

---

# PeopleTools 8.51 PeopleBook: Performance Monitor

---

**August 2010**

PeopleTools 8.51 PeopleBook: Performance Monitor  
SKU pt8.51tpfm-b0810

Copyright © 1988, 2010, Oracle and/or its affiliates. All rights reserved.

### **Trademark Notice**

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

### **License Restrictions Warranty/Consequential Damages Disclaimer**

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

### **Warranty Disclaimer**

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

### **Restricted Rights Notice**

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

#### *U.S. GOVERNMENT RIGHTS*

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

### **Hazardous Applications Notice**

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

### **Third Party Content, Products, and Services Disclaimer**

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

# Contents

## Preface

<b>Performance Monitor Preface</b> .....	<b>xi</b>
Performance Monitor .....	xi
PeopleBooks and the PeopleSoft Online Library .....	xi

## Chapter 1

<b>Getting Started with Performance Monitor</b> .....	<b>1</b>
Performance Monitor Overview .....	1
Performance Monitor Implementation .....	1
Installation-Level Steps .....	1
Post Installation-Level Steps .....	2

## Chapter 2

<b>Understanding Performance Monitor</b> .....	<b>3</b>
Performance Monitor .....	3
Performance Monitor Architecture .....	4
The Monitored System .....	6
The Monitoring System .....	8
Implementation Options .....	10
Self-Monitoring Monitoring System .....	10
Production Monitoring System .....	11
Performance Monitor Instrumentation .....	11
Understanding PMUs .....	11
Understanding Events .....	12
Performance Integration with Third Party Systems .....	12
Performance Monitor Data .....	12
Scope of the Performance Monitor .....	13

## Chapter 3

<b>Setting Up the Performance Monitor</b> .....	<b>15</b>
---	-----------

Configuration of the Performance Monitor .....	15
Enabling the Required Elements on the Monitoring System .....	15
Setting Up the Performance Monitor Database .....	16
Setting Up the Application Server .....	16
Setting Up the Web Server for the Monitoring System .....	17
Creating a PPMI User ID .....	18
Specifying the PPMI URL and PPMI User ID .....	18
Specifying the Integration Gateway URL .....	19
Setting Up the Process Scheduler Server for the Monitoring System .....	19
Enabling the Required Elements on the Monitored System .....	20
Setting Up the Database of the Monitored System .....	20
Specifying the Monitor URL .....	20
Setting Up the Web Server of the Monitored System .....	21
Setting Up the Application Server .....	21
Setting Up the Process Scheduler Server .....	22

## Chapter 4

<b>Administering the Performance Monitor .....</b>	<b>23</b>
Performance Monitor Administration .....	23
Performance Monitor Administration .....	24
Pages Used for Performance Monitor Administration .....	24
Setting Global System Options .....	25
Setting System Definition Defaults .....	29
Modifying System Definitions .....	31
Viewing Agent Definitions .....	35
Setting Agent Filter Options .....	36
Scheduling the Reaper Program .....	39
Scheduling Performance Data Archiving .....	39
Running the Performance Data Archiving Program .....	40
Working with Aborted Program Runs .....	40
Populating Performance Monitor Lookup Tables .....	40
Working with Performance Monitor Tables .....	41
Disabling Performance Monitor Agents .....	42
Working with Performance Monitor Web Profile Properties .....	43
Tracing Performance Monitor Agents .....	43
Enabling Tracing on the Application Server or Process Scheduler Server .....	43
Enabling Tracing on the Web Server .....	43
Tracing the Monitor and PPMI Servlets .....	44
Configuring Performance Monitoring Sampling Rate .....	44
Viewing Monitor Servlet Diagnostics .....	44
Monitoring Agents .....	45
Monitoring PPMI Clients .....	46

Setting Up Monitor Clusters .....	46
Using Performance Monitor Data Mover Scripts .....	48
Estimating Your Performance Database Size .....	49
Estimating Your Performance Database Size Overview .....	49
Estimating Space Requirements for Event Data .....	50
Estimating Space Requirements for PMU Data .....	52
Calculating Space Requirements .....	53

## Chapter 5

<b>Working with the Performance Trace .....</b>	<b>55</b>
Understanding the Performance Trace .....	55
Enabling a Performance Trace .....	56
Running a Performance Trace .....	56
Viewing Performance Trace Information .....	58

## Chapter 6

<b>Monitoring System Performance .....</b>	<b>59</b>
System Performance Monitoring .....	59
Monitoring System Performance .....	59
Pages Used to Monitor System Performance .....	60
Using the System Performance Home Page .....	61
Viewing Web Server Performance .....	67
Viewing Application Server Domain Performance .....	71
Viewing Analytic Server Information .....	75
Viewing Process Scheduler Server Performance .....	75
Viewing Master Scheduler Performance .....	79
Viewing Open PMUs .....	81
Viewing Open PMU Trees .....	84
Viewing Current User Sessions .....	85
Viewing Recycled Processes Diagnosis .....	86
Monitoring Resource Usage .....	86
Working with Event 150 (JVM Status) .....	86
Working with Event 151 (Network Status) .....	86
Working with Event 200 (Resources Per Process) .....	87
Working with Event 300 (Host Resource Status) .....	89
Working with Event 301 (Tuxedo "pq Rows") .....	92
Working with Event 302 (Tuxedo "psr" Rows) .....	92

**Chapter 7**

**Analyzing Historical Performance Data ..... 93**

Historical Performance Data ..... 93

    Understanding Historical Performance Data ..... 93

    Pages Used to Analyze Historical Performance Data ..... 93

Viewing Completed PMUs ..... 94

    Entering Search Criteria ..... 94

    Working with Completed PMUs ..... 96

    Completed PMU Status ..... 100

    The PMU Details Page ..... 101

Viewing Event History ..... 101

    Entering Search Criteria ..... 101

    Working with Event History Information ..... 103

Viewing User Session History ..... 105

    Entering Search Criteria ..... 105

    Working with User History Information ..... 106

Viewing Concurrent Signon Information ..... 107

    Understanding Concurrent Signon Information ..... 107

    Maintaining Chart Data ..... 107

    Viewing The Concurrent Signon Chart ..... 109

**Chapter 8**

**Working with Performance Monitor Charts and Analytics ..... 113**

Performance Monitor Charts and Analytics ..... 113

    Understanding Performance Monitor Charts and Analytics ..... 113

    Common Elements Used in This Chapter ..... 114

    Pages Used to View Performance Monitor Charts ..... 114

Viewing User Request Analytics ..... 115

    The User Requests Chart ..... 117

    The Request Time vs. Think Time Chart ..... 118

    User Request Details ..... 119

Viewing Component Trace Information ..... 123

Viewing Component Statistics ..... 127

Viewing Portal Statistics ..... 130

Viewing PIA Statistics ..... 132

Viewing Top Portal Content Requests ..... 133

Viewing Top PeopleCode Events ..... 135

Viewing Top PeopleCode Executions ..... 137

Viewing Top PeopleCode SQL Statements ..... 139

Viewing Top Components .....	140
 <b>Chapter 9</b>	
<b>Performance Monitor Security Considerations .....</b>	<b>143</b>
Setting Up PSPPMSRV Authentication .....	143
Firewall Considerations .....	143
Agent Communication with the Monitor Servlet .....	144
PSPPMSRV Communication with the Monitor Servlet .....	145
Monitor Cluster Members .....	145
Setting Up SSL .....	145
Setting Up SSL Between Agents and Performance Monitor .....	145
Setting Up SSL Between Performance Monitor and PSPPMSRV and Monitor Cluster Members .....	146
Setting Up SSL Client Authentication .....	146
Protecting the Performance Monitor Servlets .....	149
 <b>Chapter 10</b>	
<b>Performance Monitor Meta-Data .....</b>	<b>151</b>
Performance Monitor Meta-Data Definitions .....	151
Metric Definitions .....	152
Event Definitions .....	154
Context Definitions .....	155
PMU Definitions .....	155
 <b>Appendix A</b>	
<b>PMU Definition Reference .....</b>	<b>157</b>
PMU 100 Portal Request .....	157
PMU 101 PIA Request .....	158
PMU 102 Compression Request .....	159
PMU 106 PIA Request From Portal .....	160
PMU 108 User Session Ended .....	161
PMU 109 User Session Began .....	162
PMU 113 Authenticate .....	163
PMU 114 Attach File Request .....	164
PMU 115 JOLT Request .....	164
PMU 116 Redirect after Login .....	166
PMU 117 Get Content .....	166
PMU 400 Tuxedo Service PCode and SQL .....	167
PMU 401 ICPanel .....	169

PMU 402 Modal Level 1 .....	170
PMU 403 Modal Level 2 .....	171
PMU 404 PeopleCode Program Execution .....	172
PMU 406 PeopleCode SQL Execute .....	173
PMU 407 PeopleCode BuiltIn SQL Execute .....	174
PMU 408 PeopleTools SQL Execute .....	175
PMU 409 Explicit Commit .....	176
PMU 410 ICScript .....	177
PMU 411 ICQuery .....	178
PMU 412 Tuxedo Service Summary .....	179
PMU 413 GetCertificate .....	180
PMU 414 SQL Fetch Summary .....	181
PMU 415 PortalRegistry .....	182
PMU 416 ICWorklist .....	183
PMU 417 FTP Request .....	183
PMU 418 PSBusComp .....	184
PMU 419 BI GetInterlink .....	185
PMU 420 BI Execute .....	186
PMU 421 BI BulkExecute .....	187
PMU 422 LDAP Bind .....	187
PMU 423 LDAP Search .....	188
PMU 424 Call AppEngine .....	189
PMU 425 Implicit Commit .....	190
PMU 426 RemoteCall Built-in .....	190
PMU 427 Deserialization .....	191
PMU 428 Serialization .....	192
PMU 439 Workflow PCode Summary .....	194
PMU 440 Usage Monitoring .....	195
PMU 500 SavePreChange PCode Summary .....	195
PMU 501 SavePostChange PCode Summary .....	197
PMU 502 SaveEdit PCode Summary .....	198
PMU 503 SaveEdit PCode Summary .....	199
PMU 504 RowSelect PCode Summary .....	200
PMU 505 RowInsert PCode Summary .....	201
PMU 506 RowDelete PCode Summary .....	202
PMU 507 FieldChange PCode Summary .....	203
PMU 508 FieldEdit PCode Summary .....	204
PMU 509 FieldDefault PCode Summary .....	205
PMU 510 PrePopup PCode Summary .....	206
PMU 511 ItemSelected PCode Summary .....	207
PMU 512 SearchInit PCode Summary .....	208
PMU 513 SearchSave PCode Summary .....	209
PMU 514 Active PCode Summary .....	210
PMU 515 PreBuild PCode Summary .....	211
PMU 516 PostBuild PCode Summary .....	212

PMU 700 PPMI Servlet .....	213
PMU 800 Monitor Servlet Request .....	213
PMU 801 Analytic Server Request .....	215
Filter Level .....	215
PMU 802 Synchronize/Read Data from Data Cache Summary .....	216
PMU 803 Synchronize/Write Data to Data Cache Summary .....	216
PMU 804 Synchronize/Read Data from Data Cache Details .....	217
PMU 805 Synchronize/Write Data to Data Cache Details .....	218
PMU 806 Load ACE Model Trees Summary .....	219
PMU 807 Load ACE Model Trees Details .....	220
PMU 808 Recalculate ACE Model Summary .....	221
PMU 809 Recalculate ACE Model Details .....	221
PMU 810 ACE Model Get Data .....	222
PMU 811 ACE Model Set Data .....	223

## Appendix B

<b>Event Definition Reference .....</b>	<b>225</b>
Event 150 .....	225
Event 151 .....	226
Event 152 .....	226
Event 153 .....	227
Event 200 .....	228
Event 300 .....	229
Event 301 .....	230
Event 302 .....	230
Event 350 .....	231
Event 351 .....	232
Event 354 .....	233
Event 355 .....	233
Event 356 .....	234
Event 500 .....	235
Event 600 .....	235
Event 601 .....	236
Event 701 .....	237
Event 702 .....	237
Event 703 PPMI Notify Cancel Interest .....	238
Event 704 PPMI Notify Agent Change .....	238
Event 705 PPMI Notify Monitor Change .....	239
Event 801 .....	239
Event 802 Monitoring Server Client Lost .....	240
Event 803 Monitoring Server Data Lost .....	240
Event 900 Agent Buffer Overrun .....	241

Event 901 Agent Init Notification .....	241
Event 902 Agent Configuration Ack .....	242
Event 903 Agent Contact Notification .....	243
Event 904 Analytic Server Status .....	243
Event 905 Analytic Server Load Event .....	244
Event 906 Out of Analytic Servers .....	244
Event 907 Analytic Server Recycle Event .....	245

## Appendix C

<b>PeopleSoft Performance Monitor Diagnostic Plug-in .....</b>	<b>247</b>
Understanding Performance Monitor Diagnostic Plug-in .....	247
Running Performance Monitor Diagnostics .....	248
Reviewing Diagnostic Information .....	248

## Appendix D

<b>Working With Usage Monitor .....</b>	<b>251</b>
Understanding Usage Monitor .....	251
Enabling Usage Monitor .....	251
Verifying Usage Monitor Data Collection .....	252
Running a Usage Monitor Test .....	252
Integrating Usage Monitor Data with Lifecycle Management Tools .....	253

<b>Index .....</b>	<b>255</b>
--------------------	------------

# Performance Monitor Preface

This preface introduces the Performance Monitor.

---

## Performance Monitor

Performance Monitor supports the monitoring of performance throughout your PeopleSoft systems.

Readers of this PeopleBook are expected to be familiar with the following:

- PeopleSoft architecture.
- SQL.
- PeopleCode.
- Principles of performance tuning.
- PeopleSoft development tools, such as PeopleSoft Application Designer and PeopleSoft Application Engine.
- General PeopleSoft system administration.

---

## PeopleBooks and the PeopleSoft Online Library

A companion PeopleBook called *PeopleBooks and the PeopleSoft Online Library* contains general information, including:

- Understanding the PeopleSoft online library and related documentation.
- How to send PeopleSoft documentation comments and suggestions to Oracle.
- How to access hosted PeopleBooks, downloadable HTML PeopleBooks, and downloadable PDF PeopleBooks as well as documentation updates.
- Understanding PeopleBook structure.
- Typographical conventions and visual cues used in PeopleBooks.
- ISO country codes and currency codes.
- PeopleBooks that are common across multiple applications.
- Common elements used in PeopleBooks.
- Navigating the PeopleBooks interface and searching the PeopleSoft online library.
- Displaying and printing screen shots and graphics in PeopleBooks.

- How to manage the locally installed PeopleSoft online library, including web site folders.
- Understanding documentation integration and how to integrate customized documentation into the library.
- Application abbreviations found in application fields.

You can find this companion PeopleBook in your PeopleSoft online library.

## Chapter 1

# Getting Started with Performance Monitor

This chapter discusses:

- Performance Monitor overview.
- Performance Monitor implementation.

---

## Performance Monitor Overview

The Performance Monitor enables you to view real-time and historical performance data of your PeopleSoft systems. The Performance Monitor provides the information that you need to solve immediate performance issues and analyze trends in system performance.

By default, the Performance Monitor is disabled on a new PeopleSoft installation. To enable the Performance Monitor, follow the configuration instructions in this PeopleBook.

When you have a PeopleSoft application environment running and you have set up the Performance Monitor, you can begin to capture and analyze PeopleSoft performance data.

---

## Performance Monitor Implementation

Before using the Performance Monitor, you need to perform some implementation and configuration tasks, in addition to standard PeopleSoft installation and implementation tasks.

### **See Also**

[Chapter 3, "Setting Up the Performance Monitor," page 15](#)

["Performance Monitor Preface," page xi](#)

*PeopleTools 8.51 PeopleBook: Getting Started with Enterprise PeopleTools*

## Installation-Level Steps

Complete the following steps during installation.

<b>Step</b>	<b>Reference</b>
Install PeopleTools.	See PeopleTools 8.51 Installation for your platform.
Install one or more PeopleSoft applications.	See Installation documentation for your applications.

## Post Installation-Level Steps

To enable the Performance Monitor complete these steps.

<b>Step</b>	<b>Reference</b>
Configure appropriate access to the menu items that are under the PeopleTools, Performance Monitor menu.  System administrators will need access to these pages to configure the system and view the pages displaying performance information.	See <i>PeopleTools 8.51 PeopleBook: Security Administration</i> .
Configure the monitoring system.	See <a href="#">Chapter 3, "Setting Up the Performance Monitor," Enabling the Required Elements on the Monitoring System, page 15.</a>
Configure one or more monitored systems.	See <a href="#">Chapter 3, "Setting Up the Performance Monitor," Enabling the Required Elements on the Monitored System, page 20.</a>

## Chapter 2

# Understanding Performance Monitor

This chapter discusses:

- Performance Monitor.
- Performance Monitor architecture.
- Implementation options.
- Performance Monitor instrumentation.
- Performance Monitor integration with third party systems.
- Performance Monitor data.
- Scope of the Performance Monitor.

---

## Performance Monitor

Performance Monitor helps system administrators monitor PeopleSoft performance data and, if wanted, share the data with third-party monitoring tools. Performance Monitor is a diagnostic utility for monitoring the performance of the main elements of your PeopleSoft system, such as web servers, application servers, and Process Scheduler servers. You can monitor real-time performance and analyze historical data.

Performance Monitor reports:

- Durations and key metrics of PeopleTools runtime execution, such as SQL statements and PeopleCode events.
- Key resource metrics, such as host CPU use and web server execution threads.

See [Chapter 2, "Understanding Performance Monitor," Performance Monitor Instrumentation, page 11.](#)

The metrics that are provided by Performance Monitor enable system administrators to:

- Monitor real-time system performance.
- Identify poorly performing tiers, hosts, domains, servers, application code, and SQL in a PeopleSoft environment.
- Identify performance trends.
- Address and isolate performance bottlenecks.

---

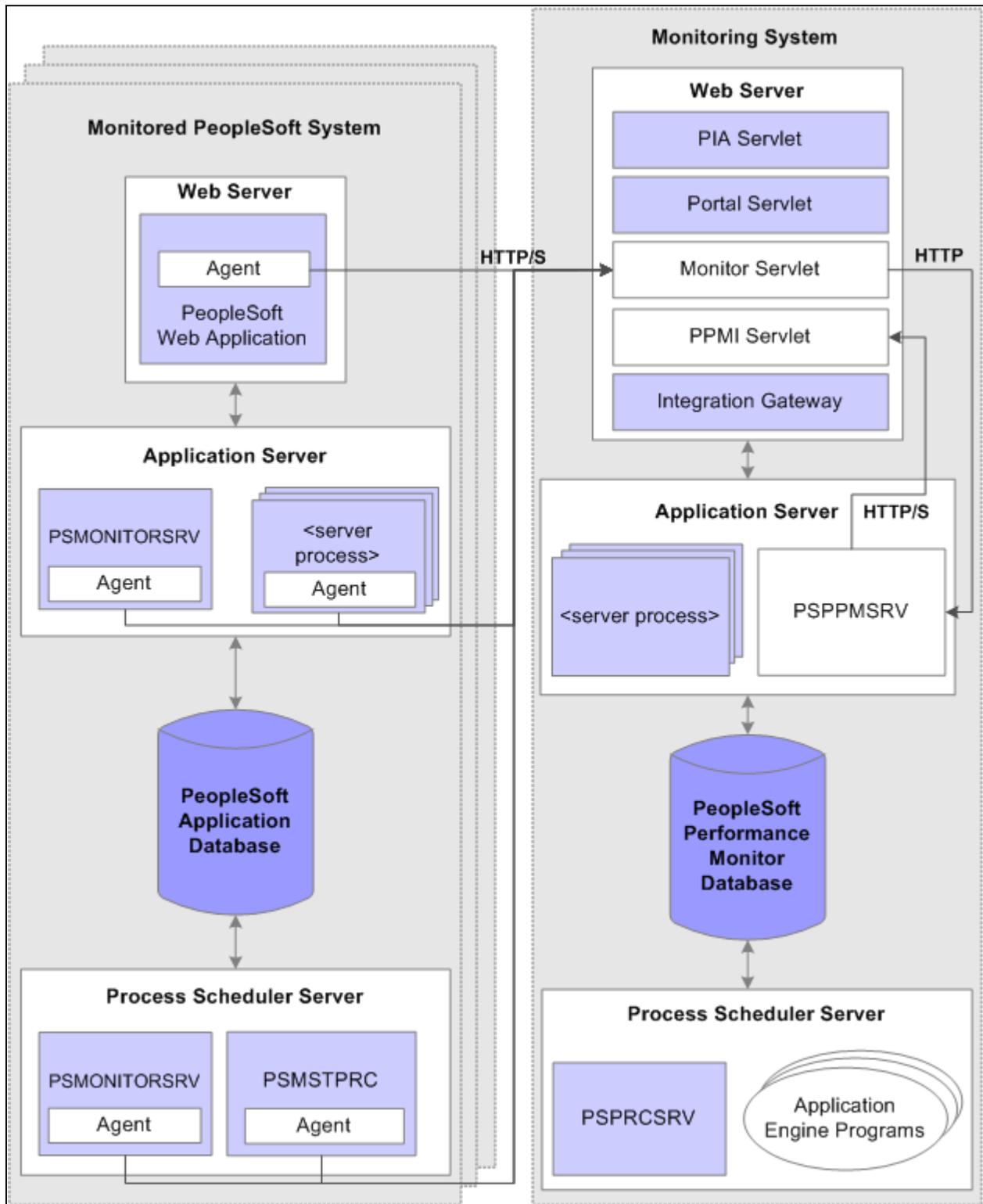
**Note.** Performance Monitor enables you to isolate and identify the causes of performance issues; however, it does not provide automatic solutions to performance issues.

---

---

## Performance Monitor Architecture

The following diagram illustrates the main elements of Performance Monitor.



Performance Monitor architecture

A logical distinction must be made between a *monitored system* and the *monitoring system*.

A monitoring system can monitor one or more PeopleSoft application systems. However, the monitored system can be monitored by just one monitoring system.

You can configure a self-monitoring system for development and testing.

## The Monitored System

The monitored system comprises the following main elements:

- PeopleSoft application database.
- Application server.
- Web server.
- Process Scheduler server.

---

**Note.** An application server is also known as a *business logic server* in some PeopleSoft materials. However, in this PeopleBook, only the term *application server* appears.

---

Performance Monitor agents reside on each main element of the monitored system. Agents run on the servers to gather and report performance information. An agent is a small piece of code operating from within an existing process. It collects performance data and uses a dedicated thread to send the data to the monitoring system. Agent threads report data to the monitoring system's monitor servlet, which in turn reports XML formatted performance data to the PSPMSRV server processes in the monitoring system's application server.

Two types of agents exist:

- *Process instrumentation agents* that report data about the specific server activities.
- *Domain resource agents* that report resource usage statistics about the process and server, such as CPU usage.

### **PeopleSoft Application Database**

The database being monitored is the database for your PeopleSoft application, such as the Human Capital Management database or the Customer Relationship Management database.

---

**Note.** Performance Monitor *does not* monitor the performance of the RDBMS or the server on which it runs.

---

### **Application Server**

The application server domain for the monitored system contains the following elements that are related to Performance Monitor:

<b>Performance Monitor Element</b>	<b>Description</b>
Server process agents	<p>One agent exists for each of the following server processes running within an application server domain:</p> <ul style="list-style-type: none"> <li>• PSAPPSRV</li> <li>• PSQCKSRV</li> <li>• PSQRYSRV</li> <li>• PSMONITORSRV</li> <li>• PSSAMSRV</li> <li>• PSANALYTICSRV</li> </ul>
Domain Monitor (PSMONITORSRV)	<p>The domain monitor runs in the PSMONITORSRV application server process. It monitors:</p> <ul style="list-style-type: none"> <li>• Host resource usage.</li> <li>• Tuxedo performance metrics.</li> </ul> <p><b>Note.</b> The domain monitor also monitors its own resource consumption.</p>

### **Web Server**

The web server contains the following elements that are related to the Performance Monitor:

<b>Performance Monitor Element</b>	<b>Description</b>
Web server agents	<p>Reports performance metrics from the instrumented code. The system starts a web server agent for a site after the site gets accessed for the first time.</p>
Domain Monitor	<p>Reports resource events that related to the web server usage. One resource monitor exists for each system that is hosted in each web application on each web server.</p>

### **Process Scheduler Server**

The Process Scheduler for the monitored system contains the following elements that are related to Performance Monitor:

<b>Component</b>	<b>Description</b>
Server process agent	<p>On the Process Scheduler server, only the Master Scheduler process, PSMSTPRC, has a server process agent recording performance data. The PSMSTPRC agent reports batch summary metrics, such as the number of and types of queued tasks.</p> <p><b>Note.</b> Performance information of this type is available only if a Master Scheduler is configured to run on the system.</p>
Domain Monitor (PSMONITORSRV)	<p>The domain monitor runs in the PSMONITORSRV server process. It monitors:</p> <ul style="list-style-type: none"> <li>• Host resource usage.</li> <li>• Tuxedo performance metrics.</li> </ul> <p><b>Note.</b> The domain monitor also monitors its own resource consumption.</p>

## The Monitoring System

Like the monitored system, the monitoring system includes the following elements:

- Performance Monitor database.
- Application server.
- Web server.
- Process Scheduler server.

### **Performance Monitor Database**

The database for the monitoring system requires at least the PeopleTools System database (PTSYS) that you create during an initial PeopleSoft installation. This database contains all of the necessary PeopleSoft objects and definitions that are required to store and display PeopleSoft performance data.

---

**Note.** The space that is required to store performance data requires increased tablespace sizes, especially in PSIMGR and the associated index tablespace.

---



---

**Note.** Any PeopleSoft system can act as a monitor for any other PeopleSoft system or for itself.

---

See [Chapter 2, "Understanding Performance Monitor," Implementation Options, page 10.](#)

### **Application Server**

The application server on the monitoring system contains the following elements that are related to Performance Monitor.

<b>Performance Monitor Element</b>	<b>Description</b>
PSAPPSRV	<p>The PSAPPSRV server process functions just as the PSAPPSRV process functions on any PeopleSoft application server. In addition, it advertises Performance Monitor services that are used by the Performance Monitor servlets to retrieve configuration and security data.</p> <p>At least two PSAPPSRV processes are required by the Performance Monitor. This ensures that:</p> <ul style="list-style-type: none"> <li>• The system can publish notifications of configuration changes through the PeopleSoft Integration Broker gateway.</li> <li>• The PPMI servlet can contact the application server to retrieve the configuration changes.</li> </ul>
PSPMSRV	<p>The PSPMSRV subscribes to performance data that is sent by the monitor servlet of the monitoring system and receives XML messages containing performance data, which it inserts into the monitoring database.</p> <p><b>Note.</b> The PSPMSRV does not receive any data directly from agents running on the monitored system.</p>

### **Web Server**

The web server on the monitoring system contains the following elements that are related to the Performance Monitor:

<b>Performance Monitor Element</b>	<b>Description</b>
PIA (psc) and Portal (psp) servlets	These servlets are the main PeopleSoft servlets that handle browser requests. Administrators use PeopleSoft pages to view performance information or set configuration options for a monitoring system.
Monitor (monitor) servlet	The monitor servlet receives all of the data that is sent by each agent running on the monitored system and relays it to the PSPMSRV processes.
Performance Monitor Integration (ppmi) servlet	The PPMI servlet accepts and authenticates registration requests from PSPMSRV processes to establish an HTTP pathway for receiving data from the monitor servlet.

<b>Performance Monitor Element</b>	<b>Description</b>
Integration Broker gateway (PeopleSoftListeningConnector) servlet	The gateway servlet is used to notify the monitor servlet of any configuration changes. The monitor servlet, in turn, notifies the agents that are running on the monitored system of any configuration changes the next time the agents interact with the monitor servlet.  <b>Note.</b> This is the same servlet that is used by PeopleSoft Integration Broker to handle integrations.

### **Process Scheduler Server**

The Process Scheduler server for the monitoring system has the following function:

<b>Performance Monitor Element</b>	<b>Description</b>
PeopleSoft Application Engine	You can schedule or invoke a variety of PeopleSoft Application Engine programs that maintain the data in your monitoring database.  See <a href="#">Chapter 4, "Administering the Performance Monitor," page 23.</a>

---

## **Implementation Options**

Two options are available for implementing the Performance Monitor:

- Self-monitoring system.
- Production monitoring system.

### **Self-Monitoring Monitoring System**

In a self-monitoring system, the same database that serves the PeopleSoft application is also being used to store the performance data. By default, all the elements that are required for the Performance Monitor are installed during the PeopleSoft server installations. So implementing the self-monitoring system requires no additional installation steps.

While this system may require less hardware, the potential impact to your online applications must be considered. Keep in mind that the same database engine is being used to store and retrieve performance data, and the same web server and application server may be used to monitor and collate performance data.

---

**Note.** The self-monitoring configuration is *not* supported for a production environment. The self-monitoring configuration is not designed for a production environment as it affects online transaction processing and overall system performance.

---

Typically, the self-monitoring system is used in development, testing, or training.

---

**Note.** The Performance Monitor has been designed so that a monitoring system does not encounter an infinite loop in a self-monitoring configuration.

---

## Production Monitoring System

With a production monitoring system, you configure a separate PeopleSoft system complete with web server, application server, and database server to act solely as a performance monitoring system. This is the recommended configuration for monitoring your production systems. Any PeopleTools system database that is delivered with PeopleSoft contains the appropriate database definitions to store and retrieve all Performance Monitor data.

With the separate monitoring system, you can monitor a single PeopleSoft system, such as your PeopleSoft Human Capital Management system, or you can configure it to monitor multiple PeopleSoft systems.

---

**Note.** Performance Monitor can monitor any PeopleSoft system running on PeopleTools 8.44 or above. Always consult PeopleTools Release Notes and PeopleBooks to become aware of any exceptions or unsupported features between release levels.

---

---

## Performance Monitor Instrumentation

This section explains the terminology that is associated with the Performance Monitor instrumentation. *Instrumentation* refers to the modifications that are made to internal PeopleSoft programming code that enables the system to report performance data to a monitor. Select areas of PeopleTools runtime source code have been instrumented.

When the system runs instrumented code, it generates time-stamped performance metrics that are relevant to that section of code. Agents running on the monitored system send the performance data to the Performance Monitor.

Instrumentation generates performance measurement units (PMUs) and events.

## Understanding PMUs

A PMU is a unit of measure that reflects the execution of a section of code. The system starts and stops a PMU at specific code locations, and the system may update a PMU anytime between the start and stop times. PeopleTools has defined a set of PMU types, and each type of PMU corresponds to the instrumentation at a specific code location, such as a SQL Execute in PSAPPSRV or a Jolt Request in the web server.

Each PMU includes:

- PMU Type.
- Instance identifier (a unique identifier for a specific PMU instance).
- Start time.
- Stop time.
- Status.

- Metrics (such as number of SQL fetches or buffer size used in a Jolt response).

See [Chapter 10, "Performance Monitor Meta-Data," PMU Definitions, page 155.](#)

Open PMUs are those that are currently being processed. These are PMUs for which a "stop" has not yet been received. Completed PMUs are those for which a "stop" has been received.

PMUs can assume a parent-child relationship. Child-PMUs start within a parent-PMU. Child-PMUs are linked to their parent by the parent instance ID. You can view PMUs within a tree structure that reveals the hierarchy of parent-child PMUs and indicates the processing times at each level.

## Understanding Events

Events are notifications containing performance metrics that are different from PMUs in that they are not hierarchical and they do not have durations. Events relate to resource usage, such as CPU usage or memory allocation. PeopleTools has defined a set of event types, and each type of event is reported at a specific location in the instrumented code.

Each event has:

- Type.
- Instance identifier (a unique identifier for a specific event instance).
- Timestamp.
- Severity.
- Metrics (such as CPU usage and memory allocation).

See [Chapter 10, "Performance Monitor Meta-Data," Event Definitions, page 154.](#)

---

## Performance Integration with Third Party Systems

Using web service technology, performance data that is generated by the Performance Monitor can be shared with third-party performance-monitoring tools by way of the Performance Monitor Integration (PPMI) API which uses the Simple Object Access Protocol (SOAP). The PPMI API protocol is not documented in PeopleBooks and is not intended for customer use. The API is made available for licensed members of the Oracle Partner Network to develop solutions that can be integrated with the PeopleSoft Performance Monitor.

---

## Performance Monitor Data

After the Performance Monitor is configured to retrieve and store data in the performance database tables, you can view the performance data using a variety of PeopleSoft pages that:

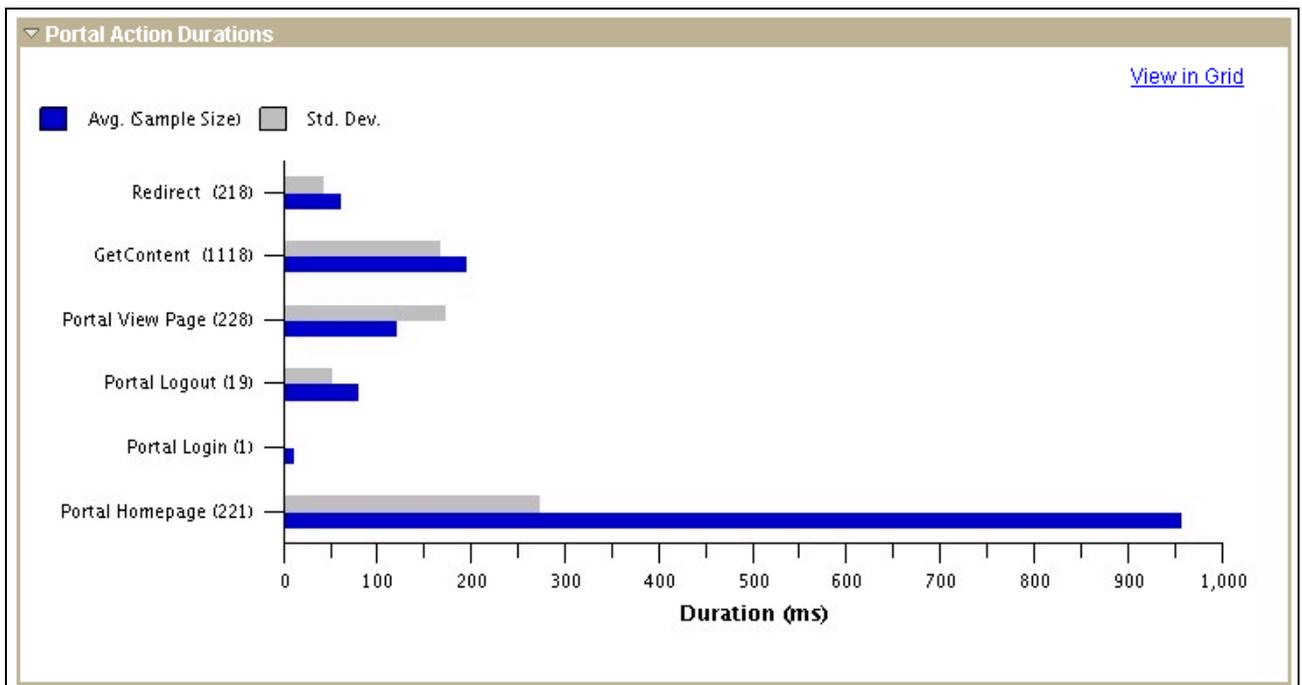
- Provide a summary of performance data.
- Present raw performance data.
- Display information in charts.

You can access all of the pages that are related to Performance Monitor using the PeopleTools Performance Monitor menu. Some performance information appears in grids.

Monitored Servers						
Customize   Find   First 1-8 of 8 Last						
Metrics Events and PMUs						
Server	Server Instance	PID	Agent Date/Time	%CPU Used	CPU Time (secs)	VM (bytes)
PSMONITORSRV	1	4140	11/16/2003 11:06:38PM	0	26.14	274948096
PSAPPSRV	2	2052	11/16/2003 11:04:45PM	11	254.734	340623360
PSAPPSRV	1	2880	11/16/2003 11:05:39PM	0	21.562	278372352
PSSAMSRV	100	1756	11/16/2003 11:03:53PM	0	84.186	274743296
PSAPPSRV	3	3040	11/16/2003 11:08:06PM	0	96.812	333246464
PSQRYSRV	70	4192	11/16/2003 11:03:39PM	0	45.437	279420928
PSSAMSRV	101	1800	11/16/2003 11:05:06PM	0	77.327	274743296
PSSAMSRV	102	4228	11/16/2003 11:04:12PM	0	92.967	273629184

Viewing grids

While other performance information appears in charts.



Viewing Charts

You have the option to view chart information in grids for sorting and downloading to Microsoft Excel.

## Scope of the Performance Monitor

The Performance Monitor provides performance metrics for:

- PeopleSoft Application Server processes (including SQL, PeopleCode, and Cache management).

- Web server servlet performance.
- Host resource usage.
- Master Scheduler performance.
- Tuxedo performance.
- Recycled server processes in the application server domain.
- User request performance.

---

**Note.** This list contains the only elements that are monitored by Performance Monitor.

---

## Chapter 3

# Setting Up the Performance Monitor

This chapter provides an overview of the configuration of the Performance Monitor and discusses how to:

- Enable the required elements on the monitoring system.
- Enable the required elements on the monitored system.

---

## Configuration of the Performance Monitor

The configuration of the Performance Monitor involves enabling the required elements on the monitored system and the monitoring system.

When you have installed the PeopleTools CD and transferred the appropriate files to your web server, application server, and Process Scheduler server, all of the required elements are in place. You do not need to install additional modules for the Performance Monitor.

---

**Note.** If you are setting up a self-monitoring system, the monitored system and the monitoring system reside on the same host machine.

---

---

## Enabling the Required Elements on the Monitoring System

This section covers the configuration of your monitoring system, which includes:

- Setting up the Performance Monitor database.
- Setting up the application server.
- Setting up the web server for the monitoring system.
- Creating a PPMI user ID.
- Specifying the PPMI URL and PPMI user ID.
- Specifying the Integration Gateway URL.

## Setting Up the Performance Monitor Database

The monitoring database requires the use of at least the PeopleTools System (PTSYS) database for the production monitoring system. Set this database up using the instructions in the PeopleTools Installation Guide.

---

**Note.** For the self-monitoring configuration, use the application database for your system. The application must be running on at least PeopleTools 8.44.

---

### See Also

*PeopleTools 8.51 Installation for your platform*

## Setting Up the Application Server

The application server on the monitoring system requires that the following server processes be enabled:

- Typical application server processes included in a default configuration.
- PSPPMSRV.

### Default Server Processes

A typical default application server configuration is suitable for Performance Monitor activity.

The PSAPPSRV server process is the main server process of the PeopleSoft Application Server.

At least two PSAPPSRV server processes must be enabled. This ensures that the monitoring system can publish notifications of configuration changes through the PeopleSoft Integration Broker gateway. You can enable more PSAPPSRVs to meet increased demand.

### PSPPMSRV

The PSPPMSRV process receives the performance data from the monitor servlet and inserts it into the monitoring database.

The following table describes the ways that you can enable this server process using the PeopleSoft Server Administration interface (PSADMIN).

<i>Method</i>	<i>Description</i>
Quick Configure menu	You can use the Quick Configure menu in PSADMIN to enable the PSPPMSRV server process. Enable the <i>Perf Collator</i> option.

<i>Method</i>	<i>Description</i>
Custom Configuration menu	<p>After you have scrolled through all of the configuration sections of the PSADMIN menu, the system prompts you to enable a variety of options. One of these prompts relates to enabling the PSPPMSRV server process. When prompted with the following:</p> <p>Do you want Performance Collators configured? (y/n)</p> <p>Enter y to indicate Yes.</p>

To add additional PSPPMSRV server processes, use the [PSPPMSRV] configuration section in PSADMIN and adjust the settings. The [PSPPMSRV] configuration section contains the following parameters.

**Min Instances** (minimum instances) Enter the minimum number of PSPPMSRV server processes that you want to start when the application server boots.

**Max Instances** (maximum instances) Enter the maximum number of PSPPMSRV server processes that you want to start when the application server boots.

---

**Note.** For the PSPPMSRV server process, the values assigned to Min Instances and Max Instances must be equal. For example, if you want three PSPPMSRV server processes to start, set Min Instances=3 and Max Instances=3. Unlike PSAPPSRV, Tuxedo does not handle load balancing between PSPPMSRVs. Decide how many PSPPMSRV instances you need operating within the appropriate number of domains and boot all of them.

---

If you have more than one PSPPMSRV configured, the monitor servlet automatically implements load balancing across the multiple PSPPMSRVs. Also, if you configure multiple application server domains, each with its own PSPPMSRVs, the monitor servlet automatically implements load balancing and failover across the domains.

## Setting Up the Web Server for the Monitoring System

By completing the internet portion of the PeopleSoft Multi-Platform Installer program, you have installed all of the elements that Performance Monitor requires on the web server. No additional installation procedures are required.

Performance Monitor servlets are not activated unless PSPPMSRVs register with them. Agent connections to a monitor are rejected until a PSPPMSRV on the monitoring system registers.

The Performance Monitor servlets that run on the web server of the monitoring system are:

- Monitor servlet.
- PPMI servlet.

---

**Note.** Make note of the URL of the web server for the monitoring system. On the monitored system, you need to specify the appropriate URL to which to send performance data.

---

---

**Note.** If you elect to implement a production monitoring system, you should set up the monitoring system itself in self-monitoring mode so that you can detect alarm events that are generated from the monitoring web server. That is, while the monitoring system monitors one or more PeopleSoft application systems, it also monitors itself.

---

If you need to configure multiple web servers for scalability and failover, you need to configure a cluster.

---

**Note.** If you notice communication errors in the log files, even though the agents have registered successfully, check the keep-alive setting on the monitoring web server. The keep-alive timeout in the monitoring web server may need to be increased to 660 or higher to resolve these issues. After changing this setting, restart the web server.

---

### **See Also**

[Chapter 4, "Administering the Performance Monitor," Setting Global System Options, page 25](#)

## **Creating a PPMI User ID**

The PSPPMSRV application server process acts as the PPMI client when you record PeopleSoft performance data. To register as a client, the PSPPMSRV requires the appropriate permissions in PeopleTools Security.

---

**Warning!** Setting up PSPPMSRV authentication is required.

---

To set up PPMI authentication:

1. Create a user profile with the name of your choice using PeopleTools Security.
2. Add the delivered *PeopleTools Perfmon Client* role to the user profile.

The *PeopleTools Perfmon Client* role contains the PTPMCLNT permission list.

3. Select PeopleTools, Performance Monitor, Administration, Global Administration and enter the user profile that you created and the associated password in the PPMI User ID and PPMI Password text boxes.

## **Specifying the PPMI URL and PPMI User ID**

PSPPMSRV processes must register with the PPMI servlet using the PPMI URL. The PSPPMSRV processes use the PPMI URL that is stored in the monitoring system's database to locate the PPMI servlet for registration.

The PSPPMSRV servers periodically read the PPMI value to determine the PPMI URL. When it is set to *NONE*, the monitoring system is disabled, which means that the PSPPMSRVs do not insert performance data into the monitoring database.

The PPMI user ID is required to authenticate the PSPPMSRV so that it can subscribe to performance data.

To specify the PPMI URL and PPMI user ID:

1. Make sure that you have created a PPMI user ID with the appropriate permissions (as described in the previous section).

2. Select PeopleTools, Performance Monitor, Administration, Global Administration.
3. On the Global Administration page, enter the appropriate value in the PPMI URL edit box.

The format is: *http://host[:port]/ppmi/[site]/*

For example: *http://server1.peoplesoft.com/ppmi/ps/*.

Where *ps* is the name of the PeopleSoft site.

4. Specify the PPMI user ID and password.
5. Click Save.
6. Click Ping Test to ensure that the URL that you entered is valid and the URL is available.

### **See Also**

[Chapter 9, "Performance Monitor Security Considerations," page 143](#)

## **Specifying the Integration Gateway URL**

A gateway URL needs to be specified so that the monitoring system can notify the monitor servlet of configuration changes. The monitoring servlet in turn notifies the monitored agents of these changes when they next communicate with the servlet. This communication uses the gateway, which you specify in the Integration Broker administration pages.

When specifying the gateway URL:

- Select LOCAL gateway.
- Enter this URL in the following format: *http://host[:port]/PSIGW/PeopleSoftListeningConnector*.

### **See Also**

*PeopleTools 8.51 PeopleBook: PeopleSoft Integration Broker Administration*, "Managing Integration Gateways," Defining Integration Gateways

## **Setting Up the Process Scheduler Server for the Monitoring System**

To invoke Performance Monitor Application Engine programs to maintain the performance data, set up a Process Scheduler server that connects to your monitoring database.

To set up a Process Scheduler server, follow the instructions that are provided in your PeopleTools Installation Guide.

See *PeopleTools 8.51 Installation for your platform*

The delivered PeopleSoft Application Engine programs that you schedule to run as part of Performance Monitor administration enable you to:

- Maintain the current PMU table (the PSPM\_REAPER program).
- Archive or delete historical performance data (the PSPM\_ARCHIVE program).
- Manage the lookup tables in the Performance Monitor interface (the PSPM\_LOOKUP program).

**See Also**

PeopleTools 8.51 Installation for your platform

[Chapter 4, "Administering the Performance Monitor," page 23](#)

---

## Enabling the Required Elements on the Monitored System

This section covers enabling the required elements on the monitored system, which is the system for which you intend to collect performance data. A monitored system requires the following items to be in place:

- PeopleSoft application database.
- Web server agent.
- Application server domain manager and agents.
- Process Scheduler server domain manager and agents.
- Valid monitor servlet URL value.

## Setting Up the Database of the Monitored System

As long as your PeopleSoft application is running against your PeopleSoft database, no further database configuration is required for the Performance Monitor.

---

**Note.** The database must be running on PeopleTools 8.44 or higher.

---

## Specifying the Monitor URL

Specifying a monitor URL enables the Performance Monitor agents and identifies the monitor servlet.

The agents on the monitored system check the monitor URL value periodically. When the monitor URL value is set to *NONE*, the Performance Monitor components on the monitored system are disabled.

The monitor URL value must reflect the URL of the monitor servlet that is running on the monitoring system.

To specify the monitor URL:

1. Select PeopleTools, Performance Monitor, Administration, Specify Monitor.

2. On the Specify Monitor page, enter the appropriate value in the Monitor URL edit box.

The format of the URL is *http[s]://host[:port]/monitor/[site]/or NONE*.

For example, enter *http://server2.domain.com/monitor/ps/*.

Where *ps* is the name of the PeopleSoft site.

3. Click Ping Test to ensure that you entered a valid URL and that the URL is available.
4. Click Save and Ping PPM to save the URL into the URL catalog, and to ensure that the associated agents are registered and able to share data with PPMI URL.

If the URL you enter is different than the previous value, the system saves the new value to the URL catalog, performs registration, and displays status of the registered agents.

In the case of a new registration, you must wait at least the duration of the Agent Heartbeat Interval.

If the agents have successfully registered, the PPMConsole (a new browser window) displays a list of all registered agents along with the agent configuration information. If no agents have registered, no list appears.

---

**Note.** In order for the new browser window (PPMConsole) to appear, the PPMConsole needs to be enabled on the Global Administration page.

---

See [Chapter 4, "Administering the Performance Monitor," Viewing Monitor Servlet Diagnostics, page 44](#)

.

See [Chapter 4, "Administering the Performance Monitor," Setting Global System Options, page 25](#).

## Setting Up the Web Server of the Monitored System

No additional installation procedure is required.

When the monitor URL value is supplied, agents on every PeopleSoft site automatically become active and begin relaying performance data to the monitoring system.

---

**Note.** Ensure that the Enable PPM Agents option is selected on the General tab of the appropriate web profile. By default, this option *is* enabled.

---



---

**Note.** If you start the web server in safe mode, then the monitoring agents in the web server are disabled.

---

## Setting Up the Application Server

The following components run and are monitored on the application server of the monitored system:

- Domain monitor: PSMONITORSRV.
- PSAPPSRV agents.
- PSQCKSRV agents.
- PSQRYSRV agents.

- PSSAMSRV agents.
- PSANALYTICSRV.

No installation or configuration procedures are required.

When the monitored system is enabled, agents in every instrumented server process automatically become active and begin relaying performance data to the monitoring system.

---

**Note.** Ensure that the EnablePPM Agent parameter in the [PSTOOLS] section of PSADMIN is set to 1 (enabled). If the EnablePPM Agent parameter is commented out or set to 0, then the agents are disabled for this server domain.

---

## Setting Up the Process Scheduler Server

The following components run and are monitored on the Process Scheduler server of the monitored system:

- Domain monitor (PSMONITORSRV).
- PSMSTPRC agents.

No installation or configuration procedures are required.

When the monitored system is enabled, agents in instrumented server processes automatically become active and begin relaying performance data to the monitoring system.

---

**Note.** Ensure that the EnablePPM Agent parameter in the [PSTOOLS] section of PSADMIN is set to 1 (enabled). By default, it is set to 1 (enabled) in the delivered domain templates. If the EnablePPM Agent parameter is commented out or set to 0, then the agents are disabled for this server domain.

---

## Chapter 4

# Administering the Performance Monitor

This chapter provides an overview of Performance Monitor and discusses how to:

- Set global system options.
- Set system definition defaults.
- Modify system definitions.
- View agent definitions.
- Set agent filter options.
- Schedule the reaper program.
- Schedule performance data archiving.
- Populate Performance Monitor lookup tables.
- Work with Performance Monitor tables.
- Disable Performance Monitor agents.
- Work with Performance Monitor web profile properties.
- Trace Performance Monitor agents.
- Trace the Monitor and PPMI servlets
- Configure performance monitoring sampling rate.
- View monitor servlet diagnostics.
- Set up monitor clusters.
- Use Performance Monitor data mover scripts.
- Estimate your performance database size.

---

## Performance Monitor Administration

This section discusses Performance Monitor administration and lists the pages used for Performance Monitor administration.

## Performance Monitor Administration

Performance Monitor administration includes:

- Specifying global settings.
- Viewing performance definitions such as those related to systems, agents, metrics, and so on.
- Setting system defaults.
- Scheduling batch programs that maintain performance data.

## Pages Used for Performance Monitor Administration

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Global Administration	PSPMMONITORGBL	PeopleTools, Performance Monitor, Administration, Global Administration	View and modify global administration settings, such as the PPMI URL value and monitor servlet clusters.
System Defaults	PSPMSYSDEFAULTS	PeopleTools, Performance Monitor, Administration, System Defaults	Set global system defaults for all monitored systems.
System Definitions	PSPMSYSDEFN	PeopleTools, Performance Monitor, Administration, System Definitions	View and modify the system definition that is associated with each of the systems that are being monitored. For example, you can set archive, PMU timeout, and agent buffer size.
Agent Definitions	PSPMAGENT	PeopleTools, Performance Monitor, Administration, Agent Definitions	View the definitions of the agents that are running on the monitored system's application server, web server, and Process Scheduler servers. It also enables you to disable the display of agent information.
Agent Filters	PSPMAGENTFILTER	PeopleTools, Performance Monitor, Administration, Agent Filters	Set the filter levels for the agents that are running on the monitored system. The filter level determines the amount and granularity of the performance data that is reported by an agent.

<b>Page Name</b>	<b>Definition Name</b>	<b>Navigation</b>	<b>Usage</b>
Schedule Reaper	PSPMREAPERRUNCNTL	PeopleTools, Performance Monitor, Administration, Schedule Reaper	Schedule the PeopleSoft Application Engine reaper program that is designed to maintain the integrity of the performance data that is in your monitoring system.
Schedule Archive	PSPMARCHIVERUNCNTL	PeopleTools, Performance Monitor, Administration, Schedule Archive	Schedule the PeopleSoft Application Engine archive program that is designed to archive or purge the performance data that is in your system.
Schedule Lookup Maintenance	PSPMLOOKUPRUNCNTL	PeopleTools, Performance Monitor, Administration, Schedule Lookup Maintenance	Schedule the PeopleSoft Application Engine program that is designed to maintain the lookup lists that are associated with the Performance Monitor interface. The lookup maintenance program ensures that the lookup lists are populated with current lookup options.

---

## Setting Global System Options

Access the Global Administration page (PeopleTools, Performance Monitor, Administration, Global Administration).

### Global Administration

\*PPMI URL:

\*PPMI User ID:   \*PPMI Password:

Archive: Clear PMUs & Events

#### Collator Row Limit

Maximum Rows:  ( 0 = Unlimited )

#### Search Row Limit

Maximum Rows:

#### Performance Monitor Cluster

Only enter cluster URLs for scalability and failover across multiple WebServers.

Cluster Members			
*Member Servlet URL			
<input type="text"/>	<input type="button" value="Register"/>	<input type="button" value="+"/>	<input type="button" value="-"/>

#### PPMConsole Settings

Enable PPMconsole      Password

Global Administration page

**PPMI URL**

The PPMI URL that is stored in the monitoring system's database is used by PSPPMSRV processes to locate the PPMI servlet with which to register.

The format of the URL is: `http[s]://<host[:port]>/ppmi/<site_name>/`

For example,

`http://server1.peoplesoft.com/ppmi/ps/`

After you specify this URL value, it is stored in the URL catalog under the ID PPM\_PPMI.

---

**Note.** If you select HTTPS, then the PSPPMSRVs use SSL encryption when sending the PPMI user ID and password to the PPMI servlet, but performance data sent from the PPMI servlet to PSPPMSRV is not encrypted. Because the data is not encrypted, the PSPPMSRV and the monitor servlet should reside in a secure network environment.

---

---

**Note.** Use the Ping Test button to verify that you have entered a valid URL and that the URL is available.

---

**PPMI User ID**

The PSPPMSRV server processes pass this user ID and password to the PPMI servlet. The servlet verifies that the user ID and password are valid, and that the user has permission to access the PPMI servlet.

The user ID that you specify must have a permission list with Performance Monitor PPMI Access selected in its PeopleTools permissions. PeopleTools recommends using the PTPMCLNT permission list, which is shipped expressly for this purpose.

---

**Note.** The PPMI User ID and PPMI Password values are required.

---

See [Chapter 9, "Performance Monitor Security Considerations," page 143.](#)

**PPMI Password**

The password associated with the PPMI user ID.

**Archive: Clear PMUs & Events** Indicates to the archive program to delete all of the performance history data that is currently stored in the monitoring database. This is in the form of an unqualified DELETE on the history tables, or for those platforms for which it is supported, the tables are truncated.

---

**Note.** If you select this option, the delivered archive program deletes all of the performance history data for *every* monitored system.

---

**Note.** Deleting and truncating performance data may cause the current user count to appear artificially low. The system bases the count on events that are found in the PSPMEVENTHIST table, and rows from this table may be deleted during the archive process.

---

**Note.** This option overrides any archive mode option that is set on the System Definition page.

---

See [Chapter 4, "Administering the Performance Monitor," Modifying System Definitions, page 31.](#)

**Collator Row Limit** Limits the amount of data that can be inserted into the Performance Monitor tables. Collators (PSPPMSRVs) periodically count the number of rows in each of the performance tables. If the total count of rows in PSPMTRANSHIST, PSPMTRANSCURR, and PSPMEVENTHIST is higher than this value, the PSPPMSRV does not insert any more rows into these tables until the number of rows deleted (by the archive program, the reaper program, manual intervention, or any combination of the three) brings the row count below this limit. If you set this value to 0, the PSPPMSRVs do not check for a row limit.

---

**Note.** If the row limit is reached, the System Performance page will report that the agents on the monitored system have stale agent data.

---

**Search Row Limit** Limits the number of rows that are returned and displayed in searches that are initiated from Performance Monitor pages. The system imposes this limit on all users who access the monitoring system.

**Performance Monitor Cluster** To provide failover and scalability, performance monitor servlets on multiple web servers can be configured as a cluster. Entering the cluster information in the Performance Monitor Cluster grid enables the load-balanced monitor and PPMI servlets to share client registration information so that PSPPMSRV instances need not be aware of the cluster configuration.

The format of the cluster member URL is `http://host[:port]/ppmi/<site_name>/`, where the host name is the actual web server machine.

Click Save and Notify Cluster to notify the cluster if you have added or removed a cluster member.

Click Ping Test to verify that you have entered a valid URL and that the URL is available.

See "Setting up Monitoring Clusters" below.

**PPMConsole Settings**

The PPMConsole settings control the display of the PPMConsole, which is an interface in a separate browser window that displays information related to servers, servlets, and agents. Click Enable PPMConsole to enable access to this interface.

To enable password security for accessing the PPMConsole, enter a password in the Password field. If you have provided a password, any system administrator who would need to access the PPMConsole will need to be aware of the password.

---

**Note.** After making any changes to the PPMConsole settings, you must restart the web server.

---

See [Chapter 3, "Setting Up the Performance Monitor," Specifying the Monitor URL, page 20.](#)

See [Chapter 4, "Administering the Performance Monitor," Viewing Monitor Servlet Diagnostics, page 44.](#)

---

## Setting System Definition Defaults

Access the System Defaults page (PeopleTools, Performance Monitor, Administration, System Defaults).

The System Defaults page enables you to set default values for all of the monitored systems. If you intend to monitor numerous systems, you can set the default values that you need for a system. When a new systems register with the monitoring system for the first time, the system adopts the default values that you have set.

Using the System Defaults page enables you to set global values for each monitored system rather than modifying the values for each system separately.

## System Defaults

Archive Mode

After:  days  
  Delete Data  
  Archive Data  
  Archive Nothing

Allow Performance Trace

PMU Timeout (days):	<input type="text" value="1"/>	Agent PMU Sample Rate (1/X):	<input type="text" value="50"/>
Agent Event Sample Rate (sec):	<input type="text" value="60"/>	Agent Heartbeat Interval (sec):	<input type="text" value="60"/>
Agent Buffering Interval (sec):	<input type="text" value="10"/>	Agent Max Buffer Size (bytes):	<input type="text" value="4194304"/>

Agent Filter Level:  ▼

Enable Usage Monitoring

Enable Usage Monitoring     
 Buffer Size

Apply to Current Systems

System Defaults page

**Note.** Except for the following page elements, the System Defaults page is identical to the System Definitions page, which is documented in the following section.

<b>Agent Filter Level</b>	Set the agent filter level for the agents of monitored systems. The default setting is 01–Standby, which means that the monitored system sends no performance information to the monitoring system.  See <a href="#">Chapter 4, "Administering the Performance Monitor," Setting Agent Filter Options, page 36.</a>
<b>Enable Usage Monitoring</b>	Select to enable the Usage Monitor so that system usage data can be collected for incorporation with the PeopleSoft Testing Framework.
<b>Buffer Size</b>	(Applies only to Usage Monitor data collection). Set the buffer limit which determines how much data should be collected in the buffer before storing the data in the database. Depending on the pages that are accessed and your testing needs, you can arrive at an optimum buffer size for your usage pattern.
<b>Apply to Current Systems</b>	Notifies the agents that are running in <i>existing</i> systems of the global configuration changes. When the agents of the existing monitored systems are notified, the existing systems adopt the new, default values.

**See Also**

Chapter 4, "Administering the Performance Monitor," Modifying System Definitions, page 31

## Modifying System Definitions

Access the System Definitions page (PeopleTools, Performance Monitor, Administration, System Definitions).

### System Definitions

**System Identifier:** 1      **Database Name:** QEDMO

**Unique Identifier:** da4f886e-7688-11de-9bb0-f071ea94d47a

**Description:**

Archive Mode

**After:**  days       Delete Data       Archive Nothing

Archive Data       Delete System

**Allow Performance Trace**

**PMU Timeout (days):**       **Agent PMU Sample Rate (1/X):**

**Agent Event Sample Rate (sec):**       **Agent Heartbeat Interval (sec):**

**Agent Buffering Interval (sec):**       **Agent Max Buffer Size (bytes):**

Save and Notify Agents

Versions

Tools Release	Valid From	Valid From Time	Valid To	Valid To Time
8.50-901.3	07/28/2009	6:33:11PM		

System Definitions page

A *system* refers to a particular monitored system. For example, you can monitor your PeopleSoft CRM system, PeopleSoft HCM system, and PeopleSoft Financials system. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system. The database name and GUID (a PeopleSoft value used to uniquely identify a PeopleSoft system) are provided by the agent during its registration process.

This section describes the properties and configuration options for each monitored system.

**System Identifier**      Identifies each monitored system. PeopleSoft automatically generates this value incrementally.

<b>Database Name</b>	The name of the PeopleSoft application database that is running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.
<b>Unique Identifier</b>	Uniquely identifies a particular PeopleSoft system. PeopleSoft assigns a unique value, referred to as a GUID, to each PeopleSoft application installation. When an agent registers with the Performance Monitor, it provides this GUID. The first time the monitoring system receives information from a monitored system, it detects the GUID. For each new GUID that is detected, the monitoring system creates a new monitored system definition. <hr/> <b>Note.</b> When copying databases, you should delete the GUID in the new (copied) database. If the new database is monitored by the same instance of the Performance Monitor that is monitoring the source database, the monitor assumes that the agents for both systems belong to the same system. This not only doubles the data that is being stored for a particular system, but also makes it unreliable. To resolve this, set the value of the GUID field in the PSOPTIONS table to <space> in the new database. The next time an application server connects to the database, the system generates a new, unique GUID. You insert the blank value in the PSOPTIONS table using the SQL tool at your site. <hr/> <b>Note.</b> After cloning a database and before booting the web server and application server, you should set the PPMI and monitor URL to NONE in the cloned database. This will prevent the agents of the cloned database from reporting into the monitoring system that was used previously. To reset the URLs, update the URL field to NONE in the PSURLDEFN table where the URL_ID field is 'PPM_MONITOR' or 'PPM_PPMI'. <hr/>
<b>Description</b>	A description of the monitoring system to assist recognition. The default value is the database name.

**Archive Mode**

The archive mode that you set specifies how the archive program (PSPM\_ARCHIVE) should process the performance data that is stored in the monitoring database. The default value is set to archive nothing after zero days.

The options are:

- **After N days.** If you select Archive Data or Delete Data, then you must specify a retention period, which determines the number of days during which performance data remains in the tables of the online monitoring system.

Performance data that is older than the specified value gets archived or deleted (depending on your selection) when the archive program runs.

- **Delete Data.** If this option is select, the next time the archive program runs, the program deletes performance history data that is older than the retention period.
- **Archive Data.** If this option is selected, the next time the archive program runs, the program archives performance history data that is older than the retention period.

The archive program moves history tables (PSPMTRANSHIST and PSPMEVENTHIST) to the archive tables (PSPMTRANSARCH and PSPMEVENTARCH).

- **Archive Nothing.** Disables performance data archiving for this monitored system.
- **Delete System.** If this option is selected, the next time the archive program runs, it deletes the system definition and its performance data.

Keep in mind that deleting a system does not prevent the monitored system from continuing to send performance data. You *must* set the Monitor URL value on the monitored system to *NONE* to disable monitoring.

---

**Note.** If the Archive: Clear PMUs & Events option is enabled in the Global Administration page, the archive settings are ignored.

---

See Schedule Archive section.

**Allow Performance Trace**

Enables authorized users who are connected to this monitored system to launch the Performance Trace Console. By default, this option is enabled on a new monitored system unless you adjust the system defaults on the System Defaults page.

See [Chapter 5, "Working with the Performance Trace," page 55](#).

**PMU Timeout**

Indicates the interval in which an open PMU is considered timed out. PMU timeouts are measured in days. The PMU Timeout value can't be set to zero.

After the specified interval, the system assumes that a PMU that has not finished encountered an error and should no longer be considered open. The PeopleSoft Application Engine reaper program (PSPM\_REAPER) moves timed out PMUs from the current PMU tables to the history PMU tables and sets the status to Reaper Timed Out.

---

**Note.** When a PMU starts, the application server agents specify an override timeout that is equal to the Tuxedo service timeout of the domain. In such cases, the reaper program uses the override value instead of the PMU Timeout that is specified here.

---

**Agent Event Sample Rate**

Specifies the rate (in seconds) that agents collect resource metrics, such as CPU usage. The default is 300 seconds (five minutes). To disable event sampling, set the value to zero.

---

**Note.** Smaller sampling rate intervals will provide more frequent updates of machine and process resource utilization on your monitored systems. However, consider that by decreasing the sampling rate interval, you increase the volume of data that is sent to and stored in the performance database. This increases the overall impact of performance monitoring.

---

**Agent Buffering Interval** Specifies the rate (in seconds) at which an agent sends performance data to the monitoring system.

This value can't be set to zero.

A smaller interval decreases the delay between the time when the monitored system generates performance data and the time that it is displayed on the monitoring system.

Larger intervals enable more efficient transmission of performance data across the network because the system can consolidate the data into packets. The larger the interval, the greater the Agent Max Buffer size should be set.

The default is 10 seconds.

**Agent PMU Sample Rate**

Enables you to reduce the amount of PMU data that is generated by monitoring just 1 of every N server trips.

Sampling does not affect PMUs that are initiated in a performance trace.

See [Chapter 4, "Administering the Performance Monitor," Configuring Performance Monitoring Sampling Rate, page 44.](#)

**Agent Heartbeat Interval**

This is the interval at which agents that do not have any performance data to report or that are in standby mode connect to the monitor servlet to be notified of any configuration changes. If agents post data to the monitoring system before this interval expires, they will be notified of any configuration changes and the timer will be reset.

The lower the number, the greater the responsiveness of the agents to configuration changes, but the greater the network traffic.

This value *can't* be set to zero.

The default value is 300.

**Agent Max Buffer Size**

Determines the maximum size of the buffer containing performance data. This enables you to cap the amount of data that is being stored by the agent on the monitored system and the amount of data that is sent across your network. If this limit is reached, the agent automatically discards new performance data until the current data has been posted to the monitor servlet.

The agent posts an alarm to the monitoring if the buffer size threshold is exceeded.

The default is 4194304 bytes (4 MB). The minimum must be 10240 bytes (10 KB).

**Save and Notify Agents**

Notifies the monitor servlet of configuration changes made on this page. First, the system saves the changes to the database. Notification of the monitor servlets occurs through the PeopleSoft Integration Broker gateway. The monitor servlet notifies the agents of changes when the agents next communicate with the monitor.

A delay occurs in publishing changes to the monitored systems. The maximum delay is the agent heartbeat interval.

---

**Note.** The Integration Broker Gateway must be configured correctly; otherwise, you will see an error message indicating that the agents were not notified of system changes.

---

**Versions**

Enables you to view a history of the PeopleTools versions that are installed on a monitored system.

The Tools Release column reveals the version numbers.

The Valid From columns indicate the date and time that an agent on a particular version first communicated with the monitoring system.

The Valid To columns indicate the date and time that an agent on a particular version last communicated with the monitoring system.

---

## Viewing Agent Definitions

Access the Agent Definitions page (PeopleTools, Performance Monitor, Administration, Agent Definitions).

Agent definitions enable you to view the details about the agents in monitored systems, and deactivate specific agents, if needed.

<b>System ID</b>	Identifies each monitored system. The PeopleSoft system automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.
<b>Database Name</b>	The name of the PeopleSoft application database that is running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.
<b>Agent ID</b>	Uniquely identifies an agent within a system. This is automatically generated by the monitor the first time an agent registers with it.
<b>Domain Name</b>	The name of the domain (application server or web server) in which the agent operates.
<b>Agent Type</b>	Indicates the type of server process the agent is monitoring.
<b>Domain Type</b>	Indicates whether the domain is an application server, a web server, or a Process Scheduler server domain.
<b>Domain Monitor</b>	Displays as <i>Yes</i> or <i>No</i> . If <i>Yes</i> , then this agent is responsible for sending resource events for its host or domain to the monitor at the specified sampling rate for the monitored system.
<b>Server Instance</b>	This number is specific to Tuxedo servers and corresponds to the Tuxedo instance number.
<b>Domain Host/Port</b>	The name or IP address of the server on which the domain resides, including the port number to which the domain listens for requests.
	<hr/> <b>Note.</b> Web server agents register with both the HTTP and the HTTPS ports. The application server agents register with the Jolt Server Listener (JSL) port. Process Schedulers do not have ports. <hr/>
<b>Domain Directory</b>	The directory in which the domain is installed on the server.
<b>Inactive Agent</b>	If this box is selected, the agent is considered inactive. That is, the agent's events and PMUs do not appear in the Performance Monitor pages showing current information. You can still view information about events and PMUs that are sent by inactive agents using the pages that display historical information.  To reactivate an agent, clear the check box, and click Save.  To deactivate an agent, select the check box, and click Save.

---

## Setting Agent Filter Options

Access the Agent Filters page (PeopleTools, Performance Monitor, Administration, Agent Filters).

## Agent Filters

**System ID:** 1                      **Database Name:** QEDMO

**Reset All Filters:**

Agent Filters			
Agent Type	Last Update User ID	Last Update Date/Time	*Filter Level
PERFMON	PTDOCGS	07/29/2009 8:36:16AM	<input type="text" value="04-Standard"/>
PSAPPSRV	PTDOCGS	07/29/2009 8:36:16AM	<input type="text" value="04-Standard"/>
PSMONITORSRV	PTDOCGS	07/29/2009 8:36:16AM	<input type="text" value="04-Standard"/>
PSMSTPRC	PTDOCGS	07/29/2009 8:36:16AM	<input type="text" value="04-Standard"/>
WEBRESOURCE	PTDOCGS	07/29/2009 8:36:16AM	<input type="text" value="04-Standard"/>
WEBSERVER	PTDOCGS	07/29/2009 8:36:16AM	<input type="text" value="04-Standard"/>

### Agent Filters page

Agent filters determine the amount of performance data that is generated and sent to the monitoring system. Depending on the situation, different levels of performance data may be needed to assist in your performance-related decisions. The levels range from no information to extremely detailed information.

Each type of PMU and event is associated with a filter level, which is the lowest level at which the system generates performance data for that PMU or event.

- System ID**                      Identifies each monitored system. The PeopleSoft system automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.
- Database Name**                      The name of the PeopleSoft application database that is running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.
- Reset All Filters**                      Enables you to uniformly adjust the agent filter levels across an entire system. Select the desired level and click Apply.

**Agent Filters**

The Agent Type column displays the types of agents that are currently known to the monitoring system.

The Last Update User ID column and Last Update Date/Time column display the user ID that last updated the filtering level and the date and time when the user made the last filtering level change.

The Filter Level column contains a drop-down list box for setting the filtering level. You have the following options:

- 01 Standby: The monitor system sends no performance data to the monitor.  
Agents on this monitored system contact the monitor at an interval that is specified by the agent heartbeat interval.
- 02 Error: Only error events are reported to the monitoring system by the agents.
- 03 Warning: Only error and warning events are reported to the monitoring system by the agents.
- 04 Standard: This is the default level of monitoring that is adopted by agents when they first register with the monitor.

In addition to errors and warnings, PMUs and events with a filter level of Standard are reported to the monitoring system. This is the recommended setting for monitoring production environments. It provides sufficient diagnostic information to isolate performance problems without inundating the network or monitoring database with performance data.

- 05 Verbose. In addition to errors and warnings, and standard PMUs and events, PMUs with a filter level of Verbose are reported to the monitoring system.

This setting provides more detailed performance metrics and may, for some production systems, be worth the overhead that is associated with this monitoring level.

- 06 Debug: In addition to errors and warnings, standard and verbose PMUs and events, PMUs with a filter level of Debug are reported to the monitoring system.

This setting provides extremely detailed performance metrics and is not suitable for monitoring production systems. It is intended only for development and test environments.

**Save and Notify Agents**

Notifies the PPMI servlet of configuration changes that are made on this page. First, the system saves the changes to the database. Notification occurs through the PeopleSoft Integration Broker gateway. The monitor servlet notifies the agents of changes when the agents next communicate with the monitor. A delay occurs in publishing changes to the monitored systems. The maximum delay is the agent heartbeat interval.

---

## Scheduling the Reaper Program

The reaper program is a delivered PeopleSoft Application Engine program named PSPM\_REAPER. The reaper program maintains the PeopleTools tables that the Performance Monitor uses to store performance data for current, real-time processing.

When the PSPPMSRV gets notified that a PMU has finished (it receives a STOP for an open PMU), it:

- Flags the corresponding start and update rows in the current PMU table (PSPMTRANSCURR) for deletion.
- Inserts a row for the completed PMU in the PSPMTRANSHIST table.

When the reaper program (PSPM\_REAPER) runs, it:

- Deletes all rows in current PMU table (PSPMTRANSCURR) that are flagged for deletion.
- Sets the status to timed out for expired PMUs in the current tables.

To run the reaper program:

1. Select PeopleTools, Performance Monitor, Administration, Schedule Reaper.
2. Select or add a run control ID.
3. Click Run.

PeopleTools delivers a recurrence definition named *PerfMon Reaper Recurrence*, which is set to run every 15 minutes. Modify this recurrence definition, if necessary, and associate it with the PSPM\_REAPER program to schedule the program to run at suitable intervals.

---

**Warning!** If you do not schedule the reaper program to run often enough, the PSPMTRANSCURR table will grow very large over time, and it may contain many old, open PMUs.

---

---

## Scheduling Performance Data Archiving

Performance data archiving options are set per system definition. So your HCM system and your CRM system may have different archiving modes. You define your archive settings in the Archive Mode group box on the System Definition page. The performance data archiving program is a PeopleSoft Application Engine program named PSPM\_ARCHIVE.

---

**Note.** The system overrides the archive options that are set in the System Definition page if you have selected the Clear PMUs & Events option on the Global Administration page.

---

See [Chapter 4, "Administering the Performance Monitor," Setting Global System Options, page 25.](#)

---

**Note.** PeopleSoft provides sample queries that demonstrate how to access data in the archive tables. Currently, no other delivered method of accessing the data is available in archive tables.

---

See [Chapter 4, "Administering the Performance Monitor," Working with Performance Monitor Tables, page 41.](#)

## Running the Performance Data Archiving Program

To run the archiving program:

1. Select PeopleTools, Performance Monitor, Administration, Schedule Archive.
2. Select or add a run control ID.
3. On the Schedule Archive page:
  - Determine whether you want to enable Run %UpdateStats at the end.

If you enable this option, the system runs %UpdateStats meta-SQL on both the history and archive tables after the archive program finishes successfully.

- Click Run to launch the archive program.

---

**Note.** You should set up a recurrence definition in Process Scheduler so that the archiving program runs at regular intervals. This can help keep the performance history tables at more manageable sizes while containing the most relevant data.

---

## Working with Aborted Program Runs

If the performance data archiving program does not finish successfully, the system automatically invokes the PeopleSoft Application Engine program named PSPM\_ARCHREC. This program is designed to return the system to the state it was in before the archive program started.

During an archive program run, the PSPPMSRVs redirect incoming PMU and event data to cloned history tables. When the archive program finishes, the system moves the data in the cloned tables to the history tables and the PSPPMSRVs resume inserting data directly into the history tables. If the archive program does not finish successfully, the PSPPMSRVs continue to insert data into the cloned tables.

---

## Populating Performance Monitor Lookup Tables

On many of the pages that are used for viewing performance information, such as the Current PMUs page, you are prompted to enter either a user ID, a component name, or a performance trace name to narrow the search to relevant performance data. Unless you are self-monitoring, the monitored system components and User IDs differ from those of the monitoring system. Therefore, the performance monitor provides its own lookup tables, and the PSPM\_LOOKUP PeopleSoft Application Engine program populates these lookup tables.

To run the lookup program:

1. Select PeopleTools, Performance Monitor, Administration, Schedule Lookup Maintenance.
2. Select or enter a run control ID.

3. On the Schedule Lookup page, click Run to launch the lookup program.

---

**Note.** You should set up a recurrence definition in Process Scheduler so that the lookup program runs at regular intervals.

---

## Working with Performance Monitor Tables

As with any PeopleTool or PeopleSoft application, the underlying application definitions and application data reside in a collection of database tables that are designed using Application Designer. Although most PeopleSoft applications provide data models that show the relationships between the database entities, typically, for PeopleTools, knowledge of the underlying database tables is not required.

However, with the Performance Monitor, knowledge of the underlying database tables may be required. For example, the Performance Monitor interface provides numerous options to use when you are viewing performance data, such as viewing by time range, viewing by user, viewing by component, and so on. In some cases, you may want a more customized view of your performance data than what the interface offers.

You can use PeopleSoft Query, or your SQL tool of choice, to build queries that run against the Performance Monitor tables and return the specific information that you require.

To assist you in creating custom queries, the Performance Monitor data model appears in the form of an entity relationship diagram (ERD) that is posted on My Oracle Support. Refer to the PeopleTools Release Notes for this release for the current location of the Performance Monitor ERD.

See Performance Monitor Database Schema and Use Cases on My Oracle Support

---

**Note.** The Performance Monitor database schema may change in future releases.

---

To view the results of sample queries running against the Performance Monitor tables, select PeopleTools, Performance Monitor, History, Sample Queries. To view the definitions and SQL of these sample queries, use PeopleSoft Query Manager. The sample queries attempt to show a realistic query while using all of the tables that you may want to include in similar queries.

The sample query definitions are:

<b>Query</b>	<b>Description</b>
PPM_COMP_BUILD_CACHE	This query returns all application server requests for a specific system that had to retrieve metadata from the database as opposed to the cache. It also shows the file cache and memory cache for comparison. This query returns information from the PMU history table.
PPM_COMP_BUILD_CACHE_ARCH	This query is similar to PPM_COMP_BUILD_CACHE, except that it returns information from the PMU archive table.

<b>Query</b>	<b>Description</b>
PPM_TIMEOUT_SQL_REQ	This query returns information from the PMU history table while joining information that is stored in the event table. This query retrieves all PMU 400s (Tuxedo Service PCode and SQL) that were running SQL statements when an Event 500 (Jolt Service Exception) was received. It is assumed that this exception occurred because of a timeout, but it could also have been due to an application server outage or a Jolt error.
PPM_TIMEOUT_SQL_REQ_ARCH	This query is similar to PPM_TIMEOUT_SQL_REQ, except that it returns information from the PMU and Event archive tables.
PPM_APPSrv_START_COUNTS	This query returns starting counts for different server processes over a period of time for a specific domain.
PPM_APPSrv_START_COUNTS_ARCH	This query is similar to PPM_APPSrv_START_COUNTS, except that it fetches information from the event archive table.

---

**Note.** When you are running a sample query, the system prompts you to enter a date. The format for the date is MM/DD/YYYY HH:MM:SS AM/PM. For example, 09/03/2003 12:00:01AM.

---

## Disabling Performance Monitor Agents

In some cases, you may not want the Performance Monitor agents to run or to be a possible factor in your online system. For example, if you have ten application server domains running against the same database, you may want only one application server domain reporting information to the monitoring system.

Agents in a domain whose monitor URL is "NONE" do not collect or transmit performance data. However, they periodically check the URL for changes. Disabling a domain prevents this small portion of Performance Monitor related processing from occurring. To prevent *any* information from being sent over the network, set the monitor URL to NONE and reboot *all* monitored domains. To completely disable monitor agents on your domains, deselect the Enable PPM Agents parameter and reboot *all* monitored domains.

**Application Server and Process Scheduler Domains** For application server and Process Scheduler domains, you disable the monitor agents using the EnablePPM Agents parameter. This parameter is in the PSTOOLS section of PSADMIN. To disable the monitor agents, set the value to 0. To enable the monitor agents, set the value to 1.

Reboot the application server domain for the change to take effect.

When disabled, the Monitor URL is ignored by that domain.

**Web Server** For the web server, you disable agents by deselecting the Enable PPM Agents option in the Web Profile interface.

Reboot the web server for the change to take effect.

---

## Working with Performance Monitor Web Profile Properties

This section alerts you to important web profile properties that are related to the Performance Monitor.

<b>Enable PPM Agent</b>	Enables a web server agent to be started on a monitored web server.
<b>PPM Monitor Buffer Size</b>	Sets the maximum buffer size for the buffer that is used by the monitor servlet for incoming performance data. The default size is 51200 KB (50 MB). If you notice in servlet trace files or other warnings that you regularly see buffer overflows, you may consider increasing this value.
<b>Trace Monitoring Server</b>	Located on the Debugging tab. Enables you to trace the ppmi servlet and the monitor servlet. The system writes the trace information to the web server log file.
<b>Trace PPM Agent</b>	Located on the Debugging tab. Enables you to trace web server agents on a monitored system. You enable this option on the web server of the monitored system.

---

## Tracing Performance Monitor Agents

You can set up tracing on the:

- Application server and Process Scheduler servers.
- Web server.

### Enabling Tracing on the Application Server or Process Scheduler Server

To enable tracing of the monitor agents on application server and Process Scheduler domains, use the TracePPM parameter in PSADMIN on the application server running on the monitored system. Set TracePPM to *1*. To disable, set to *0*.

When enabled, the agents write debug information on monitored systems to a log file in the application server LOGS directory. To view the information, open TRACEPPM\_mmdd.LOG. The LogFence setting for application server logs has no effect on this file. Error messages (such as those that are created when the monitor URL can't be reached) go directly to the APPSRV\_mmdd.LOG.

### Enabling Tracing on the Web Server

To enable tracing of the monitor agents on the web server, select Trace PPM Agent on the Debugging tab in the appropriate web profile on the monitored system.

The agents write tracing information to the web server log file.

---

## Tracing the Monitor and PPMI Servlets

In some cases, you may want to view the activity of the monitor and PPMI servlets that are running on the web server on the monitoring system.

To enable tracing for the Performance Monitor servlets, you select the Trace Monitoring Server option on the Debugging tab in the appropriate web profile. The system writes the trace results to the web server log file.

---

## Configuring Performance Monitoring Sampling Rate

To reduce the overhead that is incurred by monitoring performance, you may not want to monitor every request that is submitted to your system. You can set a sampling rate for your monitored systems so that only one out of every N server requests generate PMUs. For the Nth request, all PMUs are generated at the filter level that is set for each agent type involved in processing the request. Examples of server requests would be browser requests to a web server or Application Designer requests to an application server (when running in a three-tier configuration). You set the sampling rate for PMUs using the Agent PMU Sample Rate option on the System Definitions page.

---

**Note.** This sampling rate applies only to PMUs, not events.

---

For example, if you set the sampling rate to 1/10, the system monitors the first PIA request, but does not monitor another request until the 11th request arrives at the system.

Some PMUs are always monitored regardless of the sampling rate. The PMUs that are never ignored are those that have the Enable Sampling option *cleared* on the PMU Definitions page. Examples of such PMUs are those related to users signing on, signing off, and being redirected to other sites.

---

**Note.** Setting the sampling rate to 0 (zero) disables sampling.

---

The web server and the application server maintain separate counters. The web server counts all browser requests, and the application server counts all requests that are submitted directly to the application server, such as component interfaces or Microsoft Windows workstations running Application Designer.

### See Also

[Chapter 10, "Performance Monitor Meta-Data," PMU Definitions, page 155](#)

[Chapter 4, "Administering the Performance Monitor," Modifying System Definitions, page 31](#)

---

## Viewing Monitor Servlet Diagnostics

You can view diagnostic information that is related to the monitor servlet by accessing the servlet using the following URLs.

<b>URL</b>	<b>Description</b>
http://<host>/monitor/<site>/?cmd=agents	Reveals additional statistics about the agents sending data to the monitor servlet.
http://<host>/monitor/<site>/?cmd=ppmiclients	Reveals additional statistics about the PPMI clients receiving data from the monitor servlet.

---

**Note.** To provide access to this interface, you must enable the PPMConsole on the Global Administration page.

---

See [Chapter 4, "Administering the Performance Monitor," Setting Global System Options, page 25.](#)

## Monitoring Agents

*Agents* refers to the agents on various monitored systems that send performance data to the current monitor servlet.

The system retrieves the agent information from the monitor's cache. If an administrator has changed any agent settings and clicked Save and Notify Agents on the System Definitions page, the agent information temporarily disappears in the Agents grid. Updated agent settings appear in the Agents grid after the agent communicates with monitor servlet.

---

**Note.** You identify the monitored system, using the system ID (PeopleSoft GUID) appearing just above each grid. To identify the agents, you need to map the system GUID and agent ID with the definitions in the monitoring database.

---



---

**Note.** Agents appear in the grid if they have successfully registered. The appearance of an agent does not imply that data from the agent is being successfully inserted into the monitoring database.

---

The following information appears in the Agents grid.

<b>ID</b>	The agent ID that uniquely identifies an agent within a monitored system.
<b>Last Comm</b>	The last time the agent contacted the monitor servlet.
<b>Filter</b>	The current agent filter level.
<b>Buf-Size</b> (buffer size)	The current maximum buffer size for the agent that is specified in the system definition.
<b>Send-Itvl</b> (send interval)	The current agent buffering interval.
<b>Heartbeat</b>	The current agent heartbeat interval
<b>Sample-Itvl</b> (sample interval)	The current agent event sample rate.
<b>User Trace</b>	Indicates whether performance trace is allowed for this agent.

<b>Sampling Rate</b>	The current PMU sample rate.
<b>Sampling Filter</b>	This column is reserved for future use.

## Monitoring PPMI Clients

*PPMI clients* refer to the PSPPMSRV server processes that interact with the monitor servlet. The Clients grid shows all known PPMI clients.

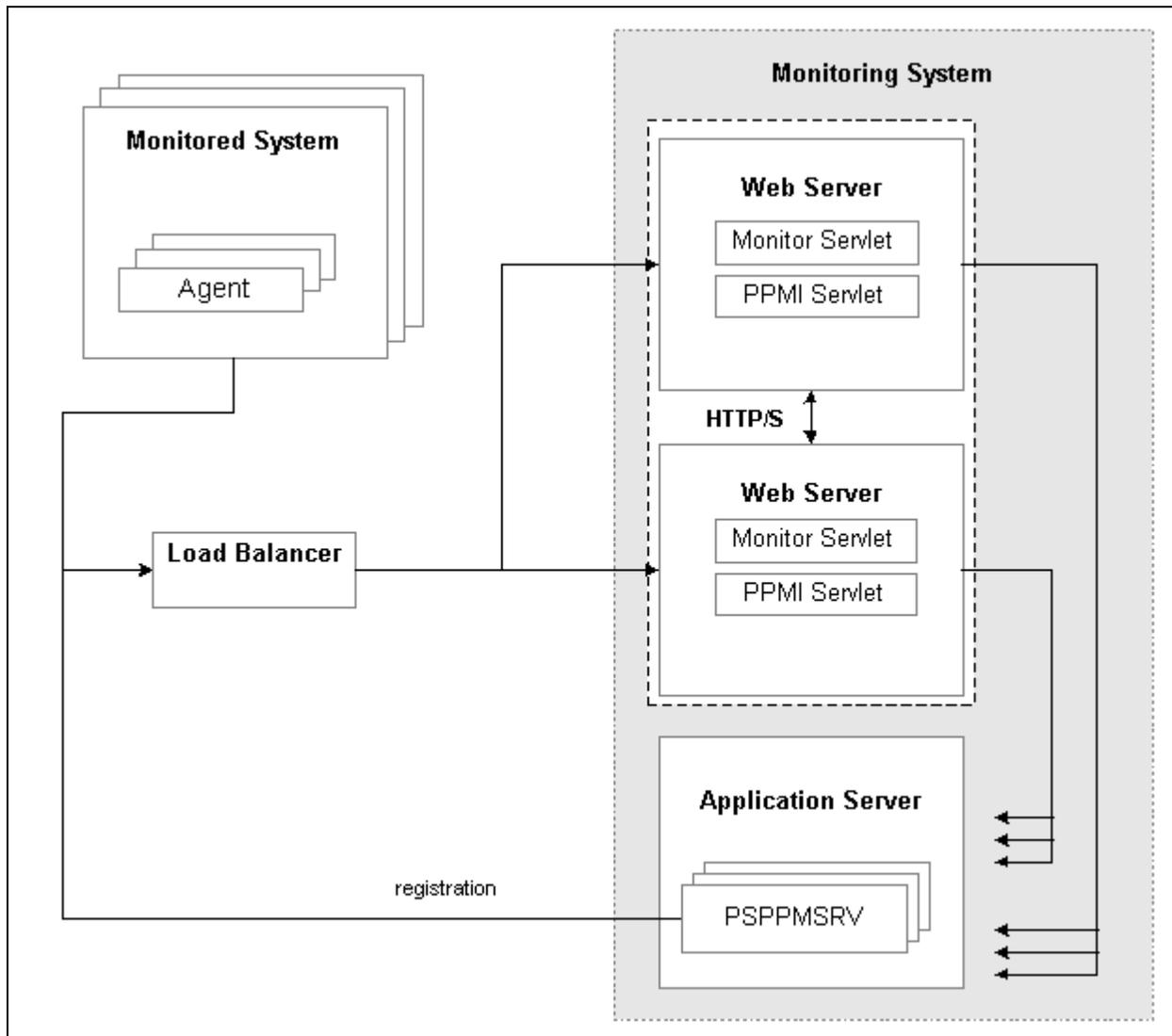
The following information appears in the Clients grid.

<b>Group</b>	The system's unique identifier (PeopleSoft GUID).
<b>ID</b>	The internal ID assigned to this PSPPMSRV.
<b>URL</b>	The IP address of the PSPPMSRV.
<b>Queue Length</b>	The number of PMUs and events not yet sent to this PSPPMSRV.
<b>Estimated Queue Size</b>	The estimated size in bytes of the PMUs and events not yet sent to this PSPPMSRV.
<b>Item Processed</b>	The number of PMUs and events sent across this PSPPMSRV connection.
<b>Estimated Bytes Processed</b>	The estimated size in bytes of the PMUs and events sent across this PSPPMSRV connection.
<b>Max Size</b>	The maximum buffer size in bytes for the PMU and event queue reached in the lifetime of this connection.
<b>Running Avg Size</b> (running average size)	The running average of the estimated size in bytes of the PMUs and events not yet sent to this PSPPMSRV.
<b>Limit</b>	The maximum buffer size in bytes for the PMU and event queue. Data is discarded when this limit is reached.

---

## Setting Up Monitor Clusters

The following diagram depicts the relationship between the elements involved in a monitor cluster. In this diagram, the cluster contains two web servers.



### Monitor cluster elements

When implementing a monitor cluster, keep these items in mind:

- A cluster must be accessed through an external third party load balancer.
- The web servers in a monitor cluster share PSPPMRSRV registration information using HTTP/S.
- Each monitor servlet (one in each web server) load balances performance information across *all* PSPPMRSRVs.
- The host and port of the PPMI URL for a clustered environment need to be set to reflect the host and port of the load balancer.

---

**Note.** External load balancers should ensure that performance information that is related to one agent is always sent to the same monitoring servlet. When sending performance data to the monitor, agents add their agent ID to the monitor URL. For example, for agent 8, the URL appears as `http://host1/monitor/ps/8`. The system administrator should set up a "sticky rule" on the load balancer so that requests from the same agent are always directed to the same web server, when available. If the sticky rule is not in place, a PMU stop time may be inserted into the monitoring database before the corresponding start time. This creates misleading open PMU information and more work for the reaper program.

---

See [Chapter 9, "Performance Monitor Security Considerations," page 143](#).

---

**Note.** If a cluster member shuts down, all performance data that is currently queued on that cluster member for transmission to a PSPMSRV is lost.

---

To set up a performance monitor cluster:

1. In the monitoring system, use the host and port of the load balancer in the PPMI URL on the Global Administration page.
2. In the monitoring system, enter the URLs of each load-balanced host in the Performance Monitor Cluster grid on the Global Administration page.

The format of the Member Servlet URL is:

```
http[s]://host/ppmi/ps
```

Where *ps* is the name of your PeopleSoft site, and *host* is the real host and port of the host on which your cluster member is running. Even though you enter *ppmi* as the servlet name, failover and scalability are implemented for both the PPMI and the monitor servlets from each site.

3. In the monitored system, use the host and port of the load balancer in the monitor URL on the Specify Monitor page.

### **See Also**

[Chapter 4, "Administering the Performance Monitor," Setting Global System Options, page 25](#)

[Chapter 3, "Setting Up the Performance Monitor," Specifying the Monitor URL, page 20](#)

---

## **Using Performance Monitor Data Mover Scripts**

PeopleSoft delivers a set of PeopleSoft Data Mover scripts for use in the administration of the Performance Monitor. The scripts are located in the following directory:

```
PS_HOME\scripts
```

The delivered scripts are described below.

- perfmondataexport.dms** Enables you to export data from your monitoring database.
- The data can be exported based on a specific system ID, between specific dates and times, or based on a specific performance trace. The default export is based on a specific system ID. The dat file that is created is named perfdata.dat.
- If you need to export performance data from a specific date and time, a performance trace, or information on all monitored systems, then open the script and edit the script as described in the comments within the script.
- perfmondataimport.dms** Enables you to import data from perfdata.dat, which is created by the perfmondataexport.dms script, into your monitoring database.
- 
- Warning!** Do not run this file on a live monitoring system because current data may be lost. The script contains the REPLACE\_DATA \* command.
- 
- PerfmonPurgeAll.dms** Enables you to purge all Performance Monitor tables in the monitoring database.
- 
- Note.** This script deletes both system definitions and all performance data that are associated with *any* monitored system.
- 
- Warning!** Shut down the monitoring system before running this script.
- 

## Estimating Your Performance Database Size

This section provides an overview of estimating your performance database size overview and discusses how to:

- Estimate space requirements for event data.
- Estimate space requirements for PMU data.
- Calculate space requirements.

### Estimating Your Performance Database Size Overview

Because performance monitoring can store a significant amount of data in your performance database, you may want to estimate the amount of data to be stored in your performance database so that you have, or are able to provide, the appropriate amount of space.

Performance database sizing estimates are based on the sum of space requirements for events and performance measurement unit (PMU) performance data. Event data resides in the PSPMEVENTHIST table. PMU data resides in the PSPMTRANSHIST and PSPMTRANSCURR tables.

This section presents formulas that you can use to estimate the potential size of your performance database.

These formulas incorporate the following assumptions and considerations:

- Performance Monitor is set to Standard agent filter mode.

- Estimates do not include space required for Verbose or Debug agent filter mode.
- Index overhead is included in the estimate.
- Performance history data is purged after a number of days (see parameter D in the following section).
- The archive mode of a monitored system is set to Delete Data after N days and the performance data archive program is scheduled to run daily.
- The reaper program is scheduled to run at least once a day.
- No performance data is stored in the archive tables (PSPMTRANSARCH and PSPMEVENTARCH).
- The calculation formulas use only the parameters that are presented in this section to calculate the estimates. In all cases, the numbers are conservative. For example, the exact formula may use  $(App - 1)$  but we choose to round up and use App instead.

## Estimating Space Requirements for Event Data

This section discusses two formulas that are used for estimating event data space requirements (in kilobytes).

- Standard formula: Use the standard formula if the application server domain configuration is based on PeopleSoft-delivered small, medium, or large template.
- Customized formula: Use the customized formula if the configuration is different from the templates.

This table describes the variables that are used in the formulas.

<b><i>Notation</i></b>	<b><i>Description</i></b>	<b><i>Performance Monitor Default Value</i></b>	<b><i>Navigation</i></b>
A	Performance Monitor agent event sampling rate.	300 seconds	PeopleTools, Performance Monitor, Administration, System Definitions
N	Number of PeopleSoft systems that are monitored by Performance Monitor. This is the number of PeopleSoft databases appearing on the System Definitions search page.	NA	PeopleTools, Performance Monitor, Administration, System Definitions
D	Performance history retention period in days. This is the value that is set for the After N days option.	NA	PeopleTools, Performance Monitor, Administration, System Definitions
W	Number of web server domains for a monitored system. This is the total number of web servers appearing on the System Performance page.	NA	PeopleTools, Performance Monitor, System Monitor, System Performance

<b>Notation</b>	<b>Description</b>	<b>Performance Monitor Default Value</b>	<b>Navigation</b>
P	Number of application server domains for a monitored system. This is the total number of application servers appearing on the System Performance page.	NA	PeopleTools, Performance Monitor, System Monitor, System Performance
App	Number of server processes running in an application server domain for a monitored system. This is the number of program names appearing on for Server Status.	Use the following number per domain template that you choose: <ul style="list-style-type: none"> <li>• Large = 60</li> <li>• Medium = 40</li> <li>• Small = 20</li> </ul>	PSADMIN, Application Server, Administer a domain, Domain, Domain status menu, Server status
S	Number of monitored PeopleSoft Process Scheduler domains for a monitored system. This is the total number of Process Scheduler domains appearing on the System Performance page.	NA	PeopleTools, Performance Monitor, System Monitor, System Performance
Prcs	Number of server processes running in a PeopleSoft Process Scheduler domain for a monitored system. This is the number of program names appearing for Server status.	8 Increase this number if more than three Application Engine processes are configured.	PSADMIN, Process Scheduler, Show Status of a Process Scheduler Server, Domain, Domain status menu, Server status
MPrCs	Number of Master Scheduler for a monitored system.	1	NA
E	Number of KB per event row in the table.	Refer to the value in the following table for the target database.	NA

The following table helps you to determine the appropriate value for E (Number of KB per event row in the table).

<b>Parameter</b>	<b>ANSI/Unicode</b>	<b>Oracle</b>	<b>Microsoft SQL Server</b>	<b>DB2 UDB</b>	<b>DB2/390</b>
E	ANSI	.4	.4	.5	.6
E	Unicode	.7	.7	.9	1.1

### **Using the Standard Formula**

The formula that you use differs depending on the template used in the application server configuration.

**Large**  $N \times D \times [8 \times W + 180 \times P + 16 \times S + 1] \times 86400 / A \times E$

**Medium**  $N \times D \times [8 \times W + 120 \times P + 16 \times S + 1] \times 86400 / A \times E$

**Small**  $N \times D \times [8 \times W + 60 \times P + 16 \times S + 1] \times 86400 / A \times E$

### **Using the Customized Formula**

Use this formula if the application server configuration is different from the standard templates.

$N \times D \times [8 \times W + 3 \times P \times \text{App} + 2 \times S \times \text{PrCs} + \text{MPrCs}] \times 86400 / A \times E$

**Note.** Eight events are reported per web server domain. Two events are reported per web server (JVM status and network status), one event per web site, and five events per web site for PeopleSoft servlets (psp, psc, cs, \*, and Scheduler Transfer).

If multiple systems are monitored and each is configured slightly differently, that is, the numbers of application server processes are different, then use the formula to estimate the requirement for each system separately.

## **Estimating Space Requirements for PMU Data**

The total space requirements, in kilobytes, for PMU data that is stored in PSPMTRANSHIST and PSPMTRANSCURR tables can be estimated using the following formula:

$N \times [D + 1] \times L \times R \times M \times T$

This table describes the variables that are used in this formula.

<b>Notation</b>	<b>Description</b>	<b>Performance Monitor Value Default</b>	<b>Navigation</b>
N	Number of PeopleSoft systems that are monitored by Performance Monitor. This is the number of PeopleSoft databases appearing on the System Definitions search page.	NA	PeopleTools, Performance Monitor, Administration, System Definitions
D	Performance history retention period in days. This is the value that is set for the After N days option.	NA	PeopleTools, Performance Monitor, Administration, System Definitions
L	Number of user sessions per day for a monitored system. A session means that a user signs on, performs a few transactions, and signs off.	NA	NA

<b>Notation</b>	<b>Description</b>	<b>Performance Monitor Value Default</b>	<b>Navigation</b>
R	Number of user interactions per session. User interactions are anything that triggers a server trip, including clicking a button, clicking TAB, and so on.	NA	NA
M	Number of PMU rows that are captured per user interaction.	9	NA
T	Number of KB per PMU row in the table.	Refer to the value in the following table for the target database.	NA

The following table helps you to determine the value for T.

<b>Parameter</b>	<b>ANSI/Unicode</b>	<b>Oracle</b>	<b>Microsoft SQL Server</b>	<b>DB2 UDB</b>	<b>DB2/390</b>
T	ANSI	1.3	1.4	1.8	2.1
T	Unicode	2.4	2.6	3.3	3.8

## Calculating Space Requirements

This section presents an example of using the formulas to estimate the performance database size for a fictitious organization.

Company ABC uses Performance Monitor to monitor two PeopleSoft Applications, Financials and HCM (N=2). Both applications use DB2 UDB Unicode databases. The company has decided that the performance history data will be kept for a 7-day period (D=7). Each system has two web server domains (W=2), two application server domains (P=2), and two PeopleSoft Process Scheduler domains (S=2). The implementation team decides to use the medium application server configuration for both domains. One Master Scheduler exists for each of the systems (MPrcs=1).

It is estimated, on the average, that 10,000 user sessions (L=10000) will be logged per day in each of the systems. During each session, 50 user interactions (clicking buttons, tab to next field or page, and so on) will occur (R=50).

This is the sample calculation for event data space (using the standard formula for a medium configuration):

$$\begin{aligned}
 & N \times D \times [8 \times W + 120 \times P + 16 \times S + 1] \times 86400 / A \times E \\
 &= 2 \times 7 \times [8 \times 2 + 120 \times 2 + 16 \times 2 + 1] \times 86400 / 300 \times E \\
 &= 1,165,248 \text{ rows} \times 0.9 \text{ KB per row} \\
 &= 1,024 \text{ MB}
 \end{aligned}$$

This is the sample calculation for PMU data space:

$$N \times [D + 1] \times L \times R \times M \times T$$

$$\begin{aligned}
 &= 2 \times [7 + 1] \times 10000 \times 50 \times 9 \times T \\
 &= 72,000,000 \text{ rows} \times 3.3 \text{ KB per row} \\
 &= 232,032 \text{ MB}
 \end{aligned}$$

This is the formula for space requirement for storing performance data on a DB2 UDB Unicode database:

$$1,024 \text{ MB} + 232,032 \text{ MB} = 233,056 \text{ MB}$$

Company ABC decides to add a 1 TB disk.

Business is going well for ABC Company. The demand for the Financial application increased by 50 percent. The IT department decided to add new web server, application server, and PeopleSoft Process Scheduler domains for the Financials application (N=1). According to the system administrator, when the application server domain is booted, a "22 processes started" message appears (App=22), and the PeopleSoft Process Scheduler domain shows a "12 processes started" message (PrCs=12). The IT department needs to estimate whether enough disk space is available to store additional performance data.

Use the customized formula to calculate the space requirement for event data that is generated by the new configuration.

$$\begin{aligned}
 &N \times D \times [8 \times W + 3 \times P \times \text{App} + 2 \times S \times \text{PrCs} + \text{MPrCs}] \times 86400 / A \times E \\
 &= 1 \times 7 \times [8 \times 1 + 3 \times 1 \times 22 + 2 \times 1 \times 12] \times 86400 / 300 \times E \\
 &= 153,216 \text{ rows} \times 3.3 \text{ KB per row} \\
 &= 494 \text{ MB}
 \end{aligned}$$

System usage increased by 50 percent for the Financials application, so the total space requirement is:

$$\begin{aligned}
 &233,056 \text{ MB} + 494 \text{ MB} + [232,032 \text{ MB}/2 \times 50\%] \\
 &= 291,558 \text{ MB}
 \end{aligned}$$

The IT department concludes that enough space is available to store the performance data.

## Chapter 5

# Working with the Performance Trace

This chapter provides an overview of the Performance Trace and discusses how to:

- Enable a performance trace.
- Run a performance trace.
- View performance trace information.

---

## Understanding the Performance Trace

The Performance Monitor enables you to monitor and record performance information for all activity on PeopleSoft systems. However, times occur when you need to monitor the performance of a specific business process or the performance issues that are reported by a specific user. In these cases, you can use a performance trace.

The Performance Trace enables you to:

- Group PMUs across server requests.
- Display PMUs from multiple systems.
- Override default agent filter levels.

A user starts and stops a performance trace from the Performance Trace console. While a performance trace is in effect, the system associates the trace name with each PMU that is created during that user's session. The trace name may then be used in the Performance Monitor pages to search for performance data that is created during the performance trace.

The trace can override the current agent filter level for PMUs that are created during the trace. The override applies to that user session only.

Performance traces are effective across multiple systems that report to the same monitor. For example, if a portal displays content from another monitored system, traces that are started from the portal also apply to the displayed content.

---

**Note.** The Performance Trace Console needs to be launched from an end user's browser.

---

---

**Note.** A performance trace is not affected by the PMU sample rate.

---

---

## Enabling a Performance Trace

Systems that are configured to allow performance traces have a Performance Trace link appearing in the Universal Navigation Header on each PeopleSoft page.

---

**Note.** In the following procedure, be sure to notice where each step needs to be completed: on the *monitoring* system or the *monitored* system.

---

To enable a performance trace:

1. On the monitoring system, select PeopleTools, Performance Monitor, Administration, System Definitions, and then select the appropriate system definition.
2. On the monitoring system, make sure Allow Performance Trace is selected on the System Definitions page.

---

**Note.** If a business process includes multiple systems, each system in the business process needs to have the system configured to allow performance traces.

---

3. On the monitoring system, click Save and Notify Agents.
4. On the *monitored* system, make sure the users who will run the performance trace have the WEBLIB\_PPM web library specified in at least one of their permission lists.

---

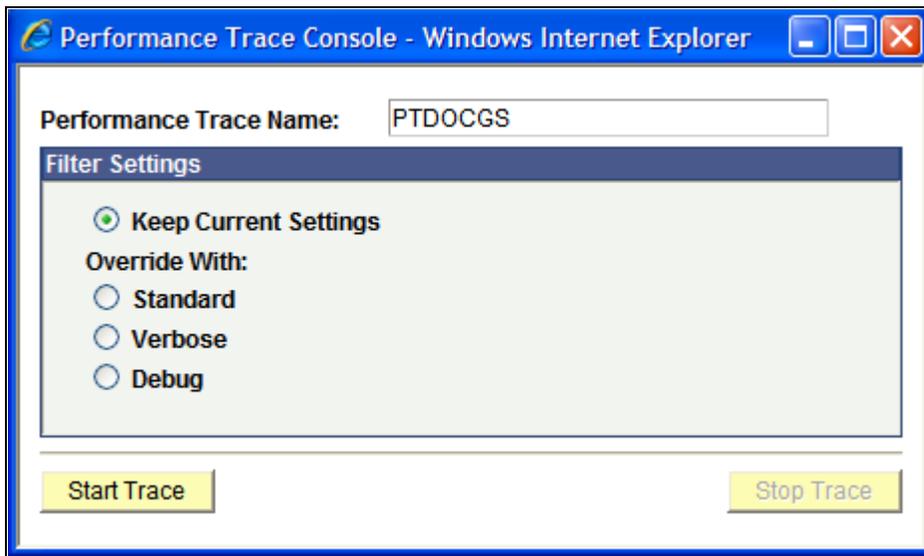
**Note.** If you don't want a user to see the Performance Trace link, you need to remove the web library WEBLIB\_PPM from that user's permission lists. Without access to WEBLIB\_PPM, the Performance Console link does not appear in the Universal Navigation Header.

---

---

## Running a Performance Trace

The Performance Trace Console launches in a small, separate window.



### Performance Trace Console

To run a performance trace:

1. If the Performance Trace link does not appear in the Universal Navigation Header, click Sign out to sign off the system, and then clear the browser's cache (remove temporary Internet files).

---

**Note.** Clearing browser cache varies depending on which supported browser you use. Refer to your browser's documentation for details.

---

2. Sign on to the PeopleSoft system.
3. Navigate to the page where you want to begin the performance trace.
4. Click the Performance Trace link, which is located in the Universal Navigation Header.

This launches the Performance Trace Console.

5. In the Performance Trace Name edit box, enter a name to identify the trace results.

Remember the name that you enter because you need to enter this name in the search criteria when you intend to view the trace results. You can accept the default name, if one appears.

6. In the Filter Level group box, set the filter level options.

You have the following options:

**Keep Current Settings** The system continues to record performance information according to the current agent filter settings.

**Override With Standard** Changes the current filter level to Standard for the current user's session.

**Override With Verbose** Changes the current filter level to Verbose for the current user's session.

**Override With Debug** Changes the current filter level to Debug for the current user's session.

7. When the appropriate filter levels have been set, click Start Trace.
8. Complete the business process that you need to monitor.
9. Click Stop Trace.

---

## Viewing Performance Trace Information

To view performance trace information:

1. Make note of the performance trace name that was entered by the user.
2. Navigate to the Performance Monitor page that displays the desired performance information.

You can use the performance trace name to view.

- Open PMUs.
  - Completed PMUs.
  - All of the performance charts.
3. In the search criteria for the page that you are on, enter the name of the performance that is trace in the Performance Trace Name edit box.
  4. Enter any additional search criteria required.
  5. Click Search.

## Chapter 6

# Monitoring System Performance

This chapter provides an overview of system performance monitoring and discusses how to:

- Use the system performance home page.
- View web server performance.
- View application server domain performance.
- View Process Scheduler server performance.
- View Master Scheduler server performance.
- View open PMUs.
- View open PMU trees.
- View current user sessions.
- View recycled processes diagnosis.
- Understand the events that monitor resource usage.

---

## System Performance Monitoring

This section contains an overview of monitoring system performance and lists the pages that are used to monitor system performance.

### Monitoring System Performance

The activities that are related to monitoring system performance are primarily for viewing and analyzing the most recent performance data that is received from agents in a monitored system.

PeopleSoft provides a collection of use cases in the form of flow charts for you to use as a framework for learning how to use Performance Monitor to detect performance issues. These flow charts do not appear in this PeopleBook; they are posted on My Oracle Support. Refer to the PeopleTools Release Notes for the current location of these flow charts.

See [Performance Monitor Database Schema and Use Cases on My Oracle Support](#)

---

**Note.** The information that is presented in the System Performance pages is as current as the last page refresh.

---

**Note.** The Standard Deviation (Std. Dev.) is provided on many pages. The standard deviation is a statistic that tells you how tightly all the values that are used to compute the average are clustered around the average. Large standard deviations warn that the averages appearing in the chart are not a reliable indicator of response times that are experienced by individual users.

## Pages Used to Monitor System Performance

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
System Performance	PSPMSYSHEALTH	PeopleTools, Performance Monitor, System Monitor, System Performance	Provides a high-level view of the performance of a monitored system displaying performance indices as well as some critical event and PMU data for the servers in your monitored system. This page acts as a "home page" for monitoring system health. For example, from the System Performance page, you can access numerous related pages, such as the Web Server page, Application Server page, the Process Scheduler page, and so on.
Web Server	PSPMWEBDOM	PeopleTools, Performance Monitor, System Monitor, Web Server	Enables you to drill down into the performance data that is related to your web server, such as JVM Status, network status, site performance, and servlet performance.
Application Server	PSPMAPPDOM	PeopleTools, Performance Monitor, System Monitor, Application Server	Enables you to drill down into the status of an application server domain and the individual server processes that are running in that domain.
Process Scheduler Server	PSPMPSCHEDDOM	PeopleTools, Performance Monitor, System Monitor, Process Scheduler Server	Enables you to drill down into the status of a Process Scheduler domain, monitor the individual server processes within a domain, and monitor the resource usage.

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Master Scheduler	PSPMMASTSCHED	PeopleTools, Performance Monitor, System Monitor, Master Scheduler	Enables you to view the status of the Master Scheduler distributing workload across multiple Process Schedulers.
Open PMUs	PSPMTRANSCURR	PeopleTools, Performance Monitor, System Monitor, Open PMUs	Provides search criteria by which you can search for open PMUs, which are PMUs that have not yet finished.
Open PMU Trees	PSPMTRANSUSER	PeopleTools, Performance Monitor, System Monitor, Open PMU Trees	Enables you to search open PMUs by user and display each open PMU in a tree format.
Current User Sessions	PSPMCURUSERS	PeopleTools, Performance Monitor, System Health, Current User Sessions	Enables you to view the users who are currently signed on to the system as well as view an individual user's history.
Recycled Processes Diagnosis	PT_PM_RECPROC_DIAG	PeopleTools, Performance Monitor, System Health, Recycled Processes Diagnosis	Shows detail about application server crashes and last service request type.

---

## Using the System Performance Home Page

Access the System Performance page (PeopleTools, Performance Monitor, System Monitor, System Performance).

The System Performance page provides various health indices as well as a high-level view of the performance of your web servers, application servers, and Process Scheduler server domains. It acts as the "home page" for system performance.

---

**Note.** A Status field appears at the top of the page only if the monitoring system detects stale data, which is data that is older than the specified event sampling rate. For example, if your event sample rate is 300 seconds, and no events have been received from a specific domain in over five minutes, then that domain is considered stale. Stale data could indicate an outage on the monitored system or that the monitored system's agent filters are set to 01–Standby. If a domain has been shut down permanently and you don't want it to appear with a stale data warning, go to the Agent Definitions page and set its domain monitor to inactive.

---

## Viewing Performance Indices

<b>User Sessions</b>	<p>Displays the number of users who have signed onto the system within the last 12 hours and have not signed off.</p> <p>Shows all PMU 109s (User Session Began) that do not have an associated 108 PMU (User Session Ended). These PMUs are associated by the session ID (Context 1). The User Sessions value includes only users who have signed on in the last 12 hours and only those users that have signed on since the last time the web server agent sent an Event 901 (Agent Initialization).</p> <p>For details on individual users, click the <a href="#">Current User Sessions</a> link.</p> <hr/> <p><b>Note.</b> If users do not click the Sign out link on a PeopleSoft page and instead close the browser or navigate to another site, then the system displays them as current users until the web-server time-out value is reached.</p> <hr/>
<b>Tuxedo Request Queued</b>	<p>Displays the number of requests queued on all application servers on the monitored system.</p> <p>The value is the sum of all the metric 5s (Total pq) for Event 300s (Host Resource Status) that are sent by all active agents for the current system that are domain monitor agents of type application server. The value includes all Event 300 data that is generated within the defined sampling interval.</p>
<b>PMUs in Past Hour</b>	<p>Displays the number of PMUs that have run in the past hour. This is an indication of the load on the monitored system, the level of monitoring, and whether the monitored system agents are currently communicating with the monitor.</p> <p>The value is the sum of the rows that are inserted into the PSPMTRANS HIST table within the last hour.</p> <p>For more details, click the <a href="#">Open PMUs</a> and the <a href="#">Completed PMUs</a> link.</p>
<b>Alarms in Past Hour</b>	<p>Displays the number of error and warning events that were posted by the monitored system's active agents in the past hour.</p> <p>This is the count of all events with filter levels of error and warning inserted into the PSPMEVENTHIST table in the last hour.</p> <p>For more details, click the <a href="#">Alarm History</a> link.</p>
<b>Batch Jobs in Process</b>	<p>Displays the number of Process Scheduler jobs that are currently running on all the Process Scheduler servers belonging to that monitored system.</p> <p>The value is the sum of all metric 1s for Event 350 (Master Scheduler Status).</p> <p>For more details, click the <a href="#">Master Scheduler</a> link.</p> <hr/> <p><b>Note.</b> If a Master Scheduler is not configured for the monitored system, this value is zero.</p> <hr/>

**Batch Jobs in Queue**

Displays the number of Process Scheduler jobs that are currently waiting to be assigned by the Master Scheduler.

The value is the sum of all metric 2s for Event 350 (Master Scheduler Status).

---

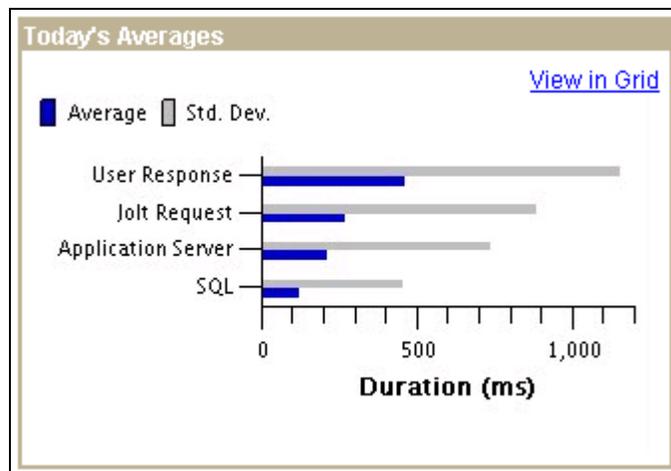
**Note.** If a Master Scheduler is not configured for the monitored system, this value is zero.

---

**Today's Averages**

The Today's Averages chart displays the average duration of various performance factors within your PeopleSoft system. The data applies only to the current day, which refers to all data that has been collected since midnight.

If progress slows in the system anywhere, this chart enables you to identify which tier is affected and enables you to monitor Tuxedo queueing.



Today's Averages chart

**User Response**

Displays the average amount of time that end users waited for server round trips to finish. A user is waiting for a server round trip to finish when *Processing* appears in the upper, right-hand corner of a PeopleSoft page.

---

**Note.** This average does not include network latency. It is the average duration of all top-level PMU 101s (PIA Request) that the monitoring system has received since midnight.

---

**Jolt Request**

Displays the average and standard deviation of all PMU 115s (Jolt Request) received since midnight.

**Application Server**

Displays the average and standard deviation of the duration of all PMU 400s (Tuxedo Service PeopleCode and SQL Summary) received since midnight.

<b>SQL</b>	<p>Displays the average and standard deviation for the time spent running SQL statements that were initiated from PeopleCode in a single application server request. This value includes the time that was required for SQL Prepare, Execute, Fetch, and Commit.</p> <p>PeopleCode SQL statements are run by application code that is written using PeopleCode and submitted to the database by way of SQLEXECs or SQL Objects.</p> <p>This value is derived from metrics 2, 3, and 4 from PMU 400 (Tuxedo Service PeopleCode and SQL Summary). Namely, this value includes PeopleCode SQL, PeopleCode Built-in SQL, and PeopleTools SQL.</p> <hr/> <p><b>Note.</b> This is the average time spent for running all SQL statements of these types in a single application server request. The system calculates this value using all data that has been received since midnight.</p> <hr/>
------------	---

### Web Server Domains

The Web Server Domains grid contains a row for each active domain-monitor web-server agent. The metric information is derived from Event 150 (JVM status) and Event 151 (Network Status). The system retrieves only the most recent event for each agent. If the most recent event's monitor date and time is older than the system's event sample rate, the Stale Data icon appears in the grid row *and* at the top of the page.

---

**Note.** Web server agents do not report performance data until the first user connects to that web server.

---

<b>Name</b>	<p>Displays the PeopleSoft web-server domain name that is being monitored.</p> <p>Click the name to go to the Web Server page for that web server. It's the same page that you access when you select PeopleTools, Performance Monitor, System Monitor, Web Server and choose that domain.</p>
<b>Host/Port</b>	<p>Identifies the web-server domain host name and listening port numbers for HTTP and HTTPS.</p>
<b>Filter Level</b>	<p>Appears on pages displaying current performance information that is related to servers, such as application servers, web servers, and so on. Displays the current agent filter level.</p> <p>The colors indicate the following filter levels:</p> <ul style="list-style-type: none"> <li>• Blue: Standby, Error, and Warning.</li> <li>• Green: Standard.</li> <li>• Orange: Verbose.</li> <li>• Red: Debug.</li> </ul> <p>Move the cursor over the icon to show a pop-up message displaying the agent filter level in text format.</p>
<b>Sessions in Web App</b>	<p>Displays the number of servlet sessions in the portal web application.</p>

<b>%JVM Memory Free</b>	A Java Virtual Machine (JVM) has a specific amount of memory allocated to it. This metric displays the percentage of the allocated memory that is available.
<b>Execute Threads</b>	The number of threads that are used by the web application to service incoming requests.
<b>Established Sockets</b>	The number of sockets in a connected state. This is a measure of the number of active users.
<b>Jolt Traffic</b>	Displays the per second amount (in bytes) of Jolt traffic between the web server and the application server.

### ***Application Server Domains/Process Scheduler Server Domains***

The application server and Process Scheduler Server grids contain a row for each active domain monitor agent that is running on the application server or Process Scheduler server. The metric information on each row is derived from Event 300 (Host Resource Status). The system retrieves only the most recent event for each agent. If the most recent event's monitor date or time is older than the system's event sample rate, the Stale Data icon appears in the grid row *and* at the top of the page.

---

**Note.** When first booted, an active server may have a stale data warning until a full sampling interval has passed.

---

<b>Domain Name</b>	<p>Displays the names of the domains that are being monitored.</p> <p>Click the domain name to access the Application Server or the Process Scheduler page for additional details on a particular domain. It's the same page that you access when you select PeopleTools, Performance Monitor, System Monitor, Application Server or Process Scheduler and choose that domain.</p>
<b>Host/Port</b>	Displays the server name or IP address and the Jolt listening port number.
<b>Filter Level</b>	<p>Appears on pages displaying current performance information that is related to servers, such as application servers, web servers, and so on. Displays the current agent filter level.</p> <p>The colors indicate the following levels:</p> <ul style="list-style-type: none"> <li>• Blue: Standby, Error, and Warning.</li> <li>• Green: Standard.</li> <li>• Orange: Verbose.</li> <li>• Red: Debug.</li> </ul> <p>Move the cursor over the icon to show a pop-up message displaying the agent filter level in text format.</p>
<b>%CPU Used</b>	Displays the percentage of the CPU capacity that is being used on the host. This percentage includes the entire processing load on the host, not just that of PeopleSoft.

<b>%Memory Used</b>	Displays the percentage of physical memory that is used on the host. This percentage includes all memory used on the host, not just the memory used by PeopleSoft.
<b>Hard Page Faults/Second</b>	This is the number of accesses to virtual memory in the last second that require disk reads on the host. This metric applies to all memory access on the host, not just the memory that is accessed by PeopleSoft.
<b>Total Tuxedo Connections</b>	This is the number of connections to this domain, and it is equivalent to the number of rows that are returned by the Tuxedo command <code>pctl (tadmin)</code> , excluding JSH and WSH connections. This value does not apply to Process Scheduler.
<b>Total Tuxedo Requests Queued</b>	The number of Tuxedo requests that are queued on that domain. A high number indicates that insufficient server processes are configured, or that the load on the host machine is too high. This value does not apply to Process Scheduler.

### ***Ping Test***

PeopleSoft Ping is a diagnostic feature that enables you to troubleshoot systemic performance issues. If you enter the ping URLs for the monitored systems in the URL catalog, you can launch the ping test from the monitoring system instead of having to sign on to the monitored system.

The following requirements must be in place:

- Single signon must be configured between the monitored and monitoring system.
- Current user ID must be a valid user ID in both the monitored and monitoring system.
- Current user ID must have permission to access the PSPing page in the monitored system.

Any system that you intend to ping must have an entry in the URL catalog for the PSPing page. The URL must contain `PTPERF_TEST` for the URL to appear in the ping lists that are associated with the Performance Monitor. For example,

```
http://server_name/psp/ps[_newwin]/EMPLOYEE/PT_LOCAL/c/UTILITIES.PTPERF_TEST.GBL
```

---

**Note.** In a self-monitoring system, you must append the text "`_newwin`" to the end of the site name on which you are running the ping test. The "`_newwin`" indicates to the PeopleSoft system that the content can run in a new window at the same time that content from a different component runs in the previous window.

---

**URL Identifier**                      Select the URL of the system that you want toping.

**Execute PSPing**                      Click to launch the PeopleSoft Ping page at the URL appearing in the URL Identifier field. The page appears in a new window. If that system is monitored, when you run Ping the system reports the results that are displayed on the Ping page to the monitoring system in the form of Event 600s (PSPING).

<b>View PSPing History</b>	<p>Launches the Event History search page displaying all Event 600 (PSPING) data for the current day on the current system.</p> <p>Verify that you ping the appropriate system. PeopleSoft runs no internal verification to verify that the system you ping is the system which this page is monitoring.</p>
----------------------------	--

---

## Viewing Web Server Performance

Access the Web Server page (PeopleTools, Performance Monitor, System Monitor, Web Server).

The Web Server page displays the most recent performance data that is received from a web server domain monitor. A domain corresponds to the domain that is specified during the PeopleSoft installation.

**Domain Name** A domain monitor starts when the first user connects to any site in a PeopleSoft web server domain. The domain name is the name of this site.

**Domain Directory** The directory on the web server where the PeopleSoft site that triggered the domain monitor resides.

**Filter Level** Appears on pages displaying current performance information that is related to servers, such as application servers, web servers, and so on. Displays the current agent filter level.

The colors indicate the following levels:

- Blue: Standby, Error, and Warning.
- Green: Standard.
- Orange: Verbose.
- Red: Debug.

Move the cursor over the icon to show a pop-up message displaying the agent filter level in text format.

**Agent Date/Time** The date and time according to the agent clock that the domain monitor used when it last sent performance data to the monitor.

---

**Note.** The date and the time appearing on this page always apply to the system on which the agent runs. Keep this in mind if you are monitoring systems in other time zones.

---

**Status** Appears only if the system detects stale data. Performance data is stale if no status events have been reported in the last sample interval for the system. For example, Status shows stale data if the web server is not booted.

**Host/Port** The host of the web server and the listening ports for HTTP and HTTPS.

<b>Monitor Date/Time</b>	The date and time that the monitoring system inserted the row of performance data into the monitoring database according to the database clock.
<b>Last Page Refresh</b>	Indicates the most recent time that the system refreshed the page either by loading the page into the browser or as a result of a user clicking the Refresh button.
<b>Web Server Snapshot</b>	This link enables you to view historical information snapshots for the current agent. By clicking this link, you see a list of all the domain agent timestamps for the current domain monitor agent Event 150 (JVM Status). After you select a particular timestamp, the system displays the Web Server page containing web server performance information for that particular time. When you are viewing snapshot information, a message appears at the top of the page reminding you that the page contains historical performance data. You can view numerous snapshots using the same method.
	<hr/> <b>Note.</b> While viewing snapshots, the Refresh button is not available. To view the most current web server information, you must access the Web Server page using the menu. <hr/>
<b>System Performance</b>	This link accesses the System Performance page (the home page for system monitoring).

### **JVM Status**

The JVM status applies to the status of the JVM in which the site runs. Multiple web sites within the same PeopleSoft domain can run within the same JVM.

These values are derived from Event 150 (JVM Status). If the latest Event 150 monitor date and time is older than the event sample rate for the current agent's system, the Stale Data icon appears at the top of the page.

<b>%JVM Memory Free</b>	A JVM has a specific amount of memory allocated to it. This metric displays the percentage of the allocated memory that is available.
<b>Max JVM Memory Available</b>	Indicates the maximum amount of memory that is available for the JVM. This value depends on the Java Runtime Environment (JRE).  The maximum JVM memory that is available equals $\text{maxMemory} - \text{totalMemory} + \text{freeMemory}$ .
<b>Sessions in Web-App</b>	Displays the number of servlet sessions in the portal web application.
<b>Execute Threads</b>	The number of threads that are used by the web application to service incoming requests.
<b>Busy Threads</b>	The number of threads that are currently servicing requests.
<b>Domain Count</b>	The number of domain monitors in this web server domain. The domain count will only be greater than 1 if a web server domain has sites belonging to different monitored systems.

The Events and PMUs tab contains links to the Event History, Open PMUs, and Completed PMUs components.

When transferring to the Event History component, the system automatically displays all of the data for the current day's Event 150 (JVM Status) rows for the current agent and the current system.

When transferring to the Open PMUs component, the system automatically displays all of the data for the current day's open PMUs for the current host/port and the current system.

When transferring to the Completed PMUs component, the system automatically displays all of the data for the current day's completed PMUs for the current host/port and the current system.

## **Network Status**

Network status data corresponds to

- Output from the domain monitor running the netstat —a command in a shell on the monitored host.
- Counters that are maintained by the domain monitor.

These values are derived from Event 151 (Network Status).

If no Event 151 or the Event 151 has an agent date/time that does not match the agent date/time of the Event 150 (JVM Status) shown at the top portion of the page, this grid appears empty.

<b>Time Wait Sockets</b>	The number of sockets in a TCP time wait state on the host on which the web server is running. A high count may mean that the time wait setting for the operating system of the server needs to be decreased, or it may just mean that a very high load is on that server.
<b>Close Wait Sockets</b>	The number of sockets in a TCP close wait state on the host on which the web server is running. A high count means that TCP clients are not closing connections and may indicate network or software configuration issues.
<b>Established Sockets</b>	The number of sockets in a connected state. This is a measure of the number of users who are connected.
<b>Jolt Traffic</b>	Displays in bytes per second the amount of Jolt traffic between the web server and the application server.
	<hr/> <b>Note.</b> The Jolt traffic value is zero unless compression is enabled. <hr/>
<b>HTTP</b>	Indicates the HTTP traffic in bytes per second that is generated by this web server.

The Events and PMUs tab contains links to the associated Event History, Open PMUs, and Completed PMUs.

When transferring to the Event History component, the system automatically displays all of the data for the current day's Event 151 (Network Status) rows for the current agent and the current system.

When transferring to the Open PMUs component, the system automatically displays all of the data for the current day's open PMUs for the current host/port and the current system.

When transferring to the Completed PMUs component, the system automatically displays all of the data for the current day's completed PMUs for the current host/port and the current system.

### **Sites**

The information on this grid applies to all sites on this web server even if they are not monitored sites.

These values are derived from Event 152 (Web Site Status).

If no Event 152 or the Event 152 rows have an agent date/time that does not match the agent date/time of the Event 150 (JVM Status), the grid appears empty.

<b>Site Path</b>	Indicates the directory on the web server where the site resides.
<b>Requests to all Servlets</b>	Indicates the number of requests that have been submitted to all of the servlets running within the site.
<b>Servlet Requests (last minute)</b>	Indicates the number of servlet requests that were submitted to the site in the last minute.
<b>Avg. Request Time (last minute)</b>	Indicates the average duration of the servlet requests that were submitted to the site in the last minute.
<b>Time in all Servlets</b>	Indicates the sum of durations of the servlet requests that were submitted to the site since the web server was last booted.
<b>Current Sessions</b>	The number of sessions that are active within the site. Active sessions are those that are currently waiting for a request to be processed. In most cases, this value is 0. However, if your site is running slowly or a transaction has a long duration, this value is greater than 0.

The Events and PMUs tab contains links to the associated Event History, Open PMUs, and Completed PMUs.

When transferring to the Event History component, the system automatically displays all of the data for the current day's Event 152 (Web Site Status) rows for the current agent and the current system.

When transferring to the Open PMUs component, the system automatically displays all of the data for the current day's open PMUs for the current agent and the current system.

When transferring to the Completed PMUs component, the system automatically displays all of the data for the current day's completed PMUs for the agent and the current system.

### **Active Servlets**

Active servlets are all servlets that are running in the portal web application.

These values are derived from Event 153 (Web Servlet Status).

<b>Servlet Name</b>	Identifies the name of a particular servlet.
<b>Requests to this Servlet</b>	Indicates the number of requests that were submitted to a particular servlet.

<b>Servlet Requests (last minute)</b>	Indicates the number of requests that were submitted to a particular servlet within the last minute.
<b>Avg. Request Time (last minute)</b>	The average duration of the requests that were handled by the servlet in the last minute.
<b>Time in this Servlet</b>	The sum of durations of requests that were handled by this servlet since the web server was last booted.

---

## Viewing Application Server Domain Performance

Access the Application Server Domain page (PeopleTools, Performance Monitor, System Monitor, Application Server).

The Application Server Domain page displays the most recent performance data that was received from:

- An application server domain monitor.
- Each active application server agent within that application server domain.

**Domain Name** Identifies the name of the current Tuxedo domain.

**Domain Directory** Identifies the directory in which the domain is installed.

**Agent Date/Time** The date and time according to the agent clock that the domain monitor used when it last sent performance data to the monitor.

---

**Note.** The date and the time appearing on this page always applies to the system on which the agent runs. Keep this in mind if you are monitoring systems in other time zones.

---

**Status** Appears only if the system detects stale data. Data is stale if no status events have been reported in the last sample interval for the system. For example, Status shows stale data if the application server is not booted.

**Host/Port** Identifies the machine name and Jolt listening port of the application server domain.

<b>Application Server Snapshots</b>	<p>This link enables you to view historical information snapshots for the current agent. By clicking this link, you see a list of all the domain agent timestamps for the current agent's Event 300 (Host Resource Status). After you select a particular timestamp, the system displays the Application Server page containing performance information for that particular time. When you are viewing snapshot information, a message appears at the top of the page reminding you that the page contains historical performance data. You can view numerous snapshots using the same method.</p> <hr/> <p><b>Note.</b> While viewing snapshots, the Refresh button is not available. To view the most current information, you must access the Application Server page using the menu.</p> <hr/>
<b>Monitor Date/Time</b>	The date and time that the monitoring system inserted the row of performance data into the monitoring database according to the database clock.
<b>Last Page Refresh</b>	Indicates the most recent time that the system refreshed the page either by loading the page into the browser or as a result of a user clicking the Refresh button.

See [Chapter 6, "Monitoring System Performance," Monitoring Resource Usage, page 86](#).

### **Host Resource Status**

The Metrics tab presents the same values that also appear on the System Performance page. The metrics show current resource utilization on the entire host machine, not just the resources that PeopleSoft uses.

The system derives these metrics from the most recent Event 300 (Host Resource Status) for the domain monitor process of that domain.

If the latest Event 300 monitor date or time is older than the event sample rate for the current agent's system, the Stale Data icon appears at the top of the page.

<b>%CPU Used</b>	Displays the percentage of the CPU capacity being utilized on the host machine.
<b>%Memory Used</b>	Displays the percentage of physical memory being utilized on the host machine.
<b>Hard Page Faults/Second</b>	This is the number of accesses to virtual memory in the last second that require disk reads.
<b>Total Tuxedo Connections</b>	This is the number of connections to this domain, and it is equivalent to the number of rows that are returned by the Tuxedo command <code>pclt (tadmin)</code> , excluding JSH and WSH connections.
<b>Total Tuxedo Requests Queued</b>	The number of Tuxedo requests that are queued on that demand. A high number indicates that insufficient server processes are configured, or that the load on the host machine is too high.

The Events and PMUs tab presents links to the Event History, Open PMUs, and Completed PMUs pages.

When transferring to the Event History component, the system displays all of the data for the current day's Event 300 (Host Resource Status) rows for the current agent and the current system.

When transferring to the Open PMUs component, the system displays all of the data for the current day's open PMUs for the current host and port and the current system.

When transferring to the Completed PMUs component, the system displays all of the data for the current day's completed PMUs for the current host and port and the current system.

### ***PQ Event Rows***

This section presents the results of the `tmadmin pq` command (print queue). It enables you to view information that is related to the Tuxedo queues that are used by the server processes running within a domain.

The grid displays all Event 301 (Tuxedo "pq" Row) rows for the current agent with the same agent date or time appearing in the top portion of the page as the Event 300 (Host Resource Status).

<b>Queue Name</b>	Identifies the Tuxedo queue that is associated with a server process.
<b>Server Name</b>	Indicates the server process that is servicing a particular queue.
<b>Server Count</b>	Indicates the number of a particular server process type that are currently running. For example, it indicates that three PSAPPSRV server processes are currently running.
<b>Queue Length</b>	Indicates the current length of the queue, which is measured by the number of requests waiting to be processed. High queue lengths may indicate that more server processes need to be configured to run.

### ***PSR Event Rows***

This section presents the results of the `tmadmin psr` command (print server processes). This command enables you to view information that is related to all the server processes (monitored and unmonitored) running within a domain.

The grid displays all Event 302 (Tuxedo "psr" Row) rows for the current agent with the same agent date or time appearing at the top of the page as the Event 300 (Host Resource Status).

<b>Server Name</b>	Indicates the name of the server process, such as PSAPPSRV, PSMONITORSRV, and so on.
<b>Server Instance ID</b>	This is the instance ID that is assigned by Tuxedo to each server process. This number remains constant across recycles. Even if the PID changes, the instance ID remains constant.
<b>PID</b>	Indicates the operating system process ID on the server.
<b>Total Requests</b>	Indicates the total number of requests that a server process has processed. Tuxedo continues to increment this number for a server instance even if the server recycles.

**Current Service** Indicates whether the server process is idle or currently handling a request.

### **Monitored Servers**

This section enables you to view information about the resources that are consumed by the monitored server processes running within a domain.

The grid displays an entry for each active application server agent within the same system and same host and port, and domain directory as the domain monitor agent. For each agent, the system retrieves the latest Event 200 (Resources Per Process) and its metrics are displayed. If no Event 200 exists for a particular agent, the system displays zeros.

The Metrics tab contains the following information.

**Server** Identifies the server process being monitored.

**Server Instance** This is the instance ID that is assigned by Tuxedo to each server process. This number remains constant across recycles. Even if the PID changes, the instance ID remains constant.

**PID** The process ID that is assigned by the server operating system.

**Agent Date/Time** The date and time according to the agent clock that the domain monitor used when it last sent performance data to the monitor.

---

**Note.** The date and the time appearing on this page always applies to the system on which the agent runs. Keep this in mind if you are monitoring systems in other time zones.

---

**%CPU Used** Indicates the percentage of the CPU capacity that a particular server process is using.

**CPU Time** The amount of CPU time that the process has consumed.

**VM** Indicates the amount of virtual memory that each server process is consuming.

The Events and PMUs tab contains links to the Event History, Open PMUs, and Completed PMUs pages.

When transferring to the Event History component, the system displays all data for the current day's Event 200 (Resources Per Process) rows for the current agent and the current system.

When transferring to the Open PMUs component, the system displays all data for the current day's open PMUs for the current agent and the current system.

When transferring to the Completed PMUs component, the system displays all data for the current day's completed PMUs for the agent and the current system.

## Viewing Analytic Server Information

The Analytic Server Summary section in the monitored Application Server Domain page provides generic performance and status information of the analytic Servers running in the application server domain.

<b>Total Configured Servers</b>	The current maximum number of configured analytic servers in the application server domain.
<b>Number Started Servers</b>	The current number of analytic servers, started up and loaded with analytic instances.
<b>Number Free Servers</b>	The current number of analytic servers, started up and free.
<b>Number Loaded Instances Today</b>	Today's total number of analytic instances loaded.
<b>Number Recycled Servers</b>	Today's total number of analytic servers recycled for reuse.
<b>Number Out Servers Today</b>	Today's total number of times that the monitored domain runs out of analytic servers. Each time a client asks to load an analytic instance, a free analytic server gets assigned and dedicated to the analytic instance.

---

## Viewing Process Scheduler Server Performance

Access the Process Scheduler Server page (PeopleTools, Performance Monitor, System Monitor, Process Scheduler).

The Process Scheduler Server page displays the most recent performance data that is received from:

- A Process Scheduler server domain agent.
- Each active Process Scheduler server agent within this Process Scheduler domain.

<b>Domain Name</b>	Identifies the name of the current domain.
<b>Domain Directory</b>	Identifies the directory in which the domain is installed.
<b>Agent Date/Time</b>	The date and time according to the agent clock that the domain monitor used when it last sent performance data to the monitor.
	<hr/> <b>Note.</b> The date and the time appearing on this page always applies to the system on which the agent runs. Keep this in mind if you are monitoring systems in other time zones. <hr/>
<b>Status</b>	Appears only if the system detects stale data. Data is stale if no status events have been reported in the last sample interval for the system. For example, Status shows stale data if the Process Scheduler is not booted.

<b>Host/Port</b>	Identifies the machine name of the Process Scheduler server domain.
<b>Process Scheduler Snapshots</b>	This link enables you to view historical information snapshots for the current agent. By clicking this link, you see a list of all the domain monitor timestamps for the current agent's Event 300 (Host Resource Status). After selecting a particular timestamp, the system displays the Process Scheduler Server page containing performance information for that particular time. When you are viewing snapshot information, a message appears at the top of the page reminding you that the page contains historical performance data. You can view numerous snapshots using the same method. <hr/> <b>Note.</b> While you are viewing snapshots, the Refresh button is not available. To view the most current information, you must access the Process Scheduler Server page using the menu. <hr/>
<b>Monitor Date/Time</b>	The date and time that the monitoring system inserted the row of performance data into the monitoring database according to the database clock.
<b>Last Page Refresh</b>	Indicates the most recent time that the system refreshed the page either by loading the page into the browser or as a result of a user clicking the Refresh button.

See [Chapter 6, "Monitoring System Performance," Monitoring Resource Usage, page 86.](#)

### **Host Resource Status**

The Metrics tab presents the values that also appear on the System Performance page.

If the latest Event 300 (Host Resource Status) monitor date and time is older than the event sample rate for the current agent's system, the Stale Data icon appears at the top of the page.

<b>%CPU Used</b>	Displays the percentage of the Central Processing Unit capacity being utilized on the host machine.
<b>%Memory Used</b>	Displays the percentage of physical memory being utilized on the host machine.
<b>Hard Page Faults/Second</b>	The number of accesses to virtual memory in the last second that require disk reads.
<b>Total Tuxedo Connections</b>	This number is always zero for a Process Scheduler domain.
<b>Total Tuxedo Requests Queued</b>	The number of Tuxedo requests that are queued on that demand. A high number indicates that insufficient server processes are configured, or that the load on the host machine is too high.

The Events and PMUs tab presents links to the Event History, Open PMUs, and Completed PMUs pages.

When transferring to the Event History component, the system displays all of the data for the current day's Event 300 (Host Resource Status) rows for the current agent and the current system.

When transferring to the Open PMUs component, the system displays all of the data for the current day's open PMUs for the current host or port and the current system.

When transferring to the Completed PMUs component, the system displays all of the data for the current day's completed PMUs for the current host port and the current system.

### ***PQ Event Rows***

This section presents the results of the `tmadmin pq` command (print queue). It enables you to view information that is related to the queues that are used by the server processes running within a domain.

The performance data displays all Event 301 (Tuxedo "pq" Row) rows for the current agent with the same agent date and time appearing in the top portion of the page as the Event 300 (Host Resource Status).

<b>Queue Name</b>	Identifies the Tuxedo queue that is associated with a server process.
<b>Server Name</b>	Indicates the server process that is servicing a particular queue.
<b>Server Count</b>	Indicates the number of a particular server process type that are currently running. For example, it indicates that three PSAESRV server processes are currently running.
<b>Queue Length</b>	Indicates the current length of the queue, which is measured by the number of requests that are waiting to be processed.

### ***PSR Event Rows***

This section presents the results of the `tmadmin psr` command (print server processes). This enables you to view information that is related to all the server processes (monitored and unmonitored) running within a domain.

The grid displays all Event 302 (Tuxedo "psr" Row) rows for the current agent with the same agent date and time appearing at the top of the page as the Event 300 (Host Resource Status).

<b>Server Name</b>	Indicates the name of the server process, such as PSAESRV, PSMONITORSRV, and so on.
<b>Server Instance ID</b>	This is the instance ID that is assigned by Tuxedo to each server process. This number remains constant across recycles. Even if the PID changes, the instance ID remains constant.
<b>PID</b>	Indicates the operating system process ID on the server.
<b>Total Requests</b>	Indicates the total number of requests that a server process has processed.
<b>Current Service</b>	Indicates whether the server process is idle or currently handling a request.

## Monitored Servers

This section enables you to view information about the resources that are consumed by the monitored server processes running within a domain.

---

**Note.** Currently, only PSMONITORSRV and PSMSTPRC are monitored.

---

The grid displays an entry for each active application server agent within the same system and same host or port and domain directory as the domain monitor agent. For each agent, the system retrieves the latest Event 200 (Resources Per Process) and its metrics are displayed. If no Event 200 exists for a particular agent, the system displays zeros.

The Metrics tab contains the following information.

<b>Server</b>	Identifies the server process that is being monitored.
<b>Server Instance</b>	This is the instance ID that is assigned by Tuxedo to each server process. This number remains constant across recycles. Even if the PID changes, the instance ID remains constant.
<b>PID</b>	The process ID that is assigned by the server operating system.
<b>Agent Date/Time</b>	The date and time according to the agent clock that the domain monitor used when it last sent performance data to the monitor.

---

**Note.** The date and the time appearing on this page always apply to the system on which the agent runs. Keep this in mind if you are monitoring systems in other time zones.

---

<b>%CPU Used</b>	Indicates the percentage of the CPU capacity that a particular server process is using.
<b>CPU Time</b>	The amount of CPU time that the process has consumed.
<b>VM</b>	Indicates the amount of virtual memory that each server process is consuming.

The Events and PMUs tab contains links to the Event History, Open PMUs, and Completed PMUs pages.

When transferring to the Event History component, the system displays all data for the current day's Event 200 (Resources Per Process) rows for the current agent and the current system.

When transferring to the Open PMUs component, the system displays all data for the current day's open PMUs for the current agent and the current system.

When transferring to the Completed PMUs component, the system displays all data for the current day's completed PMUs for the agent and the current system.

---

## Viewing Master Scheduler Performance

Access the Master Scheduler page (PeopleTools, Performance Monitor, System Monitor, Master Scheduler).

This page displays the most recent performance data received from a Master Scheduler agent.

---

**Note.** The Performance Monitor displays Master Scheduler performance data only if a Master Scheduler has been configured.

---

**System ID** Identifies each monitored system. The PeopleSoft system automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.

**Database Name** Reveals the name of the PeopleSoft application database that is running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.

**Agent Date/Time** The date and time according to the agent clock that the domain monitor used when it last sent performance data to the monitor.

---

**Note.** The date and the time appearing on this page always apply to the system on which the agent runs. Keep this in mind if you are monitoring systems in other time zones.

---

**Status** Appears only if the system detects stale data. Data is stale if it is older than the sample rate for that system. Data is stale if no status events have been reported in the last sample interval for the system. For example, Status shows stale data if the Process Scheduler is not booted.

**Monitor Date/Time** The date and time that the monitoring system inserted the row of performance data into the monitoring database according to the database clock

**Last Page Refresh** Indicates the most recent time that the system refreshed the page either by loading the page into the browser or as a result of a user clicking the Refresh button.

### ***Process Summary***

This process displays the status of the processes that are being managed by the Master Scheduler running on the selected system.

The data in this grid is derived from Event 350 (Master Scheduler Status).

**Active Processes** The number of Process Scheduler jobs that are currently processing in this monitored system.

<b>Unused Process Slots</b>	The number of available slots that this Master Scheduler can use to run process requests.
<b>Blocked Processes</b>	A process can be <i>blocked</i> for two reasons: <ul style="list-style-type: none"> <li>• The process is "mutually exclusive" with another process, and the other process is currently running.</li> <li>• The process is dependent on a file that another process has yet to create.</li> </ul>
<b>Queued Processes</b>	The number of Process Scheduler jobs that are currently waiting to be handled by an available Process Scheduler server.

## **Servers**

The Servers grid contains performance metrics for individual Process Scheduler servers that are running under the Master Scheduler.

This grid displays all Event 351 (Master Scheduler Detail) rows for all active Process Scheduler servers under the current system with the same agent date and time as the latest Event 350 (Master Scheduler Status). The system groups the data by server name and the metrics are accumulated.

---

**Note.** If the monitor date and time for an Event 351 is older than the current system's event sample rate, a Stale Data icon appears at the top of the page.

---

<b>Server Name</b>	The name of the Process Scheduler server, such as PSNT, PSUNIX, and so on. The server name is a hyperlink that launches a Server Activity secondary page for that particular server type. The Server Activity page displays all of the individual Event 351 (Master Scheduler Detail) rows. Use the Refresh button on this page to refresh the Processes grid.
<b>Active Processes</b>	The number of jobs that are currently being handled by a Process Scheduler server.

## **Queue Details**

The Queue grid displays the status of the processes that are queued to be processed by the Master Scheduler.

The data is generated from Event 354 (Batch Queue Details) rows for all active Process Scheduler agents in the current system with the same agent date and time as the latest Event 350 (Master Scheduler Status).

<b>Process Type</b>	Displays the type of process that are queued, such as SQR, Application Engine, COBOL, and so on.
<b>Queued Processes</b>	Displays the number of processes that are queued per process type.

**Blocked Processes**

A process can be *blocked* for two reasons:

- The process is "mutually exclusive" with another process, and the other process is currently running.
- The process is dependent on a file that another process has yet to create.

---

## Viewing Open PMUs

Access the Open PMUs page (PeopleTools, Performance Monitor, System Monitor, Open PMUs).

An open PMU is a PMU that started but has not finished prior to an agent reporting performance metrics to the monitor. Information regarding open PMUs are stored in the PSPMTRANSCURR table. When the PMU finishes, the PSPPMSRV flags the PMU for deletion and inserts a matching row into the PSPMTRANSHIST table, which stores completed PMU data.

See [Chapter 4, "Administering the Performance Monitor," Scheduling the Reaper Program, page 39](#).

If an end user is reporting that a PeopleSoft page is slow (displays the "Processing...." message), use the Open PMU page to see where the request has stalled. Search for that user's name to see the current state of that user's open PMUs.

Before you view current PMU details, you need to enter search criteria. If you don't specify criteria, the page displays *all* of the open PMUs for the selected system.

**User ID**

If you want information per user, enter the user ID of the user for which you want to track current PMUs.

The user ID prompts against the PSPMOPRDEFN table, which is populated by the Lookup Application Engine program.

**Performance Trace Name**

You enter the performance trace name to search on currently open PMUs running within a particular trace.

If a user has launched a performance trace in the Performance Console, the system labels every PMU that is generated in a business process with a performance trace name.

The performance trace name prompts against the PSPMPERFTRACE table, which is populated by the Lookup Application Engine program.

**Component**

If you want information per component, enter the name of the component for which you want to track completed PMUs.

The component prompts against the PSPMPNLGRPDEFN table, which is populated by the Lookup Application Engine program.

---

**Note.** Searching on component is equivalent to searching on context 1 for application server PMUs.

---

**Market**

Select the market that is associated with the component.

<b>PMU Set</b>	<p>Indicates the set to which the PMU definition belongs.</p> <p>A PMU definition set is similar to a message set.</p> <p>To view the complete set of PMU set definitions, select Performance Monitor, Utilities, PMU Definitions.</p> <p>See <a href="#">Chapter 10, "Performance Monitor Meta-Data," PMU Definitions, page 155.</a></p>
<b>PMU ID</b>	<p>Identifies the PMU definition within a PMU definition set.</p> <p>To view the complete set of PMU definitions, select Performance Monitor, Utilities, PMU Definitions.</p>
<b>Context 1, 2, 3</b>	<p>PMU metrics contain data that is specific to that PMU. Context values, on the other hand, are common to the entire user request or a specific tier. For example, the component name is stored in a context for all PMU's that are generated by an application server while it is processing that component. The system uses contexts to "flatten" a PMU tree. For example, you do not have to navigate up from a SQL PMU to an ICPanel PMU to see what component generated that SQL statement.</p> <p>In some cases, a parent PMU determines the usage of a child PMU's context fields. In such cases, the context label for the child PMU type is Generic.</p> <p>Use the Context Help button to view the context definitions for a PMU set ID and ID that you've entered.</p> <p>See <a href="#">Chapter 10, "Performance Monitor Meta-Data," Context Definitions, page 155.</a></p>
<b>Top Instance</b>	<p>Each PMU that is reported by an agent has a unique instance ID. The instance identifier of the first PMU generated by a particular user request is the top instance. The value of the top instance is then stored with every child PMU within a request. The top instance is the associating value for all PMUs that are generated by the same user request.</p> <p>Use this search field when you already know the top instance of the PMU in which you are interested.</p>
<b>Instance Identifier</b>	<p>Each PMU reported by an agent has a unique instance ID.</p>
<b>Duration &gt;=/&lt;</b>	<p>Specify criteria that is related to the duration of a PMU. The duration is the difference between the current monitor database time and the date and time that the monitor received the open PMU. The system searches for PMUs with a duration greater than or equal to the value that you enter, or less than the value that you enter. The value is in seconds.</p>
<b>Domain Name</b>	<p>Select the application server, web server, or Process Scheduler domain of the agent process that is reporting the PMUs.</p>
<b>Domain Host/Port</b>	<p>Select the host name and port number of the application server, web server, or Process Scheduler domain from which the PMUs were reported.</p>
<b>Agent ID</b>	<p>Select the agent that reported the PMUs.</p>

**Agent Type** Select agent type, such as PSAPPSRV, PSQRYSRV, or PSQCKSRV, that reported the PMUs.

---

**Note.** Some of the metrics for various PMUs don't have values until the PMU finishes, so in some cases, metrics may appear with no values.

---

### **Open PMUs**

The following information appears on the Summary tab.

<b>PMU</b>	Displays the label for that PMU type.
<b>Duration</b>	Displays the duration of the PMU.
<b>PMU Details</b>	The string that is generated by concatenating all of the contexts and metrics that are selected for display in the PMU definition.
<b>User ID</b>	Identifies the user whose request generated the PMU.
<b>Action</b>	The action could be <i>Start</i> or <i>Update</i> . Every PMU has a start, which refers to a request initiating the PMU. The update action applies to long running PMUs, which the Performance Monitor updates periodically with the latest metric values.
<b>Monitor Received Date/Time</b>	Indicates when the monitor system received notification that a PMU had been started or updated.
<b>Agent Start Date/Time</b>	Indicates that time according to the agent clock that the last update was generated.

The Identifiers tab presents the following additional information.

<b>PMU Set</b>	Indicates the set to which the PMU definition belongs. A PMU definition set is similar to a message set. To view the complete set of PMU set definitions, select PeopleTools, Performance Monitor, Administration, Meta-data, PMU Definitions.
<b>PMU ID</b>	Identifies the PMU definition within a PMU definition set. See <a href="#">Chapter 10, "Performance Monitor Meta-Data," PMU Definitions, page 155.</a>
<b>Agent ID</b>	Identifies the agent that reported the PMU information.
<b>Domain Name</b>	The domain from which the PMU was reported.
<b>Monitor Last Update Date/Time</b>	The monitor date and time of the most recent event of a specific event type that was received from a specific agent.

<b>Agent Last Update Date/Time</b>	Indicates that time according to the agent clock that the last update was received.
<b>PID</b>	The operating system process ID of the process that generated the PMU.
<b>Instance</b>	PMU instance ID. Each PMU has a unique instance ID.
<b>Parent Instance</b>	The instance ID of the parent PMU.
<b>Top Instance</b>	Each PMU that is reported by an agent has a unique instance ID. The instance identifier of the first PMU that is generated by a particular user request is the top instance. The value of the top instance is then stored with every child PMU within a request. The top instance is the associating value for all PMUs that are generated by the same user request.

The Contexts tab displays the context information that is associated with a particular PMU.

The Metrics tab displays the metric information that is associated with a particular PMU.

---

## Viewing Open PMU Trees

Access the Open PMU Trees page (PeopleTools, Performance Monitor, System Monitor, Open PMU Trees).

Every user request generates a set of PMUs that you can display as a tree. The PMU with the top instance ID is the root of the tree and PMUs with no children are the leaves. A node represents each PMU or child PMU. Clicking a node reveals its details.

The shaded text flags the node of the currently processing PMU.

A search may return multiple PMU trees. For example, if a user ID is shared across multiple users, or a single user has initiated multiple browser sessions, then more than one user request may be currently processing, and therefore more than one PMU tree open.

The data that is related to open PMUs is dynamic because the system is currently processing the PMU. Therefore, the composition of trees, and even their presence, is likely to change each time you click Refresh.

If you have an open PMU tree that was captured using the standard agent filter mode, you can drill down to the detail of PMU 400 (Tuxedo Service PCode and SQL). The Current SQL statement shows what has been submitted to the application server. This can aid in troubleshooting long-running SQL or a hung query without increasing the agent filter and asking users to repeat the process. If the open PMU finishes, then the PMU row is marked for deletion by the reaper program. You can't access the current SQL statement if the PMU has finished.

---

**Note.** So that the presentation of PMU information is readable and manageable, if a PMU is older than a day, the system displays the date that the PMU ran, not its duration in milliseconds.

---

**Note.** For each PMU in the tree, the duration value represents the period of time that has elapsed since the monitoring system received the start timestamp for each PMU. In some cases, the display may indicate that a child PMU is "older" than its parent. For example, if the application server sent PMU information before the web server sent PMU information, the child PMUs that are running on the application server will display a smaller duration than the associated parent PMUs that are running on the web server.

---

<b>User ID</b>	Enter the user ID of the user who is initiating the PMUs. You must specify a user ID to view open PMU trees.
<b>Total Trees received</b>	The number of currently open PMU trees for the user. Corresponds to the number of open requests for this user.
<b>Currently displaying</b>	Indicates the PMU tree that is currently displayed on the page, providing orientation in a list of returned PMU trees.
<b>Previous Tree/Next Tree</b>	Enables you to navigate within a list of PMU trees.
<b>Open PMU Tree</b>	You can expand or collapse PMU trees by using the folder icon to the left. Click a node to view details regarding that PMU.  Use the Left and Right links to navigate within a single tree when the tree is 20 nodes deep.

---

## Viewing Current User Sessions

Access the Current User Sessions page (PeopleTools, Performance Monitor, System Monitor, Current User Sessions).

A row appears for every PMU 109 (user session began) that was received in the last 12 hours for which a matching PMU 108 (user session ended) has not been received.

---

**Note.** If you recycle the web server during this time, the user session is considered closed.

---

The Session tab contains the following information.

<b>User ID</b>	Identifies the user.
<b>Session ID</b>	Indicates the open session on the web server that is associated with the user.
<b>Monitor Received Date/Time</b>	The date and time that the monitoring system inserted the row of performance data into the monitoring database.
<b>User History</b>	Accesses the User History page.  <u>See Chapter 7, "Analyzing Historical Performance Data," Viewing User Session History, page 105.</u>

The Details tab contains the following additional information.

The latency and user agent data is derived from PMU 116 (redirect after login) with the same session ID (context 1) as the associated PMU 109 (user session began).

<b>IP Address</b>	Indicates the IP address of the machine that the user is using to connect to the PeopleSoft system.
-------------------	---

<b>User Agent</b>	The Mozilla user-agent string specification for the user's browser.
<b>Latency</b>	The duration of PMU 116 (redirect after login). The latency value enables you to infer the network latency for a user connection.

---

## Viewing Recycled Processes Diagnosis

Access the Recycled Processes Diagnosis page (PeopleTools, Performance Monitor, System Monitor, Recycled Processes Diagnosis).

The Recycled Processes Diagnosis page is used to determine the causes of application server recycles. Specifically, it can determine what kind of server request was received by application server processes before a restart occurred.

---

## Monitoring Resource Usage

This section provides additional information that may help you to interpret the information that Performance Monitor provides that is related to the events reporting resource usage on host machines. These events are:

- Event 150 (JVM Status).
- Event 151 (Network Status).
- Event 200 (Resources Per Process).
- Event 300 (Host Resource Status).
- Event 301 (Tuxedo "pq" Row).
- Event 302 (Tuxedo "psr" Row).

These events report the usage of machine resources (CPU, memory, network, and so on). These events, except for Event 150 (JVM Status), make calls to external APIs (often specific to the operating system) to retrieve metric information. The monitored system sends each event at the sampling interval that is specified for that system.

### Working with Event 150 (JVM Status)

This event applies only to web servers.

This event does not make calls to any operating system-specific API.

### Working with Event 151 (Network Status)

This event applies only to web servers.

For Event 151 (Network Status) the system launches a separate executable from Java that invokes the "netstat -n" command. On UNIX, the command runs in a separate shell. When the command finishes, the process ends. PeopleSoft does not run "netstat" with an interval argument.

---

**Warning!** On some platforms the "netstat" command can require up to a minute (or more) to finish. If the sampling interval is shorter than the time required for the command to complete, "netstat" commands will be running continuously.

---

## Working with Event 200 (Resources Per Process)

This event applies to the application server and the Process Scheduler server.

The PeopleSoft system makes specific operating system calls to obtain metrics for %CPU that are used by the process, CPU time consumed, virtual memory size, and working set size. Operating systems have slightly different definitions for these quantities and different ways of reporting them. For instance, "working set" memory is a Windows term; "resident set" is the UNIX equivalent. PeopleSoft strives for consistency across platforms. For example, PeopleSoft expresses %CPU within a range from 0 to 100 on all machines even though some vendors scale to  $N*100\%$  if multiple CPUs ( $N$  CPUs) exist.

Microsoft Windows and Linux compute one or more resources as an average of the two measurements at the beginning and end of a sampling interval. On these platforms, the Performance Monitor does not report an Event 200 (Resources Per Process) until the second sampling interval after you boot a server.

Process resource utilization is usually sampled by the operating system and written to a memory location. Windows writes to the registry, while UNIX writes to various files. The system reads the current values for the process, so events change only when the operating system updates the statistics. Most operating systems update these statistics at least once per second.

PeopleSoft obtains all information using lightweight, C++ programmatic APIs. No additional processes or shell commands are run.

<i>Operating System</i>	<i>Description</i>
Windows	<p>Performance Monitor uses Performance Data Helper (PDH) to read registry counters. The information is identical to the Windows Performance Monitor tool. When multiple copies of a process, such as PSAPPSRV, are running, registry counters are assigned arbitrarily. For example, counter 1 and counter 2 can reverse their process assignment when a process reboots. Performance Monitor corrects for this.</p> <ul style="list-style-type: none"> <li>• CPU utilization is "% Processor Time," defined as the fraction of time that the process spends in kernel + user during the last PDH sampling interval (typically one second).</li> <li>• Process time is kernel + user, accurate to 1 millisecond.</li> <li>• Virtual memory is "Virtual Bytes."</li> <li>• Working set memory is "Working Set," which is the same as "Mem Usage," displayed by the Windows Task Manager.</li> </ul>

Operating System	Description
AIX	<p>Performance Monitor reads the psinfo files, which is the same source of information that AIX uses for its "ps" command.</p> <ul style="list-style-type: none"> <li>• CPU utilization is the same as "ps -o pcpu" or the %CPU from "ps v". AIX defines it as a lifetime average: total CPU time that is consumed by the process divided by total time that the process has been running.</li> <li>• Process time is system + user, excluding children. Accurate to 10 milliseconds.</li> <li>• Virtual memory is the same as "ps -o vsz", or the 1024 * SIZE field from "ps v".</li> <li>• Resident set memory is the same as the 1024 * RSS field from "ps v".</li> </ul>
HPUX	<p>Performance Monitor reads pst_status using pstat_getproc.</p> <ul style="list-style-type: none"> <li>• CPU utilization is the same as "top" divided by the number of CPUs (on HPUX "top" shows utilization of a single CPU).</li> <li>• Process time is system + user, excluding children.</li> <li>• Virtual memory is the same as the SIZE field from "top".</li> <li>• Resident set memory is the same as the RES field from "top".</li> </ul>
Linux	<p>Performance Monitor reads ps information from /proc files.</p> <ul style="list-style-type: none"> <li>• CPU utilization is the same as "top" divided by the number of CPUs (on Linux "top" shows utilization of a single CPU). It is the average utilization over the last sample period.</li> <li>• Process time is system + user, excluding children, and accurate to 10 milliseconds. The times for all threads in the process are added together.</li> </ul> <p><b>Warning!</b> Linux kernel 2.4 tracks only the process time of each thread. PeopleSoft searches the /proc directory to find all threads and report the total time of the process. However, on a production system with thousands of threads, accumulating this information may take up to a second of CPU time.</p> <ul style="list-style-type: none"> <li>• Virtual memory is the same as "ps -o vsz" (not visible in "top"). According to the man pages, it counts just text, data, and stack.</li> <li>• Resident set memory is the same as "ps -o rss" (not visible in "top"). This value includes just text, data, and stack.</li> </ul>

<b>Operating System</b>	<b>Description</b>
Solaris	<p>Performance Monitor reads psinfo files, which is the same source of information that Solaris uses for its "ps" command.</p> <ul style="list-style-type: none"> <li>• CPU utilization is the same as "top". Solaris computes a moving average with exponential weighting.</li> <li>• Process time is system + user, excluding children, and accurate to 10 milliseconds.</li> <li>• Virtual memory is the same as "ps -o vsz" or the SIZE field from "top".</li> <li>• Resident set memory is the same as "ps -o rss" or the RES field from "top". On Solaris, shared libraries are counted in RES but not in SIZE.</li> </ul>
OS/390	The only metric that is supported in the current release is Process Time.

## Working with Event 300 (Host Resource Status)

This event applies to the application server and the Process Scheduler server.

Performance Monitor makes specific operating system calls to obtain metrics for %CPU use on the host machine, %Memory use, and the hard page fault rate. Operating systems have slightly different definitions for these quantities, and they have different ways of reporting them. In most cases, PeopleSoft expresses %Memory use to reflect utilization of physical memory.

Performance Monitor programmatically queries the Tuxedo management information base (MIB) for total Jolt connections and total requests queued. All platforms compute one or more resources as an average of the two measurements at the beginning and end of a sampling interval. Performance Monitor does not report an Event 300 (Host Resource Status) until the second sampling interval after you boot the server.

Process resource utilization is usually sampled by the operating system and written to a memory location. Windows writes to the registry, while UNIX writes to various files. The system reads the current values for the process, so events change only when the operating system updates the statistics. Most operating systems update these statistics at least once per second.

PeopleSoft obtains all information using lightweight, C++ programmatic APIs. No additional processes or shell commands are run.

<b>Operating System</b>	<b>Description</b>
Windows	<p>Performance Monitor uses Performance Data Helper (PDH) to read registry counters. The information is identical to the Windows Performance Monitor tool.</p> <ul style="list-style-type: none"> <li>• CPU utilization is "% Processor Time (_Total)", defined as the fraction of time not run by the system idle thread in the last PDH sampling interval (typically one second).</li> <li>• Memory utilization is "% Committed Bytes In Use" and will change if the paging file is extended by the system administrator.</li> <li>• Page faults is "Pages / sec", which reports actual disk page fetches ("Page Faults /sec" reports soft faults).</li> </ul>
AIX	<p>Performance Monitor uses libperfstat API (a wrapper for knlist) to read kernel counters.</p> <ul style="list-style-type: none"> <li>• CPU utilization is the same as "topas" and "vmstat". This is an average over the last sample period, but process CPU use is an average over process lifetime. Therefore, on AIX, it is possible for machine CPU use to be momentarily smaller than CPU use of a process that was previously CPU-intensive.</li> <li>• Memory utilization is defined as "free real pages" / "total real memory pages". The "vmstat" command shows the number of free pages, but not the available total. The "topas" field, Real MEMORY, shows "real memory pages." The PAGING SPACE field shows only reserved pages, not free pages.</li> <li>• Page faults is the same as "topas" and "vmstat", with the absolute difference averaged over the last sampling time. According to Linux information, this value includes pages faults that do not cause paging activity.</li> </ul>
HPUX	<p>Performance Monitor uses pstat_getdynamic (pstat) to read kernel counters.</p> <ul style="list-style-type: none"> <li>• CPU utilization is the same as "top" and "vmstat". This is an average over the last sample period, but process CPU use is an average over a longer time period. Therefore, it is possible on HPUX for machine CPU use to be momentarily smaller than CPU use of a process that was previously CPU-intensive.</li> <li>• Memory utilization is defined as "real memory + text pages / physical memory." Neither of these quantities are exposed with vmstat.</li> <li>• Page faults is the same as "vmstat -s, zero fill page faults", with the absolute difference averaged to a rate over the last sampling time.</li> </ul>

<b>Operating System</b>	<b>Description</b>
Linux	<p>Performance Monitor reads kernel statistics from files in /proc.</p> <ul style="list-style-type: none"> <li>• CPU utilization is the same as "top", averaged over all processors. This is an average over the last sample period.</li> <li>• Memory utilization is the same as "top", or the "free" command "Mem used / (av)ailable" field. It measures physical memory utilization. The "used" field is actually "available - free". The "M" and "K" units of "top" are 1024 * 1024 bytes and 1024 bytes, respectively.</li> <li>• Page faults is the first "page" kernel counter from /proc/stat (pages swapped in from disk), with the absolute difference averaged to a rate over the last sampling time.</li> </ul>
Solaris	<p>Performance Monitor uses the Kernel Statistics API (kstat) to read kernel counters.</p> <ul style="list-style-type: none"> <li>• CPU utilization is the same as "top, user + kernel". This is an average over the last sample period, but process CPU use is a weighted average over a longer time period. Therefore, it is possible on Solaris for machine CPU use to be momentarily smaller than CPU use of a process that was previously CPU-intensive.</li> <li>• Memory utilization is defined as the "used / used + available" fields from "swap -s". This is only an approximation because of allocated but unreserved RAM. See the Sun web site for more information.</li> <li>• Page faults is the same as major page faults from "vmstat -s", with the absolute difference averaged to a rate over the last sampling time.</li> </ul>
OS/390	<p>Performance Monitor uses the ERBSMFI and CVT APIs to report resource use on the logical partition. Higher priority jobs on other partitions can "steal" resources and not appear in these metrics.</p> <ul style="list-style-type: none"> <li>• CPU utilization is the same as RMF "TOTAL/AVERAGE LPAR BUSY TIME PERC" field.</li> <li>• Memory utilization is defined from the RMF "AVAILABLE" versus RMF "TOTAL FRAMES" fields.</li> <li>• Page faults is not supported for this release.</li> </ul>

## Working with Event 301 (Tuxedo "pq Rows")

This event applies to the application server and the Process Scheduler server.

The system programmatically queries the Tuxedo management information base (MIB) for the status of each queue.

## Working with Event 302 (Tuxedo "psr" Rows)

This event applies to the application server and the Process Scheduler server.

The system programmatically queries the Tuxedo management information base (MIB) for the status of each server. Only PeopleSoft servers appear as Performance Monitor events; the BBL is not reported.

## Chapter 7

# Analyzing Historical Performance Data

This chapter provides an overview of historical performance data and discusses how to:

- View completed PMUs.
- View event history.
- View user session history.

---

## Historical Performance Data

This section provides an overview of historical performance data and lists the pages used to analyze historical performance data.

## Understanding Historical Performance Data

The system stores historical data from the moment a PMU finishes processing or an event occurs. When the archive program runs (PSPM\_ARCHIVE), it either deletes the historical data or moves it to the archive tables.

Analyzing historical data can help you:

- Identify trends.

By comparing historical data, you can spot upward and downward performance trends.

- Investigate past user complaints.

For example, a user may complain that performance was slow during the previous week. You search historical performance data to find the PMUs that were generated by the user at a specific time a week ago.

## Pages Used to Analyze Historical Performance Data

---

**Note.** The charting pages appear under the History menu; however, those pages are covered in a separate chapter.

---

<b>Page Name</b>	<b>Definition Name</b>	<b>Navigation</b>	<b>Usage</b>
Completed PMUs	PSPMTRANSHIST	PeopleTools, Performance Monitor, History, Completed PMUs	Enables you to search for and display PMU details and trees.
Event History	PSPMEVENTHIST	PeopleTools, Performance Monitor, History, Event History	Enables you to search for and display event details.
User Session History	PSPMHISTUSERS	PeopleTools, Performance Monitor, History, User History	Enables you to view a user's signon and signout activity within a specified interval of time.
Maintain Chart Data	PSPMCSOCHRT_ADMIN	PeopleTools, Performance Monitor, History, Concurrent Signon Chart, Maintain Chart Data	Specify criteria for extracting and purging stored user signon information.
Concurrent Signon Chart	PSPMCSO_CHART	PeopleTools, Performance Monitor, History, Concurrent Signon Chart, Concurrent Signon Chart	Filter chart data, and display concurrent signon chart.

**See Also**

[Chapter 8, "Working with Performance Monitor Charts and Analytics," page 113](#)

## Viewing Completed PMUs

Access the Completed PMUs page (PeopleTools, Performance Monitor, History, Completed PMUs).

---

**Note.** The search result of completed PMUs includes history information collected by agents even if they are currently marked inactive.

---

**Note.** If a "clone table" warning appears at the top of the page, the archive program is either currently running or has abnormally terminated. If this is the case, all new incoming information that is being sent to the monitoring system is currently being inserted into the clone tables. New performance data is not inserted into the current data tables until the archive program finishes successfully.

---

## Entering Search Criteria

Before you can view information about completed PMUs, you need to enter criteria so that you can locate the appropriate completed PMUs.

Use the Advanced Search link to expose all search criteria.

The maximum rows that are returned by the search depends on the value in the Search Row Limit field on the Global Administration page.

See [Chapter 4, "Administering the Performance Monitor," Setting Global System Options, page 25.](#)

<b>System ID</b>	Identifies each monitored system. PeopleSoft automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.
<b>Database Name</b>	The name of the PeopleSoft application database running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.
<b>User ID</b>	If you want information for a particular user, enter the user ID of the user for which you want to track completed PMUs.  This field prompts against the PSPMOPRDEFN table.
<b>Process ID</b>	Enables you to search completed PMUs by the process ID.
<b>Performance Trace Name</b>	If a user has launched a performance trace in the Performance Trace Console, the system labels every PMU that is generated in a business process with a performance trace name. Select the appropriate trace name to view its results.  The Performance Trace Name field prompts against the PSPMPERFTRACE table, which the Lookup program (PSPM_LOOKUP) populates.
<b>Component</b>	If you want information for a particular component, enter the name of the component for which you want to track completed PMUs.  This field prompts against the PSPMPNLGRPDEFN table. <hr/> <b>Note.</b> Searching on a component is equivalent to searching on Context 1 for application server PMUs. <hr/>
<b>Market</b>	Enter the market that is associated with the component.
<b>PMU Definition Set</b>	Indicates the set to which the PMU definition belongs. A PMU definition set is similar to a message set.  This is required if you want to generate metric and duration charts.
<b>PMU Identifier</b>	Identifies the PMU definition within a PMU definition set.  This is required if you want to generate metric and duration charts.
<b>Context 1, 2, 3</b>	PMU metrics contain data that is specific to that PMU. Context values, on the other hand, are common to the entire user request or a specific tier. For example, the component name is stored in a context. The system uses contexts to "flatten" a PMU tree. For example, you do not have to navigate up from a SQL PMU to an ICPanel PMU to see what component generated that SQL statement. In some cases, PMU types are shared across different user request types. In such cases, the label for the PMU will be generic. Use the Context Help button to view the context definitions for a PMU set ID and ID that you've entered.

<b>PMU Status</b>	Displays the PMU status. See <a href="#">Chapter 7, "Analyzing Historical Performance Data," Completed PMU Status, page 100.</a>
<b>Top Instance</b>	Each PMU that is reported by an agent has a unique instance ID. The instance identifier of the first PMU generated by a particular user request is the top instance. The value of the top instance is then stored with every child PMU within a request. The top instance is the associating value for all PMUs that are generated by the same user request.
<b>Statement Number</b>	This search field applies only when you are searching for PeopleCode SQL PMUs, which are PMU 406 (PeopleCode SQL Execute) and PMU 407 (PeopleCode BuiltIn SQL Execute). The Context 2 field contains the PeopleCode program name and Metric 2 contains the statement number in the PeopleCode where the SQL was executed.  To search by statement number, you must enter the following search criteria: PMU Definition Set = 1, PMU ID = 406 or 407, and Context 2 must have the name of a PeopleCode program.
<b>From Date/Time To Date/Time</b>	Specify a range of time in which the system should search for completed PMUs. The system searches on the monitor date and time, not the agent date and time.
<b>Duration</b>	Specify criteria that is related to the duration of a PMU in milliseconds.
<b>Domain Name</b>	Select the application server domain to generate the PMUs.
<b>Domain Host/Port</b>	Select the name and port number of the server machine on which the domain that generated the PMU resides.
<b>Agent Identifier</b>	Identify the specific agent that monitored a PMU.
<b>Agent Type</b>	Identify the type of agent that monitored a PMU.
<b>Search</b>	The Search button performs a search based on the search criteria that is defined on the page. If you want to view the most current information, use the Refresh button after running the initial search. If you click Search repeatedly, the system returns the same results from the same time range for each search.
<b>Refresh</b>	Updates the value in the To Time field to equal the current time and performs a search. This enables you to view the most current results of a search.

## Working with Completed PMUs

At the bottom of the Completed PMUs page, the Completed PMUs grid contains the information that is related to the PMUs that are retrieved by your search criteria. The tabs in the Completed PMUs page are discussed in the following sections.

## Summary

<b>PMU</b>	Indicates the name of the PMU. Click the PMU name to view the PMU Details page.
<b>Duration</b>	Indicates the duration of the PMU from start to finish in milliseconds.
<b>Complete Tree</b>	Click the tree icon to view the complete PMU tree, including all ancestor and descendant PMUs.
<b>PMU Details</b>	Provides further description of the PMU. The details comprise contexts and metrics that are helpful for identifying the PMU. Each type of PMU displays a different set of details.
<b>User</b>	Indicates the user whose request triggered the PMU.
<b>PMU Status</b>	Displays the PMU status. <a href="#">See Chapter 7, "Analyzing Historical Performance Data," Completed PMU Status, page 100.</a>
<b>Monitor Received Date/Time</b>	Indicates when the monitor system received notification that a PMU had been started or updated.
<b>Agent Start Date/Time</b>	Indicates that time according to the agent clock that the last update was received.
<b>Domain Name</b>	Indicates the domain that handled the request that is associated with the PMU.

## Identifiers

The Identifiers tab presents the following additional information.

<b>PMU Set</b>	Identifies the set to which the PMU ID belongs.
<b>PMU ID</b>	Identifies a PMU definition within a PMU set.
<b>Agent ID</b>	Identifies the agent that reported the PMU.
<b>Process ID</b>	The operating system process ID of the process that generated the PMU.
<b>Instance</b>	A unique identifier that is generated by the system for each PMU.
<b>Parent Instance</b>	The PMU that produced the PMU in this grid row.
<b>Top Instance</b>	Each PMU that is reported by an agent has a unique instance ID. The instance identifier of the first PMU that is generated by a particular user request is the top instance. The value of the top instance is then stored with every child PMU within a request. The top instance is the value that associates all PMUs that are generated by the same user request.

## **Contexts**

The Contexts tab reveals the context definition and value that is associated with each PMU in the grid. To view the details of the contexts in a PMU, examine the context and PMU definitions.

This Description column shows data for the additional description value that is part of the incoming PMU. The description stores large character values and generally stores information that exceeds the 128-character limit of Metric 7 and the 254-character limit of the contexts.

See [Chapter 10, "Performance Monitor Meta-Data," Context Definitions, page 155.](#)

## **Metrics**

The Metrics tab presents the metric values for all of the metrics that are defined for each PMU in the grid.

Metric information appears in a raw format. The values reflect exactly what is stored in the database.

See [Chapter 10, "Performance Monitor Meta-Data," Metric Definitions, page 152.](#)

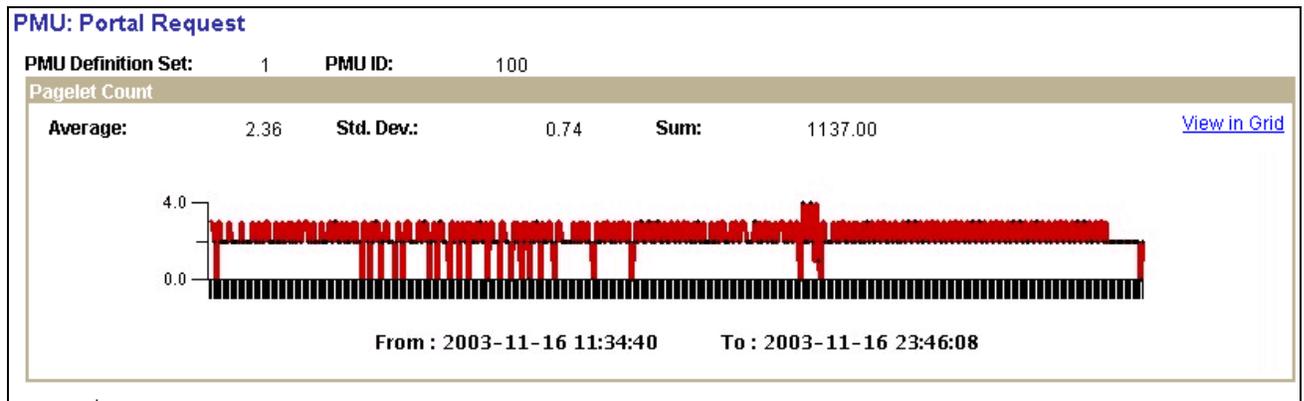
## **Trees**

The Trees tab enables you to view the PMU within the context of a PMU tree.

<b>View Ancestors</b>	Shows all parent PMUs for the selected PMU. This may be useful in determining the call tree of a child PMU. Only direct ancestors appear; sibling PMUs do not appear.
<b>View Descendants</b>	Shows all child PMUs for the selected PMU. This may be useful in determining where the time is spent for a PMU.
<b>Complete Tree</b>	Displays the entire tree representing the user request. If verbose or debug filters are set, this tree may contain a large number of nodes and may take a significant amount of time to load.

## **Metric Charts**

When your search criteria includes a PMU set and PMU identifier, all rows in the grid will be of the same type of PMU. In this case, the metric chart displays the metric values of the returned PMUs against one another. This enables you to track metrics over a period of time.



### Viewing a metric chart

Numeric metrics may be of type Counter, Gauge, or Numeric Identifier. Numeric identifiers do not appear in charts. Gauges and counters do appear in charts, but the chart does not contain a sum value for gauges.

Use the View in Grid link to view the chart information in a grid so that you can download the information to an Microsoft Excel spreadsheet if needed.

---

**Note.** If the metrics of a PMU can't be charted, a message appears stating this fact.

---

**Note.** To view the metric chart, you must specify a PMU and PMU set ID as part of the search criteria. If you click this link when performance data for multiple PMUs appears, the system displays an error message.

---

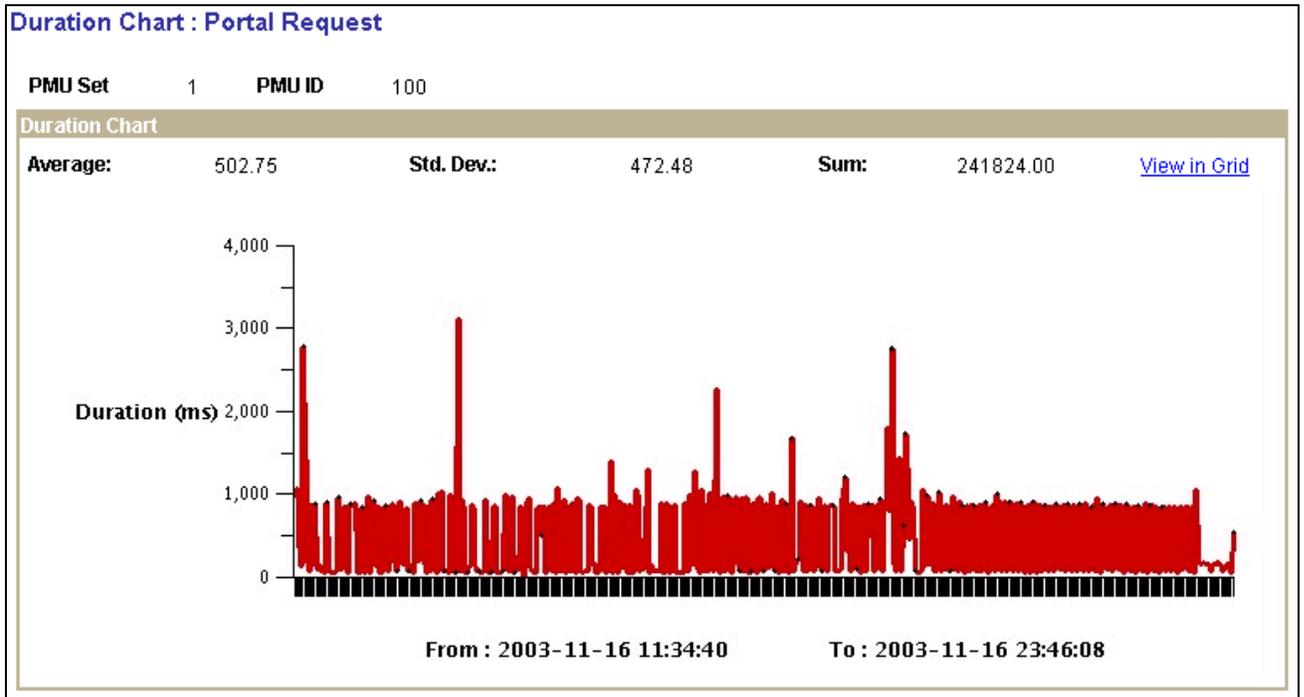
**Note.** The data appearing on this chart is not manipulated by multipliers, if any are used.

---

See [Chapter 10, "Performance Monitor Meta-Data," Metric Definitions, page 152.](#)

### ***Duration Charts***

The duration charts show all durations for all PMUs appearing in the grid. The chart is time-based.



Viewing a duration chart

Use the View in Grid link to view the chart information in a grid so that you can download the information to an Microsoft Excel spreadsheet if needed.

**Note.** To view the duration chart, you must specify a PMU and PMU set ID as part of the search criteria. If you click this link when performance data for multiple PMUs appears, the system displays an error message.

## Completed PMU Status

A completed PMU can be assigned one of the following statuses:

<b>Status</b>	<b>Description</b>
Success	The PMU finished successfully.
Failed	The PMU finished unsuccessfully.
Abort	Instrumentation code terminated a PMU due to an unexpected condition.
Unknown	Assigned in situations when an agent closes a child PMU if the top-level PMU closes out of sequence.
Timeout	Instrumentation code timed out the PMU.
Reaper Timeout	Long running PMU closed by the Reaper.

<b>Status</b>	<b>Description</b>
Admin Timeout	An administrator manually closed the PMU. This is <i>not</i> currently implemented.

## The PMU Details Page

You can access the PMU details page from a variety of Performance Monitor pages, such as the Completed PMUs page and the Open PMUs page. You can also access it readily from trees. To access the page, you click the link that is created from the PMU name.

The PMU Details page indicates:

- PMU ID and the PMU definition set to which it belongs.
- Identification information.
- Durations information, including metric values.
- Context information.
- Additional data.

The information that appears on the PMU Details page varies depending on the type of PMU that is displayed. All metrics are formatted according to information from the metric definition.

See [Chapter 10, "Performance Monitor Meta-Data," Metric Definitions, page 152.](#)

---

## Viewing Event History

Access the Event History page (PeopleTools, Performance Monitor, History, Event History).

---

**Note.** The search results of event history include history information that is collected by agents even if they are currently marked as inactive.

---

See [Chapter 4, "Administering the Performance Monitor," Viewing Agent Definitions, page 35.](#)

## Entering Search Criteria

Before you can view historical performance data that is related to events, you need to specify search criteria. This section covers the search criteria for event history.

Click the Advanced Search link to expose all of the search criteria options.

**System ID** Identifies each monitored system. PeopleSoft automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.

<b>Database Name</b>	The name of the PeopleSoft application database running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.
<b>Event Definition Set</b>	Enter the event definition set to which the event definition belongs. Currently, only one definition set exists, and it is reserved for PeopleSoft development.
<b>Event ID</b>	Select a particular event that you want to monitor, such as event 300, Host Resource Status.
<b>Process ID</b>	Specify the process ID for which you want to locate event history.
<b>From/To Date/Time</b>	Specify the range of time during which you want to locate event history. The system searches the monitor date and time, not agent date and time.
<b>Domain Host/Port</b>	If you want to monitor events on a particular server, select the server name/IP address and port number on which the domain runs.
<b>Domain Name</b>	If you want to monitor events pertaining to a particular domain, select the application server, web server, or Process Scheduler domain for which you want to view event information.
<b>Agent ID</b>	Enter specific agent information if wanted. Identifies an agent within a domain. This is automatically generated by the monitor the first time an agent registers with it.
<b>Agent Type</b>	Specify the type of server process the agent is monitoring, such as PSAPPSRV, PSQRYSRV, PSQCKSRV, and so on.
<b>Filter Level</b>	To view only the information that is returned of a specific filter level, select the desired filter level. For example, if you want to view only Error alarms, select Error, and the system returns only event information of type Error. Each event definition has a filter level that is set to one of the following levels: error, warning, standard, verbose, or debug. Error and warning levels are associated with alarms, while standard, verbose, and debug levels are associated with informational performance data. The icon that is associated with each category of filter level (alarm or informational) appears in the grid, enabling you to recognize event information easily.
<b>Metric Chart</b>	The system creates a metric chart only when you have selected a single event in the search criteria. The metrics that appear in the charts vary depending on the event definition.
<b>Search</b>	The Search button performs a search based on the search criteria that is defined on the page. If you want to view the most current information, use the Refresh button after running the initial search. If you click Search repeatedly, the system returns the same results from the same time range for each search.

**Refresh** Updates the value in the To Time field to equal the current time and performs a search. Enables you to view the most current results of a search.

## Working with Event History Information

The Completed Events section contains the results of your event history search.

---

**Note.** If a "clone table" warning appears at the top of the page, the archive program is either currently running or has abnormally terminated. If this is the case, all new incoming information that is being sent to the monitoring system is currently being inserted into the clone tables. New performance data is not inserted into the current data tables until the archive program finishes successfully.

---

See [Chapter 4, "Administering the Performance Monitor," Scheduling Performance Data Archiving, page 39.](#)

### Summary

The Summary tab displays the name of the event as well as the monitor date and time and agent date and time.

The maximum number of rows that are returned by the search is determined by the Search Row Limit on the Global Administration page.

See [Chapter 4, "Administering the Performance Monitor," Setting Global System Options, page 25.](#)

**Event** Displays the name of the event reported.  
Click the event name to view the Event Details page, which provides:

- Identification information to ensure that you can locate the exact source of the event.
- Metric information containing the metrics that are measured by a particular event and the current assigned values.
- Additional information, depending on the event definition.

---

**Note.** The Event Details page displays all metrics according to the metric definition.

---

**Monitor Date/Time** The date and time that the monitoring system inserted the row of performance data into the monitoring database according to the database clock.

**Agent Date/Time** The date and time according to the agent clock that the domain monitor used when it last sent performance data to the monitor.

---

**Note.** The date and the time appearing on this page always applies to the system on which the agent runs. Keep this in mind if you are monitoring systems in other time zones.

---

## **Identifiers**

The Identifiers tab displays information to help further identify the event and a particular event's origin.

<b>Event Set</b>	The event definition set to which the event definition belongs.
<b>Event ID</b>	The ID for a particular event definition, such as 300 for Host Resource Status.
<b>PID</b>	The operating system process ID of the process that generated the PMU.
<b>Agent ID</b>	The ID of the specific agent sending the event information.
<b>Domain Name</b>	The name of the domain in which the agent operates.
<b>Instance</b>	The system-generated identifier for a specific row of event information.
<b>Filter Level</b>	The filter level that is associated with an event definition.
<b>Description</b>	<p>If applicable according to the event definition, a description appears to place the event in context. For example, with the Event 301 (Tuxedo "pq" Row) the description contains the name of the server process that is associated with a particular queue, such as PSAPPSRV, JSL, PSSAMSRV, and so on.</p> <p>The information corresponds to the Additional Data attribute that is part of the event definition. It stores data that exceeds the 128-character limitation of Metric 7.</p>

## **Metrics**

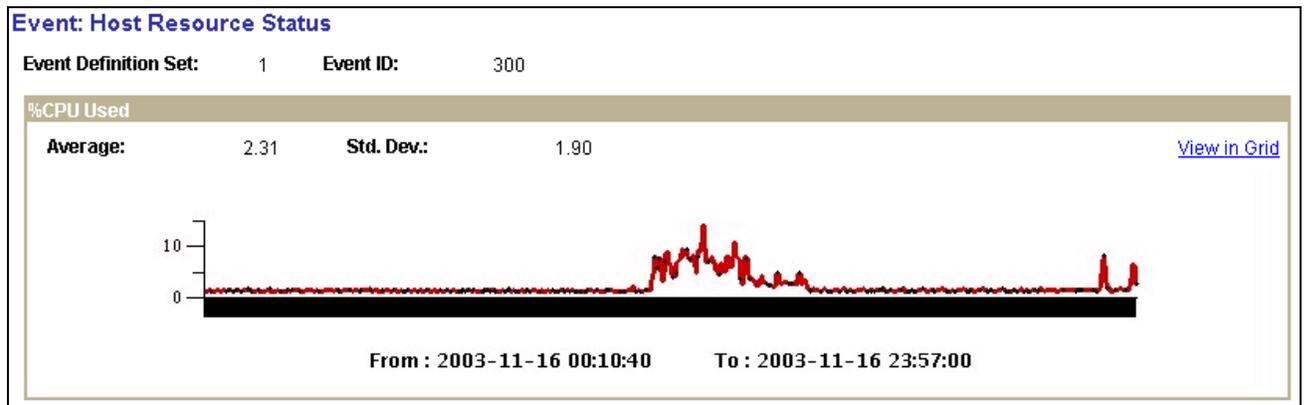
The Metrics tab displays the metric values of the event definition for a specific reported event. Only metrics that are specified in an event definition contain values. For example, event 300 (Host Resource Status) does not assign values to metric 6 or 7; therefore, those metrics appear on the Metrics page as *0.0000*.

Metric information appears in raw form.

See [Chapter 10, "Performance Monitor Meta-Data," Metric Definitions, page 152](#).

## **Metric Charts**

The system creates a metric chart only when you have selected a single event in the search criteria. The metrics that appear in the charts vary depending on the event definition.



Viewing an event metric chart

---

**Note.** %CPU Used is only one of the charts that appear in the metric charts for Host Resource Status.

---

## Viewing User Session History

Access the User Session History page (PeopleTools, Performance Monitor, History, User History).

The User Session History page enables you to monitor the signon and signout activity of a particular user within the time range specified.

A row appears for every PMU 109 (User Session Began) and every PMU 108 (User Session Ended) for that user.

---

**Note.** If you recycle the web server during this time, the user session is considered closed.

---

**Note.** Some signouts may actually be expired sessions on the web server if a user elected to close the browser rather than officially signing out of the PeopleSoft system (using the Sign out link).

---

## Entering Search Criteria

Before you can view historical performance data that is related to users, you need to specify search criteria. This section covers the search criteria for user history.

**User ID**                      Specify the user ID of a particular user.  
    This field prompts against the PSPMOPRDEFN table.

**From/To Date/Time**        Specify a range of time. The system searches on the monitor date and time, not the agent date and time.

<b>Search</b>	The Search button performs a search based on the search criteria that is defined on the page. If you want to view the most current information, use the Refresh button after running the initial search. If you click Search repeatedly, the system returns the same results from the same time range for each search.
<b>Refresh</b>	Updates the value in the To Time field to equal the current time and performs a search. Enables you to view the most current results of a search.

## Working with User History Information

The following information appears at the bottom of the User History page after you have completed a search.

### **Activity**

The activity tab presents the following information.

<b>User ID</b>	Identifies a particular user.
<b>User Activity</b>	Shows the signon and signoff activity of the user within the range of time that is specified in the search criteria.
<b>Session</b>	The session ID on the web server. This value is derived from context 1 from PMU 108 (User Session Ended) and PMU 109 (User Session Began).
<b>Monitor Received Date/Time</b>	Indicates when the monitor system received notification that a PMU had been started or updated

### **Details**

The Details tab presents the following additional information.

<b>IP Address</b>	The IP address of the computer from which the user accessed the PeopleSoft system.
<b>User Agent</b>	The Mozilla user-agent string specification for the user's browser. This value is derived from metric 7 of PMU 116 (Redirect after Login).
<b>Latency</b>	The duration of PMU 116 (Redirect After Login). The latency value enables you to infer the network latency for a user connection.

---

**Note.** The values in both the User Agent and Latency fields appear only in the rows reflecting logged on user activity.

---

---

## Viewing Concurrent Signon Information

This section provides an overview and discusses:

- Maintaining chart data.
- Viewing the concurrent signon chart.

## Understanding Concurrent Signon Information

In some situations, it is necessary to be able to track and monitor the number of users that sign on to the system concurrently. For example, if your site has implemented a cost-per-user access agreement, system administrators need a method to determine usage, and to ensure compliance with any service-level agreements. The Performance Monitor enables you to capture and store concurrent user information and display a graphical view of the number of users signed on to your system at a given time. The concurrent signon information is extracted from existing Performance Monitor user signon information, summarized, and stored separately for concurrent signon monitoring.

## Maintaining Chart Data

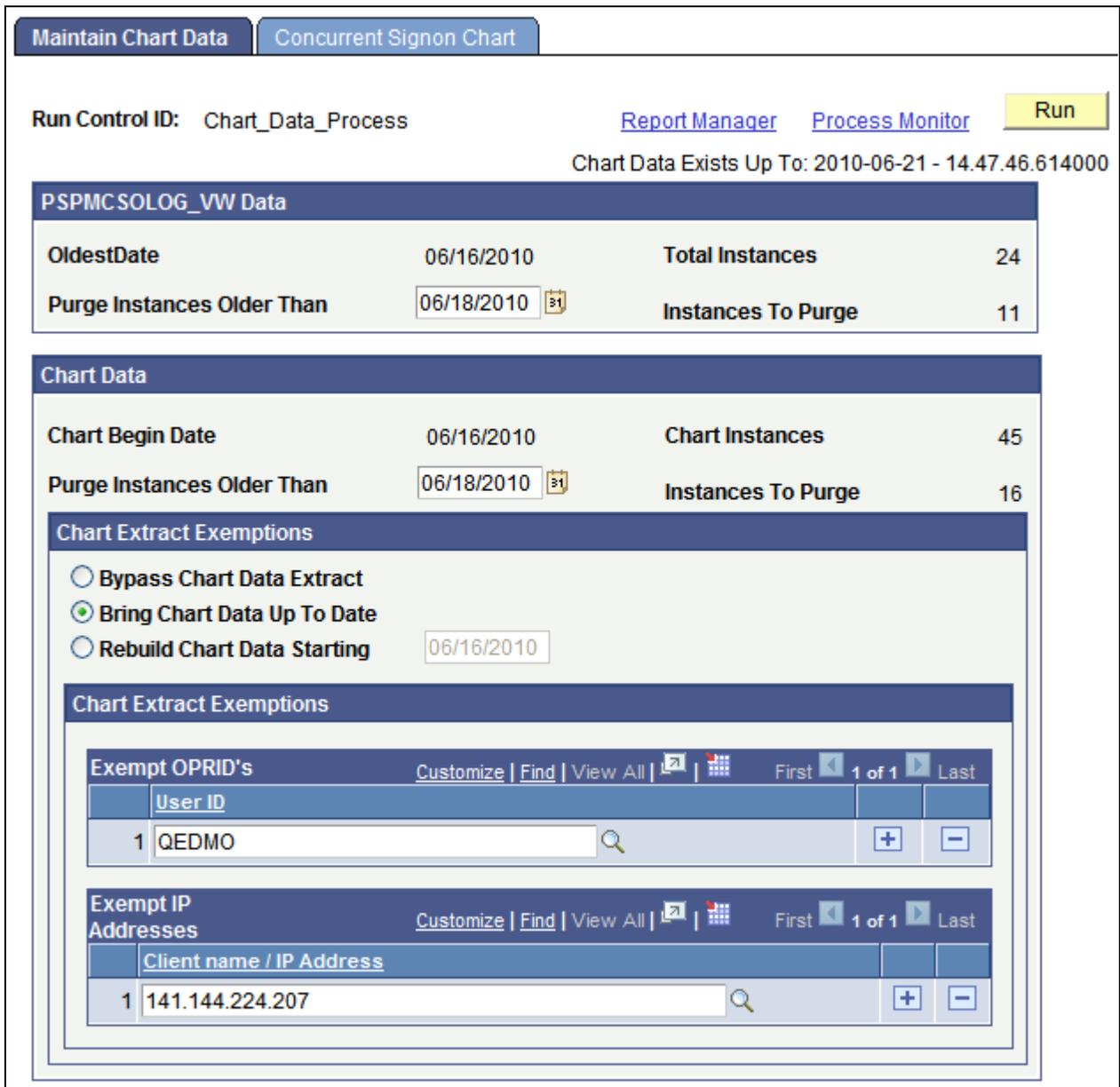
Access the Maintain Chart Data page (PeopleTools, Performance Monitor, History Concurrent Signon Chart, Maintain Chart Data).

The Maintain Chart Data enables you to set the criteria by which you purge and extract the concurrent signon user data. Once your criteria has been set, click Run to launch the PPMCSOSUM Application Engine program, which extracts summary information for PMU 108 (user session ended) and PMU 109 (user session began) from Performance Monitor PMU tables.

---

**Note.** When the PPMCSOSUM Application Engine program runs, it does not manipulate or alter existing data for PMU 108 and PMU 109. It extracts the previously stored PMU 108 and PMU 109 data required by concurrent signon monitoring and inserts the data into tables used by concurrent signon monitoring. When you purge concurrent signon information, you do not delete stored PMU data—only the summary data stored for concurrent signon monitoring is purged.

---



Maintain Chart Data page

**Run Control ID** Because this page should only have limited access, the run control is not saved by user ID. A universal run control, Chart\_Data\_Process, exists for the PSPMCSOSUM Application Engine program.

**Chart Data Exists Up To** Displays how current the signon data is. This the last time the PSPMCSOSUM Application Engine program ran.

**Oldest Date** Displays the oldest date of stored signon summary data.

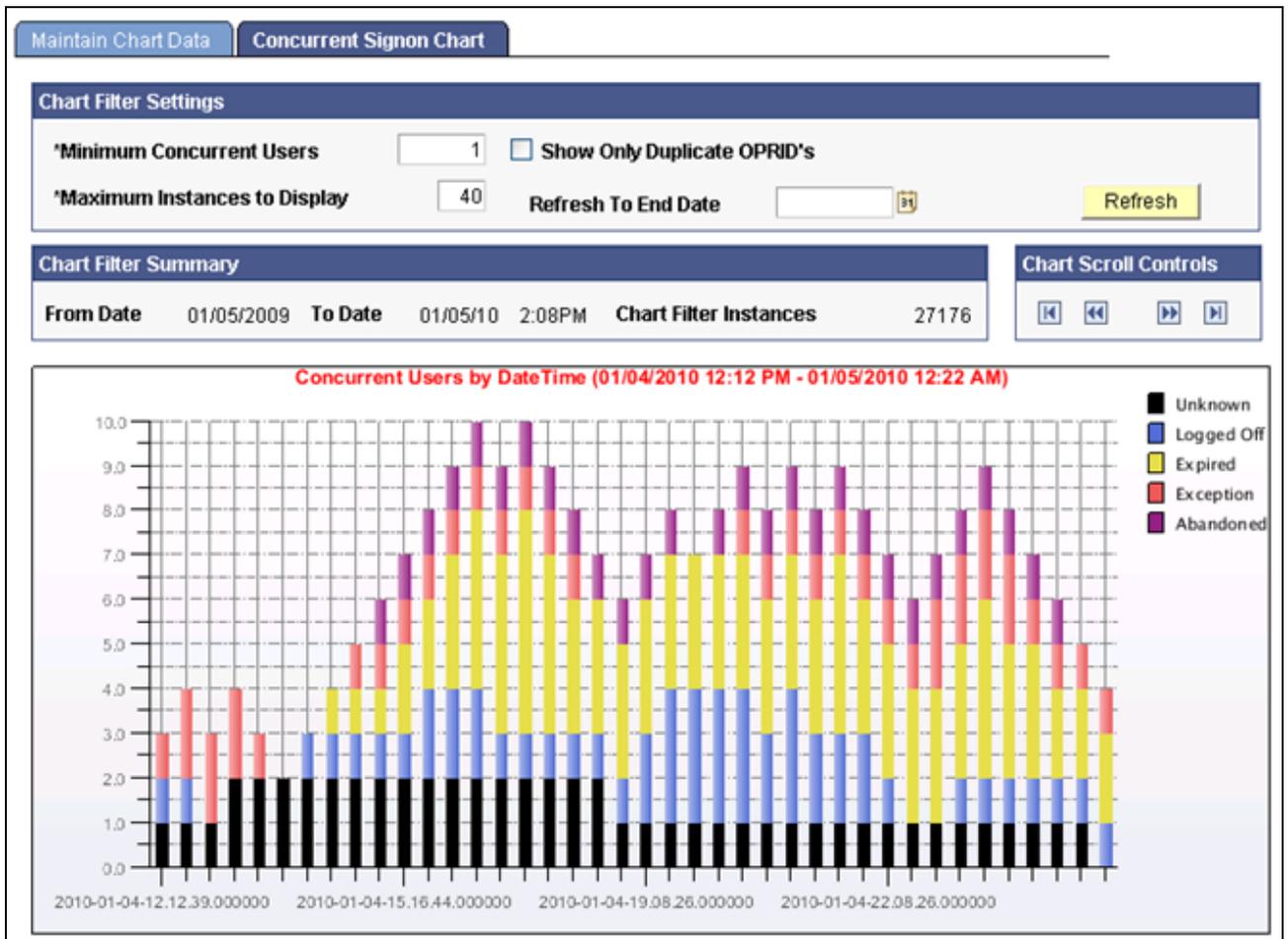
**Total Instances** Displays the number of rows of extracted PMU summary information stored in the concurrent signon tables.

<b>Purge Instances Older Than</b>	To reduce the number of rows of stored PMU summary information, specify the date of the older rows that are no longer needed.
<b>Chart Begin Date</b>	Displays the date of the oldest rows being stored for the concurrent signon chart.
<b>Chart Instances</b>	Shows the number of rows of signon data being stored for displaying the concurrent signon chart.
<b>Purge Instances Older Than</b>	To reduce the number of rows of chart data, specify the date of the older rows that are no longer needed.
<b>Instances To Purge</b>	After specifying the chart data purge date, Instances to Purge displays the number of rows that will be deleted, showing the space you will reclaim.
<b>Bypass Chart Data Extract</b>	Select if you only need to run the purge functions of the PSPMCSOSUM Application Engine program. In this case, only the specified information will be purged, and no new information will be added to the chart data.
<b>Bring Chart Up To Date</b>	(Default) Extracts all concurrent signon information to update the chart data.
<b>Rebuild Chart Data Starting</b>	Select to build the chart using data starting after the date you specify.
<b>Exempt OPRID's</b>	To make specific user IDs exempt from appearing in chart data, add those IDs to this grid. For example, exempt IDs might include those used by administrators or consultants, not actually part of the user base or the service contract.
<b>Exempt IP Addresses</b>	To make logins from specific workstations exempt from appearing in chart data, add either the IP address or the workstation name to this grid.

## Viewing The Concurrent Signon Chart

Access the Concurrent Signon Chart page (PeopleTools, Performance Monitor, History Concurrent Signon Chart, Maintain Chart Data).

Use this page to filter and view the concurrent signon chart.



Concurrent Signon Chart page

- Minimum Concurrent Users** Specify the minimum amount of concurrent users to track. Any record of current usage below this limit will not be included in the chart content.
- Show Only Duplicate OPRID's** Disables the Minimum Concurrent Users field and overrides its setting to 2. The chart content will be refreshed with only those instances where the same user ID was signed on to the system more than once simultaneously.
- Maximum Instances to Display** Use this setting to control how much information to display on the chart at a time. When viewing a chart with 100 instances (rows of data), details can become obscured, and the drill down capability becomes less effective. The chart can display a maximum of 999 bars.
- Refresh To End Date** Specify a particular date to position the chart so the very last data point for that day will be the last bar of the graph. This applies when the total number of rows in the chart data exceeds the Maximum Instances to Display value.
- Refresh** Click to refresh the chart based on the most recent filter criteria.
- Chart Filter Summary** Shows the information of the enter data set that can be displayed in the chart.

**Chart Scroll Controls** Enables you to scroll through the data and display multiple charts if not all the data is displayed at once. The scroll buttons for First, Prev, Refresh, Next and Last enable you to update the display. The First button refreshes the Chart so that the very first data point available will be shown as the first bar of the graph (or will position to the First/Last Scroll Date if specified). When the Prev or Next Button is clicked the graph moves to the prior/next set of data based on the Maximum Instances to Display setting.

Each instance (bar) in the chart displays hover text that describes the data the bar represents. You can drill down into each bar by clicking on it to display information similar to the following:

Concurrent Users as of 2010-01-04-21.16.44.000000

PSPMCSOLOG_VW						
User ID	Client name / IP Address	Log In Time	Log Out Time	Signon Type	Signoff Type	
1 USER11	10.221.161.246	01/04/10 1:22PM	01/05/10 12:22AM	PIA	Unknown	
2 USER7	10.221.161.246	01/04/10 3:17PM	01/04/10 10:17PM	PIA	Logged Off	
3 USER8	10.221.161.246	01/04/10 3:19PM	01/04/10 11:19PM	PIA	Expired	
4 USER12	10.221.161.246	01/04/10 3:23PM	01/05/10 3:23AM	PIA	Expired	
5 USER5	10.221.161.246	01/04/10 7:15PM	01/05/10 12:15AM	PIA	Abandoned	
6 USER10	10.221.161.246	01/04/10 7:21PM	01/05/10 5:21AM	PIA	Exception	
7 USER1	10.221.161.246	01/04/10 9:08PM	01/04/10 10:08PM	PIA	Logged Off	
8 USER3	10.221.161.246	01/04/10 9:12PM	01/05/10 12:12AM	PIA	Expired	

#### Concurrent Signon Drill-Down grid

**Note.** In cases where a "switch user" occurred, the original user will be signed off at the same time the new user will be signed on. You can track and confirm this by drilling down into concurrent signon instances.



## Chapter 8

# Working with Performance Monitor Charts and Analytics

This chapter provides an overview of Performance Monitor charts and analytics and discusses how to:

- View user request analytics.
- View component trace information.
- View component statistics.
- View portal statistics.
- View PIA statistics.
- View top portal content requests.
- View top PeopleCode events.
- View top PeopleCode executions.
- View top PeopleCode SQL statements.
- View top components.

---

## Performance Monitor Charts and Analytics

This section contains an overview of Performance Monitor charts and analytics, list common elements, and lists the pages used to view user request analytics.

### Understanding Performance Monitor Charts and Analytics

Performance Monitor charts enable you to view charts and graphs that help you analyze and communicate performance information.

This information is derived from completed PMU data that is stored in the historic performance data tables.

---

**Note.** This interface is not designed to assist you in diagnosing real-time, current user issues, such as a stalled user request.

---

---

**Note.** The Standard Deviation (Std. Dev.) is provided on many pages. The standard deviation is a statistic that tells you how tightly all the values that are used to compute the average are clustered around the average. Large standard deviations warn that the averages appearing in the chart are not a reliable indicator of response times experienced by individual users.

---

## Common Elements Used in This Chapter

- Chart Legends** Many of the charts use color to differentiate performance data. Always use the labels and the legend to make sure that you interpret the charts correctly.
- View in Grid** Enables you to view chart information in a grid format for sorting and downloading to Microsoft Excel.

## Pages Used to View Performance Monitor Charts

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
User Requests	PSPMTRANSUSERSUM	PeopleTools, Performance Monitor, Analytics, User Requests	This page shows top-level, completed PMUs that were initiated by a specific user ID. A <i>user request</i> is a set of PMUs that together make up a complete round-trip to the server. You can click the bars in the chart to drill down into the details of a request (User Request Details page).
Component Trace	PSPMCOMPPERF	PeopleTools, Performance Monitor, Analytics, Component Trace	Only used in conjunction with a performance trace. This page enables you to isolate the component performance that is associated with a particular performance trace.
Component Statistics	PSPMCOMPSTAT	PeopleTools, Performance Monitor, Analytics, Component Statistics	Displays performance metrics that are related to the SQL, PeopleCode, and buffer sizes of PeopleSoft components. In addition, the page displays standard deviations and average charts.
Portal Statistics	PSPMPORTALSTATS	PeopleTools, Performance Monitor, Analytics, Portal Statistics	Displays portal activity duration per user ID within a specified interval.

<b>Page Name</b>	<b>Definition Name</b>	<b>Navigation</b>	<b>Usage</b>
PIA Statistics	PSPMPIASTATS	PeopleTools, Performance Monitor, Analytics, PIA Statistics	Displays PIA activity duration and response size per user ID within a specified interval.
Top Portal Content Requests	PSPMTOPPORTAL	PeopleTools, Performance Monitor, Analytics, Top Portal Content Requests	Enables you to identify candidates for performance tuning by displaying the most requested content and the associated performance measurements.
Top PeopleCode Events	PSPMPCODEEVENTS	PeopleTools, Performance Monitor, Analytics, Top PeopleCode Events	Enables you to identify candidates for performance tuning by examining the top execution count, cumulative durations, and average durations of PeopleCode events.
Top PeopleCode Executions	PSPMTOPPCODE	PeopleTools, Performance Monitor, Analytics, Top PeopleCode Executions	Enables you to identify candidates for performance tuning by examining the top execution count, cumulative durations, and average durations of PeopleCode executions.
Top SQL Statements	PSPMSQLEVENTS	PeopleTools, Performance Monitor, Analytics, Top PeopleCode SQL Statements	Enables you to identify candidates for performance tuning by examining the top execution count, cumulative durations, and average durations of SQL statements that are executed by PeopleCode programs.
Top Components	PSPMCOMPEVENTS	PeopleTools, Performance Monitor, Analytics, Top Components	Enables you to identify candidates for performance tuning by examining the top execution count, cumulative durations and average durations of component actions.

---

## Viewing User Request Analytics

Access the User Requests page (PeopleTools, Performance Monitor, Analytics, User Requests).

This page shows top-level, completed PMUs that were initiated by a specific user ID. A *top-level* PMU is a PMU with no parent PMU. A *user request* is a set of PMUs that together make up a complete request to the monitored system.

---

**Note.** All rows in the User Request chart appear in ascending order by agent start date and time. Read the chart from the bottom to the top. If the date is the same between two adjacent rows, the system hides the date and shows only the time. If two rows have identical dates and times, the system adds a number (#x) to the label to differentiate the two bars. The charting tool requires the labels of the bars to be distinct.

---

You must enter either the user ID or the performance trace name as part of the search criteria.

<b>System ID</b>	Identifies each monitored system. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.
<b>Database Name</b>	The name of the PeopleSoft application database running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.
<b>User ID</b>	<p>The user ID of the user who generated the request.</p> <p>The User ID field prompts against the PSPMOPRDEFN table, which is populated by the lookup program (PSPM_LOOKUP). When searching on a user ID, generally, the system attempts to match a value that is stored in the OPRID (user ID) field of the PMU history table. However, because some top-level PMUs have "_unknown_" as the OPRID, the system attempts a second match. If the top-level PMU contains "_unknown_" as the OPRID, the system searches for a PMU 113 (Authenticate) in the same PMU set (same top instance ID) where PMU 113 contains the given user in the OPRID field.</p>
<b>Performance Trace Name</b>	<p>You enter the performance trace name to search on PMUs within a particular performance trace. If a user has launched a performance trace in the Performance Console, the system labels every PMU that is generated in a business process with a performance trace name.</p> <p>Performance Trace Name prompts against the PSPMPERFTRACE table, which is populated by the lookup program (PSPM_LOOKUP).</p> <hr/> <p><b>Note.</b> When searching using a performance trace name, the search ignores all agent IDs, so the current system ID is not relevant.</p> <hr/>
<b>From/To Date/Time</b>	Specify a range of time. The system searches on the monitor date and time, not the agent date and time. If you do not specify a time range, by default, the system displays information for the current day, starting at midnight (12:00 AM).

**Static Scale**

Enables you to choose the scale of the X-axis in the User Requests chart that is generated from the search results. The search results appear in result sets, which are groups of 10. Select this option to view a static scale.

---

**Note.** After you select this option, you can't clear Static Scale until you refresh the page from the browser or reenter the page.

---

The system behavior for each type of scale is:

- Dynamic scale.

The system scales the X-axis according to the longest duration in the current result set.

- Static scale.

A static scale displays the same scale for the X-axis across all of the result sets. A static scale is based on the largest duration, or bar, that exists among all of the result sets.

**Search**

The Search button performs a search based on the search criteria that is defined on the page. If you want to view the most current information, use the Refresh button after running the initial search. If you click Search repeatedly, the system returns the same results from the same time range for each search.

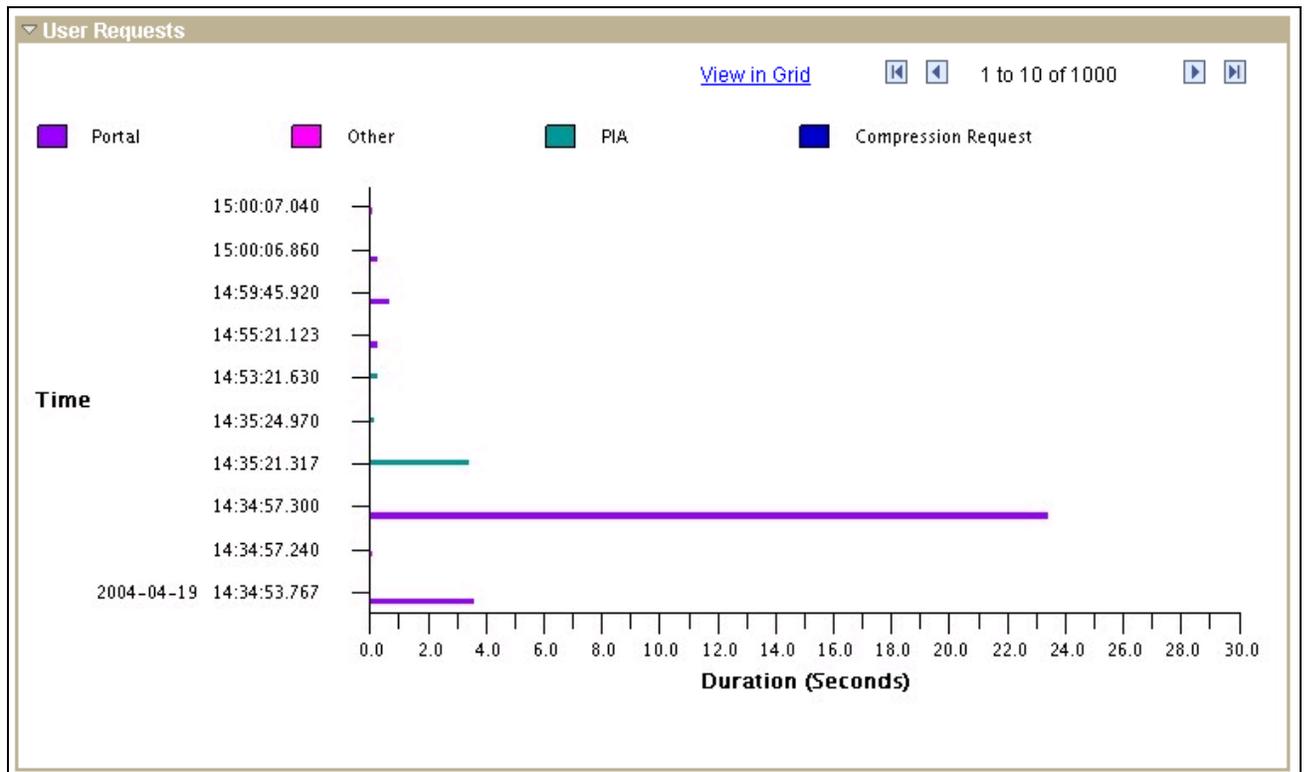
**Refresh**

Updates the To Time value to equal the current time before performing a search. This enables you to view the most current results of a search.

## The User Requests Chart

The User Requests chart reveals the time that a user initiated a specific request and its duration. The system displays user requests in sets of 10.

A single user request is represented by a bar within the bar chart. You can click any bar in the graph to view the associated User Request Detail component, which enables you to drill down into the details of a user request.



User Requests chart

On this chart, the View in Grid hyperlink displays the chart information in a grid for downloading and customized sorting. The grid contains a hyperlink that accesses the User Request Details page.

The rows on the chart contain pop-up messages that appear when you move the cursor over the bar. The messages are:

- For PMU 100 (Portal Request), the message reads Portal: [action] Duration (secs): yyy.
- For PMU 101 (PIA Request), the message reads PIA: [action] Duration (secs): yyy.
- For PMU 102 (Compression Request), the message reads Compression Request: [content type] Duration (secs): yyy.
- All other PMUs show the PMU label followed by the duration.

See [Chapter 8, "Working with Performance Monitor Charts and Analytics," User Request Details, page 119.](#)

## The Request Time vs. Think Time Chart

The Request Time vs. Think Time pie chart reveals the ratio of the amount of time that is required for the system to process the requests in the result set against the amount of network latency and user think time. This measurement is based on the time range that you entered in the search criteria.

This chart illustrates the impact that response time has on the productivity of users in the monitored system.

## User Request Details

The User Request Detail page appears *only* after you click a bar in the User Requests chart on the User Requests page. No PeopleSoft portal navigation exists for this page.

User requests are the set of PMUs that together make up a complete server round trip. All of the PMUs in a user request have the same top instance ID.

This page displays additional information for every user request that appears within the User Requests chart.

<b>Duration</b>	The duration of the top-level PMU instance for the current user request.
<b>Agent Start Date/Time</b>	The date and time that the top-level PMU began according to the clock on the monitored system.
<b>Monitor Received Date/Time</b>	The date and time that the monitoring system inserted performance information related the top-level PMU into the monitoring database.
<b>PMU Details</b>	This link accesses the completed PMU component where the system automatically loads all the PMUs with the same top PMU instance into this grid.

### *Application Server Totals*

<b>Application Server Time</b>	The total time spent in the application server.
<b>Serialization Time</b>	The time spent serializing data.
<b>Deserializaion Time</b>	The time spent deserializing data.
<b>Total Cache Misses</b>	The total number of application server cache misses for this user request. Cache misses occur when the system must retrieve metadata from the database as opposed to the application server cache. This value is the sum of metric 1 from all PMU 412 (Tuxedo Service Summary PMU) rows with the same top PMU instance.

---

**Note.** Continued high cache misses or file hits indicate cache problems that should be investigated.

---

<b>Total Cache File Hits</b>	The total number of application server cache memory hits for this user request. Cache memory hits occur when the system retrieves metadata from the application server cache file. It is the sum of metric 2 from all Tuxedo Service Summary PMU 412 (Tuxedo Service Summary) rows with the same top PMU instance.
<b>SQL Total Time</b>	The total time that is spent processing SQL for this user request.
<b>SQL Exec Count</b>	The total number of SQL statements that were run in this user request. This value is the sum of metric 6 from all PMU 400 (Tuxedo Service PCode and SQL PMU) rows with the same top PMU instance.

<b>SQL Fetch Count</b>	The total number of SQL fetches that were run in this user request. This value is the sum of metric 5 from all PMU 400 (Tuxedo Service PCode and SQL PMU) rows with the same top PMU instance.
<b>Global Variable Size</b>	The amount of memory that is used to store PeopleCode global variables. This is the sum of metric 6 from all the PMU 412 (Tuxedo Service Summary) rows with the same top PMU instance.  The greater the amount of data and memory that is required for global variables, the greater the overhead that is associated with a request.
<b>Total Cache Memory Hits</b>	The metric that is associated with the PMU 412 (Tuxedo Service Summary). Cache memory hits occur when the system retrieves metadata from the application server memory, not the cache file or the database.

### **Component PMUs**

This section displays these PMUs that are associated with a user request:

- PMU 401 (ICPanel): The ICPanel request.

Clicking the ICPanel link launches a new browser window, which contains the component statistics for that component default from midnight to the current time.

- PMU 402 (Modal Level 1): The first modal level if one exists.

The first modal level is triggered by think-time PeopleCode, such as DoModal, DoCancel, and so on.

- PMU 403 (Modal Level 2): The second modal level if one exists.

The second modal level contains the modal components that are called from the first modal level, such as secondary pages or lookup pages.

<b>PMU</b>	The PMU label that is defined in the PMU definition.  Click this link to go to the Component Statistics page for the current component so that you can compare historical performance data for that component.
<b>Name</b>	The name of the component containing the page that is running the request.
<b>Page</b>	The name of the page running the request.
<b>Action</b>	The component action string, passed from the web server to the application server, such as Save, Search, and so on.
<b>Duration</b>	Duration of the PMU on that line in the grid.

**Component Buffer Size** The component buffer size metric in PMU 401 (ICPanel), 402 (Modal Level 1), and 403 (Modal Level 2).

The component buffer must be deserialized by the application server when it is received from the web server, and then serialized when it is sent back to the web server. The larger the buffer, the greater the impact is of serialization overhead on the monitored system.

### ***Queries and Scripts***

This section displays the metrics of these PMUs that are associated with the user request:

- PMU 410 (ICScript): metrics that are related to invoking an ICScript.
- PMU 411 (ICQuery): metrics that are related to invoking an ICQuery.

<b>PMU</b>	The type of PMU: ICScript or ICQuery.
<b>Name</b>	Name of the ICScript or ICQuery.
<b>Action</b>	The action string, passed from the web server to the application server.
<b>Duration</b>	The duration of the ICScript or ICQuery.

### ***Tuxedo Service Information***

This section displays the metrics of all PMU 400s (Tuxedo Service PCode and SQL) that are associated with the current user request.

<b>Duration</b>	Duration of PMU 400 (Tuxedo Service PCode and SQL).
<b>Tuxedo Service Name</b>	The name of the Tuxedo service that was run.
<b>Server Name</b>	The name of the server process on the application server that is running the request.
<b>Host/Port</b>	The name of the server and the JSL port.

### ***User Request Durations***

The User Request Durations summarizes how time was spent servicing the user request. For example, it might show that the most time was spent in SQL.

<b>Portal Time</b>	The total time that was spent in the portal servlet for this request, including waits for calls to other servers. The sum of the duration of the PMU 100s (Portal Request) that are associated with the request.
--------------------	--

<b>PIA Time</b>	The total time that was spent in the PIA servlet. It is a subset of the portal time if the PIA request originated from Portal. The sum of the duration of the PMU 100s (Portal Request) and PMU 106s (PIA Request From Portal) that are associated with the request.
<b>Tuxedo Service Time</b>	The total time that was spent in the application server. It is a subset of the PIA time for a PIA request. A large difference between the duration of the PIA bar and the Tuxedo bar indicates that the request spent time on the web server or was queued by Tuxedo. The value is equal to the sum of the duration of the PMU 400 (Tuxedo Service PCode and SQL) rows.
<b>PeopleCode Exec Time</b>	The total time that was spent by the application server running PeopleCode, including any SQL that was triggered from within the PeopleCode. It is a subset of the Tuxedo Service Time. The value is equal to the sum of metric 1 of the PMU 400 (Tuxedo Service PCode and SQL) rows that are associated with the request.
<b>PeopleCode SQL Time</b>	This is the total time that was spent by the application server running SQLExecs and SQL objects that were triggered from within the PeopleCode. It is a subset of the PeopleCode Exec Time. The value is equal to the sum of metric 2 of the PMU 400 (Tuxedo Service PCode and SQL) rows that are associated with the requests.
<b>PeopleCode Built-in SQL Time</b>	This is the total time that was spent by the application server running SQL that was triggered from within PeopleCode built-in functions such as GetNextNumber(). It is a subset of the PeopleCode Exec Time. The value is equal to the sum of metric 3 of the PMU 400s (Tuxedo Service PCode and SQL) that are associated with the requests.
<b>PeopleTools SQL Exec Time</b>	This is the total time that was spent by the application server running SQL that was triggered by the PeopleTools runtime. It is a subset of the Tuxedo Service Time. It may overlap with the PeopleCode Exec time. The value is equal to the sum of metric 4 of the PMU 400 (Tuxedo Service PCode and SQL) rows that are associated with the requests.

---

**Note.** The number of bars appearing in the graph vary per request type.

---

### ***PeopleCode Durations***

This chart is a summary showing which PeopleCode event types (such as SavePreChange, SavePostChange) were triggered for this user request, how many times each type was triggered, and how much time was spent in running PeopleCode for each event type.

The PeopleCode Durations chart contains data only if a user accessed a component or ran an iScript while the filter level was set at verbose. The data is derived from PMUs 500–516.

These PMUs contain metrics for the cumulative execution count and execution time at the record field, component record, page, and component-level for each PeopleCode event type.

The number of times that a particular PMU ran appears in parenthesis.

All charts have a View in Grid hyperlink to show the chart information in a grid. This enables you to sort the information and download the information to Excel spreadsheets.

---

## Viewing Component Trace Information

Access the Component Trace page (PeopleTools, Performance Monitor, Analytics, Component Trace).

To view the Component Trace page, a user must have completed a performance trace. This page enables you to isolate the component performance information within a particular performance trace. The information presented applies *only* to component processing.

---

**Note.** To gain the most performance data and a clearer insight into component activity, you should run the performance trace that is generating the trace information in debug or verbose mode.

---

<b>Performance Trace Name</b>	Displays the name of the trace, which is typically the end user's user ID, but can be whatever name was entered by the end user in the Performance Trace Name edit box on the Performance Trace Console.
<b>Monitor Start Date and Time</b>	Displays the date and time that the monitor began receiving performance information for a particular performance trace.
<b>First Component</b>	Identifies the first component that is accessed during the performance trace.
<b>Component Cache Status</b>	Enables you to determine whether the objects that are associated with a component were accessed from the application server cache or retrieved from the database during the performance trace.
<b>Server Round Trips</b>	Displays the number of server round trips that are related to component processing that were completed during a particular performance trace.
<b>SQL Executes</b>	Displays the number of SQL programs that were run during component processing.
<b>SQL Fetches</b>	Displays the number of SQL fetches that were run during component processing.
<b>PeopleCode Program Executions</b>	Displays the number of PeopleCode programs that were run during component processing.
<b>Email XML File</b>	Enables you to email the data from the Component Trace page in an XML format so that you can share the information with colleagues or PeopleSoft consultants.

## Warnings

The Warnings section contains any warnings that may alert you to situations in the trace that may cause discrepancies in or misunderstandings of the trace results. For example, if particular component objects were not fully cached at the time of the trace, a warning appears notifying you of this. Production systems usually cache all frequently accessed objects for better performance. Objects that are not cached can skew results because after an object is fully cached, performance when accessing that object typically increases dramatically. The Description column provides details of the warning and provides considerations or remedies where appropriate.

A warning message appears if the performance trace is not run in debug mode.

## Duration Summary

You see a duration summary if the user ran the performance trace with standard or higher filter level.

The Duration section contains information regarding the duration of the entire performance trace as well as other elements of the performance trace, such as the time spent processing SQL and the time spent processing PeopleCode. For example, the duration information may show that the time for the total trace was 50 seconds, and 22 seconds were spent processing SQL while 15 seconds were spent processing PeopleCode.

---

**Note.** The duration total is not a sum of the measurements appearing here. The PeopleSoft system does not instrument every aspect of the runtime, and these areas that are not instrumented account for the difference between the total duration and the total of the measurements that are displayed here.

---

<b>Measurement</b>	Displays the measurement appearing in the row, such as SQL, PeopleCode, or Total Trace.
<b>Duration</b>	Displays the total time that was consumed by each element of a performance trace. For example, it may show the total trace consuming 50 seconds and 22 seconds were attributed to SQL.
<b>% of Total</b>	Displays the percentage of the total trace value that each element of the trace consumed. For example, if the total trace value is 50 seconds and the SQL value is 22 seconds, the % of total value is 44.

## Server Round Trips

The Server Round Trip section contains information for each server round trip that was completed during a particular performance trace. It contains two tabs: Durations and PeopleTools State Management.

You see a duration summary if the user ran the performance trace with standard or higher agent filter level.

The Durations tab contains the following information:

<b>Seq (Sequence)</b>	Displays the sequential order in which the server round trips were run, with 1 indicating the first round trip of the performance trace.
-----------------------	--

<b>Action</b>	Displays the action that was completed during the server round trip, such as Load Page/Search Page, Load Search Result, Click PeopleCode Command Button for field FUNCLIB <table.field>.  This column is a hyperlink that launches the Round Trip Details page.
<b>Component</b>	Displays the component from which the user invoked the action.
<b>Page</b>	Displays the page within the component from which the user invoked the action.
<b>Duration</b>	Displays the amount of time that each action, or server round trip, consumed.
<b>% of Total Trace</b>	Displays the percentage of the total performance trace duration that each server round trip consumed. For example, if the total trace value is 50 seconds and a particular action consumes 3.5 seconds, the % of total trace is 7.

The PeopleTools State Management tab contains the following additional information:

<b>Pack/Unpack Time</b>	Reflects the time that was required for serialization (pack) and deserialization (unpack). The value is derived as follows:  PMU 427 + 428 duration - (time spent performing SQL in 427 and 428)  The value reflects the serialization duration plus the deserialization duration minus any time spent performing SQL as part of serialization and deserialization.
<b>Component Buffer Size</b>	The component buffer size, metric 1, in PMU 401 (ICPanel). The component buffer has to be deserialized by the application server when it is received from the web server, and then serialized when it is sent back to the web server.  The larger the buffer, the greater is the impact of serialization overhead on the monitored system.
<b>PeopleCode Global Size</b>	This is the size of the memory that is used for PeopleCode global variables, which are sent to and from the application server during a Jolt request. This is included in the send buffer size and receive buffer size.  This is metric 5 for all PMU 412 (Tuxedo Service Summary) rows.

### **SQL Summary**

The SQL Summary section contains additional details regarding the SQL activity during component processing. It contains two tabs: SQL Executes and SQL Fetches.

You see a SQL summary if the user ran the performance trace with verbose or higher agent filter level.

The SQL Executes tab contains the following information:

<b>Seq (Sequence)</b>	The system orders the rows by total duration. The sequence in this case is a ranking of total duration.
<b>SQL Operation and Tables</b>	Displays the SQL operation and tables (such as UPDATE PS_QUERY_RUN_QRYVW).

<b>SQL Type</b>	Reflects the type of source code making the SQL call. The SQL types are: <ul style="list-style-type: none"> <li>• PeopleTools (PMU 408).</li> <li>• Inline PeopleCode (PMU 406).</li> <li>• PeopleCode Built-In (PMU 407).</li> </ul>
<b>Execution Count</b>	Displays the number of times that the SQL was run during the performance trace.
<b>Total Duration</b>	The total time required by all executions.
<b>Total Duration Average</b>	The average execution duration, as in Total Duration / Execution Count.

The SQL Fetches tab contains the following additional information for each SQL statement that was run:

<b>Fetch Count</b>	The total count of all SQL fetches that were run during component processing within a particular performance trace.
<b>Fetch Count Average</b>	The average number of fetches per SQL Execute.
<b>Fetch Duration</b>	The sum total of all durations for SQL fetches.
<b>Fetch Duration Average</b>	The system average for SQL fetch durations during component processing.

### ***PeopleCode Summary***

The PeopleCode Summary section contains additional details regarding the PeopleCode activity during component processing. These values relate to PMU 404 (PeopleCode Program Execution).

You see a PeopleCode summary only if the user ran the performance trace in debug mode.

The following metrics apply only to the performance information that was collected during the performance trace.

<b>Seq (Sequence)</b>	The system orders the rows by .
<b>PeopleCode Program</b>	Displays the name of the PeopleCode program that was run.
<b>Execution Count</b>	Displays the number of times this program has been run.
<b>Duration Total</b>	Displays the sum of all durations of the PeopleCode program.
<b>Duration Average</b>	Displays the average duration of the PeopleCode program.
<b>Duration Maximum</b>	Displays the longest duration of this PeopleCode program.
<b>Duration Minimum</b>	Displays the shortest duration of this PeopleCode program.

## Round Trip Details

The Round Trip Details page contains a subset of the performance data that appears on the Component Trace page. It enables you to drill down into a specific round trip to gather duration, SQL, and PeopleCode information. The data that is presented in the page is similar to the data in the Performance Trace page. The PMU Details link accesses the completed PMU page with all PMUs that were captured in this action.

---

## Viewing Component Statistics

Access the Component Statistics page (PeopleTools, Performance Monitor, Analytics, Component Statistics).

The Component Statistics page enables you to gather performance information that is related to a specific PeopleSoft component.

<b>System ID</b>	Identifies each monitored system. PeopleSoft automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.
<b>Database Name</b>	The name of the PeopleSoft application database that is running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.
<b>User ID</b>	The user ID of the user who generated the request. The user ID field prompts against the PSPMOPRDEFN table, which is populated by the Lookup program (PSPM_LOOKUP).
<b>Performance Trace Name</b>	You enter the performance trace name to search on PMUs within a particular performance trace. If a user has launched a performance trace in the Performance Console, the system labels every PMU that is generated in a business process with a performance trace name. Performance Trace Name prompts against the PSPMPERFTRACE table, which is populated by the Lookup program (PSPM_LOOKUP).
<hr/>	
	<b>Note.</b> When searching using a performance trace name, the search ignores all agent IDs, so the current system ID is not relevant.
<b>From/To Date/Time</b>	Specify a range of time. The system searches on the monitor date and time, not the agent date and time. If you do not specify a time range, by default, the system displays information for the current day, starting at midnight (12:00 AM).
<b>Component</b>	Select the component name.
<b>Market</b>	Select the market to which the component belongs.

<b>Search</b>	The Search button performs a search based on the search criteria that is defined on the page. It should be pressed only once per search. If you want to view the most current information, use the Refresh button after running the initial search. If you click Search repeatedly, the system returns the same results from the same time range for each search.
<b>Refresh</b>	Updates the To Time value to equal the current time. This enables you to view the most current results of a search.
<b>Component Execution Count</b>	The number of times that a component was accessed within the specified time period.  This value is calculated by selecting the count of PMU 400 (Tuxedo Service PCode and SQL) rows that were found with the current search results.

### ***Component Averages***

The Component Averages section displays the average and standard deviation of the following metrics.

<b>Duration</b>	The average duration of the PMU 400 (Tuxedo Service PCode and SQL) rows that were run from this component.
<b>SQL Fetch Count</b>	The average number of SQL fetches that were run for this component.  The value is the average of metric 5 from all PMU 400 (Tuxedo Service PCode and SQL) rows with Context 1 equal to this component's name.
<b>SQL Execute Count</b>	The average number of SQL statements that were run for this component. The average of metric 6 from all PMU 400 (Tuxedo Service PCode and SQL) rows with Context 1 equal to this component's name.

### ***Component Sizes***

The Component Sizes chart displays the average and standard deviation of the component buffer sizes.

<b>Receive Buffer Size</b>	The average size of the data that the application server received from the web server during a JOLT request.  This is metric 2 for PMU 115 (JOLT Request).
<b>Send Buffer Size</b>	Size of the data that was sent from the web server to the application server during a JOLT request.  This is metric 1 for PMU 115 (JOLT Request).
<b>Component Buffer Size</b>	This is the average component buffer size (metric 1) in PMU 401 (ICPanel).  The component buffer has to be deserialized by the application server when it is received from the web server, and then serialized when it is sent back to the web server.

**PeopleCode Global Size** This is the average size of the memory that is used for PeopleCode global variables, which are sent to and from the application server during a JOLT request. This is included in the send buffer size and receive buffer size.

This is metric 6 for all PMU 412 (Tuxedo Service Summary) rows.

### **Component Durations**

This chart shows the average and the standard deviation of these metrics:

**Portal Time** The average time that was spent in the portal servlet for this component, including waits for calls to other servers. The value is the duration of the PMU 100 (Portal Request) rows that are associated with requests that are accessing this component.

**PIA Time** The average time that was spent in the PIA servlet for this component. The value is the duration of the PMU 100 (Portal Request) rows and PMU 106 (PIA Request From Portal) rows that are associated with the request accessing this component.

**Tuxedo Service Time** The average time that was spent in the application server for this component. The value is the duration of the PMU 400 (Tuxedo Service PCode and SQL) rows that are associated with the request that is accessing this component.

**PeopleCode Exec Time** The average time that was spent by the application server running PeopleCode, including any SQL that was triggered from within the PeopleCode.

The value is the average of metric 1 of the PMU 400 (Tuxedo Service PCode and SQL) rows that are associated with the component.

**PeopleCode SQL Time** This is the average time that was spent by the application server running SQLExecs and SQL objects that were triggered from within the PeopleCode. It is a subset of the PeopleCode Exec Time.

The value is equal to the average of metric 2 of the PMU 400 (Tuxedo Service PCode and SQL) rows that are associated with the requests.

---

**Note.** This is the average time that was spent for running all SQL statements of this type in a single application server request, not the average duration of one SQL statement of this type.

---

**PeopleCode Built-in SQL Time** This is the average time that was spent by the application server running SQL that was triggered from within PeopleCode built-in functions such as GetNextNumber(). It is a subset of the PeopleCode Exec Time.

The value is the average of metric 3 of the PMU 400 (Tuxedo Service PCode and SQL) rows that are associated with the requests.

---

**Note.** This is the average time that was spent for running all SQL statements of this type in a single application server request, not the average duration of one SQL statement of this type.

---

**PeopleTools SQL Exec Time**

This is the average time that was spent by the application server running SQL that was triggered by the PeopleTools runtime. It is a subset of the Tuxedo Service Time. This value may overlap with the PeopleCode Exec time.

The value is the average of metric 4 of the PMU 400 (Tuxedo Service PCode and SQL) rows that are associated with the requests.

---

**Note.** This is the average time that was spent for running all SQL statements of this type in a single application server request, not the average duration of one SQL statement of this type.

---

**PeopleCode Durations**

The PeopleCode Durations chart contains data only if a user accessed a component while the monitoring level was set at verbose. The data is derived from PMUs 500–516.

This chart is a summary showing:

- Which PeopleCode event types (such as SavePreChange and SavePostChange) were triggered for this component at the record field, component record, page, and component levels.
- The average time that was spent running each event instance.
- The sample count that was used to calculate these averages.

The standard deviation is not calculated for this chart.

---

## Viewing Portal Statistics

Access the Portal Statistics page (PeopleTools, Performance Monitor, Analytics, Portal Statistics).

This page enables you to view the performance of the portal servlet. All charts have a View in Grid hyperlink to show the chart information in a grid. This enables you to sort the information and to download to Microsoft Excel spreadsheets.

The chart on this page is generated from all Portal Request PMUs 100 (Portal Request) rows grouped by action (login, logout, homepage, target page, error), all PMU 117 (GetContent) rows, and all PMU 116 (Redirect after Login) rows.

Before you view the charts, you must enter search criteria.

**System ID**

Identifies each monitored system. PeopleSoft automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.

**Database Name**

The name of the PeopleSoft application database running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.

<b>User ID</b>	The user ID of the user who generated the request. The User ID field prompts against the PSPMOPRDEFN table, which is populated by the Lookup program (PSPM_LOOKUP).
<b>Performance Trace Name</b>	You enter the performance trace name to search on PMUs within a particular performance trace. If a user has launched a performance trace in the Performance Console, the system labels every PMU that was generated in a business process with a performance trace name. Performance Trace Name prompts against the PSPMPERFTRACE table, which is populated by the Lookup program (PSPM_LOOKUP).
<hr/>	
<b>Note.</b> When searching using a performance trace name, the search ignores all agent IDs, so the current system ID is not relevant.	
<hr/>	
<b>From/To Date/Time</b>	Specify a range of time. The system searches on the monitor date and time, not the agent date and time. If you do not specify a time range, by default, the system displays information for the current day, starting at midnight (12:00 AM).
<b>Search</b>	The Search button performs a search based on the search criteria that is defined on the page. It should be pressed only once per search. If you want to view the most current information, use the Refresh button after running the initial search. If you click Search repeatedly, the system returns the same results from the same time range for each search.
<b>Refresh</b>	Updates the To Time value to equal the current time. This enables you to view the most current results of a search.

### ***Portal Action Durations***

This chart reveals the average and standard deviation of the durations of the following metrics. The sample size that is used to derive the average appears in parentheses to the right of the bar label.

<b>Redirect</b>	The time that was required for the browser to load a page that is requested by way of a URL redirection.
<b>Get Content</b>	The time that was required for the system to display target content.
<b>Portal View page</b>	The time that was spent to assemble a portal page, excluding the homepage.
<b>Portal login</b>	The time that was spent on processing a login request.
<b>Portal Homepage</b>	The time that was spent assembling the homepage.
<b>Portal error</b>	The time that was spent assembling the error reporting page.
<b>Portal Expire</b>	The time that was spent processing a portal page expiration.
<b>Portal Logout</b>	The time that was spent on processing a logout request.

**Portal Refresh Pagelet** The time that was spent refreshing a pagelet. A pagelet refresh gets triggered by the refresh button on a pagelet from the homepage.

---

## Viewing PIA Statistics

Access the PIA Statistics page (PeopleTools, Performance Monitor, Analytics, PIA Statistics).

This page enables you to view the performance of the PIA servlet.

All charts have a View in Grid hyperlink to show the chart information in a grid. This also enables users to sort the information and to download to Microsoft Excel spreadsheets.

**System ID** Identifies each monitored system. The PeopleSoft system automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.

**Database Name** The name of the PeopleSoft application database running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.

**User ID** The user ID of the user who generated the request.  
The User ID field prompts against the PSPMOPRDEFN table, which is populated by the Lookup program (PSPM\_LOOKUP).

**Performance Trace Name** You enter the performance trace name to search on PMUs within a particular performance trace. If a user has launched a performance trace in the Performance Console, the system labels every PMU that is generated in a business process with a performance trace name.

Performance Trace Name prompts against the PSPMPERFTRACE table, which is populated by the Lookup program (PSPM\_LOOKUP).

---

**Note.** When searching using a performance trace name, the search ignores all agent IDs, so the current system ID is not relevant.

---

**From/To Date/Time** Specify a range of time. The system searches on the monitor date and time, not the agent date and time. If you do not specify a time range, by default, the system displays information for the current day, starting at midnight (12:00 AM).

**Search** The Search button performs a search based on the search criteria that is defined on the page. It should be pressed only once per search. If you want to view the most current information, use the Refresh button after running the initial search. If you click Search repeatedly, the system returns the same results from the same time range for each search.

**Refresh** Updates the To Time value to equal the current time. This enables you to view the most current results of a search.

### ***PIA Action Durations***

The average time for a PIA servlet to handle a direct request from a browser (PIA View Page), and handle a request from a Portal.

The data is generated from PMU 101 (PIA Request), and the data is grouped by context 3, which is action (login, logout, homepage, target page, error). Additional performance data is generated from PMU 102 (Compression Request), PMU 106 (PIA Request From Portal), and PMU 114 (Attach File Request).

### ***PIA Action Response Sizes***

The average size of the content sent to a browser when fulfilling a PIA request for the same rows of data that appear in the PIA Action Durations chart.

The data is generated from the metric 1 values for PMU 101 (PIA Request), metric 1 values for 106 (PIA Request from Portal), metric 1 values for PMU 114 (Attach File Request), and metric 1 values for PMU 102 (Compression Request).

## **Viewing Top Portal Content Requests**

Access the Top Portal Content Requests page (PeopleTools, Performance Monitor, Analytics, Top Portal Content Requests).

This page enables you to identify, view, and rank the performance data that is associated with individual requests for content through the PeopleSoft portal. The charts enable you to identify content requests that are both popular and performing poorly.

All charts have a View in Grid hyperlink to show the chart information in a grid. This enables you to sort the information and to download to Microsoft Excel spreadsheets.

Before you can view portal content requests, you must first enter search criteria.

<b>System ID</b>	Identifies each monitored system. PeopleSoft automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.
<b>Database Name</b>	The name of the PeopleSoft application database running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.
<b>User ID</b>	The user ID of the user who generated the request. The User ID field prompts against the PSPMOPRDEFN table, which is populated by the Lookup program (PSPM_LOOKUP).

<b>Performance Trace Name</b>	<p>Enter the performance trace name to search on PMUs within a particular performance trace. If a user has launched a performance trace in the Performance Console, the system labels every PMU that is generated in a business process with a performance trace name.</p> <p>Performance Trace Name prompts against the PSPMPERFTRACE table, which is populated by the Lookup program (PSPM_LOOKUP).</p> <hr/> <p><b>Note.</b> When searching using a performance trace name, the search ignores all agent IDs, so the current system ID is not relevant.</p> <hr/>
<b>From/To Date/Time</b>	Specify a range of time. The system searches on the monitor date and time, not the agent date and time. If you do not specify a time range, by default, the system displays information for the current day, starting at midnight (12:00 AM).
<b>Search</b>	The Search button performs a search based on the search criteria that is defined on the page. It should be pressed only once per search. If you want to view the most current information, use the Refresh button after running the initial search. If you click Search repeatedly, the system returns the same results from the same time range for each search.
<b>Number to Retrieve</b>	The number of top ranked statistics to be displayed on the charts.
<b>Refresh</b>	Updates the To Time value to equal the current time. Enables you to view the most current results of a search.

### ***Top Portal Content Requests***

This chart indicates the most popular content that is requested by users through the portal.

The chart shows the top counts of content reference instances for PMU 117 (GetContent) that were accessed within the search period specified. The instances are grouped by context 3, which assumes one of the following values:

- Portal CREF ID of the content.
- Pagelet (appears when the template references the pagelet by URL rather than by CREF ID).
- Menu (the CREF ID of the menu pagelet).
- DEFAULT\_TEMPLATE (the CREF ID of the default template CREF).
- Homepage Tab <tab name>.
- Target content (appears when the content is unregistered target content).
- Logout (appears when a portal issues a logout request to another portal).

### ***Top Cumulative Durations***

This chart shows the portal content references that in total consumed the most time on the Portal servlet and their request counts. The system selects the top duration sums for PMU 117 (GetContent) grouped by Context3 (Portal CREF ID) that were accessed within the search period specified.

### ***Duration Averages***

This chart shows the portal content references that on average consumed the most time on the Portal servlet and their request counts. The system selects the top duration averages for PMU 117 (GetContent) grouped by Context3 (Portal CREF ID) that were accessed within the search period specified.

### ***Top Cumulative Response Sizes***

This chart indicates the largest cumulative response sizes. A response size refers to the size, in bytes, of the HTML response that the system sends to the browser when fulfilling a user request.

The data is generated from metric 1 in PMU 117 (GetContent).

### ***Top Average Response Sizes***

This chart indicates the top average of all of the response sizes. A response size refers to the size, in bytes, of the HTML response that the system sends to the browser when fulfilling a user request.

The data is generated from metric 1 in PMU 117 (GetContent).

## **Viewing Top PeopleCode Events**

Access the Top PeopleCode Events page (PeopleTools, Performance Monitor, Analytics, Top PeopleCode Events).

These charts show the PeopleCode events that have highest durations and execution counts, grouped by component and event type. Durations and counts are summed across levels (record field, component record, page and component) for each component and event type.

All charts have a View in Grid hyperlink to show the chart information in a grid. This also enables users to sort the information and to download to Microsoft Excel spreadsheets.

**Note.** Performance data must be captured in verbose agent filter mode or higher to be displayed.

Before you can view PeopleCode events, you must first enter search criteria.

<b>System ID</b>	Identifies each monitored system. PeopleSoft automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.
<b>Database Name</b>	The name of the PeopleSoft application database running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.
<b>User ID</b>	The user ID of the user who generated the request. The User ID field prompts against the PSPMOPRDEFN table, which is populated by the Lookup program (PSPM_LOOKUP).

<b>Performance Trace Name</b>	<p>You enter the performance trace name to search on PMUs within a particular performance trace. If a user has launched a performance trace in the Performance Console, the system labels every PMU that is generated in a business process with a performance trace name.</p> <p>Performance Trace Name prompts against the PSPMPERFTRACE table, which is populated by the Lookup program (PSPM_LOOKUP).</p> <hr/> <p><b>Note.</b> When searching using a performance trace name, the search ignores all agent IDs, so the current system ID is not relevant.</p> <hr/>
<b>Component</b>	Select the component name.
<b>Market</b>	Select the market to which the component belongs.
<b>From/To Date/Time</b>	Specify a range of time. The system searches on the monitor date and time, not the agent date and time. If you do not specify a time range, by default, the system displays information for the current day, starting at midnight (12:00 AM).
<b>Search</b>	The Search button performs a search based on the search criteria that is defined on the page. It should be pressed only once per search. If you want to view the most current information, use the Refresh button after running the initial search. If you click Search repeatedly, the system returns the same results from the same time range for each search.
<b>Number to Retrieve</b>	The number of top ranked statistics to be displayed on the charts.
<b>Refresh</b>	Updates the To Time value to equal the current time. Enables you to view the most current results of a search.

---

**Note.** If the filter level of the PSAPPSRV agent type is set to a level that is lower than the level of PMU 500–516 (Verbose), a warning message appears at the top of the page indicating that information for the current component may not be available. This data is available only in the following situations: the agent filters on the application server are set to verbose or debug, *or* performance traces were run with a filter level set to debug or verbose. A small, yellow triangle appears at the top of the page alerting you if application servers for that system are currently at a higher filter level than verbose. This, however, does not mean that you would not see data; PMUs may exist from performance traces or lower filter levels from past monitoring.

---

### ***Top PeopleCode Events***

This chart indicates the PeopleCode events that are run most often.

It shows the highest counts for all PMU 500–516. The counts consist of adding metrics 1, 3, and 5. The data is grouped by component and event name (contexts 1 and 2).

See [Chapter 4, "Administering the Performance Monitor," Setting Agent Filter Options, page 36.](#)

### ***Top Cumulative Durations***

This chart reveals the largest sum of all the durations of a specific PeopleCode event.

It shows the top sum of PMU 500–516 durations. The duration is not the duration of the PMU; the duration is the sum of metrics 2, 4, and 6. The data is grouped by component and event name (contexts 1 and 2).

### ***Duration Averages***

This chart shows the components with event types that on average consume the most time in application server requests.

Duration averages are the averages and standard deviations of the sum of durations of PMU 500–516. The duration is not the duration of the PMU; the duration is the sum of metrics 2, 4, and 6. The data is grouped by component and event name (contexts 1 and 2). The sample count that is used to calculate these averages is not the execution counts of the events (metrics 1, 3 and 5) but the number of times this component reported that event type.

---

## **Viewing Top PeopleCode Executions**

Access the Top PeopleCode Executions page (PeopleTools, Performance Monitor, Analytics, Top PeopleCode Executions).

This page enables you to identify the PeopleCode programs that the system runs the most and the performance of the PeopleCode programs.

The data on this page is generated from PMU 404 (PeopleCode Program Execution), and it is based on all PeopleCode function calls and all calls to application packages.

All charts have a View in Grid hyperlink to show the chart information in a grid. This enables you to sort the information and to download to Microsoft Excel spreadsheets.

---

**Note.** Performance data must be captured in debug agent filter mode to be displayed in this chart.

---

Before you can view top PeopleCode executions, you must first enter search criteria.

<b>System ID</b>	Identifies each monitored system. PeopleSoft automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.
<b>Database Name</b>	The name of the PeopleSoft application database running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.
<b>User ID</b>	The user ID of the user who generated the request. The User ID field prompts against the PSPMOPRDEFN table, which is populated by the Lookup program (PSPM_LOOKUP).

<b>Performance Trace Name</b>	<p>You enter the performance trace name to search on PMUs within a particular performance trace. If a user has launched a performance trace in the Performance Console, the system labels every PMU that is generated in a business process with a performance trace name.</p> <p>Performance Trace Name prompts against the PSPMPERFTRACE table, which is populated by the lookup program (PSPM_LOOKUP).</p> <hr/> <p><b>Note.</b> When searching using a performance trace name, the search ignores all agent IDs, so the current system ID is not relevant.</p> <hr/>
<b>Component</b>	Select the component name.
<b>Market</b>	Select the market to which the component belongs.
<b>From/To Date/Time</b>	Specify a range of time. The system searches on the monitor date and time, not the agent date and time. If you do not specify a time range, by default, the system displays information for the current day, starting at midnight (12:00 AM).
<b>Search</b>	The Search button performs a search based on the search criteria that is defined on the page. It should be pressed only once per search. If you want to view the most current information, use the Refresh button after running the initial search. If you click Search repeatedly, the system returns the same results from the same time range for each search.
<b>Number to Retrieve</b>	The number of top ranked statistics to be displayed on the charts.
<b>Refresh</b>	Updates the To Time value to equal the current time. Enables you to view the most current results of a search.

---

**Note.** This data is available only when the agent filters on the application server are set to debug, *or* performance traces were run with a filter level set to debug. A small, yellow triangle appears at the top of the page alerting you if application servers for that system are currently at a higher filter level than debug. This, however, does not mean that you would not see data; PMUs may exist from performance traces or lower filter levels from past monitoring.

---

### ***Top PeopleCode Executions***

Indicates the PeopleCode programs that the system runs the most.

Top PeopleCode executions refers to the highest counts for all PMU 404 (PeopleCode Program Execution) rows. The data is grouped by component and program name (contexts 1 and 2).

### ***Top Cumulative Durations***

Identifies the PeopleCode programs with the longest cumulative durations. The duration of each execution of a particular PeopleCode program is recorded and added to the sum total to create a cumulative duration value.

The data is generated from the sum of PMU 404 (PeopleCode Program Execution) durations, and the data is grouped by component and program name (contexts 1 and 2).

### ***Duration Averages***

Identifies the PeopleCode programs with the highest average duration in the system.

The data is generated from the sum of PMU 404 (PeopleCode Program Execution) durations, and the data is grouped by component and program name (contexts 1 and 2).

---

## **Viewing Top PeopleCode SQL Statements**

Access the Top PeopleCode SQL page (PeopleTools, Performance Monitor, Analytics, Top PeopleCode SQL).

The data for this page is generated from the sum of all PMU 406 (PeopleCode SQL Execute) rows, and the data relates to the SQL that PeopleCode programs run.

The bars on the chart are links that you can click to access the Completed PMU history component. The system automatically populates the search criteria with values derived from the Top PeopleCode SQL Statements page. For example, PMU Identifier is set to *406*, and the Context 2 and Statement Number values are set appropriately.

All charts have a View in Grid hyperlink to show the chart information in a grid. This also enables you to sort the information and to download to Microsoft Excel spreadsheets.

<b>System ID</b>	Identifies each monitored system. PeopleSoft automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.
<b>Database Name</b>	The name of the PeopleSoft application database running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.
<b>User ID</b>	The user ID of the user who generated the request. The User ID field prompts against the PSPMOPRDEFN table, which is populated by the Lookup program (PSPM_LOOKUP).
<b>Performance Trace Name</b>	You enter the performance trace name to search on PMUs within a particular performance trace. If a user has launched a performance trace in the Performance Console, the system labels every PMU that is generated in a business process with a performance trace name. Performance Trace Name prompts against the PSPMPERFTRACE table, which is populated by the Lookup program (PSPM_LOOKUP).
	<hr/> <b>Note.</b> When searching using a performance trace name, the search ignores all agent IDs, so the current system ID is not relevant. <hr/>
<b>Component</b>	Select the component name.
<b>Market</b>	Select the market to which the component belongs.

<b>From/To Date/Time</b>	Specify a range of time. The system searches on the monitor date and time, not the agent date and time. If you do not specify a time range, by default, the system displays information for the current day, starting at midnight (12:00 AM).
<b>Search</b>	The Search button performs a search based on the search criteria that is defined on the page. It should be pressed only once per search. If you want to view the most current information, use the Refresh button after running the initial search. If you click Search repeatedly, the system returns the same results from the same time range for each search.
<b>Number to Retrieve</b>	The number of top ranked statistics to be displayed on the charts.
<b>Refresh</b>	Updates the To Time value to equal the current time. Enables you to view the most current results of a search.

### ***Top SQL Statements***

This chart indicates the PeopleCode SQL statements that the system runs the most.

SQL statements are identified by PeopleCode locations, such as PeopleCode Program and PeopleCode Statement Number (context 2 and metric 2).

### ***Top Cumulative Durations***

Identifies the PeopleCode SQL statements with the longest cumulative durations.

SQL statements are identified by PeopleCode locations, such as PeopleCode Program and PeopleCode Statement Number (context 2 and metric 2).

### ***Duration Averages***

Identifies the PeopleCode SQL statements with the longest average durations.

SQL statements are identified by PeopleCode locations, such as PeopleCode Program and PeopleCode Statement Number (context 2 and metric 2).

---

**Note.** This data is available only when the agent filters on the application server are set to verbose or debug, or performance traces were run with a filter level set to debug or verbose. A small, yellow triangle appears at the top of the page alerting you if application servers for that system are currently at a higher filter level than verbose. This, however, does not mean that you would not see data; PMUs may exist from performance traces or lower filter levels from past monitoring.

---

---

## **Viewing Top Components**

Access the Top Components page (PeopleTools, Performance Monitor, Analytics, Top Components).

All charts have a View in Grid hyperlink to show the chart information in a grid. This enables you to sort the information and to download to Microsoft Excel spreadsheets.

The data on this page is generated from PMU 401 (ICPanel), and it is grouped by action, as in ICSave. The system passes the "action" to the application server as part of an ICPanel service request specifying the reason for the request.

On this page, the charts are grouped by action. The "action" refers to the action of a user in a component. The following are examples of how the actions appear:

- "#KEY" translates into Load Component.
- "#ICPanel1" translates into "Tab to Page."
- Launch Page/Search Page.
- Click PeopleCode Command Button for Field DERIVED\_PSPM.PM\_FILTERCHANGE.
- Switch to Add Action Mode.
- Launch Secondary Page.
- Change Drop Down List Value for Field PRCSRQSTDLG\_WRK.RECURNAME.
- Click OK or Save to Save Secondary Page.
- Click Cancel on Secondary Page.
- Save Component.
- Select Row0 From Search Result.
- Select Row1 From Search Result.
- Sort Column on Grid.
- Tab to Next Page.
- View Page.

Before viewing top components, you must first enter search criteria.

<b>System ID</b>	Identifies each monitored system. PeopleSoft automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.
<b>Database Name</b>	The name of the PeopleSoft application database running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.
<b>User ID</b>	The user ID of the user who generated the request. The User ID field prompts against the PSPMOPRDEFN table, which is populated by the Lookup program (PSPM_LOOKUP).

<b>Performance Trace Name</b>	<p>You enter the performance trace name to search on PMUs within a particular performance trace. If a user has launched a performance trace in the Performance Console, the system labels every PMU that is generated in a business process with a performance trace name.</p> <p>Performance Trace Name prompts against the PSPMPERFTRACE table, which is populated by the Lookup program (PSPM_LOOKUP).</p> <hr/> <p><b>Note.</b> When searching using a performance trace name, the search ignores all agent IDs, so the current system ID is not relevant.</p> <hr/>
<b>Component</b>	Select the component name.
<b>Market</b>	Select the market to which the component belongs.
<b>From/To Date/Time</b>	Specify a range of time. The system searches on the monitor date and time, not the agent date and time. If you do not specify a time range, by default, the system displays information for the current day, starting at midnight (12:00 AM).
<b>Search</b>	The Search button performs a search based on the search criteria defined on the page. It should be pressed only once per search. If you want to view the most current information, use the Refresh button after running the initial search. If you click Search repeatedly, the system returns the same results from the same time range for each search.
<b>Number to Retrieve</b>	The number of top ranked statistics to be displayed on the charts.
<b>Refresh</b>	Updates the To Time value to equal the current time. This enables you to view the most current results of a search.

### ***Top Components***

This chart indicates the components that users access and use the most.

The data is grouped by component, page, and action (contexts 1, 2, and 3).

### ***Top Cumulative Durations***

Identifies the component PMUs with the longest cumulative durations.

The data is grouped by component, page, and action (contexts 1, 2, and 3).

### ***Duration Averages***

This chart displays the average duration and standard deviation of the PMUs that are associated with a particular component.

The data is grouped by component, page, and action (contexts 1, 2, and 3).

## Chapter 9

# Performance Monitor Security Considerations

This chapter discusses how to:

- Set up PSPPMSRV authentication.
- Work with firewalls.
- Set up SSL.
- Set up SSL client authentication.
- Protect the Performance Monitor servlets.

---

## Setting Up PSPPMSRV Authentication

The PSPPMSRV application server process acts as the PPMI client when you record PeopleSoft performance data. To register as a client, the PSPPMSRV requires the appropriate permissions in PeopleTools Security.

---

**Warning!** Setting up PSPPMSRV authentication is required.

---

To set up PPMI Client Authentication:

1. Create a user profile using PeopleTools Security.
2. Add the *PeopleTools Perfmon Client* role to the user profile.

The *PeopleTools Perfmon Client* role contains the PTPMCLNT permission list.

3. Select PeopleTools, Performance Monitor, Administration, Global Administration and enter the user profile that you created and the associated password in the PPMI User ID and PPMI Password text boxes.

---

## Firewall Considerations

When setting up firewalls in a Performance Monitor environment, consider:

- Agent communication with the monitor servlet.
- PSPPMSRV communication with the monitor servlet.

- Monitor cluster members.

## Agent Communication with the Monitor Servlet

If you require a forward proxy to create a "bridge" for a firewall residing between the monitored system and the monitoring system, configure your web server, application server, and Process Scheduler server accordingly.

### **Web Server**

The process varies depending on which web server you use.

To set up a forward proxy on WebLogic:

1. Open the setenv.cmd file.
2. Set HTTP\_PROXY\_HOST and HTTP\_PROXY\_HTTPPORT, or HTTP\_PROXY\_HTTPSHOST and HTTP\_PROXY\_HTTPSPORT.
3. Restart the application server and Process Scheduler domain.

To set up a forward proxy on WebSphere:

1. Open WebSphere Administration console at `http://<machine-name>:9090/admin` and login.
2. Expand Servers, Application Servers, server1, Process Definition, JavaVirtualMachine, CustomProperties.
3. Click New Key, Value pair and add the following new pairs:
  - Key="http.proxyHost", Value="<forward proxy hostname>"
  - Key="http.proxyPort", Value="<forward proxy HTTP port>"
  - Key="https.proxyHost", Value="<forward proxy hostname>"
  - Key="https.proxyPort", Value="<forward proxy HTTPS port>"
4. Save the configuration changes and logout.
5. Restart WebSphere.

### **Application Server**

To configure forward proxy on the application server:

1. Open the PSAPPSRV.CFG file.
2. Complete the Proxy Host and Proxy Port under the [PSTOOLS] section.
3. Restart the application server and Process Scheduler domain.

---

**Note.** The agents do not use the Proxy Host settings in the PSAPPSRV.CFG file.

---

### **Process Scheduler**

To configure forward proxy on the Process Scheduler server:

1. Open the PSPRCS.CFG file.
2. Enter the Proxy Host and Proxy Port under the [PSTOOLS] section.

---

**Note.** The agents do not use the Proxy Host settings in the PSPRCS.CFG file.

---

## **PSPPMRSRV Communication with the Monitor Servlet**

You can't have a firewall between the PSPPMRSRV processes and the monitoring web server. When PSPPMRSRV starts, it binds to the next free port that is allocated by the operating system. As such, no static port exists. This saves configuring ports for multiple PSPPMRSRVs.

## **Monitor Cluster Members**

The monitor cluster members communicate with each other on their allotted ports. If the cluster members are on different sides of a firewall, then these port numbers need to remain open for HTTP/S.

---

## **Setting Up SSL**

This section discusses how to:

- Set up SSL between the agents and the Performance Monitor.
- Set up SSL between the Performance Monitor and PPMI clients.

---

**Note.** Setting up SSL encryption is optional.

---

## **Setting Up SSL Between Agents and Performance Monitor**

This configuration encrypts data that is sent from the agents to the Performance Monitor.

To set up SSL between agents and Performance Monitor:

1. Install a digital root certificate on the web server.

A digital root certificate from a trusted certificate authority (CA) must be installed on the web server that is hosting the monitor URL.

---

**Note.** If the root certificate that is installed on your monitoring web server is from a nonstandard CA, then a copy of that certificate must be installed in the application server key store for the monitored databases. The agents load this certificate when they start.

---

2. Specify *https* for the Monitor URL.
  - Select PeopleTools, Performance Monitor, Administration, Specify Monitor.
  - Enter the URL that is specified for the Monitor URL beginning with *https*.

## Setting Up SSL Between Performance Monitor and PSPPMSRV and Monitor Cluster Members

This configuration encrypts the PPMI user ID and password when it is passed to the monitor and communication between monitor cluster members. Performance data that is published by the monitor *is not* encrypted.

To set up SSL between Performance Monitor and PSPPMSRV and between cluster members:

1. Install a digital root certificate on the web server.

A digital root certificate from a trusted CA must be installed on the web server that is hosting the monitor URL.
2. Specify *HTTPS* for the PPMI URL.
  - Select PeopleTools, Performance Monitor, Administration, Global Administration.
  - Enter the URL that is specified for the PPMI URL beginning with *https*. If you elect to use PSPPMSRV as the PPMI client, only the communication from the PSPPMSRV to the PPMI servlet is encoded.
3. Specify *HTTPS* for the member servlet URLs in the Global Administration page.

---

**Note.** If the root certificate that is installed on your monitoring web server is from a nonstandard CA, then a copy of that certificate must be installed in the application server key stores for the monitoring databases.

---

---

## Setting Up SSL Client Authentication

SSL client authentication validates that the client is trusted by the server.

---

**Note.** Setting up SSL client authentication is optional.

---

To set up SSL client authentication:

1. Set up SSL (as described in the previous section).
2. Set up your own certificate authority.

Remove all other certificate authorities from the monitoring web server's key store. All certificates that are signed by this authority will be trusted by the monitoring web server.

- Configure the monitoring web server so that client authentication is required and HTTP requests are disabled.

---

**Note.** This configuration prevents web browser connections to the web server unless the browser has loaded the client certificate. In particular, the Ping buttons that you use when you set up the PPMI URL and the Monitor URL require the browser to have a trusted client certificate loaded.

---

- Configure Client Authentication on all elements that must access the monitoring system through HTTPS.

The following internal elements must have client certificates in their key stores. Each of these certificates must be signed by your certificate authority. The client authentication ensures that the data that an element receives is authentic in that no third-party could have inserted any incorrect data.

<i>Element</i>	<i>Description</i>
Agents	Configuring client authentication ensures that performance information that is sent between agents and the monitoring system is authentic.
Monitor cluster members	Monitor cluster members exchange information regularly. Configuring client authentication ensures that performance information that is sent between the cluster members is authentic.
Integration gateway	The gateway makes HTTP/S requests to notify the monitoring system of configuration changes. Configuring client authentication ensures that configuration notifications that are sent through the gateway are authentic.
PSPPMRSRV	PSPPMRSRV instances make HTTP/S requests to register with the monitoring servlet. Configuring client authentication ensures that the registration process is authentic.
PIA to Integration Gateway	During notification of configuration changes, PIA makes an HTTP/S request to the gateway. Configuring client authentication ensures that data that is sent between PIA and the gateway is authentic.

The following client certificates are used by these elements. The PSPPMRSRV instances and the Monitor Cluster members use the same certificate.

<i>Certificate</i>	<i>Description</i>
Agent certificate	This certificate resides in the key store in the database of the monitored system. The agents use this certificate.

<b>Certificate</b>	<b>Description</b>
Monitor certificate	This certificate resides in the key store in the database of the monitoring system. PSPPMRSRV instances and monitor cluster members use this certificate.
Integration gateway certificate	This certificate resides in the monitoring system gateway. This certificate is used during notification of configuration changes.
PIA to Integration Gateway Certificate	This certificate resides in the key store in the database of the monitoring system. PIA uses this certificate to make a request to the gateway.

The following table describes where each certificate is configured.

<b>Certificate</b>	<b>Procedure</b>
Agent certificate	Create a client certificate in the key store in the monitored database, using the Digital Certificates page (PeopleTools, Security, Security Objects, Digital Certificates). The certificate type must be "Local Node" and the alias must be "PerfMon".
Monitor certificate	Create a client certificate in the key store in the monitoring database, using the Digital Certificates page (PeopleTools, Security, Security Objects, Digital Certificates). The certificate type must be "Local Node" and the alias must be "PerfMon".
Integration Gateway certificate	Create a client certificate in the key store for gateway, using the pskeymanager utility. Edit the integrationGateway.properties file to include the certificate alias and encrypted certificate password in the ig.certificateAlias and ig.certificatePassword properties.
PIA to Integration Gateway Certificate	Create a client certificate in the key store in the monitoring database, using the Digital Certificates page (PeopleTools, Security, Security Objects, Digital Certificates). The certificate type must be "Local Node" and the alias must be the name of the default local node (messaging node) in the monitoring database.  Discover the name of the local node by selecting PeopleTools, Integration Broker, Node Definitions. Click Search and find the node marked as the default local node.  <b>Note.</b> While the alias of the certificate must be the same as the name of the default local node, the name of the certificate does not have to match. In particular, the certificate name can't contain the underscore character.  Configure the Integration Broker Gateway URL to use HTTPS.

---

## Protecting the Performance Monitor Servlets

If you do not intend to use the Performance Monitor servlets, and you want to make sure that they are not exposed to hackers, you can disable the servlets permanently.

To disable the servlets, edit the web.xml file in the Portal Web-Application and remove the servlet definitions and servlet mappings for the monitor and PPMI servlet.



## Chapter 10

# Performance Monitor Meta-Data

This section discusses:

- Performance Monitor meta-data definitions.
- Metric definitions.
- Event definitions.
- Context definitions.
- PMU definitions.

---

## Performance Monitor Meta-Data Definitions

The Performance Monitor uses the following meta-data definitions:

- Metrics.
- Events.
- Contexts.
- PMUs.

### ***Metrics***

Performance Monitor agents send PMUs and events to the monitor. Each PMU and event contains one or more metrics. Each metric has a unique identifier. Metric definitions are the building blocks for creating PMUs and events. PMUs and events comprise up to six numeric values and one string metric value. One metric definition can appear in multiple event and PMU definitions.

### ***Events***

Events are notifications containing performance metrics that are different from PMUs in that they are not hierarchical, and they do not have durations. PeopleTools has defined a set of event types, and each type of event is reported at a specific location in the instrumented code.

Each event has:

- Type.
- Instance identifier (a unique identifier for a specific event instance).

- Timestamp.
- Severity.
- Metrics (such as CPU usage and memory allocation).

Event definitions group as many as seven metrics to measure the intended performance data. Some events do not have metrics.

### **Contexts**

A context definition applies only to PMU definitions. Contexts provide additional information so that PMU performance data can be displayed and searched more effectively. For example, a context definition enables you to group and sort numeric values, such as an execute count, by page name. Contexts enable the system to assign the data to various elements such as pages, components, service calls, and so on. Without contexts, you have only numeric data in no understandable grouping.

PMU metrics contain data that is specific to that PMU. Context values, on the other hand, are common to the entire user request or a specific tier. The system uses contexts to "flatten" a PMU tree. For example, you do not have to navigate up from a SQL PMU to an ICPanel PMU to see what component generated that SQL statement.

### **PMUs**

A PMU is a unit of measure that reflects the execution of a section of code. The system starts and stops a PMU at specific code locations, and the system may update a PMU anytime between the start and stop times. PeopleTools has defined a set of PMU types, and each type of PMU corresponds to the instrumentation at a specific code location, such as a SQL Execute in PSAPPSRV or a Jolt Request in the web server.

Each PMU includes:

- PMU Type.
- Instance identifier (a unique identifier for a specific PMU instance).
- Start time.
- Stop time.
- Status.
- Metrics (such as number of SQL fetches or buffer size that is used in a Jolt response).

A PMU represents a section of code that is bracketed by calls to an internal instrumentation API that signal the start and stop of that logical unit of code.

PMU definitions group as many as seven metrics to measure the intended performance data. Some PMUs do not have metrics.

---

## **Metric Definitions**

This section discusses the attributes of a metric definition.

---

**Note.** Only PeopleSoft should modify metric definitions. Modifying metric definitions at your site could cause unexpected results.

---

<b>Metric Identifier</b>	A numeric value acting as the unique identifier, or key, for a metric definition.
<b>Metric Type</b>	<p>Displays the type of the metric definition. Metric types are:</p> <ul style="list-style-type: none"> <li>• <b>Counter:</b> A counter metric is designed to enable sums of values from a specific time range to be calculated.  Examples are bytes printed and records written. The values can also be averaged, maximums and minimums can be calculated, and other kinds of statistical calculations can be performed</li> <li>• <b>Gauge:</b> A gauge metric is designed to be used instead of a counter when it is not meaningful to calculate sum values that are recorded within a time range.  Calculations that are performed on gauge metrics include: average, standard deviation, median, maximum value, and minimum value. For example, the amount of memory that is used on a server is a gauge metric type. If you measure the amount of memory that is used over 20 transactions in a time range, the sum of the memory that is used is not necessarily useful. However, the average, median, and standard deviation provide insight into usage per transaction.</li> <li>• <b>Numeric Identifier:</b> A numeric identifier is a numeric value that is used as an identifier, not as a measurement value.  Creating sums and averages, or manipulating these values in any arithmetic way is not meaningful. For example, message numbers and error codes are numeric identifier metric types.</li> <li>• <b>String:</b> Used with metric definition attributes that need to be represented as text, not a numeric value.  Arithmetic operations are not performed on string metric types. For example, descriptive attributes, such as site path, file name, and so on, are string metric types.</li> </ul>
<b>Metric Label</b>	This metric label appears on any page that displays a metric to describe the metric value.
<b>Description</b>	Displays a more detailed description of the purpose of the metric.
<b>Metric Multiplier</b>	<p>Enables you to manipulate metric values using a multiplier.</p> <p>The multiplier determines how the metrics appears on a PeopleSoft page. For example, if a metric is in milliseconds and you want to change it to display in seconds, you would specify a multiplier of 1000.</p> <p>If the value does not need to be manipulated (increased or decreased), the multiplier is 1.0.</p>

**Display Metric As an Integer** Metrics can appear as a real number (with decimals) or integers (without decimals).  
 For example, an average duration appears as a real number, while a sum of SQL Statement executions must appear as an integer.  
 To display the metric as an integer, select this check box. To display the metric as a real number, clear this check box.

### ***User Defined Display Text***

This section provides flexibility when you are defining metrics. For example, you can assign labels to particular values that are returned by functions. Based on a particular return value, the system displays various label values. Typically, this would be used in the case of Boolean values, such as the case with metric 23, *Is this a Pagelet*. If a 0 is returned, the system displays *No*. If a 1 is returned, the system displays *Yes*.

**Display Value/Label** Enables you to determine what appears on the pages displaying a metric. These options are mutually exclusive.  
 Select Display Value to display the actual value. Select Display Label to display the label describing the metric value.

**Metric Value** The actual value of a user-defined metric value.

**Metric Value Label** The label that is associated with a user-defined metric value.

---

## **Event Definitions**

This section describes the attributes of an event definition.

**Event Definition Set** Event definitions belong to a particular set. This is similar to message definitions in the message catalog belonging to a message set.

---

**Note.** Currently, only one set exists, set 1, which is reserved for internal PeopleSoft development.

---

**Description** Explains the purpose of a particular event definition set.

### ***Definitions***

**Event ID** Identifies an event definition within an event set.

**Event Label** The name of the event definition. This value appears with any event metric values on the pages displaying the event information.

**Description** Provides additional identification, if needed.

<b>Additional Data Label</b>	If the optional long character field is populated for this event type, it requires a display label.
<b>Filter Level</b>	Sets the level at which the system begins recording data about a particular event definition. The level of the overall monitoring system must equal or exceed the level of the event filter level before the system records the event data. For example, if the overall system filter level is set to <i>Standard</i> , the system records information from event definitions with a filter level set to <i>Standard</i> , <i>Warning</i> , and <i>Error</i> .
<b>Metrics</b>	An event comprises up to six, predefined numeric metrics and one string metric. Select the metric to include using the lookup button.  See <a href="#">Chapter 10, "Performance Monitor Meta-Data," Metric Definitions, page 152</a> .

---

## Context Definitions

This section discusses the attributes of a context definition.

<b>Context Identifier</b>	A numeric value that uniquely identifies a specific context definition.
<b>Context Label</b>	The label that appears on any page displaying the context definition to identify it.
<b>Description</b>	A more detailed explanation of the context definition, if needed.

---

## PMU Definitions

This section describes the attributes of PMU definition.

<b>PMU Definition Set</b>	PMU definitions belong to a particular set. This is similar to message definitions in the message catalog belonging to a message set.
---------------------------	---

---

**Note.** Currently, only one set exists, set 1, which is reserved for internal PeopleSoft development.

---

<b>Description</b>	Explains the purpose of a particular PMU definition set.
--------------------	--

### **Definitions**

<b>PMU ID</b>	Identifies a PMU definition within an event set.
---------------	--

<b>PMU Label</b>	The name of the PMU definition. This value appears with any metric values on the pages displaying the PMU information.
<b>Description</b>	Provides additional identification, if needed.
<b>Additional Data Label</b>	Every PMU has an optional long data field for additional information or labeling. The system displays this label only if this field is populated.
<b>Filter Level</b>	<p>Sets the level at which the system begins recording data about a particular PMU definition. The level of the overall monitoring system must equal or exceed the level of the PMU filter level before the system records the event data. For example, if the overall system filter level is set to <i>Verbose</i>, the system records information from event definitions with a filter level set to <i>Standard</i> and <i>Verbose</i>, but it does not record information that is related to PMU definitions with a filter level set to <i>Debug</i>.</p> <p>See <a href="#">Chapter 4, "Administering the Performance Monitor," Setting Agent Filter Options, page 36.</a></p>
<b>Enable Sampling</b>	<p>Some PMUs that are associated with a user connection are always monitored regardless of the sampling rate. The PMUs that are never ignored are those that have the Enable Sampling option cleared. Examples of such PMUs are those that are related to users signing on, signing off, and being redirected to other sites.</p> <p>See <a href="#">Chapter 4, "Administering the Performance Monitor," Configuring Performance Monitoring Sampling Rate, page 44.</a></p>
<b>Contexts 1, 2, 3</b>	<p>A PMU has up to three context fields.</p> <p>See <a href="#">Chapter 10, "Performance Monitor Meta-Data," Context Definitions, page 155.</a></p> <hr/> <p><b>Note.</b> If the contents of a context field vary according to the "parent" PMU, then the system uses Context 14 "Generic." For example, Context 1 of a SQL PMU such as PMU 407 (PeopleCode Built-In SQL Execute) is the component name for an ICPanel request or an iScript name for an ICScript request.</p> <hr/>
<b>Metrics</b>	<p>A PMU comprises up to six, predefined numeric metrics and one string metric.</p> <p>Select the metric to include using the lookup button.</p>
<b>Display</b>	Select this option if you intend to display the metric label, the value, or both on a PeopleSoft page.

## Appendix A

# PMU Definition Reference

This appendix contains the structure of each PMU that is defined in the PeopleSoft system.

---

**Note.** Currently, only one PMU set definition exists, set 1.

---

---

## PMU 100 Portal Request

PMU 100 has the following identification attributes.

<i>Attribute</i>	<i>Value</i>
ID	100
Label	Portal Request
Description	Reported at entry and exit of portal servlet

### ***Filter Level***

PMU 100 has a filter level of 4.

### ***Sampling Enabled***

Yes.

### ***Contexts***

<i>Context</i>	<i>Value</i>	<i>Description</i>
1	Session ID	Web server Session ID.
2	IP Address	Client IP Address.
3	Action	Action - "cmd" parameter (such as login, logout, or homepage).

### ***Metrics***

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Response Code	HTTP response code.
2	Pagelet Count	Pagelet Count.
7	Target CREF	Target CREF of the portal page.

### **Additional Label**

Additional Data.

---

## **PMU 101 PIA Request**

PMU 101 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	101
Label	PIA Request
Description	Reported at entry and exit of PIA servlet

### **Filter Level**

PMU 101 has a filter level of 4.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Session ID	Web server Session ID.
2	IP Address	Client IP Address.
3	Action	Action - "cmd" parameter (such as login, logout, or homepage).

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Response Size (bytes)	Size of HTML Response in bytes.
2	Response Code	HTTP response code.
3	Static Content Count	Number of items of static content downloaded with request.
4	Is this a Pagelet?	Boolean - is this is a pagelet or not?

### ***Additional Label***

Additional Data.

---

## **PMU 102 Compression Request**

PMU 102 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	102
Label	Compression Request
Description	Compression servlet run when the browser requests content

### ***Filter Level***

PMU 102 has a filter level of 5.

### ***Sampling Enabled***

Yes.

### ***Contexts***

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Session ID	Web server Session ID.
2	IP Address	Client IP Address.
3	Content Type	Content Type of Response.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Response Size (bytes)	Size of HTML Response in bytes.
2	Response Code	HTTP response code.
3	Compression Ratio	Compression ratio.
4	Compressed?	Boolean - was the file compressed?

**Additional Label**

Additional Data.

---

## PMU 106 PIA Request From Portal

PMU 106 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	106
Label	PIA Request From Portal
Description	Reported at Entry and Exit when Portal calls PIA directly

**Filter Level**

PMU 106 has a filter level of 4.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Session ID	Web server Session ID.
2	IP Address	Client IP Address.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Response Size (bytes)	Size of HTML Response in bytes.
2	Response Code	HTTP Response Code.
3	Static Content Count	Number of items of static content downloaded with request.
4	Is this a Pagelet?	Boolean - is this is a pagelet or not?

---

## PMU 108 User Session Ended

PMU 108 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	108
Label	User Session Ended
Description	User Session logout, expiration, timeout, or error

### **Filter Level**

PMU 108 has a filter level of 4.

### **Sampling Enabled**

No.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Session ID	Web server Session ID.
2	IP Address	Client IP Address.
3	Not used	Not used.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Reason	Reason for session termination.
2	Response Code	HTTP Response Code.
7	User ID	User ID of user logging on.

## PMU 109 User Session Began

PMU 109 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	109
Label	User Session Began
Description	User Session began (user logged in)

### **Filter Level**

PMU 109 has a filter level of 4.

### **Sampling Enabled**

No.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Session ID	Web server Session ID.
2	IP Address	Client IP Address.
3	Not used	Not used.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
7	User ID	User ID of user logging on.

## PMU 113 Authenticate

PMU 113 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	113
Label	Authenticate
Description	Entry and Exit for application server authentication (GetCertificate)

### **Filter Level**

PMU 113 has a filter level of 4.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Session ID	Web server Session ID.
2	IP Address	Client IP Address.
3	Not used	Not used.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Bypass Signon?	Boolean - logging in with a guest ID?
2	Authtoken Cookie Present?	Boolean - is the Authtoken Cookie in the request?
7	User ID	User ID of user logging on.

### **Additional Label**

Previously logged in user.

---

## PMU 114 Attach File Request

PMU 114 has the following identification attributes.

<i>Attribute</i>	<i>Value</i>
ID	114
Label	Attach File Request
Description	Entry and Exit for file upload from browser client

### ***Filter Level***

PMU 114 has a filter level of 4.

### ***Sampling Enabled***

Yes.

### ***Contexts***

<i>Context</i>	<i>Value</i>	<i>Description</i>
1	Session ID	Web server Session ID.
2	IP Address	Client IP Address.
3	File Name	File Name.

### ***Metrics***

<i>Metric</i>	<i>Value</i>	<i>Description</i>
1	Total File Size (bytes)	Total size of incoming files in bytes.

### ***Additional Label***

All File Names.

---

## PMU 115 JOLT Request

PMU 115 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	115
Label	JOLT Request
Description	JOLT request to the application server

### **Filter Level**

PMU 115 has a filter level of 4.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Session ID	Web server Session ID.
2	IP Address	Client IP Address.
3	Tuxedo Service Name	Tuxedo Service Name, such as ICPanel, ICScript, and so on.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	JOLT SendBuf Size (bytes)	Size in bytes of the buffer sent to the application server.
2	JOLT RecvBuf Size	Size in bytes of the buffer received from the application server.
3	JOLT Return Code	JOLT Return Code.
4	JOLT Request Retried	JOLT Request Retried.

### **Additional Label**

Error Status Code.

---

## PMU 116 Redirect after Login

PMU 116 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	116
Label	Redirect after Login
Description	Redirected round trip time (network latency)

### **Filter Level**

PMU 116 has a filter level of 4.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Session ID	Web server session ID.
2	IP Address	Client IP address.
3	URL redirected to	URL redirected to, after logon.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
7	User Agent.	User Agent.

---

## PMU 117 Get Content

PMU 117 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	117
Label	Get Content
Description	Portal requests for content or pagelet

**Filter Level**

PMU 117 has a filter level of 4.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Session ID	Web server session ID.
2	IP Address	Client IP address.
3	Portal CREF ID	Portal CREF ID of this content.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Response Size (bytes)	Size of HTML Response in bytes.
2	Is this a Pagelet?	Boolean - is this is a pagelet or not?
3	Cookie Count	Number of cookies received in the response.
7	Target CREF	Target CREF of the portal page.

**Additional Label**

URL for which request issued.

---

## PMU 400 Tuxedo Service PCode and SQL

PMU 400 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	400
Label	Tuxedo Service PCode and SQL
Description	Current SQL and PeopleCode and SQL summary metrics

**Filter Level**

PMU 400 has a filter level of 4.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	Server Name	Server Name (such as PSAPPSRV, PSQRYSRV, PSSAMSRV).
3	Tuxedo Service Name	Tuxedo Service Name (such as ICPanel, ICScript, and so on).

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	PeopleCode Exec Time (ms)	PeopleCode execution time (milliseconds).
2	PeopleCode SQL Time (ms)	Execution Time for SQLExec and SQL object (milliseconds).
3	PCode BuiltIn SQL Time (ms)	Total PeopleCode BuiltIn SQL Execution Time(milliseconds).
4	PeopleTools SQL Time (ms)	SQL time excluding PeopleCode and BuiltIn SQL (milliseconds).
5	SQL Fetch Count	Total number of rows fetched from the database.
6	SQL Execute Count	SQL Execute Count.
7	Trace Level	SQL, PeopleCode, and Component Processor trace levels.

**Additional Label**

Current SQL statement.

## PMU 401 ICPanel

PMU 401 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	401
Label	ICPanel
Description	Entry and exit for component search and processing on application server

### **Filter Level**

PMU 401 has a filter level of 4.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Component	Component name.
2	Page	Page name.
3	Action	Request Action (such as #ICSave) from web server to application server

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Component Buffer Size (bytes)	Size of the component buffer (bytes).
2	Component/Page Deferred Modes	Component and Page Processing Modes for this Request. For example, Off, Page Only, Component Only, and so on.
3	Field set to Interactive Mode.	Boolean - field disallows Deferred Processing for the Page?

<b>Metric</b>	<b>Value</b>	<b>Description</b>
4	Deferred Field Edit Count	Number of deferred field edits processed in this request.
7	Action String	String passed by web server (such as QE_ABSENCE_HIST\$new\$\$\$0).

**Additional Label**

Component Key.

---

## PMU 402 Modal Level 1

PMU 402 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	402
Label	Modal Level 1
Description	Triggered by Think-Time PeopleCode (DoModal, DoCancel, and so on).

**Filter Level**

PMU 402 has a filter level of 4.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Component	Component name.
2	Page	Page name.
3	Action	Request Action (such as #ICSave) from web server to application server.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Component Buffer Size (bytes)	Size of the component buffer (bytes).
7	Action String	String passed by web server (such as QE_ABSENCE_HIST\$new\$0\$\$0).

### **Additional Label**

Component Key.

---

## **PMU 403 Modal Level 2**

PMU 403 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	403
Label	Modal Level 2
Description	Modal component/secondary page/lookup page from a Level 1.

### **Filter Level**

PMU 403 has a filter level of 4.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Component	Component name.
2	Page	Page name.
3	Action	Request Action (such as #ICSave) from web server to application server.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Component Buffer Size (bytes)	Size of the component buffer (bytes).
7	Action String	String passed by web server (such as QE_ABSENCE_HIST\$new\$\$\$0).

**Additional Label**

Component Key.

---

## PMU 404 PeopleCode Program Execution

PMU 404 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	404
Label	PeopleCode Program Execution
Description	PeopleCode Events, External Function Calls and Application Class Methods

**Filter Level**

PMU 404 has a filter level of 6.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Program	PeopleCode Program Name.

**Additional Label**

Component Key.

## PMU 406 PeopleCode SQL Execute

PMU 406 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	406
Label	PeopleCode SQL Execute
Description	Execution of PeopleCode SQLExec and SQL Objects

### **Filter Level**

PMU 406 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Program	PeopleCode Program name.
3	SQL Origin	SQLExec or SQL Object.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Cursor Number	Unique ID for a SQL cursor.
2	PeopleCode Statement Number	PeopleCode statement number (excluding comments and white space).
3	SQL Was Compiled	Boolean - was SQL compiled before executing?
4	SQL Compile Time (ms)	Compile time for this SQL (milliseconds).

<b>Metric</b>	<b>Value</b>	<b>Description</b>
7	SQL Operation and Tables	SQL Operation and Tables (such as UPDATE PS_QUERY_RUN_QRYVW).

**Additional Label**

SQL Statement.

---

## PMU 407 PeopleCode BuiltIn SQL Execute

PMU 407 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	407
Label	PeopleCode BuiltIn SQL Execute
Description	SQL executed during a PeopleCode BuiltIn or Method call

**Filter Level**

PMU 407 has a filter level of 5.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Program	PeopleCode Program Name.
3	PeopleCode BuiltIn	PeopleCode Built-In Name (such as GetNextNumber).

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Cursor Number	Unique ID for a SQL cursor.
2	PeopleCode Statement Number	PeopleCode statement number (excluding comments and white space).
3	SQL Was Compiled	Boolean - was SQL compiled before executing?
4	SQL Compile Time (ms)	Compile time for this SQL (milliseconds).
7	SQL Operation and Tables	SQL Operation and Tables (such as UPDATE PS_QUERY_RUN_QRYVW).

**Additional Label**

SQL Statement.

---

## PMU 408 PeopleTools SQL Execute

PMU 408 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	408
Label	PeopleTools SQL Execute
Description	All SQL calls excluding PMUs 406 and 407

**Filter Level**

PMU 408 has a filter level of 5.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Cursor Number	Unique ID for a SQL cursor.
2	PeopleCode Statement Number	PeopleCode statement number (excluding comments and white space).
3	SQL Was Compiled	Boolean - was SQL compiled before executing?
7	SQL Operation and Tables	SQL Operation and Tables (such as UPDATE PS_QUERY_RUN_QRYVW).

### **Additional Label**

SQL Statement.

---

## **PMU 409 Explicit Commit**

PMU 409 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	409
Label	Explicit Commit
Description	PeopleCode controlled Commit (such as GetNextNumberWithGaps)

### **Filter Level**

PMU 409 has a filter level of 5.

### **Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Program	PeopleCode Program name.
3	PeopleCode Builtin	PeopleCode BuiltIn Name (such as GetNextNumber).

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	PeopleCode Statement Number	PeopleCode statement number (excluding comments and white space).
2	Cursor Number	Unique ID for a SQL cursor.

---

## PMU 410 ICSript

PMU 410 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	410
Label	ICScript
Description	Entry and exit for ICSript execution on the application server

**Filter Level**

PMU 410 has a filter level of 4.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	ICScript URL	ICScript URL.
2	PeopleCode Program	PeopleCode Program name.

---

## PMU 411 ICQuery

PMU 411 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	411
Label	ICQuery
Description	Entry and exit for "Run to HTML" queries on the application server.

### **Filter Level**

PMU 411 has a filter level of 4.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Query	Query name.
2	Action.	Request Action (such as #ICSave) from web server to the application server.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	SQL Fetch Count	Total number of rows fetched from the database.
2	Prompt Count	How many prompts the query defines as its criteria.

## PMU 412 Tuxedo Service Summary

PMU 412 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	412
Label	Tuxedo Service Summary
Description	Cache and other metrics (continuation of PMU 400)

### **Filter Level**

PMU 412 has a filter level of 4.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	Server Name	Server Name (such as PSAPPSRV, PSQRYSRV, PSSAMSRV, and so on).
3	Tuxedo Service Name	Tuxedo Service Name (such as ICPanel, ICScript, and so on).

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Cache Misses	Manager cache misses resulting in object fetches from the database.
2	Memory Cache Hits	Manager cache hits resulting in object fetches from memory.
3	File Cache Hits	Manager cache hits resulting in object fetches from file.

<b>Metric</b>	<b>Value</b>	<b>Description</b>
4	Deserialization Time (ms)	Deserialization Request Time.
5	Serialization Time (ms)	Serialization Reply Time. Serialization time includes HTML generation, building state blob, and sending of the blob to the web server. <ul style="list-style-type: none"> <li>• HTML generation.</li> <li>• Building state blob.</li> <li>• Sending of the blob to the web server.</li> </ul>
6	PeopleCode Global Size (bytes)	Size of serialized global and component variables (bytes).

---

## PMU 413 GetCertificate

PMU 413 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	413
Label	GetCertificate
Description	Metrics for application server user authentication service

### **Filter Level**

PMU 413 has a filter level of 4.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Tuxedo Service Name	Tuxedo Service Name (such as ICPanel, ICScript, and so on).
2	Reason	Reason for Service Request.

<b>Context</b>	<b>Value</b>	<b>Description</b>
3	Not used	Not used.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Authenticated?	Boolean - Authentication succeeded?
7	Timestamp	Timestamp.

---

## **PMU 414 SQL Fetch Summary**

PMU 414 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	414
Label	SQL Fetch Summary
Description	Metrics for all SQL fetches for parent SQL (406, 407, 408)

### **Filter Level**

PMU 414 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Program	PeopleCode Program name.
3	SQL Fetch Origin	SQL Fetch Origin.

---

**Note.** Context 2 and Context 3 are populated only for PeopleCode and SQL built-in functions.

---

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	SQL Fetch Time (ms)	SQL fetch time.
2	Cursor Number	Unique ID for a SQL cursor.
3	SQL Fetch Count	Total number of rows fetched from the database.
7	SQL Operation and Tables	SQL Operation and Tables (such as UPDATE PS_QUERY_RUN_QRYVW).

**PMU 415 PortalRegistry**

PMU 415 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	415
Label	PortalRegistry
Description	Metrics for Portal registry service call on the application server

**Filter Level**

PMU 415 has a filter level of 5.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Portal	Generic - Varies based upon server and service request.
2	Portal Command	Portal Command (such as EPortalCRefByURL).

---

## PMU 416 ICWorklist

PMU 416 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	416
Label	ICWorklist
Description	Metrics for CICWorkListService::DoService()

### **Filter Level**

PMU 416 has a filter level of 4.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Component	Component name.
2	Page	Page name.
3	Action	Request Action (such as #ICSave) from the web server to the application server.

---

## PMU 417 FTP Request

PMU 417 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	417
Label	FTP Request
Description	Metrics for file transfer between application server and FTP server

**Filter Level**

PMU 417 has a filter level of 4.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	FTP Server URL	FTP Server URL.
3	FTP Action name	FTP Action name (such as Get, Put, Delete, and so on).

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
7	FTP Log File	FTP log file (such as c:\temp\ftp28975.log).

**Additional Label**

FTP script file name.

---

## PMU 418 PSBusComp

PMU 418 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	418
Label	PSBusComp
Description	External Component Interface call into the application server

**Filter Level**

PMU 418 has a filter level of 4.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Component Interface	Component Interface name.
2	CI Method Name	Component Interface method name.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Component Buffer Size (bytes)	Size of the component buffer (bytes).

---

## PMU 419 BI GetInterlink

PMU 419 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	419
Label	BI GetInterlink
Description	PeopleCode Business Interlink GetInterlink call.

**Filter Level**

PMU 419 has a filter level of 5.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	BI Name	Business Interlink Name (such as PT_GETSTOCKQUOTE).

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Component Buffer Size (bytes)	Size of the component buffer (bytes).

---

## **PMU 420 BI Execute**

PMU 420 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	420
Label	BI Execute
Description	PeopleCode Business Interlink GetInterlink call

### **Filter Level**

PMU 420 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	BI Name	Business Interlink Name (such as PT_GETSTOCKQUOTE).

---

## PMU 421 BI BulkExecute

PMU 421 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	421
Label	BI BulkExecute
Description	PeopleCode Business Interlink BulkExecute call

### **Filter Level**

PMU 421 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	BI Name	Business Interlink Name (such as PT_GETSTOCKQUOTE).

---

## PMU 422 LDAP Bind

PMU 422 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	422
Label	LDAP Bind
Description	PeopleCode LDAP ExecuteBind function call

**Filter Level**

PMU 422 has a filter level of 5.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Tuxedo Service Name	Tuxedo Service Name (such as ICPanel, ICScript, and so on).
2	LDAP Object Name	LDAP Object Name.
3	LDAP DN	LDAP Distinguished Name.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
7	LDAP Host:Port	Host and port of the LDAP server.

---

## PMU 423 LDAP Search

PMU 423 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	423
Label	LDAP Search
Description	Metrics for LdapDirBIHandler::ExecuteSearch()

**Filter Level**

PMU 423 has a filter level of 5.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Tuxedo Service Name	Tuxedo Service Name (such as ICPanel, ICScript, and so on).
2	LDAP Object Name	LDAP Object Name.

<b>Context</b>	<b>Value</b>	<b>Description</b>
3	LDAP DN	LDAP Distinguished Name.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	LDAP Attribute Count	Number of attributes found in the LDAP catalog.
7	LDAP Host:Port	Host and port of the LDAP server.

---

## **PMU 424 Call AppEngine**

PMU 424 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	424
Label	Call AppEngine
Description	PeopleCode CallAppEngine Builtin call

### **Filter Level**

PMU 424 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Program	PeopleCode program name.

### **Metrics**

<i>Metric</i>	<i>Value</i>	<i>Description</i>
7	AppEngine Program Name	Application Engine program name.

---

## PMU 425 Implicit Commit

PMU 425 has the following identification attributes.

<i>Attribute</i>	<i>Value</i>
ID	425
Label	Implicit Commit
Description	PeopleTools-controlled SQL commit

### **Filter Level**

PMU 425 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<i>Context</i>	<i>Value</i>	<i>Description</i>
1	Generic	Generic - Varies based upon server and service request.

### **Metrics**

<i>Metric</i>	<i>Value</i>	<i>Description</i>
1	Cursor Number	Unique ID for a SQL cursor

---

## PMU 426 RemoteCall Built-in

PMU 426 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	426
Label	RemoteCall Built-in
Description	PeopleCode RemoteCall Built-in called

### **Filter Level**

PMU 426 has a filter level of 4.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Program	PeopleCode program name.
3	Dispatcher Program Name	Dispatcher Program Name.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
7	COBOL Program Name	COBOL Program Name.

---

## **PMU 427 Deserialization**

PMU 427 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	427
Label	Deserialization
Description	SQL summary metrics during Deserialization

**Filter Level**

PMU 427 has a filter level of 6.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	Server Name	Server Name (such as PSAPPSRV, PSQRYSRV, PSSAMSRV).
3	Tuxedo Service Name	Tuxedo Service Name, such as ICPanel, ICScript, and so on.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	PeopleCode SQL Time	Execution Time for SQLExec and SQL object (milliseconds).
2	PCode Builtin SQL Time	Total PeopleCode BuiltIn SQL Execution Time(milliseconds).
3	PeopleTools SQL Time	SQL time excluding PeopleCode and BuiltIn SQL (milliseconds).
4	SQL Fetch Count	Total number of rows fetched from the database.
5	SQL Execute Count	SQL Execute Count.

---

**PMU 428 Serialization**

PMU 428 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	428

<b>Attribute</b>	<b>Value</b>
Label	Serialization
Description	SQL summary metrics during serialization

### **Filter Level**

PMU 428 has a filter level of 6.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	Server Name	Server Name (such as PSAPPSRV, PSQRYSRV, PSSAMSRV).
3	Tuxedo Service Name	Tuxedo Service Name, such as ICPanel, ICScript, and so on.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	PeopleCode SQL Time	Execution Time for SQLExec and SQL object (milliseconds).
2	PCode Builtin SQL Time	Total PeopleCode Builtin SQL Execution Time(milliseconds).
3	PeopleTools SQL Time	SQL time excluding PeopleCode and Builtin SQL (milliseconds).
4	SQL Fetch Count	Total number of rows fetched from the database.
5	SQL Execute Count	SQL Execute Count.

## PMU 439 Workflow PCode Summary

PMU 439 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	439
Label	Workflow PCode Summary
Description	Component, Component Record and Record Field Events

### **Filter Level**

PMU 439 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Record Field PCode Exec Count	Execution count for record field level PeopleCode.
2	Record Field PCode Exec Time	Total execution time for record field-level PCode (ms).
5	Component PCode Exec Count	Execution count for component-level PeopleCode.
6	Component Level PCode Time	Total execution time for component-level PeopleCode (ms).
7	Not used	Not used.

**Additional Label**

None.

---

**PMU 440 Usage Monitoring**

PMU 440 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	440
Label	Usage Monitoring
Description	Usage Monitoring data

**Filter Level**

PMU 440 is collected at all filter levels if Usage Monitoring is enabled, otherwise PMU 440 information is not collected.

**Sampling Enabled**

Yes.

**Contexts**

Determined by PeopleSoft Test Framework (PTF).

---

**Note.** Usage Monitor information is intended to be viewed only within the context of PTF reports.

---

**Metrics**

Determined by PeopleSoft Test Framework.

---

**Note.** Usage Monitor information is intended to be viewed only within the context of PTF reports.

---

**Additional Label**

None.

---

**PMU 500 SavePreChange PCode Summary**

PMU 500 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	500
Label	SavePreChange PCode Summary
Description	Component, Component Record and Record Field Events

### **Filter Level**

PMU 500 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Record Field PCode Exec Count	Execution count for record field level PeopleCode.
2	Record Field PCode Exec Time	Total execution time for record field-level PCode (ms).
3	Component Rec PCode Exec Count	Execution count for component record-level PeopleCode.
4	Component Rec PCode Exec Time	Total execution time for component record PeopleCode (ms).
5	Component PCode Exec Count	Execution count for component-level PeopleCode.
6	Component Level PCode Time	Total execution time for component-level PeopleCode (ms).

## PMU 501 SavePostChange PCode Summary

PMU 501 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	501
Label	SavePostChange PCode Summary
Description	Component, Component Record and Record Field Events

### **Filter Level**

PMU 501 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Record Field PCode Exec Count	Execution count for record field level PeopleCode.
2	Record Field PCode Exec Time	Total execution time for record field-level PCode (ms).
3	Component Rec PCode Exec Count	Execution count for component record-level PeopleCode.
4	Component Rec PCode Exec Time	Total execution time for component record PeopleCode (ms).

<b>Metric</b>	<b>Value</b>	<b>Description</b>
5	Component PCode Exec Count	Execution count for component-level PeopleCode.
6	Component Level PCode Time	Total execution time for component-level PeopleCode (ms).

## PMU 502 SaveEdit PCode Summary

PMU 502 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	502
Label	SaveEdit PCode Summary
Description	Component, Component Record and Record Field Events.

### **Filter Level**

PMU 502 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Record Field PCode Exec Count	Execution count for record field level PeopleCode.

<b>Metric</b>	<b>Value</b>	<b>Description</b>
2	Record Field PCode Exec Time	Total execution time for record field-level PCode (ms).
3	Component Rec PCode Exec Count	Execution count for component record-level PeopleCode.
4	Component Rec PCode Exec Time	Total execution time for component record PeopleCode (ms).

---

## PMU 503 SaveEdit PCode Summary

PMU 503 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	503
Label	SaveEdit PCode Summary
Description	Component Record and Record Field Events

### **Filter Level**

PMU 503 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Record Field PCode Exec Count	Execution count for record field level PeopleCode.
2	Record Field PCode Exec Time	Total execution time for record field-level PCode (ms).
3	Component Rec PCode Exec Count	Execution count for component record-level PeopleCode.
4	Component Rec PCode Exec Time	Total execution time for component record PeopleCode (ms).

## PMU 504 RowSelect PCode Summary

PMU 504 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	504
Label	RowSelect PCode Summary
Description	Component Record and Record Field Events

### **Filter Level**

PMU 504 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Record Field PCode Exec Count	Execution count for record field level PeopleCode.
2	Record Field PCode Exec Time	Total execution time for record field-level PCode (ms).
3	Component Rec PCode Exec Count	Execution count for component record-level PeopleCode.
4	Component Rec PCode Exec Time	Total execution time for component record PeopleCode (ms).

## PMU 505 RowInsert PCode Summary

PMU 505 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	505
Label	RowInsert PCode Summary
Description	Component Record and Record Field Events

### **Filter Level**

PMU 505 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Record Field PCode Exec Count	Execution count for record field level PeopleCode.
2	Record Field PCode Exec Time	Total execution time for record field-level PCode (ms).
3	Component Rec PCode Exec Count	Execution count for component record-level PeopleCode.
4	Component Rec PCode Exec Time	Total execution time for component record PeopleCode (ms).

## PMU 506 RowDelete PCode Summary

PMU 506 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	506
Label	RowDelete PCode Summary
Description	Component Record and Record Field Events

### **Filter Level**

PMU 506 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).
3	Not used	Not used.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Record Field PCode Exec Count	Execution count for record field level PeopleCode.
2	Record Field PCode Exec Time	Total execution time for record field-level PCode (ms).
3	Component Rec PCode Exec Count	Execution count for component record-level PeopleCode.
4	Component Rec PCode Exec Time	Total execution time for component record PeopleCode (ms).

**PMU 507 FieldChange PCode Summary**

PMU 507 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	507
Label	FieldChange PCode Summary
Description	Component Record and Record Field Events

**Filter Level**

PMU 507 has a filter level of 5.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Record Field PCode Exec Count	Execution count for record field level PeopleCode.
2	Record Field PCode Exec Time	Total execution time for record field-level PCode (ms).
3	Comp Field PCode Exec Count	Execution count for component field level PeopleCode.
4	Comp Field PCode Time (ms)	Total execution time for component-field PeopleCode (ms).

**PMU 508 FieldEdit PCode Summary**

PMU 508 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	508
Label	FieldEdit PCode Summary
Description	Component Record and Record Field Events

**Filter Level**

PMU 508 has a filter level of 5.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Record Field PCode Exec Count	Execution count for record field level PeopleCode.
2	Record Field PCode Exec Time	Total execution time for record field-level PCode (ms).
3	Comp Field PCode Exec Count	Execution count for component field level PeopleCode.
4	Comp Field PCode Time (ms)	Total execution time for component-field PeopleCode (ms).

**PMU 509 FieldDefault PCode Summary**

PMU 509 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	509
Label	FieldDefault PCode Summary
Description	Component Record and Record Field Events

**Filter Level**

PMU 509 has a filter level of 5.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Record Field PCode Exec Count	Execution count for record field level PeopleCode.
2	Record Field PCode Exec Time	Total execution time for record field-level PCode (ms).
3	Comp Field PCode Exec Count	Execution count for component field level PeopleCode.
4	Comp Field PCode Time (ms)	Total execution time for component-field PeopleCode (ms).

**PMU 510 PrePopup PCode Summary**

PMU 510 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	510
Label	PrePopup PCode Summary
Description	Component Record and Record Field Events

**Filter Level**

PMU 510 has a filter level of 5.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).
3	Not used	Not used.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Record Field PCode Exec Count	Execution count for record field level PeopleCode.
2	Record Field PCode Exec Time	Total execution time for record field-level PCode (ms).
3	Comp Field PCode Exec Count	Execution count for component field level PeopleCode.
4	Comp Field PCode Time (ms)	Total execution time for component-field PeopleCode (ms).

**PMU 511 ItemSelected PCode Summary**

PMU 511 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	511
Label	ItemSelected PCode Summary
Description	Popup Menu Event

**Filter Level**

PMU 511 has a filter level of 5.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Popup Menu PCode Exec Count	Execution count for Popup menu-level PeopleCode.
2	Popup Menu PCode Time (ms)	Total execution time for Popup menu-level PCode(ms).

**PMU 512 SearchInit PCode Summary**

PMU 512 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	512
Label	SearchInit PCode Summary
Description	Component Record and Record Field Events

**Filter Level**

PMU 512 has a filter level of 5.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Popup Menu PCode Exec Count	Execution count for Popup menu-level PeopleCode.
2	Popup Menu PCode Time (ms)	Total execution time for Popup menu level PCode (ms).
3	Component Rec PCode Exec Count	Execution count for component record-level PeopleCode.
4	Component Rec PCode Exec Time	Total execution time for component record PeopleCode (ms).

## PMU 513 SearchSave PCode Summary

PMU 513 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	513
Label	SearchSave PCode Summary
Description	Component Record and Record Field Events

### **Filter Level**

PMU 513 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).
3	Not used	Not used.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Record Field PCode Exec Count	Execution count for record field level PeopleCode.
2	Record Field PCode Exec Time	Total execution time for record field-level PCode (ms).
3	Component Rec PCode Exec Count	Execution count for component record-level PeopleCode.
4	Component Rec PCode Exec Time	Total execution time for component record PeopleCode (ms).

**Additional Label**

None.

---

**PMU 514 Active PCode Summary**

PMU 514 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	514
Label	Active PCode Summary
Description	Page Events

**Filter Level**

PMU 514 has a filter level of 5.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.

<b>Context</b>	<b>Value</b>	<b>Description</b>
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).
3	Not used	Not used.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Page PCode Exec Count	Execution count for page-level PeopleCode.
2	Page PCode Time (ms)	Total execution time for page-level PeopleCode (ms).

---

## **PMU 515 PreBuild PCode Summary**

PMU 515 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	515
Label	PreBuild PCode Summary
Description	Component Events

### **Filter Level**

PMU 515 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
5	Component PCode Exec Count	Execution count for component-level PeopleCode.
6	Component Level PCode Time	Total execution time for component-level PeopleCode (ms).

**PMU 516 PostBuild PCode Summary**

PMU 516 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	516
Label	PostBuild PCode Summary
Description	Component Events

**Filter Level**

PMU 516 has a filter level of 5.

**Sampling Enabled**

Yes.

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Generic	Generic - Varies based upon server and service request.
2	PeopleCode Event	PeopleCode Event (such as SavePreChange).

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
5	Component PCode Exec Count	Execution count for component-level PeopleCode.
6	Component Level PCode Time	Total execution time for component-level PeopleCode (ms).

---

## PMU 700 PPMI Servlet

PMU 700 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	700
Label	PPMI Servlet
Description	Reported at entry and exit of PPMI servlet

### **Filter Level**

PMU 700 has a filter level of 5.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Session ID	Web server session ID.
2	IP Address	Client IP address.

---

## PMU 800 Monitor Servlet Request

PMU 800 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	800

<b>Attribute</b>	<b>Value</b>
Label	Monitoring Servlet Request
Description	Monitoring Servlet Incoming Buffer from Agent

### **Filter Level**

PMU 800 has a filter level of 6.

### **Sampling Enabled**

Yes.

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Session ID	Web server session ID.
2	IP Address	Client IP address.
3	Agent ID	PeopleSoft Performance Monitor Agent ID.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Monitor Buffer Size (bytes)	Number of data objects (Events & PMUs) in incoming buffer.
2	Collator Group Count	Number of collator groups to which events were reported.
3	Agent Version	Agent version number.
4	Not used	Not used
5	Not used	Not used
6	Not used	Not used
7	Not used	Not used

### **Additional Label**

None.

## PMU 801 Analytic Server Request

PMU 801 has the following identification attributes.

<i>Attribute</i>	<i>Value</i>
ID	801
Label	Analytic Server Request
Description	Request to Analytic Server

## Filter Level

PMU 801 has a filter level of 5.

### Contexts

<i>Context</i>	<i>Value</i>	<i>Description</i>
1	Analytic Instance	Analytic instance describing the request.
2	Request Name	Describes the specific request code. For example, Load Analytic Instance — LOAD INSTANCE.
3	Analytic Model	If available, the analytic model for which the request was made.

### Metrics

<i>Metric</i>	<i>Value</i>	<i>Description</i>
1	VM Bytes at Request Start	Total virtual memory consumed by the process at the start of the request.
2	Working Set (bytes) at Request Start	Amount of physical memory consumed by the process at the completion of the request.
3	VM Bytes at Request End	Total virtual memory consumed by the process at completion of the request.
4	Working Set (bytes) at Request End	Amount of physical memory consumed at the completion of the request.

## PMU 802 Synchronize/Read Data from Data Cache Summary

PMU 802 has the following identification attributes.

<i>Attribute</i>	<i>Value</i>
ID	802
Label	Analytic Data Cache Read Request Summary
Description	Read data request to the Analytic Server Data Cache Summary

### **Filter Level**

PMU 802 has a filter level of 5 (Verbose).

### **Contexts**

<i>Context</i>	<i>Value</i>	<i>Description</i>
1	Analytic Instance	Analytic instance describing the request.
2	Transaction Name	Optimization/ACE (Analytic Calculation Engine) transaction.

### **Metrics**

<i>Metric</i>	<i>Value</i>	<i>Description</i>
1	Total Records Read	Total number of records read.
2	Total Rows Read	Total number of rows read.

## PMU 803 Synchronize/Write Data to Data Cache Summary

PMU 803 has the following identification attributes.

<i>Attribute</i>	<i>Value</i>
ID	803
Label	Analytic Data Cache Write Request Summary

<b>Attribute</b>	<b>Value</b>
Description	Write data request to Analytic Server Data Cache Summary

### **Filter Level**

PMU 803 has a filter level of 5 (Verbose).

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Analytic Instance	Analytic instance describing the request.
2	Transaction Name	Optimization/ACE (Analytic Calculation Engine) transaction.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Total Records Written	Total number of records written to the database.
2	Total Rows Written	Total number of rows written to the database.

## **PMU 804 Synchronize/Read Data from Data Cache Details**

PMU 804 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	804
Label	Analytic Data Cache Read Request Details.
Description	Read data request to Analytic Server Data Cache Details.

### **Filter Level**

PMU 804 has a filter level of 6 (Debug).

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Analytic Instance	Analytic instance describing the request.
2	Transaction Name	Optimization/ACE transaction.
3	Record Name	Name of the record to be read from the Data Cache.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Total Records Read	Total number of rows read for the record.
2	Total New Rows Inserted in Memory	Total number of rows updated in memory.
3	Total Rows Updated in Memory	Total number of rows updated in memory.
4	Total Rows Deleted in Memory	Total number of rows deleted in memory.

## **PMU 805 Synchronize/Write Data to Data Cache Details**

PMU 805 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	805
Label	Analytic Data Cache Write Request Details
Description	Write data request to Analytic Server Data Cache Details

### **Filter Level**

PMU 805 has a filter level of 6 (Debug).

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Analytic Instance	Analytic instance describing the request.

<b>Context</b>	<b>Value</b>	<b>Description</b>
2	Transaction Name	Optimization/ACE transaction.
3	Record Name	Name of the record to be written to the Data Cache.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Total Records Modified	Total number of records modified for the record.
2	Total New Rows Inserted into the Data Cache	Total number of new rows inserted into the Data Cache.
3	Total Rows Updated in the Data Cache	Total number of rows updated in the Data Cache.
4	Total Rows Deleted in the Data Cache	Total number of rows deleted in the Data Cache.

## **PMU 806 Load ACE Model Trees Summary**

PMU 806 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	806
Label	Load ACE Model Trees Summary
Description	Load ACE Model Trees Summary

### **Filter Level**

PMU 806 has a filter level of 5 (Verbose).

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Analytic Instance	Analytic instance describing the request.
2	Analytic Model	Analytic Model Name.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Total ACE Trees Loaded	Total number of ACE trees loaded.
2	Total Number of Nodes Read	Total number of tree nodes read.
3	Total Number of Tree Node Override Values	Total number of tree node override values read.

**PMU 807 Load ACE Model Trees Details**

PMU 807 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	807
Label	Load ACE Model Trees Details
Description	Load ACE Model Trees Details

**Filter Level**

PMU 807 has a filter level of 6 (Debug).

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Analytic Instance	Analytic instance describing the request.
2	Analytic Model	Analytic Model Name.
3	Dimension Name	Dimension for which tree is read.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Total Number Nodes Read for Tree	Total number of tree nodes read for this tree.
2	Total Number of Tree Node Override Values	Total number of tree node override values read for this tree.

## PMU 808 Recalculate ACE Model Summary

PMU 808 has the following identification attributes.

<i>Attribute</i>	<i>Value</i>
ID	808
Label	Recalculate ACE Model Summary
Description	Recalculate ACE Model Summary

### **Filter Level**

PMU 808 has a filter level of 5 (Verbose).

### **Contexts**

<i>Context</i>	<i>Value</i>	<i>Description</i>
1	Analytic Instance	Analytic instance describing the request.
2	Analytic Model	Analytic Model Name.

### **Metrics**

<i>Metric</i>	<i>Value</i>	<i>Description</i>
1	Number of Modified Cubes	Total number of modified cubes.
2	Number of Cells Modified	Total number of cells (if one cube is modified).
3	Total Number of Cubes Evaluated	Total number of cubes evaluated.
4	Total Number of Cells Evaluated	Total number of cells evaluated.
5	Number of Strongly Connected Cube Systems Processed	Total number of strongly connected cube systems evaluated.

## PMU 809 Recalculate ACE Model Details

PMU 809 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	809
Label	Recalculate ACE Model Details
Description	Recalculate ACE Model Details

### **Filter Level**

PMU 809 has a filter level of 6 (Debug).

### **Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Analytic Instance	Analytic instance describing the request.
2	Analytic Model	Analytic Model Name.
3	First Cube in Strongly Connected Cubes	First cube in strongly connected cubes representing the evaluation unit.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Total Number of Cubes Evaluated	Total number of cubes evaluated.
2	Total Number of Cells Evaluated	Total number of cells evaluated.
3	Number of Iterations of a Strongly Connected Cube System	Total number of iterations on the strongly connected cube system.

## **PMU 810 ACE Model Get Data**

PMU 810 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	810
Label	ACE Model Get Data
Description	ACE Model Get Data

**Filter Level**

PMU 810 has a filter level of 5 (Verbose).

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Analytic Instance	Analytic instance describing the request.
2	Analytic Model	Analytic model name.
3	Cube Collection Name	Cube collection for which the data is read.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Total Number of Rows	Total number of rows in the cube collection.
2	Number of Data Rows of Returned	Total number of data rows returned.

**PMU 811 ACE Model Set Data**

PMU 811 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	811
Label	ACE Model Set Data
Description	ACE Model Set Data

**Filter Level**

PMU 811 has a filter level of 5 (Verbose).

**Contexts**

<b>Context</b>	<b>Value</b>	<b>Description</b>
1	Analytic Instance	Analytic instance describing the request.
2	Analytic Model	Analytic Model Name.
3	Cube Collection Name	Cube collection for which the data is updated.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Total Number of Rowset Rows	Total number of rows in the rowset.
2	Number of Data Rows of Updated	Number of rows updated in the cube collection.

## Appendix B

# Event Definition Reference

This appendix contains the structure of each event that is defined in the PeopleSoft system.

---

**Note.** Currently, only one event set definition exists, set 1.

---

---

## Event 150

Event 150 has the following identification attributes.

<i>Attribute</i>	<i>Value</i>
ID	150
Label	JVM Status
Description	JVM Status

### ***Filter Level***

Event 150 has a filter level of 4.

### ***Metrics***

<i>Metric</i>	<i>Value</i>	<i>Description</i>
1	%JVM Memory Free	JVM % of available memory.
2	Max JVM Memory Available	JVM Max bytes available.
3	Sessions in Web-App	JVM Number of sessions in Web Application.
4	Execute Threads	JVM Number of execute threads.
5	Busy Threads	JVM Number of busy threads.
6	Domain Count	Number of domains on the web server.

---

## Event 151

Event 151 has the following identification attributes.

<i>Attribute</i>	<i>Value</i>
ID	151
Label	Network Status
Description	Netstat socket counts and other network traffic metrics

### **Filter Level**

Event 151 has a filter level of 4.

### **Metrics**

<i>Metric</i>	<i>Value</i>	<i>Description</i>
1	Time_Wait Sockets	Number of sockets in Time_Wait state reported by Netstat.
2	Close_Wait Sockets	Number of sockets in Close_Wait state reported by Netstat.
3	Established Sockets	Number of sockets in Established state reported by Netstat.
4	JOLT Traffic (bytes/sec)	Network Status - Jolt bytes per sec.
5	HTTP (bytes/sec)	Network Status - HTTP bytes per sec.

---

## Event 152

Event 152 has the following identification attributes.

<i>Attribute</i>	<i>Value</i>
ID	152
Label	Web Site Status
Description	Web site status reported by web server Domain Monitor

**Filter Level**

Event 152 has a filter level of 4.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Requests to all Servlets	Total requests to all servlets (web server Domain Monitor).
2	Servlet Requests (last minute)	Servlet requests in last minute (web server Domain Monitor).
3	Avg Request Time (last minute)	Average request time in last minute (web server Domain Monitor).
4	Time in all Servlets (ms)	Total servlet request time (web server Domain Monitor).
5	Current Sessions	Current number of sessions (web server Domain Monitor).
7	Site Path	Site Path.

---

**Event 153**

Event 153 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	153
Label	Web Servlet Status
Description	Web servlet status reported by Web Server Domain Monitor

**Filter Level**

Event 153 has a filter level of 4.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Requests to all Servlets	Total requests to this servlet (Web Server Domain Monitor).
2	Servlet Requests (last minute)	Servlet requests in last minute (Web Server Domain Monitor).
3	Avg Request Time (last minute)	Average request time in last minute (Web Server Domain Monitor).
4	Time in this Servlet (ms)	Total time in this servlet (Web Server Domain Monitor).
7	Servlet Name	Servlet name for reported event (Web Server Domain Monitor).

### **Additional Data Label**

Servlet Path

---

## **Event 200**

Event 200 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	200
Label	Resources Per Process
Description	Machine resource utilization metrics from C++ processes

### **Filter Level**

Event 200 has a filter level of 4.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	%CPU Used	%CPU currently consumed.
2	CPU Time (secs)	Total User Time - CPU time consumed by this process (secs).

<b>Metric</b>	<b>Value</b>	<b>Description</b>
3	VM (bytes)	Total Virtual Memory consumed by process.
4	Working Set (bytes)	Amount of physical memory consumed by the process.

## Event 300

Event 300 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	300
Label	Host Resource Status
Description	Host Resource metrics from PSMONITORSRV

### **Filter Level**

Event 300 has a filter level of 4.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	%CPU Used	%CPU currently consumed.
2	%Memory Used	%Memory Consumed.
3	Hard Page Faults/Second	Hard page faults per second.
4	Total Tuxedo Connections	Total Tuxedo "pclt" connections.
5	Total Tuxedo Requests Queued	Total Tuxedo Requests Queued (Sum of "pq").

### **Additional Data Label**

Tuxedo Domain Trace Levels.

---

## Event 301

Event 301 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	301
Label	Tuxedo "pq" Row
Description	Reported in groups to simulate a Tuxedo command line "pq"

### **Filter Level**

Event 301 has a filter level of 4.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Server Count	Number of Tuxedo servers.
2	Queue Length	Number of Tuxedo service requests in queue.
3	Avg. Queue Length	Average Tuxedo queue length.
7	Queue Name	Tuxedo Queue Name.

### **Additional Data Label**

Server Name.

---

## Event 302

Event 302 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	302
Label	Tuxedo "psr" Row

<b>Attribute</b>	<b>Value</b>
Description	Reported in groups to simulate a Tuxedo command line "psr"

**Filter Level**

Event 302 has a filter level of 4.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Server Instance	Tuxedo Server Instance ID.
2	Total Requests	Total Tuxedo Requests to this Server.
3	PID	O/S Process Identifier (PID).
7	Server Name	Tuxedo Server Name.

**Additional Data Label**

Current Service.

**Event 350**

Event 350 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	350
Label	Master Scheduler Status
Description	Master Scheduler Process Status

**Filter Level**

Event 350 has a filter level of 4.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Active Processes	Process Scheduler - Number of processes running.
2	Queued Processes	Process Scheduler - Number of processes queued.
3	Blocked Processes	Process Scheduler - Blocked Processes.
4	Unused Process Slots	Process Scheduler - Available Process Slots.

## Event 351

Event 351 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	351
Label	Master Scheduler Detail
Description	Master Scheduler Details per Process Scheduler Server

### **Filter Level**

Event 351 has a filter level of 4.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Active Processes	Process Scheduler - Number of processes running
4	Unused Process Slots	Process Scheduler - Available Process Slots
7	Server Name	Server Name

### **Additional Data Label**

Process Type.

---

## Event 354

Event 354 has the following identification attributes.

<i>Attribute</i>	<i>Value</i>
ID	354
Label	Batch Queue Details
Description	Master Scheduler Queue Details per Process Type

### ***Filter Level***

Event 354 has a filter level of 4.

### ***Metrics***

<i>Metric</i>	<i>Value</i>	<i>Description</i>
1	Queued Processes	Process Scheduler - Number of processes queued.
2	Blocked Processes	Process Scheduler - Blocked Processes.
7	Process Type	Process Scheduler - Process type.

---

## Event 355

Event 355 has the following identification attributes.

<i>Attribute</i>	<i>Value</i>
ID	355
Label	Killed Query
Description	Query killed by PSMONITORSRV at user request

### ***Filter Level***

Event 355 has a filter level of 3.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Timed-out?	Boolean - has the query timed-out?
7	Query Name	Name of the Query that was killed.

**Additional Data Label**

Query details.

---

**Event 356**

Event 356 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	356
Label	Server Process Recycle
Description	Server processes recycle/shutdown event

**Filter Level**

Filter level is 2 (Warning).

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	167	Service timeout setting.
2	168	Recycle counts setting.
3	169	Allowed consecutive service failure.
4	170	Recycle reason.
5		Normal shutdown.

---

## Event 500

Event 500 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	500
Label	JOLT Service Exception
Description	JOLT Service Exception (for example, an application server service timeout)

### **Filter Level**

Event 500 has a filter level of 3.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	PMU instance affected	PMU instance effected.
2	JOLT Error Number	JOLT Error Number.
7	Jolt Error Detail	Jolt Error Detail

---

## Event 600

Event 600 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	600
Label	PSPING
Description	PSPING metrics forwarded from browser

### **Filter Level**

Event 600 has a filter level of 3.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Network Latency (ms)	PSPING - Network latency (milliseconds).
2	WebServer Latency (ms)	PSPING - Web Server latency (milliseconds).
3	AppServer Latency (ms)	PSPING - application server latency (milliseconds).
4	DB Latency (milliseconds)	PSPING - database latency (milliseconds).
7	IP Address	

**Additional Data Label**

OPRID (user ID), JSessionID.

**Event 601**

Event 601 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	601
Label	User Monitoring Level Override
Description	Performance Trace user override of filter level

**Filter Level**

Event 601 has a filter level of 4.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Agent Filter Mask	Agent Filter Mask.
7	User Initiated PMU Name	User Initiated PMU Name.

---

## Event 701

Event 701 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	701
Label	PPMI Notify Interest
Description	PPMI Client registration - registerNotificationInterest

### **Filter Level**

Event 701 has a filter level of 4.

### **Additional Data Label**

Additional Data.

---

## Event 702

Event 702 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	702
Label	PPMI Notify Listener
Description	PPMI Client registration - registerNotificationListener

### **Filter Level**

Event 702 has a filter level of 4.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
7	PPMI Group Name	PPMI Group Name.

**Additional Data Label**

Additional Data.

**Event 703 PPMI Notify Cancel Interest**

Event 703 has the following identification attributes.

<i>Attribute</i>	<i>Value</i>
ID	703
Label	PPMI Notify Cancel Interest
Description	PPMI Client disconnected - cancelNotificationInterest

**Filter Level**

Event 703 has a filter level of 4.

**Metrics**

<i>Metric</i>	<i>Value</i>	<i>Description</i>
7	PPMI Group Name	PPMI Group Name

**Additional Data Label**

Additional Data.

**Event 704 PPMI Notify Agent Change**

Event 704 has the following identification attributes.

<i>Attribute</i>	<i>Value</i>
ID	704
Label	PPMI Notify Agent Change
Description	Administrator altered agent state -notifyStateChange

**Filter Level**

Event 704 has a filter level of 4.

**Event 705 PPMI Notify Monitor Change**

Event 705 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	705
Label	PPMI Notify Monitor Change
Description	Administrator altered cluster state - notifyStateChange

**Filter Level**

Event 705 has a filter level of 4.

**Event 801**

Event 801 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	801
Label	Monitor Server Buffer Overrun
Description	Monitoring Server Alarm - Buffer Overrun

**Filter Level**

Event 801 has a filter level of 3.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Dropped Items	Number of objects that were dropped from the buffer.
7	PPMI Client URL	URL of PPMIClient.

---

## Event 802 Monitoring Server Client Lost

Event 802 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	802
Label	Monitoring Server Client Lost
Description	Monitoring Server Alarm - Unable to Contact PPMIClient

### **Filter Level**

Event 802 has a filter level of 3.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
7	PPMI Client URL	URL of PPMIClient.

---

## Event 803 Monitoring Server Data Lost

Event 803 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	803
Label	Monitoring Server Data Lost
Description	Monitor Server Alarm - Data lost due to PPMI Client error

### **Filter Level**

Event 803 has a filter level of 3.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
7	PPMI Client URL	URL of PPMIClient.

---

## Event 900 Agent Buffer Overrun

Event 900 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	900
Label	Agent Buffer Overrun
Description	Agent Alarm - Buffer Overrun Alarm

### **Filter Level**

Event 900 has a filter level of 3.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Dropped Items	Number of objects that were dropped from the buffer.
2	Buffer Size (bytes)	Estimated size of current buffer, when item was dropped.

---

## Event 901 Agent Init Notification

Event 901 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	901
Label	Agent Init Notification
Description	Agent Event - Agent Initialization

**Filter Level**

Event 901 has a filter level of 4.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Contact Attempts	Number of failed contact attempts before success.

---

## Event 902 Agent Configuration Ack

Event 902 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	902
Label	Agent Configuration Ack
Description	Agent Event - Agent Dynamic Configuration State Change Ack

**Filter Level**

Event 902 has a filter level of 4.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Agent Buffer Max Size (bytes)	Agent Buffer Maximum Size.
2	Agent Send Interval (secs)	Agent Buffer Send Interval.
3	Agent HeartBeat Interval (secs)	Agent HeartBeat Interval.
4	Agent Filter Mask	Agent Filter Mask.
5	Agent Idle Time (secs)	Agent Idle Time.

---

## Event 903 Agent Contact Notification

Event 903 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	903
Label	Agent Contact Notification
Description	Agent Event - Contact Initiated with Monitoring Server

### **Filter Level**

Event 903 has a filter level of 6.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Contact Attempts	Number of failed contact attempts before success.

---

## Event 904 Analytic Server Status

Event 904 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	904
Label	Analytic Server Status
Description	Analytic Server Usage Status

### **Filter Level**

Event 904 has a filter level of 3.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Total Number of Analytic Servers	Total number of configured analytic servers.
2	Total Number of Started Analytic Servers	Number of started analytic servers.
3	Number Of Free Analytic Server	Number of Analytic Servers, started up, but free.

---

## Event 905 Analytic Server Load Event

Event 905 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	905
Label	Analytic Server Load Event
Description	Analytic Instance loading in a Analytic Server

### **Filter Level**

Event 905 has a filter level of 3.

### **Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	VM Bytes	Total Virtual Memory Consumed by the process after the load.
2	Working Set(bytes)	Amount of physical memory consumed by the process after the load.

### **Additional Data Label**

Analytic Instance Name.

---

## Event 906 Out of Analytic Servers

Event 906 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	906
Label	Out of Analytic Servers
Description	Out of Analytic Servers Alarm

**Filter Level**

Event 906 has a filter level of 3.

**Metrics**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
1	Total Number of Analytic Servers	Total number of configured analytic servers.
2	Total Number of Started Analytic Servers	Number of started analytic servers.

**Additional Data Label**

Process Type.

---

## Event 907 Analytic Server Recycle Event

Event 907 has the following identification attributes.

<b>Attribute</b>	<b>Value</b>
ID	907
Label	Analytic Server Recycle Event
Description	Analytic Server Recycling Event Details

**Filter Level**

Event 907 has a filter level of 4.

**Additional Data Label**

Analytic Server Tuxedo Instance Number



## Appendix C

# PeopleSoft Performance Monitor Diagnostic Plug-in

This section provides an overview and discusses:

- Running Performance Monitor diagnostics.
- Reviewing diagnostic information.

---

## Understanding Performance Monitor Diagnostic Plug-in

When configuring the Performance Monitor, you are required to specify or verify a variety of field values, on multiple PeopleTools pages, within several PeopleTools administrative interfaces, including:

- Performance Monitor pages.
- Integration Broker pages.
- Web Profile pages.
- Security pages.

If troubleshooting is ever required, or you need to review your system settings, the system administrator needs to access all of these settings in various locations. To view Performance Monitor settings in a single report, PeopleTools provides the Performance Monitor Diagnostic Plug-in. This enables you to view all the key settings in a central location to determine any necessary modifications or corrections.

---

**Note.** The Performance Monitor Diagnostic Plug-in generates a read-only report. It does not physically aggregate the settings in a central location for modification. If settings need to be modified, you need to navigate to the appropriate administrative interface to make changes.

---

### See Also

*PeopleTools 8.51 PeopleBook: Data Management, "Working With The Diagnostics Framework"*

*PeopleTools 8.51 PeopleBook: Data Management, "Developing Diagnostic Plug-Ins"*

---

## Running Performance Monitor Diagnostics

To run the Performance Monitor diagnostic plug-in:

1. Register the Performance Monitor diagnostic plug-in.

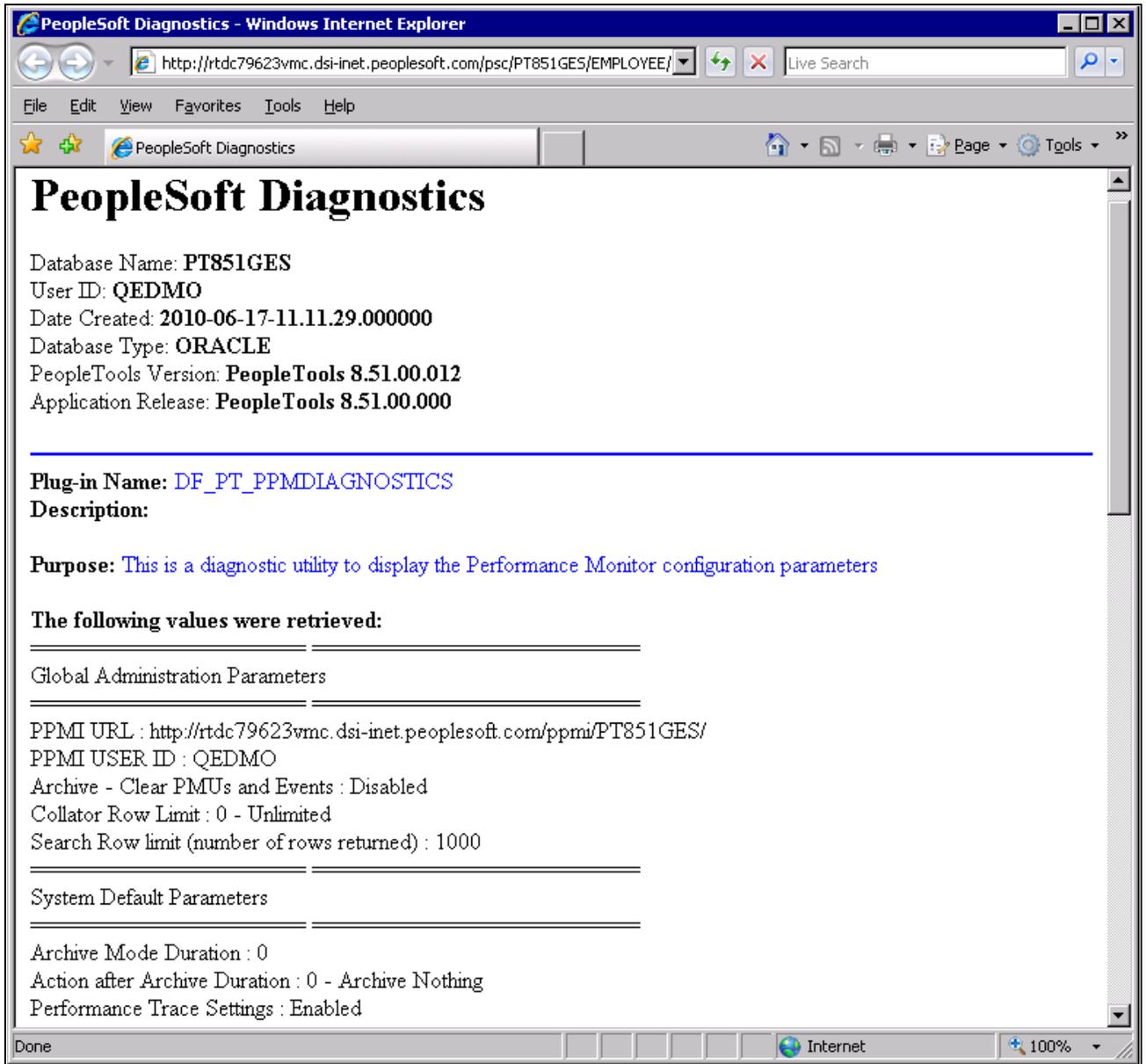
Navigate to Application Diagnostics, Register Diagnostics, and use the Plug-in Name drop-down list to select *DF\_PT\_PPMDIAGNOSTICS*.

2. Select Application Diagnostics, Launch Diagnostics.
3. Select the check box next to *DF\_PT\_PPMDIAGNOSTICS*.
4. Specify how you want the report delivered:
  - Email report: delivers the report to the specified email address(s). SMTP must be configured for this option to succeed.
  - Display report in browser: delivers the report in a separate browser window.
5. Click Generate Diagnostics.

---

## Reviewing Diagnostic Information

The generated report appears similar to the following:



Sample Performance Monitor diagnostic plug-in output

The Performance Monitor diagnostic report displays the following settings, grouped by where they appear in the PeopleTools interface.

<i>Diagnostic Report Section</i>	<i>Configuration Parameters Displayed</i>
Global Administration Parameters	PPMI URL PPMI USER ID Archive - Clear PMUs and Events Collator Row Limit Search Row limit (number of rows returned)

<b>Diagnostic Report Section</b>	<b>Configuration Parameters Displayed</b>
System Default Parameters	Archive Mode Duration Action after Archive Duration Performance Trace Settings PMU Timeout (in days) Agent Event Sample Rate (sec) Agent Buffering Interval (sec) Agent Filter level Agent PMU sample rate (1/X) Agent Heartbeat Interval (sec) Agent Maximum Buffer size (bytes)
System Definitions	Database Name Tools release Archive Mode Duration (days) Archive Mode Operation Performance Trace Settings PMU Timeout (in days) Agent Event Sample Rate (sec) Agent Buffering Interval (sec) Agent PMU sample rate (1/X) Agent Heartbeat Interval (sec) Agent Maximum Buffer size (bytes) Agent Filter Levels (Server/Filter Level)
Specify Monitor	Monitor URL
IB Information (Integration Broker)	IB Gateway URL Number of IB Gateway Connectors
Web Profile Information	Configured Webprofile Webprofile Buffer Limit Webprofile PPM Agent Enabled status
Security	Permission List for the current user

## Appendix D

# Working With Usage Monitor

This section provides an overview and discusses:

- Enabling Usage Monitor.
- Verifying Usage Monitor data collection.
- Running a Usage Monitor test.
- Integrating Usage Monitor data with Lifecycle Management Tools.

---

## Understanding Usage Monitor

The Usage Monitor provides additional information for analysis when determining test coverage with the PeopleSoft Test Framework. In the PeopleSoft Test Framework (PTF), a test coverage report correlates PeopleTools project data with PTF test metadata to identify components, menus, pages, records and fields that are referenced in PTF tests. When used in conjunction with Usage Monitor, test coverage correlations can be extended to include information on all definitions (including PeopleCode, Application Engine programs, and so on) from a usage perspective. A test coverage report identifies which objects included in the change project are referenced by which PTF test. Definitions included in the change project that are not referenced in the PTF test metadata or Usage Monitor data appears in the report identified as a coverage gap.

---

**Note.** In Performance Monitor, you enable Usage Monitor and verify the usage data is being collected. You can only incorporate the usage data into meaningful reports by using the features provided by the PeopleSoft Test Framework. To access the PeopleSoft Test Framework, select PeopleTools, Lifecycle Tools, Test Framework.

---

### See Also

*PeopleTools 8.51 PeopleBook: PeopleSoft Test Framework, "Identifying Change Impacts," Using Usage Monitor Data with PTF*

---

## Enabling Usage Monitor

To enable Usage Monitor:

- Make sure that the Usage Monitoring State parameter is set correctly for the application server domain.

See *PeopleTools 8.51 PeopleBook: System and Server Administration*, "Setting Application Server Domain Parameters," Usage Monitoring State.

- Select the Enable Usage Monitoring check box on the System Defaults page (PeopleTools, Performance Monitor, Administration, System Defaults).

See [Chapter 4, "Administering the Performance Monitor," Setting System Definition Defaults, page 29.](#)

---

## Verifying Usage Monitor Data Collection

The Usage Monitor data is collected by PMU 440.

To verify that Performance Monitor is collecting Usage Monitor data:

1. Select PeopleTools, Performance Monitor, History, Completed PMUs.
2. On the Completed PMUs page, in the PMU Identifier field, enter *440*.
3. Click Search.
4. Verify that the grid populates with Usage Monitoring rows.

---

**Note.** This information is not intended to be analyzed outside of PTF.

---

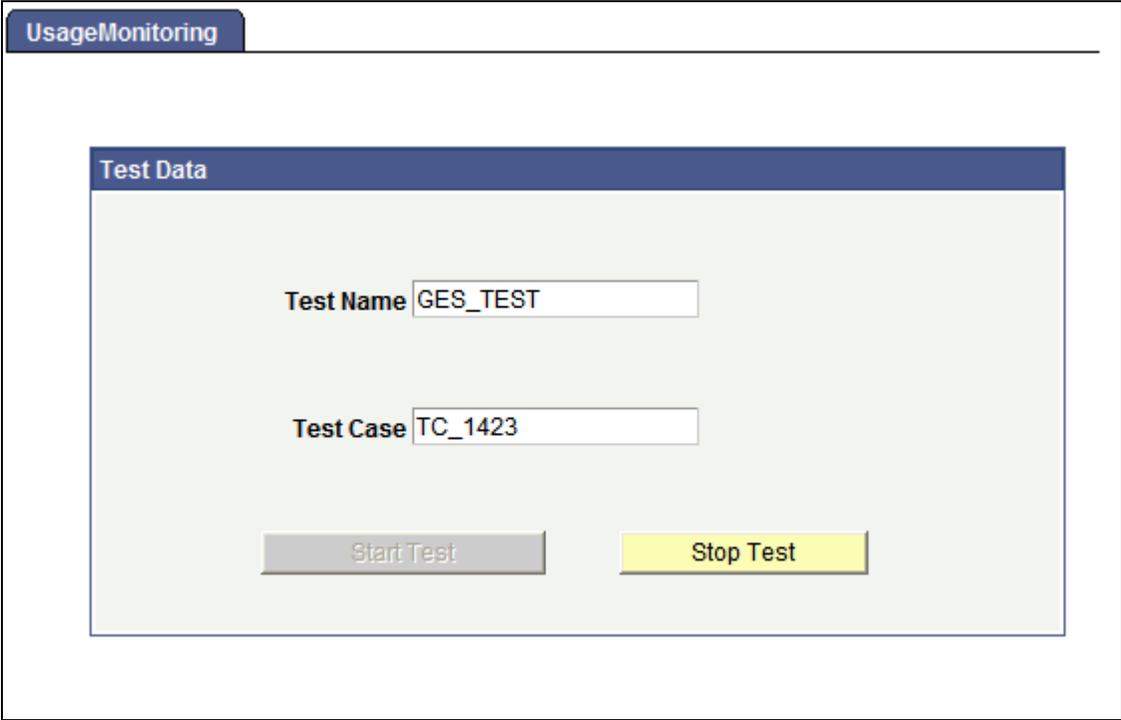
---

## Running a Usage Monitor Test

To run a Usage Monitor test, use the Usage Monitoring page, which you access by using the Usage Monitoring top-level menu item.



Usage Monitoring menu item



The screenshot shows a web interface for 'UsageMonitoring'. Inside, there is a 'Test Data' section with two input fields: 'Test Name' containing 'GES\_TEST' and 'Test Case' containing 'TC\_1423'. Below these fields are two buttons: 'Start Test' (grey) and 'Stop Test' (yellow).

Usage Monitoring page

Enter a Test Name and Test Case value to associated specific usage data with individual tests and test cases. These values are required.

To begin a test, click Start Test.

To end a test, click Stop Test.

---

## Integrating Usage Monitor Data with Lifecycle Management Tools

The usage data generated by Usage Monitor within the Performance Monitor framework, is not intended to be viewed outside of the PeopleSoft Test Framework. The PeopleSoft Test Framework provides data aggregation programs and reports that group and display the data in a meaningful way for enhanced test coverage analysis.

### See Also

*PeopleTools 8.51 PeopleBook: PeopleSoft Test Framework, "Understanding PeopleSoft Test Framework"*



# Index

## A

Agent Definitions page 35

Agent Filters page 36

agents

- activating/deactivating 36
- communicating with monitor servlets 144
- configuring client authentication 147
- connecting to monitors 17
- disabling 42
- disabling in safe mode 21
- domain resource agents 6
- identifying for events 104
- identifying for PMUs 83, 97
- identifying for users 106
- Master Scheduler 79
- monitoring 45
- monitoring agent history for application
  - server domains 72
- monitoring agent history for Process Scheduler servers 76
- monitoring agent history for web servers 68
- monitoring definitions 35
- process instrumentation agents 6
- setting buffering intervals 34
- setting buffer sizes 35
- setting event sampling rates 34
- setting filtering 36
- setting heartbeat intervals 35
- setting up SSL 145
- understanding 6
- understanding monitor URLs 20
- using certificates 147

AIX

- using Event 200 87
- using Event 300 89

Application Engine programs

- maintaining performance data 19
- PSPM\_ARCHIVE 33, 39, 93
- PSPM\_ARCHREC 40
- PSPM\_LOOKUP 40
- reaper program *See Also* reaper program

application server

- monitoring Jolt requests 63
- recycled processes 86

Application Server Domain page 71

application server domains

- monitored system elements 6
- monitoring 65
- monitoring host resource status 72
- monitoring performance 71
- monitoring PQ event rows 73
- monitoring PSR event rows 73
- monitoring server processes 74

application servers

- disabling agents 42
- domains *See Also* application server domains
- enabling tracing 43
- monitoring system elements 8
- setting up for monitored systems 21
- setting up for monitoring systems 16

using Event 200 87

using Event 300 89

using Event 301 92

using Event 302 92

archiving

- deleting performance history data 28
- scheduling for performance data 39
- setting archive mode 33
- understanding aborted programs 40
- understanding performance history data 93
- "clone table" warning 94, 103

authentication

- setting up for PPMI client 143
- setting up PPMI authentication 18
- setting up SSL for clients 146
- understanding PMU 113 163
- understanding PMU 413 180

## B

buffers 216, 217, 218, 219, 220, 221, 222, 223

- monitoring for agents 241, 242
- monitoring for components
  - 121, 128, 169, 170, 171, 184, 185
- monitoring for monitor servlets 213, 239
- monitoring for PMU/event queue 46
- setting agent buffering intervals 34
- setting for performance data 35

built-in SQL

- monitoring execution durations for components 129
- monitoring execution durations for user requests 122

## C

caches

- cache memory hits 119, 179
- cache misses 119, 179
- clearing browser caches 57
- sample queries 41

Central Processing Unit (CPU) usage

*See* CPU usage

certificates

- certificate authorities (CAs) 145, 146
- client *See Also* client certificates
- digital 145, 146
- root 145, 146

charts

- Component Durations 129
- Component Sizes 128
- downloading to Excel 130
- Duration Averages 135, 137, 139, 140, 142
- duration charts 99
- metric charts 98, 102, 104
- monitoring performance data 12, 59
- PeopleCode Durations 122, 130
- PIA Action Durations 133

- PIA Action Response Sizes 133
- Portal Action Durations 131
- Request Time 118
- Think Time 118
- Top Average Response Sizes 135
- Top Components 142
- Top Cumulative Durations 134, 136, 138, 140, 142
- Top Cumulative Response Sizes 135
- Top PeopleCode Events 136
- Top PeopleCode Executions 138
- Top Portal Content Requests 134
- Top SQL Statements 140
- understanding 113
- understanding legends 114
- User Request *See Also* User Request chart viewing in grid format 114, 130
- client certificates
  - setting up SSL client authentication 146
  - understanding types 147
- clone table warning 94, 103
- Completed PMUs page 94
- component buffers
  - monitoring 121, 128
  - understanding PMU 401 169
  - understanding PMU 402 170
  - understanding PMU 403 171
  - understanding PMU 418 184
  - understanding PMU 419 185
- Component Durations chart 129
- components
  - buffers *See Also* component buffers
  - Duration Averages chart 142
  - monitoring 140
  - monitoring averages/standard deviations 128
  - monitoring durations 129
  - monitoring PMUs 120
  - monitoring sizes 128
  - monitoring statistics 127
  - PeopleCode Durations chart 130
  - Top Components chart 142
  - Top Cumulative Durations chart 142
  - viewing performance of 123
- Component Sizes chart 128
- Component Statistics page 127
- Component Trace page 123
- contexts
  - monitoring for PMUs 98
  - PMU 100 157
  - PMU 101 158
  - PMU 102 159
  - PMU 106 160
  - PMU 108 161
  - PMU 109 162
  - PMU 113 163
  - PMU 114 164
  - PMU 115 164
  - PMU 116 166
  - PMU 117 166
  - PMU 400 167
  - PMU 401 169
  - PMU 402 170
  - PMU 403 171
  - PMU 404 172
  - PMU 406 173
  - PMU 407 174
  - PMU 408 175
  - PMU 409 176
  - PMU 410 177
  - PMU 411 178
  - PMU 412 179
  - PMU 413 180
  - PMU 414 181
  - PMU 415 182
  - PMU 416 183
  - PMU 417 183
  - PMU 418 184
  - PMU 419 185
  - PMU 420 186
  - PMU 421 187
  - PMU 422 187
  - PMU 423 188
  - PMU 424 189
  - PMU 425 190
  - PMU 426 190
  - PMU 427 191
  - PMU 428 192
  - PMU 439 194
  - PMU 440 195
  - PMU 500 195
  - PMU 501 197
  - PMU 502 198
  - PMU 503 199
  - PMU 504 200
  - PMU 505 201
  - PMU 506 202
  - PMU 507 203
  - PMU 508 204
  - PMU 509 205
  - PMU 510 206
  - PMU 511 207
  - PMU 512 208
  - PMU 513 209
  - PMU 514 210
  - PMU 515 211
  - PMU 516 212
  - PMU 700 213
  - PMU 800 213
  - PMU 801 215
  - PMU 802 216
  - PMU 803 216
  - PMU 804 217
  - PMU 805 218
  - PMU 806 219
  - PMU 807 220
  - PMU 808 221
  - PMU 809 221
  - PMU 810 222
  - PMU 811 223
  - understanding 152
  - understanding definitions 155
- counter metrics 153
- CPU usage
  - monitoring for application server domains 65, 72, 74
  - monitoring for Process Scheduler server domains 65
  - monitoring for Process Scheduler servers 76, 78
  - using Event 200 87
  - using Event 300 89
- Current User Sessions page 85

## D

databases

- estimating size of 49
- identifying for agents 36
- identifying for Master Scheduler 79
- setting up for monitored systems 20
- setting up monitoring databases 16
- understanding monitoring system databases 8
- understanding PeopleSoft application databases 6

Data Mover 48

deviation, standard *See* standard deviation

domain monitors

- application servers 7
- monitoring for web servers 68
- Process Scheduler servers 8
- understanding Event 152 226
- understanding Event 153 227
- understanding network status 69
- web servers 7

domain resource agents 6

domains

- application server
  - See Also* application servers, application server domains
- disabling 42
- disabling agents 42
- domain resource agents 6
- identifying for agents 36
- identifying for agents sending events 104
- identifying for PMU requests 97
- identifying for PMUs 83
- monitors *See Also* domain monitor
- Process Scheduler server
  - See Also* Process Scheduler server domains
- understanding stale domains 61
- web server *See Also* web server domains

Duration Averages chart

- monitoring components 142
- monitoring PeopleCode events 137
- monitoring PeopleCode programs 139
- monitoring PeopleCode SQL 140
- monitoring portal content references 135

## E

encryption

- PSPMSRV 27
- SSL *See Also* SSL

errors

- Jolt 235
- logging 43
- monitoring 62
- reporting to monitoring systems 38

event

- size calculations 50
- space requirements 50

Event History page 101

events

- Event 150 86, 225
- Event 151 86, 226
- Event 152 226
- Event 153 227

- Event 200 87, 228
- Event 300 89, 229
- Event 301 92, 230
- Event 302 92, 230
- Event 350 231
- Event 351 232
- Event 354 233
- Event 355 233
- Event 356 234
- Event 500 235
- Event 600 235
- Event 601 236
- Event 701 237
- Event 702 237
- Event 703 238
- Event 704 238
- Event 705 239
- Event 801 239
- Event 802 240
- Event 803 240
- Event 900 241
- Event 901 241
- Event 902 242
- Event 903 243
- Event 904 243
- Event 905 244
- Event 906 244
- Event 907 245
- monitoring history 101, 103
- monitoring metrics 104
- monitoring PQ event rows for application server domains 73
- monitoring PQ event rows for Process Scheduler servers 77
- monitoring PSR event rows for application server domains 73
- monitoring PSR event rows for Process Scheduler servers 77
- monitoring resource usage 86
- PeopleCode *See Also* PeopleCode events
- searching for history 101
- understanding 12, 151
- understanding definitions 154
- understanding set definitions 225

Excel, downloading charts to 130

exporting data 48

## F

failover

- configuring servlets as clusters 28
- implementing automatically 17

filtering

- monitoring for agents 45
- monitoring for events 104
- overriding for agents 55
- setting agent filters to debug 38
- setting agent filters to verbose 38
- setting for agents 36
- setting for events 155, 225
- setting for performance tracing 57
- setting for PMUs 156, 157
- setting for PSAPPSRV agents 136

firewalls

- agent communication with monitor servlets 144

- communication between monitor cluster members 145
- PSPPMRSRV communication with monitor servlets 145
- using 143
- forward proxies, setting up 144

## G

- gateway, integration *See* integration gateway
- gateway servlets 10
- gauge metrics 153
- Global Administration page 25
- global variables
  - monitoring for components 128
  - monitoring for user requests 120
- grids
  - monitoring performance data 12
  - viewing chart data 114
- GUIDs, monitored system 32, 45

## H

- host resource status
  - monitoring for application server domains 72
  - monitoring for Process Scheduler servers 76
  - using Event 300 89, 229
- HPUX
  - using Event 200 87
  - using Event 300 89
- HTTP traffic, monitoring 69

## I

- importing data 48
- instrumentation
  - process instrumentation agents 6
  - understanding 11
  - understanding PMUs 152
- integration gateway
  - configuring client authentication 147
  - servlets 10
  - specifying URLs 19
  - using certificates 148

## J

- Java Runtime Environment (JRE) 68
- Java Virtual Machine (JVM) *See* JVM
- Jolt
  - identifying the host/port for application server domains 71
  - identifying the listening port for application server domains 65
  - identifying the listening port for Process Scheduler server domains 65
  - monitoring execution times 63
  - monitoring traffic for web server domains 65
  - monitoring traffic for web servers 69

- JRE 68
- JVM
  - memory 65
  - status *See Also* JVM status
- JVM memory, monitoring 65, 68
- JVM status
  - Event 150 225
  - monitoring 68
  - using Event 150 86

## K

- key stores 147

## L

- labels
  - setting for metrics 154
  - viewing for component PMUs 120
  - viewing for contexts 155
  - viewing for events 154, 225
  - viewing for metrics 153, 156
  - viewing for PMUs 83, 156, 157
- latency
  - Event 600 235
  - monitoring user history 106
  - monitoring user sessions 86
  - PMU 116 166
- Linux
  - using Event 200 87
  - using Event 300 89
- load balancing
  - implementing automatically 17
  - implementing monitor clusters 47
- look-up tables, populating 40

## M

- Master Scheduler
  - monitoring details (Event 351) 232
  - monitoring details (Event 356) 234
  - monitoring performance 79
  - monitoring processes 79
  - monitoring Process Scheduler jobs 62
  - monitoring queue details 80, 233
  - monitoring servers 80
  - monitoring status (Event 350) 231
  - PSMSTPRC process 8
- Master Scheduler page 79
- Master Scheduler servers 80
- memory
  - JVM 65
  - monitoring for application server domains 66, 72
  - monitoring for Process Scheduler server domains 66
  - monitoring for Process Scheduler servers 76
  - using Event 300 89
  - virtual *See Also* virtual memory
- metadata
  - contexts *See Also* contexts
  - events *See Also* events

- metrics *See Also* metrics
- PMUs *See Also* PMUs
- understanding 151
- metrics
  - defining display text 154
  - Event 150 225
  - Event 151 226
  - Event 152 226
  - Event 153 227
  - Event 200 228
  - Event 300 229
  - Event 301 230
  - Event 302 230
  - Event 350 231
  - Event 351 232
  - Event 354 233
  - Event 355 233
  - Event 356 234
  - Event 500 235
  - Event 600 235
  - Event 601 236
  - Event 701 237
  - Event 702 237
  - Event 703 238
  - Event 704 238
  - Event 705 239
  - Event 801 239
  - Event 802 240
  - Event 803 240
  - Event 900 241
  - Event 901 241
  - Event 902 242
  - Event 903 243
  - Event 904 243
  - Event 905 244
  - Event 906 244
  - Event 907 245
  - modifying definitions 153
  - monitoring events 104
  - monitoring for PMUs 98
  - PMU 100 157
  - PMU 101 158
  - PMU 102 159
  - PMU 106 160
  - PMU 108 161
  - PMU 109 162
  - PMU 113 163
  - PMU 114 164
  - PMU 115 164
  - PMU 116 166
  - PMU 117 166
  - PMU 400 167
  - PMU 401 169
  - PMU 402 170
  - PMU 403 171
  - PMU 404 172
  - PMU 406 173
  - PMU 407 174
  - PMU 408 175
  - PMU 409 176
  - PMU 410 177
  - PMU 411 178
  - PMU 412 179
  - PMU 413 180
  - PMU 414 181
  - PMU 415 182
  - PMU 416 183
  - PMU 417 183
  - PMU 418 184
  - PMU 419 185
  - PMU 420 186
  - PMU 421 187
  - PMU 422 187
  - PMU 423 188
  - PMU 424 189
  - PMU 425 190
  - PMU 426 190
  - PMU 427 191
  - PMU 428 192
  - PMU 439 194
  - PMU 440 195
  - PMU 500 195
  - PMU 501 197
  - PMU 502 198
  - PMU 503 199
  - PMU 504 200
  - PMU 505 201
  - PMU 506 202
  - PMU 507 203
  - PMU 508 204
  - PMU 509 205
  - PMU 510 206
  - PMU 511 207
  - PMU 512 208
  - PMU 513 209
  - PMU 514 210
  - PMU 515 211
  - PMU 516 212
  - PMU 700 213
  - PMU 800 213
  - PMU 801 215
  - PMU 802 216
  - PMU 803 216
  - PMU 804 217
  - PMU 805 218
  - PMU 806 219
  - PMU 807 220
  - PMU 808 221
  - PMU 809 221
  - PMU 810 222
  - PMU 811 223
  - selecting for PMUs 156
  - understanding 151
  - understanding definitions 152
- monitor clusters
  - communicating between members 145
  - configuring client authentication 147
  - setting up 46
  - setting up SSL between members 146
  - using certificates 148
- monitored systems
  - default settings 29
  - defining 31
  - enabling required elements 20
  - identifying databases for agents 36
  - identifying for agents 36
  - identifying for Master Scheduler 79
  - monitoring current/assigned Process Scheduler jobs 62
  - monitoring errors/warnings 62
  - monitoring PeopleTools version history 35
  - monitoring processes for Master Scheduler 79
  - monitoring requests queued 62
  - setting up application servers 21
  - setting up databases 20

- setting up Process Scheduler servers 22
- setting up web servers 21
- specifying monitor URLs 20
- understanding 5, 6
- understanding application servers 6
- understanding databases 6
- understanding Process Scheduler servers 7
- understanding web server 7
- monitoring systems
  - creating PPMI user IDs 18
  - enabling required elements 15
  - entering descriptions 32
  - setting archive mode 33
  - setting up application servers 16
  - setting up monitoring databases 16
  - setting up Process Scheduler servers 19
  - setting up web servers 17
  - specifying integration gateway URL 19
  - specifying PPMI URLs/userIDs 18
  - understanding 5, 8
  - understanding application servers 8
  - understanding databases 8
  - understanding Process Scheduler servers 10
  - understanding production monitoring systems 11
  - understanding self-monitoring systems
    - See Also* self-monitoring systems
  - understanding web servers 9
- monitoring usage 251
- monitor servlets
  - communicating with agents 144
  - communicating with agents and monitoring systems 19
  - communicating with PSPMSRV 145
  - disabling 149
  - enabling tracing 44
  - encryption considerations 27
  - identifying 20
  - load balancing and failover 17
  - monitoring diagnostics 44
  - protecting 149
  - understanding 9
- MS Excel, downloading charts to 130
- Multi-Platform Installer program 17

## N

- network status
  - monitoring 69
  - using Event 151 86, 226
- notifications
  - events *See Also* events
  - notifying monitor servlets of configuration changes 35
  - notifying PPMI servlets of configuration changes 38
- numeric identifier metrics 153

## O

- Open PMUs page 81
- Open PMU Trees page 84
- OS/390
  - using Event 200 87

- using Event 300 89

## P

- PeopleCode
  - Duration Averages chart 139
  - Durations chart 122
  - events *See Also* PeopleCode events
  - monitoring execution durations for components 129
  - monitoring execution durations for user requests 122
  - monitoring executions 137
  - monitoring execution times 63
  - PeopleCode Durations chart 130
  - SQL *See Also* PeopleCode SQL
  - Top Cumulative Durations chart 138
  - Top PeopleCode Executions chart 138
- PeopleCode Durations chart 122, 130
- PeopleCode events
  - charts 136
  - monitoring 135
- PeopleCode global variables *See* global variables
- PeopleCode SQL
  - charts 140
  - monitoring execution durations for components 129
  - monitoring execution durations for user requests 122
  - monitoring execution times 63
  - monitoring statements 139
  - using PMU 400 167
  - using PMU 406 173
  - using PMU 427 191
  - using PMU 428 192
- PeopleSoft Application Engine programs
  - See* Application Engine programs
- PeopleSoft Internet Architecture (PIA) *See* PIA
- PeopleSoft Multi-Platform Installer program 17
- PeopleSoft Performance Monitor
  - See* Performance Monitor
- PeopleSoft Test Framework 251
- PeopleTools
  - events *See Also* events, events
  - maintaining performance data tables 39
  - monitoring PeopleCode execution times 63
  - monitoring SQL execution times 63
  - monitoring version history 35
  - PMUs *See Also* PMUs
  - system database (PTSYS) 8, 16
  - version support for Performance Monitor 11
- performance measurement units (PMUs) *See* PMUs
- Performance Monitor
  - administering 23
  - agents *See Also* agents
  - analyzing historical performance data 93
  - architecture 4
  - configuring 15
  - data model 41
  - instrumentation *See Also* instrumentation
  - metadata *See Also* metadata
  - monitored systems
    - See Also* monitored systems
  - monitoring data 12
  - monitoring system performance 59
  - monitoring systems

- See Also* monitoring systems
  - performance metrics coverage 13
  - populating look-up tables 40
  - production monitoring systems
    - See Also* production monitoring systems
  - security *See Also* security
  - self-monitoring systems 10
  - setting global system options 25
  - setting up 1, 15
  - sharing performance data with third-party systems 12
  - understanding xi, 3
  - working with tables 41
- Performance Monitor Integration (PPMI)
  - See* PPMI, PPMI
- Performance Monitor servlets *See* servlets
- performance trace
  - viewing component information 123
- Performance Trace Console
  - launching 33
  - running a trace 56
  - understanding 55
- PIA
  - configuring client authentication 147
  - monitoring statistics 132
  - PIA Action Durations chart 133
  - PIA Action Response Sizes chart 133
  - servlets *See Also* PIA servlets
  - using certificates 148
- PIA Action Durations chart 133
- PIA Action Response Sizes chart 133
- PIA servlets
  - monitoring action durations 133
  - monitoring durations for components 129
  - monitoring durations for user requests 121
  - monitoring performance 132
  - understanding 9
- PIA Statistics page 132
- pinging
  - cluster web server URLs 28
  - PPMI URLs 27
  - system performance 66
- PMU
  - size calculations 52
  - space requirements 52
- PMU details page 101
- PMUs
  - configuring sampling rate 44
  - entering search criteria 94
  - identifying definitions/sets 83
  - monitoring completed 94, 96
  - monitoring component PMUs 120
  - monitoring context definitions 98
  - monitoring details 101
  - monitoring durations 99
  - monitoring durations for components 128
  - monitoring executed PMUs 62
  - monitoring metrics 98
  - monitoring open PMUs 81
  - monitoring open PMU trees 84
  - monitoring query/script PMUs 121
  - monitoring Tuxedo PMUs 121
  - monitoring user request analytics
    - See Also* User Request chart
  - monitoring user request durations 121
  - monitoring user requests 119
  - monitoring within tree context 98
  - PMU 101 157
  - PMU 102 159
  - PMU 106 160
  - PMU 108 161
  - PMU 109 162
  - PMU 113 163
  - PMU 114 164
  - PMU 115 164
  - PMU 116 166
  - PMU 117 166
  - PMU 400 167
  - PMU 401 169
  - PMU 402 170
  - PMU 403 171
  - PMU 404 172
  - PMU 406 173
  - PMU 407 174
  - PMU 408 175
  - PMU 409 176
  - PMU 410 177
  - PMU 411 178
  - PMU 412 179
  - PMU 413 180
  - PMU 414 181
  - PMU 415 182
  - PMU 416 183
  - PMU 417 183
  - PMU 418 184
  - PMU 419 185
  - PMU 420 186
  - PMU 421 187
  - PMU 422 187
  - PMU 423 188
  - PMU 424 189
  - PMU 425 190
  - PMU 426 190
  - PMU 427 191
  - PMU 428 192
  - PMU 439 194
  - PMU 440 195
  - PMU 500 195
  - PMU 501 197
  - PMU 502 198
  - PMU 503 199
  - PMU 504 200
  - PMU 505 201
  - PMU 506 202
  - PMU 507 203
  - PMU 508 204
  - PMU 509 205
  - PMU 510 206
  - PMU 511 207
  - PMU 512 208
  - PMU 513 209
  - PMU 514 210
  - PMU 515 211
  - PMU 516 212
  - PMU 700 213
  - PMU 800 213
  - PMU 801 215
  - PMU 802 216
  - PMU 803 216
  - PMU 804 217
  - PMU 805 218
  - PMU 806 219
  - PMU 807 220
  - PMU 808 221
  - PMU 809 221

- PMU 810 222
  - PMU 811 223
  - setting sampling rates 34
  - setting timeouts 34
  - understanding 11, 152
  - understanding completed PMU status 100
  - understanding definitions 155
  - Portal Action Durations chart 131
  - portals
    - monitoring action durations 131
    - monitoring portal content references 134
    - monitoring response sizes 135
    - monitoring statistics 130
    - monitoring top portal content requests 133, 134
    - servlets *See Also* portal servlets
  - portal servlets
    - monitoring durations for components 129
    - monitoring durations for user requests 121
    - monitoring performance 130
    - monitoring portal content references 134
    - understanding 9
  - Portal Statistics page 130
  - PPMConsole
    - enabling 28
    - securing 28
  - PPMI
    - clients *See Also* PSPPMSRV processes
    - creating user IDs 18
    - encrypting user IDs and passwords 27, 146
    - pinging URLs 27
    - servlets *See Also* PPMI servlets
    - setting up client authentication 143
    - specifying passwords 27
    - specifying URLs 18, 27
    - specifying user IDs 27
    - understanding 12
    - user IDs 18
  - PPMI servlets
    - accessing 27
    - enabling tracing 44
    - registering PSPPMSRV processes 18
    - understanding 9
    - using PMU 700 213
  - process instrumentation agents 6
  - Process Scheduler
    - disabling agents 42
    - monitoring current/assigned jobs 62
    - server *See Also* Process Scheduler server
  - Process Scheduler server domains
    - identifying 75
    - monitoring 65
  - Process Scheduler servers
    - domains
      - See Also* Process Scheduler server domains
      - enabling tracing 43
      - identifying for Master Scheduler 80
      - monitored systems 7
      - monitoring host resource status 76
      - monitoring performance 75
      - monitoring PQ event rows 77
      - monitoring PSR event rows 77
      - monitoring server processes 78
      - monitoring systems 10
      - setting up for monitored systems 22
      - setting up for monitoring systems 19
      - using Event 200 87
      - using Event 300 89
    - using Event 301 92
    - using Event 302 92
  - production monitoring systems
    - setting up web servers 18
    - understanding 11
    - using the PTSYS database 16
  - PSAPPSRV processes 9, 16
  - PSMONITORSRV processes 7, 8
  - PSMSTPRC processes 8
  - PSPM\_ARCHIVE program 33, 39, 93
  - PSPM\_ARCHREC program 40
  - PSPM\_LOOKUP program 40
  - PSPM\_REAPER program *See* reaper program
  - PSPPMSRV processes
    - archiving 40
    - communicating with monitor servlets 145
    - configuring client authentication 147
    - monitoring 46
    - registering 18
    - registering servlets 17, 27
    - setting table row limits 28
    - setting up authentication 18, 143
    - setting up SSL 146
    - understanding 9, 16
    - using certificates 148
    - using PMUs 39
  - PTSYS database 8, 16
  - purging data 48
- ## Q
- queries
    - monitoring PMUs 121
    - Performance Monitor tables (sample queries) 41
  - queues
    - identifying Tuxedo queues for application server domains 73
    - identifying Tuxedo queues for Process Scheduler servers 77
    - monitoring length for application server domains 73
    - monitoring length for Process Scheduler servers 77
- ## R
- reaper program
    - Reaper Timeout status 100
    - scheduling 39
    - understanding PMU timeouts 34
  - Recycled Processes Diagnosis page 86
  - refreshes, page
    - monitoring for application server domains 72
    - monitoring for Master Scheduler 79
    - monitoring for Process Scheduler servers 76
    - monitoring for web servers 68
    - System Performance pages information 59
  - registration
    - agents 36
    - PSPPMSRV 18
    - servlets 17, 27
  - resource usage 86

# S

- safe mode 21
- sampling
  - enabling for PMUs 156
  - PMUs with enabled sampling 157
  - setting for agent events 34, 87
  - setting for agent PMUs 34
  - setting for PMUs 44
  - understanding stale data 61
- scalability 28
- scripts
  - exporting 48
  - importing 48
  - purging 48
- scripts, monitoring 121
- searches
  - component statistics 127
  - event history 101
  - PIA statistics 132
  - PMUs 94
  - portal statistics 130
  - top components 140
  - top PeopleCode events 135
  - top PeopleCode executions 137
  - top PeopleCode SQL statements 139
  - top portal content requests 133
  - user history 105
  - user requests 115
- Secure Sockets Layer (SSL) *See* SSL
- security
  - firewalls *See Also* firewalls
  - PPMConsole 28
  - setting up PPMI client authentication 143
  - setting up PPMI user IDs 18
  - SSL *See Also* SSL
  - understanding 143
- self-monitoring systems
  - configuring 15
  - pinging 66
  - setting up monitoring databases 16
  - understanding 10
- server processes
  - default 16
  - identifying for agents 36
  - identifying for application server domains 73, 74
  - identifying for Process Scheduler servers 77, 78
  - identifying Tuxedo queues 73, 77
  - list of agents 7
  - monitoring for Master Scheduler server 80
  - monitoring queued/blocked processes for Master Scheduler 80
  - monitoring requests/status for application server domains 73
  - monitoring requests/status for Process Scheduler servers 77
  - monitoring resource consumption for application server domains 74
  - monitoring resource consumption for Process Scheduler servers 78
  - printing 73, 77
  - PSAPPSRV 9, 16
  - PSMONITORSRV 7, 8
  - PSPPMSRV 9
  - setting up application servers 16
- servers
  - application *See Also* application servers
  - Master Scheduler 80
  - monitoring server round trip times 63
  - processes *See Also* server processes
  - Process Scheduler
    - See Also* Process Scheduler server
  - web *See Also* web servers
- servlets
  - configuring as clusters 28
  - gateway 10
  - monitor *See Also* monitor servlets
  - monitoring active servlets for web servers 70
  - monitoring requests 70
  - monitoring sessions for web server domains 64
  - monitoring sessions for web servers 68
  - PIA *See Also* PIA servlets
  - portal *See Also* portal servlets
  - PPMI *See Also* PPMI servlets
  - running on web servers 17
- sessions
  - identifying 106
  - monitoring 105
  - monitoring active 70
  - monitoring sessions 85
- Simple Object Access Protocol (SOAP) 12
- SOAP 12
- sockets
  - monitoring for web server domains 65
  - monitoring for web servers 69
- Solaris
  - using Event 200 87
  - using Event 300 89
- SQL
  - monitoring executions/fetches for components 128
  - monitoring SQL execution times 63
  - monitoring SQL statements/fetches in user requests 119
  - PeopleCode *See Also* PeopleCode SQL
- SSL
  - setting up 145
  - setting up client authentication 146
  - setting up for agents 145
  - setting up for PSPPMSRV 146
- stale data
  - monitoring for application server domains 71
  - monitoring for Master Scheduler 79
  - monitoring for Process Scheduler servers 75
  - monitoring for web servers 67
  - understanding 61
- standard deviation
  - Duration Averages chart 137, 142
  - monitoring for component buffers 128
  - monitoring for component durations 129
  - monitoring for components 128
  - monitoring for portal action durations 131
  - understanding 60
- status
  - host resource *See Also* host resource status
  - JVM *See Also* JVM status
  - Master Scheduler processes 79
  - Master Scheduler queued processes 80
  - Master Scheduler status (Event 350) 231
  - monitoring for PMUs 97
  - network *See Also* network status
  - querying for queue status 92

- querying for server status 92
- stale data *See Also* stale data
- understanding for completed PMUs 100
- web servlet status (Event 153) 227
- web site status (Event 152) 226
- string metrics 153
- System Defaults page 29
- System Definitions page 31
- System Performance page
  - monitoring application server domains 65
  - monitoring average durations 63
  - monitoring performance Indices 62
  - monitoring Process Scheduler server domains 65
  - monitoring web server domain information 64
  - pinging 66
  - using 61

## T

- tables
  - cloned history tables 40
  - maintaining performance data tables 39
  - moving timed-out PMUs 34
  - populating look-up tables 40
  - setting row limits for monitor tables 28
  - understanding Performance Monitor tables 41
  - "clone table" warning 94, 103
- testing 66
- third-party systems, sharing performance data 12
- timeouts
  - setting for PMUs 34
  - understanding PMU timeout statuses 100
- Today's Averages page 63
- Top Average Response Sizes chart 135
- Top Components chart 142
- Top Components page 140
- Top Cumulative Durations chart 134, 136, 138, 140, 142
- Top Cumulative Response Sizes chart 135
- Top PeopleCode Events chart 136
- Top PeopleCode Events page 135
- Top PeopleCode Executions chart 138
- Top PeopleCode Executions page 137
- Top Portal Content Requests chart 134
- Top Portal Content Requests page 133
- Top SQL Statements chart 140
- Top SQL Statements page 139
- tracing
  - enabling 56
  - enabling for monitored systems 33
  - enabling for servlets 44
  - enabling on application servers 43
  - enabling on Process Scheduler servers 43
  - enabling on web servers 43
  - monitoring trace information 58
  - Performance Trace Console
    - See Also* Performance Trace Console
  - running 56
  - understanding 55
- trees, PMU 84, 98
- Tuxedo
  - identifying domains for application servers 71
  - identifying queues for application server domains 73
  - identifying queues for Process Scheduler servers 77
  - monitoring connections/requests for application server domains 66, 72
  - monitoring connections/requests for Process Scheduler server domains 66
  - monitoring connections/requests for Process Scheduler servers 76
  - monitoring PMUs 121
  - monitoring service durations for components 129
  - monitoring service durations for user requests 122
  - using Event 301 92
  - using Event 302 92

## U

- uniform resource locators (URLs) *See* URLs
- URLs
  - pinging 66
  - specifying for integration gateway 19
  - specifying for PPMI 18
  - specifying for PPMI servlets 27
  - specifying monitor URLs 20
- Usage Monitor
  - displaying data 253
  - enabling 251
  - overview 251
  - running tests 252
  - verifying 252
- User History page 105
- User Request chart
  - understanding 115
  - using 117
- User Request Detail page 119
- User Requests page 115
- users
  - identifying for PMU requests 83
  - identifying for PMUs 97
  - monitoring component PMUs 120
  - monitoring for PMU trees 85
  - monitoring for web server domains 65
  - monitoring for web servers 69
  - monitoring history 105, 106
  - monitoring query/script PMUs 121
  - monitoring request analytics
    - See Also* User Request chart
  - monitoring request durations 121
  - monitoring requests 119
  - monitoring sessions 85
  - monitoring signed on 62
  - monitoring Tuxedo PMUs 121
  - PeopleCode Durations chart 122
  - Request Time and Think Time charts 118
  - searching for history 105

## V

- virtual memory
  - monitoring for application server domains 72, 74
  - monitoring for Process Scheduler server domains 66
  - monitoring for Process Scheduler servers

76, 78

## W

### warnings

- clone table 94, 103
- monitoring 62
- reporting to monitoring system 38
- stale data *See Also* stale data

### web.xml 149

### web applications

- monitoring execute threads for web server domains 65
- monitoring execute threads for web servers 68
- monitoring servlet sessions for web server domains 64
- monitoring servlet sessions for web servers 68

### web server domains

- identifying 67
- monitoring 64

### Web Server page 67

### web servers

- disabling agents 42
- domains *See Also* web server domains
- enabling tracing 43
- monitored system elements 7
- monitoring active servlets 70
- monitoring JVM status 68
- monitoring network status 69
- monitoring performance 67
- monitoring sites 70
- monitoring systems 9
- recycling 105
- setting up for monitored systems 21
- setting up for monitoring systems 17
- starting in safe mode 21
- using Event 150 86
- using Event 151 86

### Windows

- using Event 200 87
- using Event 300 89

