



M6/WS/PMI 600.003.1

AutoPilot® M6 Plug-in for WebSphere Application Server (PMI)

Installation and User's Guide

Version 6.0.2

CONFIDENTIALITY STATEMENT: THE INFORMATION WITHIN THIS MEDIA IS PROPRIETARY IN NATURE AND IS THE SOLE PROPERTY OF NASTEL TECHNOLOGIES, INC. ALL PRODUCTS AND INFORMATION DEVELOPED BY NASTEL ARE INTENDED FOR LIMITED DISTRIBUTION TO AUTHORIZED NASTEL EMPLOYEES, LICENSED CLIENTS, AND AUTHORIZED USERS. THIS INFORMATION (INCLUDING SOFTWARE, ELECTRONIC AND PRINTED MEDIA) IS NOT TO BE COPIED OR DISTRIBUTED IN ANY FORM WITHOUT THE EXPRESSED WRITTEN PERMISSION FROM NASTEL TECHNOLOGIES, INC.

PUBLISHED BY:

RESEARCH AND DEVELOPMENT DEPARTMENT
NASTEL TECHNOLOGIES, INC.
48 SOUTH SERVICE ROAD, SUITE 205
MELVILLE, NY 11747
COPYRIGHT © 2001–2022

ALL RIGHTS RESERVED. NO PART OF THE CONTENTS OF THIS DOCUMENT MAY BE PRODUCED OR TRANSMITTED IN ANY FORM, OR BY ANY MEANS WITHOUT THE WRITTEN PERMISSION OF NASTEL TECHNOLOGIES.

DOCUMENT TITLE: **AUTOPILOT M6 PLUG-IN FOR WEBSPHERE APPLICATION SERVER (PMI) INSTALLATION AND USER'S GUIDE**

DOCUMENT RELEASE DATE: **JUNE 2022**

NASTEL DOCUMENT NUMBER: **M6/WS/PMI 600.003.1**

CONFIDENTIALITY STATEMENT: THE INFORMATION WITHIN THIS MEDIA IS PROPRIETARY IN NATURE AND IS THE SOLE PROPERTY OF NASTEL TECHNOLOGIES, INC. ALL PRODUCTS AND INFORMATION DEVELOPED BY NASTEL ARE INTENDED FOR LIMITED DISTRIBUTION TO AUTHORIZED NASTEL EMPLOYEES, LICENSED CLIENTS, AND AUTHORIZED USERS. THIS INFORMATION (INCLUDING SOFTWARE, ELECTRONIC AND PRINTED MEDIA) IS NOT TO BE COPIED OR DISTRIBUTED IN ANY FORM WITHOUT THE EXPRESSED WRITTEN PERMISSION FROM NASTEL TECHNOLOGIES, INC.

ACKNOWLEDGEMENTS:

THE FOLLOWING TERMS ARE TRADEMARKS OF NASTEL TECHNOLOGIES CORPORATION IN THE UNITED STATES OR OTHER COUNTRIES OR BOTH: NASTEL AUTOPILOT, AUTOPILOT M6, M6 WEB SERVER, M6 WEB CONSOLE, M6 FOR WMQ, MQCONTROL.

THE FOLLOWING TERMS ARE TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES OR OTHER COUNTRIES OR BOTH: IBM, MQ, WEBSPHERE MQ, WIN-OS/2, AS/400, OS/2, DB2, INFORMIX, AIX, AND Z/OS.

JAVA, J2EE, AND THE JAVA LOGOS ARE TRADEMARKS OF SUN MICROSYSTEMS INC. IN THE UNITED STATES OR OTHER COUNTRIES, OR BOTH.

INSTALLANYWHERE IS A REGISTERED TRADEMARK OF ZEROG SOFTWARE IN THE UNITED STATES OR OTHER COUNTRIES, OR BOTH.

THIS PRODUCT INCLUDES SOFTWARE DEVELOPED BY THE APACHE SOFTWARE FOUNDATION ([HTTP://WWW.APACHE.ORG/](http://www.apache.org/)); INCLUDING DERBY DATABASE SERVER. THE "JAKARTA PROJECT" AND "TOMCAT" AND THE ASSOCIATED LOGOS ARE REGISTERED TRADEMARKS OF THE APACHE SOFTWARE FOUNDATION.

INTEL, PENTIUM AND INTEL486 ARE TRADEMARKS OR REGISTERED TRADEMARKS OF INTEL CORPORATION IN THE UNITED STATES, OR OTHER COUNTRIES, OR BOTH.

MICROSOFT, WINDOWS, WINDOWS NT, WINDOWS XP, THE WINDOWS LOGOS, MS SQL SERVER, AND VISUAL SOURCESAFE ARE REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION.

UNIX IS A REGISTERED TRADEMARK IN THE UNITED STATES AND OTHER COUNTRIES LICENSED EXCLUSIVELY THROUGH X/OPEN COMPANY LIMITED.

MAC, MAC OS, AND MACINTOSH ARE TRADEMARKS OF APPLE COMPUTER, INC., REGISTERED IN THE U.S. AND OTHER COUNTRIES.

"LINUX" AND THE LINUX LOGOS ARE REGISTERED TRADEMARKS OF LINUS TORVALDS, THE ORIGINAL AUTHOR OF THE LINUX KERNEL. ALL OTHER TITLES, APPLICATIONS, PRODUCTS, AND SO FORTH ARE COPYRIGHTED AND/OR TRADEMARKED BY THEIR RESPECTIVE AUTHORS.

JASPERREPORTS IS A PRODUCT OF JASPERSOFT CORPORATION.

ORACLE IS A REGISTERED TRADEMARK OF ORACLE CORPORATION AND/OR ITS AFFILIATES.

SYBASE IS A TRADEMARK OF SYBASE, INC. ® INDICATES REGISTRATION IN THE UNITED STATES OF AMERICA.

MYSQL IS A REGISTERED TRADEMARK OF MYSQL AB IN THE UNITED STATES, THE EUROPEAN UNION AND OTHER COUNTRIES.

OPENREPORTS IS DISTRIBUTED UNDER THE GNU GENERAL PUBLIC LICENSE.

OTHER COMPANY, PRODUCT, AND SERVICE NAMES, MAY BE TRADEMARKS OR SERVICE MARKS OF OTHERS.

Table of Contents

CHAPTER 1: INTRODUCTION	1
1.1 HOW THIS GUIDE IS ORGANIZED	1
1.2 HISTORY OF THIS DOCUMENT	1
1.2.1 <i>User Feedback</i>	1
1.3 RELATED DOCUMENTS	1
1.4 RELEASE NOTES	2
1.5 INTENDED AUDIENCE	2
1.6 SYSTEM REQUIREMENTS.....	2
1.6.1 <i>Platforms</i>	2
1.6.2 <i>WebSphere Application Servers</i>	2
1.6.3 <i>Other Requirements</i>	2
1.7 TERMS AND ABBREVIATIONS.....	2
1.8 TECHNICAL SUPPORT.....	2
1.9 CONVENTIONS	3
CHAPTER 2: ABOUT WEBSHERE APPLICATION SERVER.....	5
2.1 INTRODUCTION	5
2.2 WAS PMI SUPPORT FOR MONITORING.....	5
2.2.1 <i>WAS Features</i>	5
CHAPTER 3: INSTALLATION.....	6
3.1 INSTALLATION PREPARATION.....	6
3.1.1 <i>Installation Materials</i>	6
3.1.2 <i>Licensing Information</i>	6
3.2 INSTALLATION	6
3.2.1 <i>Before Installing WAS</i>	6
3.2.2 <i>Installing WAS PMI Expert from Nastel Resource Center</i>	6
CHAPTER 4: USING WEBSHERE APPLICATION SERVER.....	7
4.1 WAS PMI EXPERTS.....	7
4.1.1 <i>Deploying WAS PMI Experts</i>	7
4.1.2 <i>Configuring WAS PMI Experts</i>	8
4.2 GLOBAL SECURITY	14
CHAPTER 5: WEBSHERE APPLICATION SERVER PMI CONFIGURATION METRICS	15
5.1 FACTS PUBLISHED BY WAS PMI EXPERT	15
5.1.1 <i>Enterprise Beans</i>	16
5.1.2 <i>JDBC Connection Pool</i>	18
5.1.3 <i>J2C Connection Pool (WAS Version 5 only)</i>	19
5.1.4 <i>Java Virtual Machine</i>	20
5.1.5 <i>Object Request Broker</i>	20
5.1.6 <i>Servlet Session</i>	21
5.1.7 <i>Transaction</i>	22
5.1.8 <i>ThreadPool</i>	23
5.1.9 <i>Web Application</i>	23
5.1.10 <i>System</i>	24
5.1.11 <i>Dynamic Cache</i>	24
5.1.12 <i>High Availability Manager</i>	25
5.1.13 <i>Object Pools</i>	25
APPENDIX A: REFERENCES	27
A.1 NASTEL DOCUMENTATION	27
A.2 IBM DOCUMENTATION.....	27
APPENDIX B: CONVENTIONS.....	29
B.1 TYPOGRAPHICAL CONVENTIONS.....	29
GLOSSARY	31

INDEX	35
-------------	----

Figures

FIGURE 4-1. DEPLOYMENT TOOL AND NETWORKED NODES.....	7
FIGURE 4-2. DEPLOYING WEBSPHERE EXPERTS.....	7
FIGURE 4-3. WAS PMI EXPERT GENERAL CONFIGURATION.....	8
FIGURE 4-4. WAS PMI EXPERT ABOUT CONFIGURATION.....	9
FIGURE 4-5. WAS PMI EXPERT DEPENDENCIES CONFIGURATION.....	9
FIGURE 4-6. WAS PMI EXPERT FACT OPTIONS CONFIGURATION.....	10
FIGURE 4-7. WAS PMI EXPERT LOGGING CONFIGURATION.....	11
FIGURE 4-8. WAS PMI EXPERT RESTART-RECOVERY CONFIGURATION.....	12
FIGURE 4-9. WAS PMI EXPERT SECURITY CONFIGURATION.....	12
FIGURE 4-10. WAS PMI EXPERT WEBSPHERE SERVER CONFIGURATION.....	13

Tables

TABLE 1-1. DOCUMENT HISTORY.....	1
TABLE 4-1. WAS PMI EXPERT PROPERTIES: GENERAL.....	8
TABLE 4-2. WAS PMI EXPERT PROPERTIES: ABOUT.....	9
TABLE 4-3. WAS PMI EXPERT PROPERTIES: DEPENDENCIES.....	9
TABLE 4-4. WAS PMI EXPERT PROPERTIES: FACT OPTIONS.....	10
TABLE 4-5. WAS PMI EXPERT PROPERTIES: LOGGING.....	11
TABLE 4-6. WAS PMI EXPERT PROPERTIES: RESTART-RECOVERY.....	12
TABLE 4-7. WAS PMI EXPERT PROPERTIES: SECURITY.....	12
TABLE 4-8. WAS PMI EXPERT PROPERTIES: WEBSPHERE SERVER.....	13
TABLE 5-1. STATISTIC TYPES.....	15
TABLE 5-2. WEBSPHERE FACT STATUS.....	15
TABLE 5-3. ENTERPRISE BEANS.....	16
TABLE 5-4. JDBC CONNECTION POOL.....	18
TABLE 5-5. J2C CONNECTION POOL.....	19
TABLE 5-6. JAVA VIRTUAL MACHINE RUNTIME.....	20
TABLE 5-7. OBJECT REQUEST BROKER PERFORMANCE.....	20
TABLE 5-8. SERVLET SESSION MANAGER.....	21
TABLE 5-9. TRANSACTION MANAGER.....	22
TABLE 5-10. THREADPOOL.....	23
TABLE 5-11. WEB APPLICATIONS.....	23
TABLE 5-12. SYSTEM.....	24
TABLE 5-13. DYNAMIC CACHE.....	24
TABLE 5-14. HIGH AVAILABILITY MANAGER.....	25
TABLE 5-15. OBJECT POOLS.....	25
TABLE A-1. NASTEL DOCUMENTATION.....	27
TABLE B-1. TYPOGRAPHICAL CONVENTIONS.....	29

Chapter 1: Introduction

Welcome to the AutoPilot M6 for WebSphere Application Server PMI Installation and User's Guide. This guide explains how to update and perform new installations. It is designed to work with AutoPilot M6, its components and other plug-ins, and run simultaneously without interference or performance degradation. It will hereinafter be referred to as M6 for WAS PMI or WAS PMI. Please review this guide carefully before installing the product.

1.1 How This Guide is Organized

- [Chapter 1:](#)** Identifies the users and history of the document. System requirements are outlined. All other system and platform information is listed in the AutoPilot M6 Installation and User's Guides.
- [Chapter 2:](#)** Contains a brief description of WAS PMI along with listings of features and facts published.
- [Chapter 3:](#)** Provides instructions for new installations of the WAS PMI.
- [Chapter 4:](#)** Provides instruction for using the WAS PMI Expert.
- [Chapter 5:](#)** Describes the WebSphere Performance Monitoring Infrastructure metrics collected by the WAS expert. (PMI)
- [Appendix A:](#)** Provides a detailed list of all reference information required for the installation of AutoPilot.
- [Appendix B:](#)** Contains conventions used in AutoPilot M6 and documents typographical conventions.
- [Glossary:](#)** Contains a listing of unique and common acronyms and words and their definitions.
- [Index:](#)** Contains alphanumeric cross-reference of all topics and subjects of importance within this guide.

1.2 History of This Document

Release Date	Document Number	For AutoPilot Version	Summary
March 2006	AP/WS 400.001	AP/IT 4.0 or higher	Initial document release.
September 2008	M6/WS/PMI 600.002	M6	Update for M6 and WAS 6.1.
March 2011	M6/WS/PMI 600.003	M6	Errata
June 2022	M6/WS/PMI 600.003.1	M6	Changed title to <i>AutoPilot® M6 Plug-in for WebSphere Application Server (PMI) Installation and User's Guide</i>

1.2.1 User Feedback

Nastel encourages all Users and Administrators of AutoPilot to submit comments, suggestions, corrections and recommendations for improvement for all AutoPilot documentation. Please send your comments via Post/Mail, or by e-mail. Send messages to: support@nastel.com. You will receive a written response, along with status of any proposed change, update, or correction.

1.3 Related Documents

The complete listing of related and referenced documents is listed in [Appendix A](#) of this guide.

1.4 Release Notes

See INSTALL.TXT and README.TXT files on installation media. These files should be accessible wherever AutoPilot M6 is installed. (For example: [C:\nastel\AutoPilotM6].) README.TXT is also available on the Nastel Resource Center.

1.5 Intended Audience

WAS PMI Installation and User's Guide is intended for use by installers and administrators of Nastel's AutoPilot and AutoPilot WebSphere. There are three user groups defined for installation and use.

- **Installer:** Should be familiar with Java Run Time Environment 1.5.1 (JRE 1.5.1) or higher (included in AutoPilot M6 for Windows, Solaris, AIX, HP-UX and Linux). Procedures for installing software on the target platform such as Windows and/or UNIX. Basic understanding of TCP/IP.
- **Administrator:** Should have a working knowledge of middleware, TCP/IP, and system management. The administrator should also have an understanding of Java Runtime Environment (JRE), TCP/IP and installation procedures for the platform where AutoPilot is installed (for example, Windows, NT, UNIX, etc.)
- **User:** Requires only local operating system knowledge and basic knowledge of AutoPilot M6.

1.6 System Requirements

This section defines system and platform prerequisite support requirements for WAS.

1.6.1 Platforms

WAS is compatible with the following platforms:

- Windows NT/2000/XP
- Unix (Solaris, AIX, HP-UX, Linux)

1.6.2 WebSphere Application Servers

WAS is compatible with the following version of WebSphere:

- V7.x
- V6.x

1.6.3 Other Requirements

WAS requires the following conditions:

- Performance Monitoring of the WebSphere Server is enabled, and set to maximum.
- AutoPilot CEP server is running on the same machine as the WebSphere server and no other application servers or plug-ins are installed on this node.
- AutoPilot M6 or higher.
- Java Run Time Environment 1.5.1 (JRE 1.5.1) or higher (included in AutoPilot M6)
- Installer may need administrative privileges for the target platform.

1.7 Terms and Abbreviations

A list of Terms and Abbreviations used in this document is located in the [Glossary](#).

1.8 Technical Support

If you need additional technical support, you can contact Nastel Technologies by telephone or by e-mail. To contact Nastel technical support by telephone, call **(800) 963-9822 ext. 1**, if you are calling from outside the United States dial **001-631-761-9190**. To contact Nastel technical support by e-mail, send a message to <mailto:support@nastel.com>. To access the Nastel automated support system (user id and

password required), go to <http://support.nastel.com/>. Contact your local AutoPilot Administrator for further information.

1.9 Conventions

Refer to [Appendix B](#) for conventions used in this guide.

This page intentionally left blank.

Chapter 2: About WebSphere Application Server

2.1 Introduction

AP M6 for WebSphere Application Server (WAS) was designed to enable AutoPilot M6 to monitor and manage *e*Business applications for continuous operations in addition to its standard features.

2.2 WAS PMI Support for Monitoring

WAS PMI provides monitoring for the following categories:

- Enterprise Beans
- JDBC Connection Pool
- J2C Connection Pool
- Java Virtual Machine
- Object Request Broker (ORB)
- Servlet Session Manager
- Transaction Manager
- Thread Pool
- Web Applications
- System
- Dynamic Cache
- High Availability Manager
- Object Pool

2.1.1 WAS Features

WAS provides the following management features:

- Monitors and controls status of WebSphere services and processes such as HTTP DAEMON, DB/2 and Servlet Engines, memory usage, CPU usage, I/O rates, and thread counts.
- Monitors and manages Bean Data, Bean Object Pools, and Bean Method performance.
- Monitors and manages the performance of Servlet Engine and Servlet sessions.
- Monitors WebSphere runtime performance including ORB, Thread Pool, Database Connection Manager, Connections Pool Usage, Memory Usage and Transaction Manager.
- Automatic fault recovery.
- Create custom monitoring.
- Collect system level performance information, such as CPU, I/O, and memory.
- Collect historical data for subsequent analysis/capacity planning.
- Instantly manage-enable any Java-Bean application.
- Organize management information into views, customized for specific users.

Chapter 3: Installation

3.1 Installation Preparation

This section contains general information related to preparing for and installing the installation of WAS software.

3.1.1 Installation Materials

Installation can be performed by download through the Nastel Resource Center, installation media or FTP. The media installation can be automatically initiated, then continued using the installation wizard, or manually installed. The installation media contains all required AutoPilot M6 components for Java 2 platforms.

3.1.1.1 Technical Documents

Prior to installation, review all text files and installation procedures provided on the Nastel Resource Center. It is recommended that all installation related materials are printed to allow the installer to review prior to installation, and better follow the detailed instructions within.

3.1.2 Licensing Information

A copy of the standard Licensing Agreement is imbedded in the installation software. The formal licensing agreement has been furnished in the purchase agreement package.

3.2 Installation

This section provides instructions for installing WAS on the compatible platforms. Review all installation related materials prior to commencing installation procedures. Reviewing materials will allow installers to pre-determine installation options and familiarize themselves with associated requirements.

3.2.1 Before Installing WAS

This procedure outlines the typical procedures for installing M6 for WAS PMI on all supported platforms.

1. WAS 5.x and 6.x.
2. Ensure Performance Monitoring for WebSphere server is enabled and set to Maximum.
3. AutoPilot CEP Server is running on same machine as WAS.

3.2.2 Installing WAS PMI Expert from Nastel Resource Center

1. Download M6 for WAS PMI from the Nastel Resource Center.
2. Save your work and logoff AutoPilot and WebSphere Application Server.

	NOTE:	There are no specific logoff procedures required to exit AutoPilot Console.
---	--------------	---

3. Stop Nodes and/or Domain Servers that will be updated as specified in the *AutoPilot M6 User's Guide*.
4. Copy AP_PMI_WAS-6.0.2.pkg into [AUTOPILOT_HOME]\updates directory.
5. At command prompt run:


```
[AUTOPILOT_HOME]\bin\pkgman ..\updates\AP_PMI_WAS-6.0.2.pkg
```
6. Enter path of WAS. For example: **C:\Program Files\WebSphere\AppServer**.
7. Verify installation: [AUTOPILOT_HOME]\bin\pkgman -libinfo. The details of the library are listed.

	NOTE:	Make sure there are no errors posted at the bottom of the screen.
---	--------------	---

Chapter 4: Using WebSphere Application Server

4.1 WAS PMI Experts

All experts are agents knowledgeable about the subject they are assigned to monitor. Experts are mobile and act as connectors to other AutoPilot and optional plug-in experts. WAS PMI experts are contained in WebSphere Application Server's target managed resources (Nodes) and collect system's status information, which are analyzed by assigned policies, then filtered by AutoPilot managers to be viewed in a Business View viewer. Experts publish facts. Facts in AutoPilot are basic true statements about current run-time values, indicating an object's state in a managed resource.

WAS experts maintain the same strengths and qualities as all AutoPilot experts, and allow the user to deploy them in a WebSphere environment. WAS is supplied with a generic expert. The set-up and configuration of the supplied Expert is addressed in this section. Refer to the *AutoPilot M6 User's Guide* for additional information on expert deployment and configuration.

4.1.1 Deploying WAS PMI Experts

1. Click **Tools** from the toolbar and then **Deployment Tool** to display *Directory Viewer*.



Figure 4-1. Deployment Tool and Networked Nodes

2. Right-click CEP server to display *Node Menu*. Click **Deploy Expert**. Expert menu is displayed.
3. Click **WebSphere >> WebSphere PMI Monitor**.

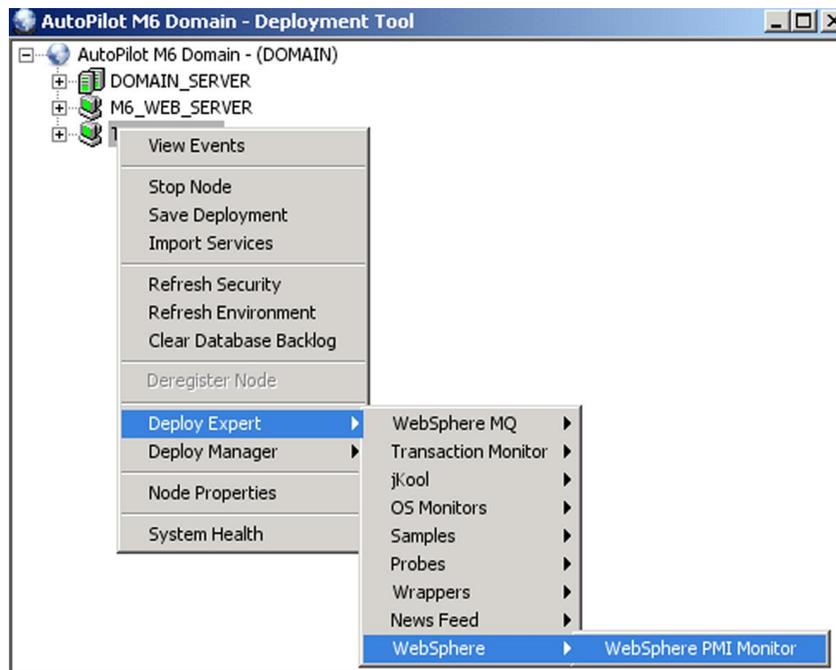


Figure 4-2. Deploying WebSphere Experts

4. Review the WAS PMI Expert configuration in the screens displayed. Refer to Configuring [WAS PMI Experts](#) instructions for further detail.
5. Click **Apply**. The WAS PMI Expert will be deployed only on CEP server selected in step 2 above.

4.1.2 Configuring WAS PMI Experts

This paragraph provides detailed instructions and information for configuring WAS PMI Expert. WAS is supplied with a typical expert, the set-up and configuration of this supplied expert will be addressed. Other Software/Platform specific and standard experts are addressed in supporting documentation for that software/platform and the *AutoPilot User’s Guide*. Only WAS PMI Expert is discussed in this Guide.

This expert consists of eight screens of user definable configuration data. The following procedure explains the entries for each category of the WAS PMI Expert.

1. **General:** This screen is common to all Experts. Review (if updating existing Expert) or configure data elements as follows:

Table 4-1. WAS PMI Expert Properties: General	
Element	Description
Brief description	Short description of the service. This name will appear in brackets (example: [Process Monitor]) on screen with listed Node.
Content	User defined category that will be registered in the Domain Server. Context is displayed as folder icon under each CEP Server. The default name is WebSphere Experts.
Name	Name that uniquely identifies the service in the Domain Server. Enter or modify the Service Name as required, or in accordance with local guidelines. Variations of names are used when deploying services on multiple Nodes. No spaces or blanks are allowed in Service Name formats. The default name is “Service_”, plus a 12 digit random number (example: Service_123456789012).

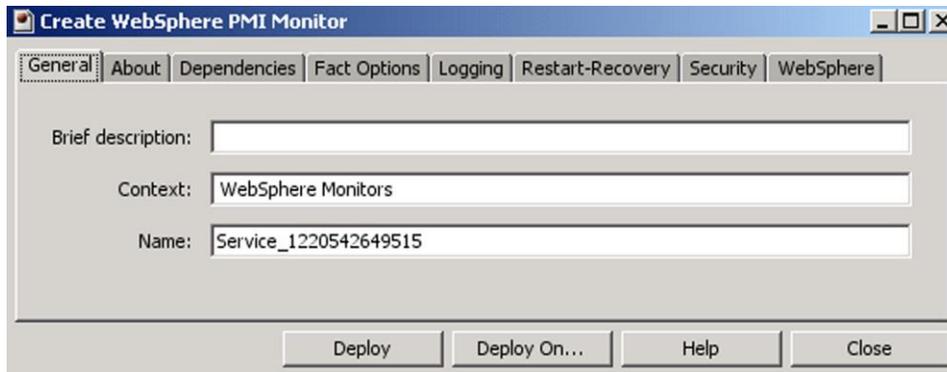


Figure 4-3. WAS PMI Expert General Configuration

2. **About:** This screen is common to all Experts and cannot be changed.

Table 4-2. WAS PMI Expert Properties: About	
Element	Description
Package Title	Implementation title of source package.
Package vendor	Name of implementation vendor.
Package version	Package version as assigned by the vendor.

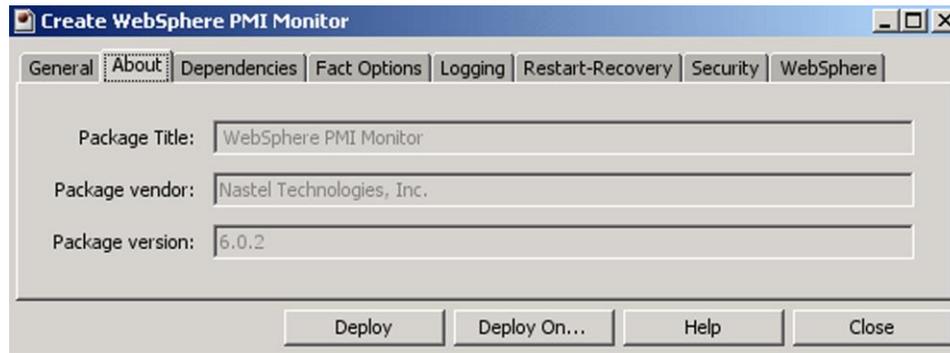


Figure 4-4. WAS PMI Expert About Configuration

3. **Dependencies:** (Reserved) Leave blank unless required. Dependencies are comma delimited lists of services within a CEP Server. The lists must be available to the specified service prior to loading. The sequence of service loading is determined by the list.

Table 4-3. WAS PMI Expert Properties: Dependencies	
Element	Description
Platform dependencies	Comma separated list of operating system platforms this expert is dependant on.
Service dependencies	Comma separated list of services this expert is dependant on.

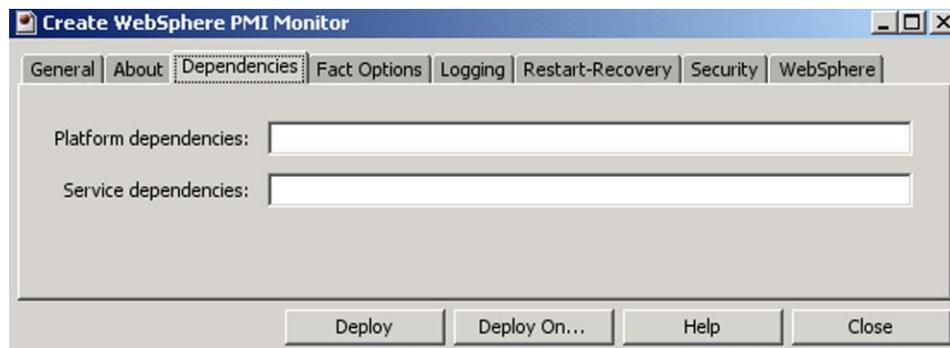


Figure 4-5. WAS PMI Expert Dependencies Configuration

4. **Fact Options:** This screen is common to all Experts. Review (if updating existing Expert) or configure data elements as follows:

Table 4-4. WAS PMI Expert Properties: Fact Options	
Element	Description
Exclude Fact Filters	Comma separated list of fact paths to exclude during publishing
Expire facts(ms)	Automatically expires facts that have not been updated in the specified time (ms).
Fact History Size	Automatically maintains specified number of samples for each published fact in memory.
Fact History Time (ms)	Automatically maintains fact history not exceeding specified time (ms).
Include Fact Filters	Comma separated list of fact paths to include during publishing.

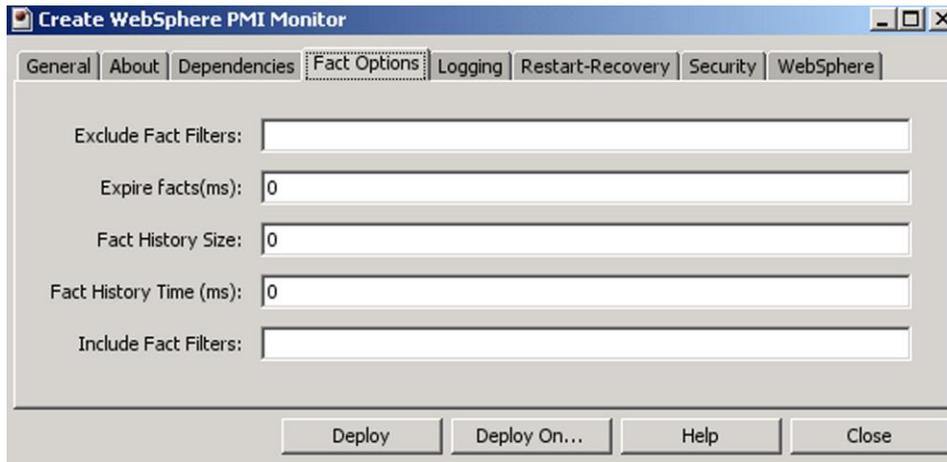


Figure 4-6. WAS PMI Expert Fact Options Configuration

5. **Logging:** This screen is common to all Experts. Review (if updating existing Expert) or configure data elements as follows:

Table 4-5. WAS PMI Expert Properties: Logging	
Element	Description
Audit	Enable/Disable service audit trace. Default is disabled.
Log name	Log name associated with the service. The default name is Services, but may be changed as required. (Example: Task_Progress_Process_Log).
Log service activity	Enable/Disable service activity trace. Default is disabled.
Log size (bytes)	Log size in bytes. Real log size is the maximum value of the server.log.size and logsize. Default is 200000.
Record Facts	Enable/Disable fact recording for this service. Default is enabled. CEP Server records all facts produced by this service into an .fct file which can be replayed back using the apfact utility. Recording occurs only when CEP Server is started with -logfacts option or the environmental variable property server.service.fact.logging=true is defined in global.properties or node.properties. Default is enabled.

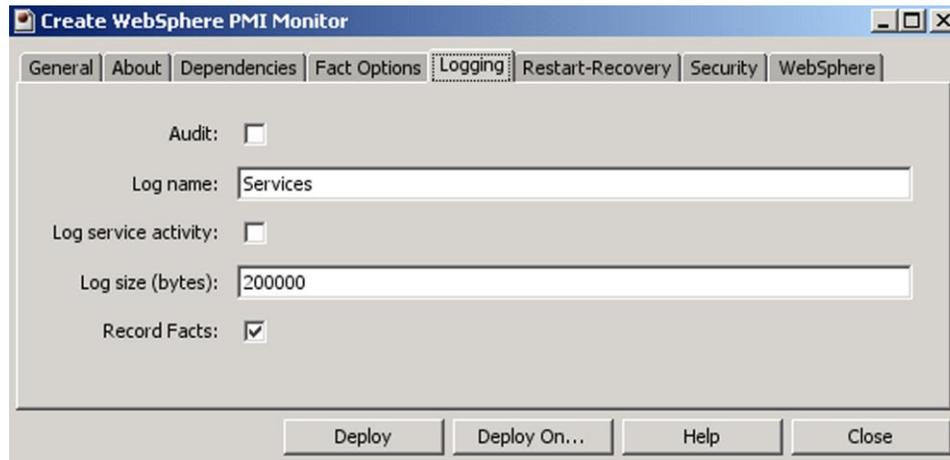


Figure 4-7. WAS PMI Expert Logging Configuration

6. **Restart/Recovery:** This screen is common to all Experts. Review (if updating existing Expert) or configure data elements as follows:

Table 4-6. WAS PMI Expert Properties: Restart-Recovery	
Element	Description
Automatic start	Enable/disable automatic start. Default is enabled.
Save in registry	Persistent services are saved in Registry .xml file. Default is enabled.
Synchronous Control	Enable/Disable synchronous service initiation. Default is enabled.

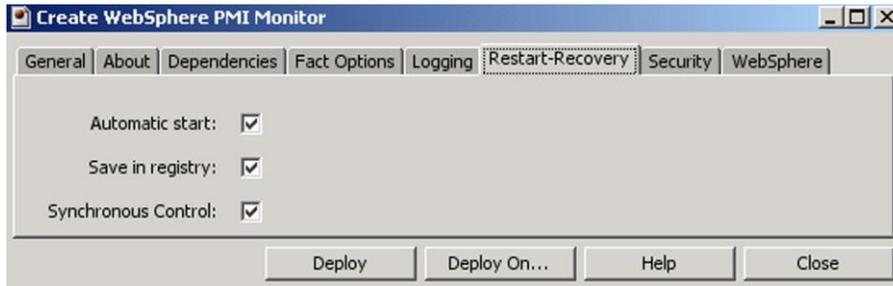


Figure 4-8. WAS PMI Expert Restart-Recovery Configuration

7. **Security:** This screen is common to all Experts. Review (if updating existing Expert) or configure data elements as follows:

Table 4-7. WAS PMI Expert Properties: Security		
Property	Description	
Inherit permissions from owner	Enable/disable inherit permission from owner’s permission masks. Default is enabled.	
Owner	User that owns the object.	
Permissions	Permissions for users in same group and users in other groups. Enable/disable as required.	
	Group:	Other:
Read	Group members may read/view attributes of an object.	Other users may read/view attributes of an object.
Change	Group members may change the attributes of an object.	Other users may change the attributes of an object.
Delete	Group members may delete the object.	Other users may delete the object.
Control	Group members may execute control actions such as start, stop, and disable.	Other users may execute control actions such as start, stop, and disable.
Execute	Group members may execute operational commands on the object.	Other users may execute operational commands on the object.

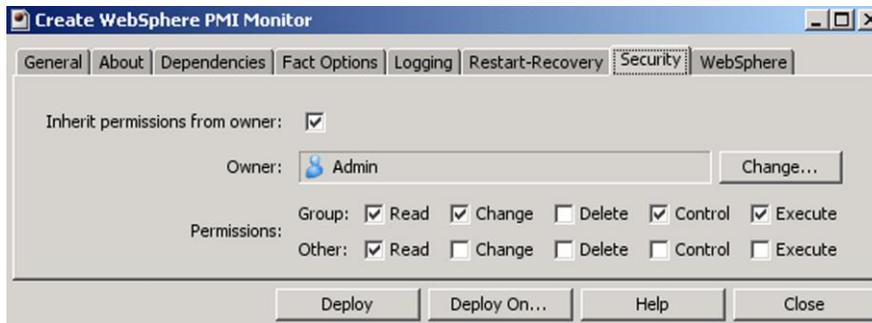


Figure 4-9. WAS PMI Expert Security Configuration

8. **WebSphere Server:** Review (if updating existing Expert) or configure data elements as follows:

Table 4-8. WAS PMI Expert Properties: WebSphere Server	
Element	Description
Connection method	The type of connection to use (SOAP or RMI). Default is SOAP.
Connection port	The TCP/IP port number to connect to the WebSphere server. Default is 8880.
Sampling interval (sec.)	Number of seconds between each sampling interval. Default is 300.
SAS configuration URL	URL to the SAS configuration file.
SOAP configuration URL	URL to the SOAP configuration file.
SSL configuration URL	URL to the SSL configuration file.
WebSphere home directory	Home directory of the WAS installation.
WebSphere server host name	The name of the machine running the WebSphere server. Default is localhost.
WebSphere server version	The version number of WAS that is being monitored; 5.x/6.x for all versions of WebSphere Application Server including V7.

	NOTE: AutoPilot Experts for WAS will not run if the identified server in WebSphere Server is an IP address. Ensure the hostname is accessible from the node and perform a Ping Test to verify.
---	---

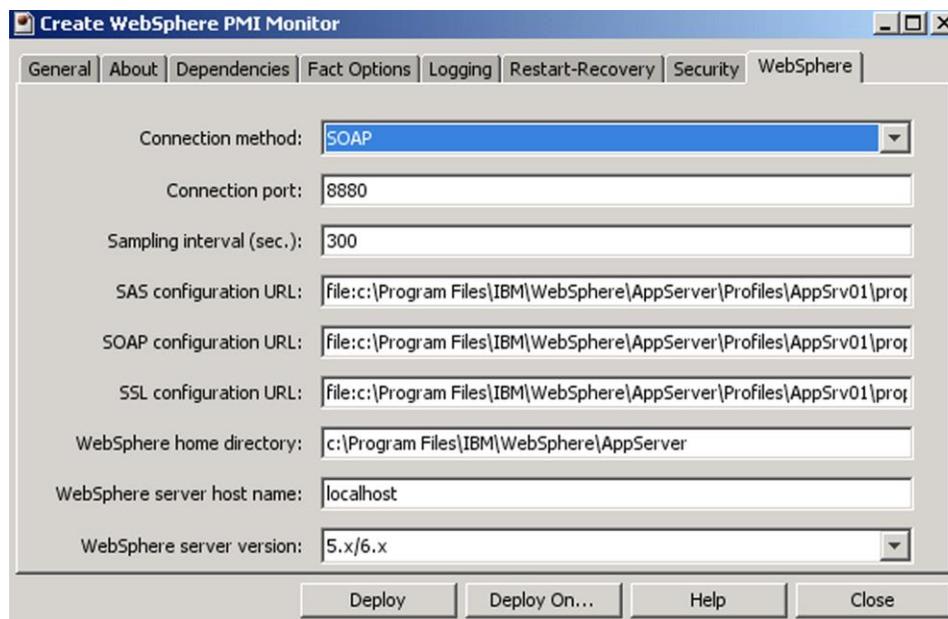


Figure 4-10. WAS PMI Expert WebSphere Server Configuration

4.2 Global Security

If Global Security is turned on for the WAS, the following settings in

[*WAS_HOME*]/properties/sas.client.props must be set:

```
com.ibm.CORBA.securityEnabled=true
com.ibm.CORBA.loginUserid=<user>
com.ibm.CORBA.loginPassword=<password>
```

The following settings in [*WAS_HOME*]/properties/soap.client.props must also be set:

```
com.ibm.SOAP.securityEnabled=true
com.ibm.SOAP.loginUserid=<user>
com.ibm.SOAP.loginPassword=<password>
```

The above passwords may be encoded with

[*WAS_HOME*]/bin/PropFilePasswordEncoder.bat for enhanced security.

Chapter 5: WebSphere Application Server PMI Configuration Metrics

This section describes the WebSphere Performance Monitoring Infrastructure (PMI) metrics collected by the WAS expert. PMI provides server-side data collection and client-side API to retrieve performance data. Performance data has two components: static and dynamic. The static component consists of a name, ID and other descriptive attributes to identify the data. The dynamic component contains information that changes over time, such as the current value of a counter and the time stamp associated with that value.

The PMI data can be one of the following statistic types (these statistic types follow the J2EE 1.4 Performance Data Framework):

Statistic Type	Description	Example
CountStatistic	Represents a running count of a given value.	Total number of connections created.
AverageStatistic	Represents a simple average. Keeps track of total, count, min, and max. The average can be derived by total and count. (This type is WebSphere extension to J2EE Performance Data Framework.)	Average HttpSession size in bytes.
TimeStatistic	Represents a simple average. Keeps track of total, count, min, and max. The average can be derived from total and count. Unit of measure is in milliseconds or seconds. (This type is WebSphere extension to J2EE Performance Data Framework.)	Average session lifetime.
RangeStatistic	Represents a time-weighted average. Keeps track of current, low water mark, high water mark, time-weight total, and integral.	Average percent of the pool that is in use.
BoundedRangeStatistic	Same as RangeStatistic, with lower bound and upper bound.	JDBC average pool size.

5.1 Facts Published By WAS PMI Expert

The WAS PMI Expert publishes the following facts to indicate its own status:

Fact Name	Fact Values	Description
Status	Connecting	Whether the expert is currently connecting to the WebSphere server.
	Connected	Whether the expert is currently connected to the WebSphere server.
	Working	Whether the expert is currently getting information from the WebSphere server.
	Not Connected	Whether the expert is currently unconnected from the WebSphere server.

WAS PMI Expert publishes the following facts for each WebSphere server:

	NOTE:	This set of facts is published for each Data Source.
---	--------------	--

- Enterprise Beans
- JDBC Connection Pool
- J2C Connection Pool
- Java Virtual Machine
- Object Request Broker
- Servlet Session Manager
- Transaction Manager
- Thread Pool
- Web Applications
- System
- Dynamic Cache
- High Availability Manager
- Object Pools

5.1.1 Enterprise Beans

The Enterprise Bean data counters report load values, response times, and life cycle activities for enterprise beans. Examples include the average number of active beans and the number of times bean data is loaded or written to the database. Information is provided for enterprise bean methods and the remote interfaces used by an enterprise bean.

Table 5-3. Enterprise Beans

Name	Type	Description
CreateCount	CountStatistic	Number of times that beans were created.
Remove Count	CountStatistic	Number of times that beans were removed.
PassivateCount	CountStatistic	Number of times that beans were passivated (entity and stateful).
ActivateCount	CountStatistic	Number of times beans were activated (entity and stateful).
LoadCount	Count Statistic	Number of times that bean data were loaded from persistent storage (entity).
StoreCount	CountStatistic	Number of times that bean data were stored in persistent storage (entity).
InstantiateCount	CountStatistic	Number of times that bean objects were instantiated.
FreedCount	CountStatistic	Number of times that bean objects were freed.
ReadyCount	RangeStatistic	Number of concurrently ready beans (entity and session).
LiveCount	RangeStatistic	Number of concurrently live beans.
MethodResponseTime	TimeStatistic	Average response time, in milliseconds, on the bean methods (home, remote, and local).
CreateTime	TimeStatistic	Average time, in milliseconds, that a bean create call takes including the time for the load, if any.
LoadTime	TimeStatistic	Average time, in milliseconds, for loading the bean data from persistent storage.
StoreTime	TimeStatistic	Average time, in milliseconds, for storing the bean data to persistent storage (entity).

Table 5-3. Enterprise Beans

Name	Type	Description
RemoveTime	TimeStatistic	Average time, in milliseconds, that a bean entries call takes including the time at the database, if any.
MethodCallCount	CountStatistic	Total number of method calls.
ActivationTime	TimeStatistic	Average time, in milliseconds, that a beanActivate call takes including the time at the database, if any.
PassivationTime	TimeStatistic	Average time, in milliseconds, that a beanPassivate call takes including the time at the database, if any.
ActiveMethodCount	TimeStatistic	Number of concurrently active methods - the number of methods called at the same time.
RetrieveFromPoolCount	CountStatistic	Number of calls retrieving an object from the pool (entity and stateless).
RetrieveFromPoolSuccess Count	CountStatistic	Number of times that a retrieve found an object available in the pool (entity and stateless).
ReturnsToPoolCount	CountStatistic	Number of calls returning an object to the pool (entity and stateless).
ReturnsDiscardCount	CountStatistic	Number of times that the returning object was discarded because the pool was full (entity and stateless).
DrainsFromPoolCount	CountStatistic	Number of times that the daemon found the pool was idle and attempted to clean it (entity and stateless).
DrainSize	TimeStatistic	Average number of object discarded in each drain (entity and stateless).
PooledCount	RangeStatistic	Number of objects in the pool (entity and stateless).
MessageCount	CountStatistic	Number of messages delivered to the bean onMessage method (message driven beans).
MessageBackoutCount	CountStatistic	Number of messages that failed to be delivered to the bean onMessage method (message driven beans).
WaitTime	TimeStatistic	Average time to obtain a ServerSession from the pool (message driven bean).
ServerSessionPoolUsage	RangeStatistic	Percentage of the server session pool in use (message driven bean).

5.1.2 JDBC Connection Pool

JDBC Connection Pool data counters contain usage information about connection pools for a database. Examples include the average size of the connection pool or number of connections, the average number of threads waiting for a connection, the average wait time in milliseconds for a connection, and the average time the connection is in use.

Table 5-4. JDBC Connection Pool

Name	Type	Description
CreateCount	CountStatistic	Total number of connections created.
PoolSize	BoundedRangeStatistic	Average pool size.
FreePoolSize	BoundedRangeStatistic	Average free pool size.
AllocateCount	CountStatistic	Total number of connections allocated.
ReturnCount	CountStatistic	Total number of connections returned.
WaitingThreadCount	RangeStatistic	Number of threads that are concurrently waiting for a connection.
FaultCount	CountStatistic	Total number of faults, such as timeouts, in connection pool.
WaitTime	TimeStatistic	Average waiting time, in milliseconds, until a connection is granted.
UseTime	TimeStatistic	Average time a connection is used. (Difference between the time at which the connection is allocated and returned. This includes the JDBC operation time.)
PercentUsed	RangeStatistic	Average percent of the pool that is in use.
PercentMaxed	RangeStatistic	Average percent of the time that all connections are in use.
PrepStmtCacheDiscardCount	CountStatistic	Total number of statements discarded by the LRU algorithm of the statement cache.
ManagedConnectionCount	CountStatistic	Number of ManagedConnection objects in use.
JDBCTime	TimeStatistic	Amount of time, in milliseconds, spent executing in the JDBC driver (includes time spent in JDBC driver, network and database.)

5.1.3 J2C Connection Pool (WAS Version 5 only)

The J2C Connection Pool data counters contain usage information about the Java 2 Platform, Enterprise Edition (J2EE) Connector architecture and are used to monitor the J2C connection pool performance.

Name	Type	Description
ManagedConnectionCount	CountStatistic	Number of ManagedConnection objects in use.
ConnectionHandleCount	CountStatistic	Number of connections that are associated with ManagedConnections (physical connections) objects in this pool.
CreateCount	CountStatistic	Total number of managed connections created.
CloseCount	CountStatistic	Total number of managed connections destroyed.
AllocateCount	CountStatistic	Total number of times that a managed connection is allocated to a client. The total is maintained across the pool, not the connection.
FaultCount	CountStatistic	Number of faults, such as timeouts, in connection pool.
FreePoolSize	BoundedRangeStatistic	Number of free connections in the pool.
Pool size	BoundedRangeStatistic	Average number of managed connections in the pool.
WaitingThreadCount	RangeStatistic	Average number of threads concurrently waiting for a connection.
PercentUsed	RangeStatistic	Average percent of pool that is in use. The value is determined by total number of configured connections in ConnectionPool, not current number of connections.
PercentMaxed	RangeStatistic	Average percent of time that all connections are in use.
WaitTime	TimeStatistic	Average waiting time, in milliseconds, until a connection is granted.
UseTime	TimeStatistic	Average time, in milliseconds, that connections are in use.

5.1.4 Java Virtual Machine

The Java Virtual Machine (JVM) data counters contain memory used by a process as reported by JVM run time. Examples are the total memory available and the amount of free memory for the JVM and are used to monitor the JVM performance.

Table 5-6. Java Virtual Machine Runtime

Name	Type	Description
FreeMemory	CountStatistic	Free memory in JVM run time
UsedMemory	CountStatistic	Used memory in JVM run time
UpTime	CountStatistic	Amount of time the JVM is running
HeapSize	BoundedRangeStatistic	Total memory in the JVM run time

5.1.5 Object Request Broker

The Object Request Broker data counters contain information for the ORB. Examples include the object reference lookup time, the total number of requests, and the processing time for each interceptor.

Table 5-7 Object Request Broker Performance

Name	Type	Description
LookupTime	TimeStatistic	Time, in milliseconds, to look up an object reference before method dispatch can be carried out.
RequestCount	CountStatistic	Total number of requests sent to the ORB.
ConcurrentRequestCount	RangeStatistic	Number of requests that are concurrently processed by the ORB.
ProcessingTime	TimeStatistic	Time, in milliseconds, it takes a registered portable interceptor to run.

5.1.6 Servlet Session

The Servlet Session data counters contain usage information for HTTP sessions. Examples include the total number of accessed sessions, the average amount of time it takes for a session to perform a request, and the average number of concurrently active HTTP sessions

Table 5-8. Servlet Session Manager

Name	Type	Description
CreateCount	CountStatistic	Number of sessions created.
InvalidateCount	CountStatistic	Number of sessions invalidated.
LifeTime	TimeStatistic	Average session lifetime in milliseconds (time invalidated - time created).
ActiveCount	RangeStatistic	Number of concurrently active sessions. A session is active if WAS is currently processing a request which uses that session.
LiveCount	RangeStatistic	Number of sessions that are currently cached in memory.
NoRoomForNewSessionCount	CountStatistic	Applies only to session in memory with AllowOverflow=false. The number of times that a request for a new session can not be handled because it would exceed the maximum session count.
CacheDiscardCount	CountStatistic	Number of session objects that have been forced out of the cache. (A Least Recently Used (LRU) algorithm removes old entries to make room for new sessions and cache misses). Applicable only for persistent sessions.
ExternalReadTime	TimeStatistic	Time, in milliseconds, taken in reading the session data from persistent store. For multirow sessions, the metrics are for the attribute; for single row sessions, the metrics are for the whole session. Applicable only for persistent sessions. When using a JMS persistent store, the user has the choice of whether to serialize the data being replicated. If they choose not to serialize the data, the counter will not be available.
ExternalReadSize	TimeStatistic	Size of session data read from persistent store. Applicable only for (serialized) persistent sessions; similar to externalReadTime above.
ExternalWriteTime	TimeStatistic	Time, in milliseconds, taken to write the session data from the persistent store. Applicable only for (serialized) persistent sessions. Similar to externalReadTime above.
ExternalWriteSize	TimeStatistic	Size of session data written to persistent store. Applicable only for (serialized) persistent sessions. Similar to externalReadTime above.
AffinityBreakCount	CountStatistic	Number of requests received for sessions that were last accessed from another Web application. This can indicate failover processing or a corrupt plug-in configuration.
SessionObjectSize	TimeStatistic	Size in bytes of (the serializable attributes) in-memory sessions. Only count session objects that contain at least one serializable attribute object. Note that a session may contain some attributes that are serializable and some that are not. The size in bytes is at a session level.
TimeSinceLastActivated	TimeStatistic	Time difference in milliseconds between previous and current access time stamps. Does not include session time out.
TimeoutInvalidationCount	CountStatistic	Number of sessions that are invalidated by timeout.
ActivateNonExistSessionCount	CountStatistic	Number of requests for a session that no longer exists, presumably because the session timed out. Use this counter to help determine if the timeout is too short.

5.1.7 Transaction

The Transaction data counter contains performance information for the transaction manager. Examples include the average number of active transactions, the average duration of transactions, and the average number of methods per transaction. Definitions are per transaction manager or server.

Table 5-9. Transaction Manager

Name	Type	Description
GlobalBegunCount	CountStatistic	Total number of global transactions started on server.
GlobalInvolvedCount	CountStatistic	Total number of global trans involved on server (for example, begun and imported).
LocalBegunCount	CountStatistic	Total number of local transactions started on server.
ActiveCount	CountStatistic	Number of concurrently active global transactions.
LocalActiveCount	CountStatistic	Number of concurrently active local transactions.
GlobalTranTime	TimeStatistic	Average duration of global transactions.
LocalTranTime	TimeStatistic	Average duration of local transactions.
GlobalbeforeCompletionTime	TimeStatistic	Average duration of before_completion for global transactions.
GlobalCommitTime	TimeStatistic	Average duration of commit for global transactions.
GlobalPrepareTime	TimeStatistic	Average duration of prepare for global transactions.
LocalBeforeCompletionTime	TimeStatistic	Average duration of before_completion for local transactions.
LocalCommitTime	TimeStatistic	Average duration of commit for local transactions.
CommittedCount	CountStatistic	Total number of global transactions committed.
RolledbackCount	CountStatistic	Total number of global transactions rolled back.
OptimizationCount	CountStatistic	Number of global transactions converted to single phase for optimization.
LocalCommittedCount	CountStatistic	Number of local transactions committed.
LocalRolledbackCount	CountStatistic	Number of local transactions rolled back.
GlobalTimeoutCount	CountStatistic	Number of global transactions timed out.
LocalTimeoutCount	CountStatistic	Number of local transactions timed out.

5.1.8 ThreadPool

The ThreadPool data counters contain information about the thread pools for Object Request Broker (ORB) threads and the Web container pools used to process HTTP requests. Examples include the number of threads created and destroyed, the maximum number of pooled threads allowed, and the average number of active threads in the pool.

Name	Type	Description
CreateCount	CountStatistic	Total number of threads created.
DestroyCount	CountStatistic	Total number of threads destroyed.
ActiveCount	RangeStatistic	Number of concurrently active threads.
Pool size	BoundedRangeStatistic	Average number of threads in pool.
Percent maxed	RangeStatistic	Average percent of time that all threads are in use.
DeclaredthreadHungCount	CountStatistic	Number of threads declared hung
ClearedThreadHangCount	CountStatistic	Number of thread hangs cleared
ConcurrentHungThreadCount	RangeStatistic	Number of concurrently hung threads
ActiveTime	TimeStatistic	Average time, in milliseconds, threads are in an active state.

5.1.9 Web Application

The Web Application data counters contain information for the selected server. Examples include the number of loaded servlets, the average response time for completed requests, and the number of requests for the servlet.

Name	Type	Description
LoadedServletCount	CountStatistic	Number of servlets that were loaded.
ReloadCount	CountStatistic	Number of servlets that were reloaded.
RequestCount	CountStatistic	Total number of requests a servlet processed.
ConcurrentRequests	RangeStatistic	Number of requests that are concurrently processed.
ServiceTime	TimeStatistic	Response time, in milliseconds, of a servlet request.
ErrorCount	CountStatistic	Total number of errors in a servlet or Java Server Page (JSP).

5.1.10 System

The System data counters contain information for a machine (node). Examples include the CPU utilization and memory usage. Note that this category is available at node level, which means it is only available at node agent in the multiple servers' version.

Table 5-12. System

Name	Type	Description
CPUUsageSince Last Measurement	CountStatistic	Average system CPU utilization taken over the time interval since the last reading. Because the first call is required to perform initialization, an invalid value such as 0 will be returned. All subsequent calls will return the expected value. On SMP machines, the value returned will be the utilization averaged over all CPUs.
FreeMemory	CountStatistic	Amount of real free memory available on the system. Real memory that is not allocated is only a lower bound on available real memory, since many operating systems take some of the otherwise unallocated memory and use it for additional I/O buffering. The exact amount of buffer memory which can be freed up is dependent on both the platform and the application(s) running on it.
CPUUsageSinceServerStarted	TimeStatistic	Average percentCpuUsage that is busy after the server is started.

5.1.11 Dynamic Cache

The Dynamic Cache data counters contain information for the dynamic cache service. Examples include in-memory cache size, the number of invalidations, and the number of hits and misses.

Table 5-13. Dynamic Cache

Name	Type	Description
maxInMemoryCacheEntryCount	CountStatistic	Maximum number of in-memory cache entries.
InMemoryCacheEntryCount	CountStatistic	Current number of in-memory cache entries.

5.1.12 High Availability Manager

The High Availability Manager maintains statistical data within the entire WAS. Examples include number of bulletin-board subject and subscriptions and number of local groups.

Table 5-14. High Availability Manager

Name	Type	Description
Number of local groups	RangeStatistic	Total number of local groups.
Group state rebuild time	TimeStatistic	Time taken in milliseconds to rebuild the global group state. During the rebuild time, no fail-over can happen. If this time is too high and is unacceptable for the desired availability, you may want to increase the number of coordinators. For proper operation of this counter, you must host the active coordinator in an application server other than the deployment manager.
Number of bulletin-board subjects	RangeStatistic	Total number of bulletin-board subjects managed.
Number of bulletin-board subscriptions	RangeStatistic	Total number of bulletin-board subscriptions.
Bulletin-board rebuild time	TimeStatistic	Time taken in milliseconds to rebuild the global state of the bulletin-board. During this time no messages will be received by the subscribers. If this time is too high, and is unacceptable, you may want to increase the number of coordinators. For proper operation of this counter, you must host the active coordinator in an application server other than the deployment manager.
Number of local bulletin-board subjects	RangeStatistic	Total number of subjects being posted to locally. The number includes the proxy postings (if any) done by the core group bridge service on behalf of servers belonging to different WebSphere cells.

5.1.13 Object Pools

The Object Pools contain information for the Object Pools. Examples include objects created, allocated and returned, along with idle size.

Table 5-15. Object Pools

Name	Type	Description
ObjectsCreatedCount	CountStatistic	Total number of objects created.
ObjectsAllocatedCount	CountStatistic	Number of objects requested from the pool.
ObjectsReturnedCount	CountStatistic	Number of objects returned to the pool.
IdleObjectsSize	RangeStatistic	Average number of idle object instances in the pool.

This page intentionally left blank.

Appendix A: References

A.1 Nastel Documentation

Table A-1. Nastel Documentation	
Document Number (or higher)	Title
M6/INS 600.007	<i>AutoPilot® M6 Installation Guide</i>
M6/USR 600.009	<i>AutoPilot® M6 Administrator's Guide</i>
M6WMQ/INS 620.003	<i>AutoPilot® M6 for WebSphere MQ Installation Guide</i>
M6WMQ/ADM 620.002	<i>AutoPilot® M6 for WebSphere MQ Administrator's Guide</i>
M6WMQ/SM 600.002	<i>AutoPilot® M6 for WebSphere MQ Security Manager User's Guide</i>
M6/WMQ 600.002	<i>AutoPilot® M6 Plug-in for WebSphere MQ</i>
M6/WMQ 600.001	<i>AutoPilot® M6 Plug-in for WebSphere MQ</i>

A.2 IBM Documentation

SC33-1872 WebSphere MQ Intercommunications

SC33-1369 WebSphere MQ MQSC Command Reference

SC34-5456 WebSphere MQ Using Java

<http://www-306.ibm.com/software/integration/mqfamily/library/manualsa/>

<http://www-306.ibm.com/software/integration/wmq/support/>

This page intentionally left blank.

Appendix B: Conventions

B.1 Typographical Conventions

Table B-1. Typographical Conventions	
Convention	Description
Blue/Underlined	Used to identify links to referenced material or websites. Example: support@nastel.com
Bold Print	Used to identify topical headings, glossary entries, toggles and buttons used in procedural steps. Example: Click EXIT .
<i>Italic Print</i>	Used to place emphasis on titles, menus, screen names, or other categories.
Monospaced Bold	Used to identify keystrokes/data entries, file names, directory names, etc.
<i>Monospaced Italic</i>	Used to identify variables in an address location. Example: [<i>AutoPilot_Home</i>]\documents, where portion of the address in brackets [] is variable.
Monospaced Text	Used to identify addresses, commands, scripts, etc.
Normal Text	Typically used for general text throughout the document.
Table Text	Table text is generally a smaller size to conserve space. 10, 9, and 8 point type is used in tables through the AutoPilot product family of documents.

This page intentionally left blank.

Glossary

Application Programming Interface (API): a source code interface that an operating system, library or service provides to support requests made by computer programs.

API: *see* Application Programming Interface.

Asynchronous: Communication between computer and devices that can occur at any time and at irregular intervals.

AutoPilot M6: Nastel Technologies' Enterprise Application Management Platform. AutoPilot M6 is designed to monitor and control distributed IT services such as application servers, middleware, user applications, workflow engines, brokers, Service Oriented Architecture (SOA) and Enterprise Service Bus (ESB) based applications and their impact on business services.

AutoPilot M6 for WMQ: Nastel Technologies' WebSphere MQ management solution. Re-designated as M6 for WMQ with release 6.0, prior releases retain the AP-WMQ or MQControl trademark.

AutoPilot M6 Web: AutoPilot M6 Web is a browser-based interface that provides monitoring and operational control over managed resources and applications. It allows users to monitor health, recover from a failure, view historical performance graphs and visualize impacts of a failure.

BSV: *see* Business View

Business View (BSV): A collection of rules that define a desired state of an eBusiness environment. Business Views can be tailored to present information in the form most suited to a given user, as defined by the user.

CEP: *see* Complex Event Processing.

CEP Server: A container that can host any number of AutoPilot M6 services such as experts, managers, policies, etc.

Client: Any programming component that uses the AutoPilot M6 infrastructure; for example, the AutoPilot M6 User Console.

Complex Event Processing (CEP): A technology for building and managing event-driven information systems. CEP is primarily an event processing concept that deals with the task of processing multiple events from an event cloud with the goal of identifying the meaningful events within the event cloud. CEP employs techniques such as detection of complex patterns of many events, event correlation and abstraction, event hierarchies, and relationships between events such as causality, membership, and timing, and event-driven processes.

Console: The console acts as the graphical interface for AutoPilot M6.

Contacts: A subordinate to a given Manager or Expert.

Data Source Name (DSN): Logical name that is used by Open Database Connectivity (ODBC) to refer to the drive and other information that is required to access data. The name is use by Internet Information Services (IIS) for a connection to an ODBC data source, (Example: Microsoft SQL Server database). The ODBC tool in Control Panel is used to set the DSN. When ODBC DSN entries are used to store the connection string values externally, you simplify the information that is needed in the connection string. This makes changes to the data source completely transparent to the code itself.

Decision Support System (DSS): An AutoPilot M6-based service designed to monitor, store, and display any event information generated by AutoPilot M6-enabled middleware and applications.

Deploy: To put to use, to position for use or action.

Domain Server: A specialized CEP server that maintains the directory of CEP servers, experts etc. The domain server is also capable of hosting experts, managers etc.

DSN: *see* Data Source Name

DSS: *see* Decision Support System

Enterprise Service Bus (ESB): A software architecture construct, implemented by technologies found in a category of middleware infrastructure products usually based on standards that provides foundational services for more complex architectures via an event-driven and standards-based messaging engine.

ESB: *see* Enterprise Service Bus

EVT: Event Log file extension (e.g.: sample.evt)

Event: An Event is something that happens to an object. Events are logged by AutoPilot M6 and are available for use by AutoPilot M6 Policies or the user.

Expert: Services that monitor specific applications such as an applications server, web-server or specific components within the applications -- (Example - channels in WebSphere MQ). Experts generate facts.

Facts: Facts are single pieces of data that have a unique name and value. One or more facts are used to determine the health of the object, application or server.

File Monitor: Built-in expert that monitors application error logs and publishes logs to AutoPilot M6 as facts and events.

Graphical User Interface (GUI): A type of environment that represents programs, files, and options by means of icons, menus, and dialog boxes on the screen. The user can select and activate these options by pointing and clicking with a mouse or, often, with the keyboard. Because the graphical user interface provides standard software routines to handle these elements and report the user's actions (such as a mouse click a particular icon or at a particular location in text, or a key press); applications call these routines with specific parameters rather than attempting to reproduce them from scratch.

Grids: In AutoPilot M6, a collection of clusters that allows users to define and automate CEP server failover. Defined under domain server\naming\grid folder.

GUI: *see* Graphical User Interface.

IIS: *see* Internet Information Services

Internet Information Services (IIS): Microsoft's brand of Web server software, utilizing HTTP to deliver World Wide Web documents. It incorporates various functions for security, allows CGI programs, and also provides for Gopher and FTP services.

J2EE: *see* Java Platform, Enterprise Edition.

Java: A platform-independent, object-oriented programming language developed and made available by Sun Microsystems.

Java Database Connectivity (JDBC): The JDBC API provides universal data access from the Java programming language. Using the JDBC 2.0 API, you can access virtually any data source, from relational databases to spreadsheets and flat files. JDBC technology also provides a common base on which tools and alternate interfaces can be built. The *JDBC Test Tool* that was developed by Merant and Sun Microsystems may be used to test drivers, to demonstrate executing queries and getting results, and to teach programmers about the JDBC API.

Java Developer's Kit (JDK): A set of software tools developed by Sun Microsystems, Inc., for writing Java applets or applications. The kit, which is distributed free, includes a Java compiler, interpreter, debugger, viewer for applets, and documentation.

Java Management Extensions (JMX): Java technology that supplies tools for managing and monitoring applications, system objects, devices (e.g. printers) and service oriented networks. Those resources are represented by objects called MBeans (for *Managed Bean*).

Java Message Service (JMS): Java message-oriented middleware API for sending messages between two or more clients.

Java Platform, Enterprise Edition (J2EE): A widely used platform for server programming in the Java programming language. The Java EE Platform differs from the Standard Edition (SE) of Java in that it adds additional libraries which provide functionality to deploy fault-tolerant, distributed, multi-tier Java software, based largely on modular components running on an application server.

Java Server Pages (JSP): JSP technology enables rapid development of web-based applications that are platform independent. Java Server Pages technology separates the user interface from content generation enabling designers to change the overall page layout without altering the underlying dynamic content. Java Server Pages technology is an extension of the Java™ Servlet technology.

Java Virtual Machine (JVM): The “virtual” operating system that JAVA-written programs run. The JVM is a hardware- and operating system-independent abstract computing machine and execution environment. Java programs execute in the JVM where they are protected from malicious programs and have a small compiled footprint.

JDBC: *see* Java Database Connectivity

JDK: *see* Java Developer’s Kit.

JMS: *see* Java Message Service.

JMX: *see* Java Management Extensions.

JRE: JAVA Run-time Environment. The minimum core JAVA required to run JAVA Programs.

JSP: *see* Java Server Pages

JVM: *see* JAVA Virtual Machine.

Kerberos Authentication: Mutual authentication between server and client without exchanging a password. Kerberos prevents eavesdropping or replay attacks, and ensures the integrity of the data. Its designers aimed primarily at a client-server model, and it provides mutual authentication — both the user and the server verify each other's identity. Kerberos builds on symmetric key cryptography and requires a trusted third party.

LDAP: *see* Lightweight Directory Access Protocol.

Lightweight Directory Access Protocol (LDAP): A networking protocol for querying and modifying directory services running over TCP/IP.

Manager: Managers are the home or container for policies. All business views must reside on managers, and managers must be deployed prior to deploying a business view or policy.

Message-Oriented Middleware (MOM): A client/server infrastructure that increases interoperability, portability, and flexibility of an application by allowing the application to be distributed over multiple heterogeneous platforms.

Message Queue Interface (MQI): Part of IBM’s Networking Blueprint. It is a method of program-to-program communication suitable for connecting independent and potentially non-concurrent distributed applications.

MOM: *see* Message-Oriented Middleware.

MQI: *see* Message Queue Interface

Naming Service: A common server records “names” of objects and associates them with references, locations and properties.

Object Request Broker (ORB): A piece of middleware software that allows programmers to make program calls from one computer to another via a network.

ODBC: *see* Open Database Connectivity.

Open Database Connectivity (ODBC): Provides a standard software API method for using database management systems.

ORB: *see* Object Request Broker.

Package Manager: The command line utility that allows users to list, install, uninstall, verify and update AutoPilot M6 installation on any CEP server.

PKGMAN: *see* Package Manager

Policy/Business Views: Policies are a collection of one or more sensors. Business views are used to visually present the health and status of the different systems as well as automatically issue remedial actions.

Process Wrapper: Built-in wrapper that monitors a process or script started by AutoPilot M6.

Sensor: A rule that is used to determine the health of an object or application based on one or more facts. Actions can then be issued, based on the health.

Service-Oriented Architecture (SOA): An evolution of distributed computing and modular programming. SOAs build applications out of software services. Services are relatively large, intrinsically unassociated units of functionality, which have no calls to each other embedded in them. They typically implement functionalities most humans would recognize as a service, such as filling out an online application for an account, viewing an online bank statement, or placing an online book or airline ticket order. Instead of services embedding calls to each other in their source code, protocols are defined which describe how one or more services can talk to each other. This architecture then relies on a business process expert to link and sequence services, in a process known as orchestration, to meet a new or existing business system requirement.

Simple Mail Transfer Protocol (SMTP): A TCP/IP protocol for sending messages from one computer to another on a network. This protocol is used on the Internet to route e-mail. *See also* communications protocol, TCP/IP. *Compare* CCITT X series, Post Office Protocol.

SMTP: *see* Simple Mail Transfer Protocol

SOA: *see* Service-Oriented Architecture

Synchronous: Communication within a computer that occurs at regular intervals. It is usually governed by the microprocessor clock.

TCP/IP: *see* Transmission Control Protocol/Internet Protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP): A protocol developed by the Department of Defense for communications between computers. It is built into the UNIX system and has become the de facto standard for data transmission over networks, including the Internet.

UDP: *see* Universal Datagram Protocol.

Universal Datagram Protocol (UDP): A connectionless protocol that runs on top of IP networks. Provides very few error recovery services, offering instead a direct way to send and receive datagrams over an IP network. It's used primarily for broadcasting messages over a network.

Virtual Machine (VM): Software that mimics the performance of a hardware device, such as a program that allows applications written for an Intel processor to be run on a Motorola chip. *Also See* Java Virtual Machine.

Visual Source Safe (VSS): Microsoft VSS helps you manage your projects by saving them to a database. When you need to share files between two or more projects, you can share them quickly and efficiently. When you add a file to VSS, the file is backed up on the database, made available to other people, and changes that have been made to the file are saved so you can recover an old version at any time. Members of your team can see the latest version of any file, make changes, and save a new version in the database.

VM: *see* Virtual Machine

VSS: *see* Visual Source Safe.

WebSphere MQ: IBM's message queuing product.

Websphere_MQ_Manager: A specialized manager capable of hosting one or more MQSeries specific policies, apart from the regular policies.

Wireless Application Protocol (WAP): An open global specification that is used by most mobile telephone manufacturers. WAP determines how wireless devices utilize Internet content and other services. WAP enables devices to link diverse systems contents and controls.

XEN: A free software virtual machine monitor for IA-32, x86-64, IA-64 and PowerPC architectures. It is software that runs on a host operating system and allows several guest operating systems to be run on top of the host on the same computer hardware at the same time. Modified versions of Linux and NetBSD can be used as hosts.

Index

A	
Abbreviation	2
Automatic Start.....	12
B	
BoundedRangeStatistic.....	15
C	
configuring experts.....	8
CountStatistic	15
D	
data counters.....	16
Deploy Expert.....	7
Dynamic Cache Data Counters.....	24
E	
Enterprise Beans Data Counters	16
F	
Fact Status	15
Feedback, User	1
G	
Global Security.....	14
H	
High Availability Manager Data Counters	25
I	
INSTALL.TXT.....	1
Installation Preparation.....	5
installation,AutoPilot/WebSphere	6
J	
J2C Connection Pool Data Counters	19
Java Virtual Machine Runtime Data Counters	20
JDBC Connection Pool Data Counters.....	18
L	
Licensing Information	5
log name	11
M	
Materials, installation	5
O	
Object Pools Data Counters.....	25
Object Request Broker Performance Data Counters	20
P	
PMI.....	1, 15
Preparation, installation.....	5
R	
RangeStatistic	15
README.TXT	1
recovery	12
Registry, save in	12
Resource Center installation.....	6
restart.....	12
S	
<i>service activity</i>	11
<i>service audit trace</i>	11
Servlet Session Manager Data Counters.....	21
Statistic Types	15
System Data Counters	24
T	
technical support.....	2
Terms.....	2
text files	5
ThreadPool Data Counters	23
TimeStatistic.....	15
Transaction Manager Data Counters	22
U	
User Feedback	1
W	
Web Applications Data Counters	23
WebSphere Expert configuration	8
<i>websphere server</i>	11
<i>WebSphere WSPC</i>	11
<i>WSPC, WebSphere</i>	11

This page intentionally left blank.