you may find easier to browse.

Configuration data example

Node	Server	Name	Value	Description
hostaNode01	server1	startup-customProperties	Null	WebContainer global name and value pairs for configuring non-default behavior inside of WebSphere.
hostaNode01	server1	startup-defaultVirtualHostName	default_host	The default virtual hostname for the webcontainer.
hostaNode01	server1	startup-servletCachingEnabled	false	A boolean value containing true if servlet caching is enabled on the webcontainer, false otherwise.
hostaNode01	server1	startup-vhosts-default_host-aliases	*:9080;*:80;*:9443;*:5060;*:5061;*:443;	The set of aliases associated with this virtual host.

This example shows a few of the many configuration values dumped by the Web container Diagnostic Provider, including the host aliases defined for use by the “default_host” virtual host, and the fact that Servlet caching was not enabled at server startup time. If the current values for these items are different than the startup values, those may also be found in this configuration data dump. The “Save” button is used to save the configuration data to a file in the “logs” directory.

State data

- Displays data that reflects the current state of the component being queried, for example:
 - ▶ Servlets currently cached in the Web container
 - ▶ Number of available connections in a connection pool
 - ▶ Number of threads waiting for an available connection
- By default, data that is readily available to the component at runtime is displayed
 - ▶ Additional state data can be gathered for troubleshooting by changing the state collection specification



A state data dump displays data about the current state of the component being queried, such as the Servlets that are currently cached in the Web container, or the number of threads waiting for an available connection from a particular JDBC™ connection pool. By default, only state data that is readily available as part of the component's normal operation is displayed, meaning that there is no overhead involved with providing this information. Components can be configured to track additional state information by changing the “state collection specification”.

State collection specifications

- The state collection specification dictates if components should record additional state data during runtime
 - ▶ Some performance impact
 - ▶ Can be changed on the fly using the “runtime” tab
- Specification format:
 - ▶ *provider name:data name=level*
 - ▶ Provider name and data name are regular expressions
 - ▶ Level is 1 (enabled) or 0 (disabled)
 - ▶ Multiple specifications separated by semicolon (;)



A state collection specification is a string that specifies whether or not components should keep track of additional state data during runtime that would not ordinarily be available as a part of the component's normal operations. This data collection does have some performance impact. It can be enabled on-the-fly using the “runtime” tab in the administrative console, so you can enable it only when you need it, without requiring a restart. The string is specified in the format shown here; you list the name of a Diagnostic Provider, the name of a particular value that can be collected by that provider, and a collection level, which is either “1” or “0”. Both the provider name and the data name are processed as regular expressions, so you can use patterns or wildcards. You can provide multiple specifications with a single string by separating them with semicolons.

Specification examples

- .*:.*=0
 - ▶ Do not collect any state data that is not readily available
- WebContainerDP:.*=1
 - ▶ Collect all state data that can be collected for WebContainerDP
- ConnMgrDP_MyDS:totalNumberConnections=1
 - ▶ Keep a count of the total number of connections for the Data Source named MyDS
- AdvisorDP:.*=1;WebContainerDP:.*=1
 - ▶ Collect all state data for AdvisorDP and WebContainerDP



Some sample state collection specification strings are shown here. The default specification is “dot-star, colon, dot-star equals zero” which disables all extra data collection, meaning data that is not readily available will not be collected. The second example uses a wildcard to specify that all data that can be collected for the WebContainerDP should be collected. The third example explicitly states that the total number of connections for a particular data source should be collected, but no other values. Names of specific data can be browsed and selected in the administrative console. The last example shows how you can combine multiple collection specifications using a semicolon.

Demonstration

- Demonstration of Diagnostic Providers in the Administrative Console:



To view a demonstration of how to use Diagnostic Providers in the administrative console, click the “pause” button on this presentation, and then click the “show me” link shown here.

Section

Summary and reference



This section will present summary and reference information.

Summary

- Diagnostic Providers
 - ▶ New framework that enables WebSphere Application Server components to provide more useful diagnostic data
 - ▶ State data, configuration data, and self-tests



In summary, the Diagnostic Provider framework is a new framework in WebSphere Application Server V6.1 that enables individual components to provide useful diagnostic information upon request. Components can provide state data, configuration data, and can run self-diagnostic test routines. At this point, only a few major components implement Diagnostic Providers.

Resources

- The information center contains much more detail on Diagnostic Providers
 - ▶ Diagnostic Providers main:
 - /help/topic/com.ibm.websphere.base.doc/info/aes/ae/ctrb_diagnosticprov.html (or search for “Diagnostic Providers”)
 - ▶ Using the Diagnostic Service interface from wsadmin:
 - /help/topic/com.ibm.websphere.base.doc/info/aes/ae/ctrb_dpdisermbean.html (or search for “Diagnostic Service”)



You can find some useful information about the Diagnostic Provider framework in the WebSphere Application Server information center.

Trademarks, Copyrights, and Disclaimers

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

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
eflago/business	DB2	iSeries	OS/400	xSeries
ALX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

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

Linux is a registered trademark of Linus Torvalds.

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

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

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2005,2006. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

