

## **Progress Fathom Management Standard Edition**

### **OpenEdge Server Management Guide**

Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. This manual is also copyrighted and all rights are reserved. This manual may not, in whole or in part, be copied, photocopied, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Progress Software Corporation.

The information in this manual is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear in this document. The references in this manual to specific platforms supported are subject to change.

Allegrix, A [Stylized], ObjectStore, Progress, Powered by Progress, Progress Fast Track, Progress Profiles, Partners in Progress, Partners en Progress, Progress en Partners, Progress in Progress, P.I.P., Progress Results, ProVision, ProCare, ProtoSpeed, SmartBeans, SpeedScript, and WebSpeed are registered trademarks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and/or other countries. AccelEvent, A Data Center of Your Very Own, Allegrix & Design, AppsAlive, AppServer, ASPen, ASP-in-a-Box, BusinessEdge, Business Empowerment, Empowerment Center, eXcelon, Fathom, Future Proof, IntelliStream, ObjectCache, OpenEdge, PeerDirect, POSSE, POSSENET, Progress Business Empowerment, Progress Dynamics, Progress Empowerment Center, Progress Empowerment Program, Progress for Partners, Progress OpenEdge, Progress Software Developers Network, PSE Pro, PS Select, SectorAlliance, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Technical Empowerment, WebClient, and Who Makes Progress are trademarks or service marks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and other countries.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Any other trademarks or service marks contained herein are the property of their respective owners.

Fathom Management includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Copyright © 1999 The Apache Software Foundation. All rights reserved (Xalan XSLT Processor) and Copyright © 2000-2002 The Apache Software Foundation. All rights reserved (Jakarta-Oro). The names “Apache,” “Xerces,” “Jakarta-Oro,” and “Apache Software Foundation” must not be used to endorse or promote products derived from this software without prior written permission. Products derived from this software may not be called “Apache” or “Jakarta-Oro,” nor may “Apache” or “Jakarta-Oro” appear in their name, without prior written permission of the Apache Software Foundation. For written permission, please contact [apache@apache.org](mailto:apache@apache.org). Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

Fathom Management includes software developed by ACME Labs. Copyright © 2000 by Jef Poskanzer <[jef@acme.com](mailto:jef@acme.com)>. All rights reserved. Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

Fathom Management includes software developed by Sun Microsystems, Inc. Copyright © 2003 Sun Microsystems, Inc. All Rights Reserved. Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

Fathom Management includes the Jetty Package Copyright © 1998 Mort Bay Consulting Pty. Ltd. (Australia).

Fathom Management includes software developed by the ModelObjectsGroup (<http://www.modelobjects.com>). Copyright © 2000-2001 ModelObjects Group. All rights reserved. The name “ModelObjects” must not be used to endorse or promote products derived from the SSC Software without prior written permission. Products derived from the SSC Software may not be called “ModelObjects”, nor may “ModelObjects” appear in their name, without prior written permission. For written permission, please contact [djacobs@modelobjects.com](mailto:djacobs@modelobjects.com).

Fathom Management includes files that are subject to the Netscape Public License Version 1.1 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at (<http://www.mozilla.org/NPL>). Software distributed under the License is distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License. The Original Code is Mozilla Communicator client code, released March 31, 1998. The Initial Developer of the Original Code is Netscape Communications Corporation. Portions created by Netscape are Copyright © 1998-1999 Netscape Communications Corporation. All Rights Reserved.

Fathom Management contains copyright material licensed fromAdventNet, Inc. <http://www.adventnet.com>. All rights to such copyright material rest with AdventNet.

Fathom Management includes the RSA Data Security, Inc. MD5 Message-Digest Algorithm. Copyright © 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

August 2004



Product Code: 4496

Release: V3.0A

---

# Contents

---

<b>Preface</b> .....	<b>Preface–1</b>
<b>1. Fathom Management Support for OpenEdge Servers</b> .....	<b>1–1</b>
OpenEdge servers overview .....	1–2
AppServer .....	1–2
WebSpeed Transaction Server .....	1–2
NameServer .....	1–3
Managing broker resources .....	1–3
Server and agent details .....	1–4
Managing NameServer resources .....	1–4
Fathom features to support OpenEdge server resources .....	1–5
OpenEdge server resource-related prerequisites .....	1–6
Installation .....	1–6
Discovering and enabling local resources .....	1–6
Discovering and enabling remote resources .....	1–7
Role authorization and OpenEdge tasks .....	1–7
<b>2. Getting Started</b> .....	<b>2–1</b>
Fathom Management console and OpenEdge information .....	2–2
Fathom menu bars .....	2–2
Using the main menu bar for OpenEdge server tasks .....	2–5
Introducing the OpenEdge main page .....	2–8
Main page format and content .....	2–8
Accessing OpenEdge resource information .....	2–12
Accessing OpenEdge resources from the list frame .....	2–12
Accessing an OpenEdge main page .....	2–14
Deleting OpenEdge resources .....	2–15
Effects of an AdminServer warm start on Fathom .....	2–18
Stages of a warm start .....	2–18

Warm start implications . . . . .	2-19
Initiating an AdminServer warm start . . . . .	2-20
AdminServer warm start process . . . . .	2-20
Understanding OpenEdge server graphs . . . . .	2-21
Graphs available on Performance View pages . . . . .	2-21
Displaying OpenEdge viewlets on a collection view. . . . .	2-22
Changing OpenEdge pinup graphical displays . . . . .	2-25
<b>3. Managing WebSpeed Transaction Server Data . . . . .</b>	<b>3-1</b>
Overview . . . . .	3-2
Reviewing WebSpeed broker status . . . . .	3-2
Modifying WebSpeed control settings . . . . .	3-4
Broker Control page content . . . . .	3-5
Changing WebSpeed Broker controls . . . . .	3-8
Viewing broker process details . . . . .	3-11
Agent Pool Control page content . . . . .	3-13
Agent pool data initial configuration section . . . . .	3-15
Adding or trimming agents . . . . .	3-18
Killing a WebSpeed agent process . . . . .	3-19
Accessing and reviewing WebSpeed-related log file data . . . . .	3-20
Getting started with log files for WebSpeed resources . . . . .	3-21
Characteristics of WebSpeed resource log file monitors . . . . .	3-22
WebSpeed log file monitor default values . . . . .	3-23
Reviewing predefined log file monitor search criteria . . . . .	3-24
Customizing a WebSpeed broker log file monitor . . . . .	3-25
Using the WebSpeed log file viewers . . . . .	3-28
Refreshing log file data. . . . .	3-30
Examining WebSpeed-related Operational views . . . . .	3-31
Accessing and reviewing the Broker Performance View . . . . .	3-31
Accessing and reviewing the Agent Performance View . . . . .	3-35
Examining WebSpeed-related Informational views . . . . .	3-37
<b>4. Managing AppServer Data . . . . .</b>	<b>4-1</b>
Overview . . . . .	4-2
Reviewing AppServer broker status . . . . .	4-2
Modifying AppServer control settings . . . . .	4-4
Broker Control page content . . . . .	4-5
Changing AppServer Broker controls . . . . .	4-7
Viewing broker process details . . . . .	4-10
Server Pool Control page content . . . . .	4-13
Adding or trimming AppServers . . . . .	4-17
Killing an AppServer process . . . . .	4-18
Accessing and reviewing AppServer-related log file data . . . . .	4-20
Getting started with log files for AppServer resources . . . . .	4-21

Characteristics of AppServer resource log file monitors .....	4-21
AppServer log file monitor default values .....	4-22
Reviewing predefined log file monitor search criteria .....	4-24
Customizing an AppServer Broker log file monitor .....	4-25
Using the AppServer log file viewers .....	4-28
Refreshing log file data .....	4-30
Examining AppServer-related Operational views .....	4-31
Accessing and reviewing the Broker Performance View .....	4-32
Accessing and reviewing the Server Performance View .....	4-35
Examining AppServer-related Informational views .....	4-37
 <b>5. Managing NameServer Data .....</b>	<b>5-1</b>
Overview .....	5-2
Reviewing NameServer status .....	5-2
Modifying NameServer control settings .....	5-3
NameServer Control page content .....	5-4
Changing NameServer controls .....	5-5
Accessing and reviewing NameServer-related log file data .....	5-6
Getting started with NameServer log files .....	5-7
Characteristics of a NameServer log file monitor .....	5-7
NameServer log file monitor default values .....	5-7
Reviewing predefined log file monitor search criteria .....	5-8
Customizing a NameServer log file monitor .....	5-9
Using the NameServer log file viewer .....	5-13
Refreshing log file data .....	5-15
Examining NameServer Operational and Informational views .....	5-16
Accessing and reviewing Operational views .....	5-16
Accessing and reviewing Informational views .....	5-21
 <b>6. Monitoring Plans and Rules for OpenEdge Server Resources .....</b>	<b>6-1</b>
Fathom resource monitoring overview .....	6-2
Key terms and definitions .....	6-3
Default polling and trend values .....	6-5
Default monitoring plan details .....	6-8
Monitoring plan default values .....	6-8
OpenEdge default monitoring plan examples .....	6-11
Maintaining monitoring plans .....	6-13
Updating monitoring plans .....	6-13
General rule conventions .....	6-19
Understanding and using resource monitor rules .....	6-20
Common rule characteristics .....	6-20
NameServer rules .....	6-21
AppServer rules .....	6-22
WebSpeed rules .....	6-23

Average Procedure Duration High rule .....	6-24
Rejected Request Percent High rule .....	6-25
Queued Request Percent High rule .....	6-26
Agent (Server) Unavailable rule .....	6-27
Working with rule sets .....	6-29
Characteristics of and benefits associated with rule sets .....	6-30
Editing a rule set .....	6-32
Copying a rule set .....	6-32
Deleting a rule set .....	6-33
Adding rule sets that have one or more rules in common .....	6-33
Associating a rule set with a monitoring plan .....	6-33
<b>7. Calculating Rule Threshold Settings Using the Configuration Advisor .....</b>	<b>7-1</b>
Overview .....	7-2
Rule details .....	7-2
Data analysis and recommended values overview .....	7-4
Generating and applying threshold rule settings .....	7-8
Setting rules-related criteria .....	7-9
Understanding the recommended threshold settings .....	7-12
Evaluating recommended settings .....	7-13
Comparing and selecting threshold settings .....	7-16
Submitting your threshold setting selections .....	7-17
Determining the effectiveness of your selections .....	7-17
<b>8. Analyzing OpenEdge Application Performance .....</b>	<b>8-1</b>
Overview .....	8-2
Investigating application performance issues .....	8-3
Fathom in the workplace .....	8-3
Implementing business requirements with Fathom Management at XYZ Corporation .....	8-4
Consulting Fathom Management documentation .....	8-5
Planning an application performance review .....	8-5
Problem definition .....	8-6
Initial investigation .....	8-6
Drilling deeper into Fathom-supplied data .....	8-7
Testing and documenting your potential solutions .....	8-12
Responding to an application crisis .....	8-12
Problem definition .....	8-13
Initial investigation .....	8-13
Drilling deeper into Fathom-supplied data .....	8-14
Testing and documenting your potential solutions .....	8-20
For more Fathom assessment information .....	8-21
<b>Index .....</b>	<b>Index-1</b>

## Figures

Figure 2-1:	Main menu bar . . . . .	2-2
Figure 2-2:	Collection views — detail menu bar . . . . .	2-3
Figure 2-3:	Resource category view — detail menu bar . . . . .	2-3
Figure 2-4:	Library→New selections from the detail menu bar . . . . .	2-4
Figure 2-5:	WebSpeed main page sample . . . . .	2-8
Figure 2-6:	Broker statistics not available information . . . . .	2-11
Figure 2-7:	Binoculars icon . . . . .	2-22
Figure 3-1:	WebSpeed Status section sample . . . . .	3-2
Figure 3-2:	Command and control section example . . . . .	3-4
Figure 3-3:	Broker Control page example . . . . .	3-6
Figure 3-4:	Agent Pool Control page example . . . . .	3-14
Figure 3-5:	WebSpeed-related search criteria . . . . .	3-22
Figure 3-6:	WebSpeed broker log file viewer example . . . . .	3-29
Figure 3-7:	WebSpeed Operational views section example . . . . .	3-31
Figure 3-8:	WebSpeed Informational views section example . . . . .	3-37
Figure 4-1:	AppServer Status section sample . . . . .	4-2
Figure 4-2:	Command and control section example . . . . .	4-4
Figure 4-3:	Broker Control page example . . . . .	4-5
Figure 4-4:	Server Pool Control page example . . . . .	4-13
Figure 4-5:	AppServer-related search criteria . . . . .	4-22
Figure 4-6:	AppServer Broker log file viewer example . . . . .	4-29
Figure 4-7:	Operational views section example . . . . .	4-31
Figure 4-8:	Informational views section example . . . . .	4-37
Figure 5-1:	NameServer Status section sample . . . . .	5-2
Figure 5-2:	Command and control section example . . . . .	5-3
Figure 5-3:	NameServer Control page example . . . . .	5-4
Figure 5-4:	NameServer log file viewer example . . . . .	5-14
Figure 6-1:	NameServer instance default monitoring example . . . . .	6-11
Figure 6-2:	AppServer broker default monitoring plan example . . . . .	6-12
Figure 6-3:	WebSpeed broker default monitoring plan example . . . . .	6-12
Figure 6-4:	Available NameServer Rules page . . . . .	6-21
Figure 6-5:	Available AppServer Rules page . . . . .	6-22
Figure 6-6:	Available Webspeed Rules page . . . . .	6-23
Figure 6-7:	Accessing rule sets from the detail frame . . . . .	6-32
Figure 7-1:	Configuration Advisor page example . . . . .	7-4
Figure 7-2:	Configuration Advisor recommend thresholds page . . . . .	7-7
Figure 7-3:	Configuration Advisor recommended values . . . . .	7-12
Figure 7-4:	Recommended Values field content . . . . .	7-14
Figure 7-5:	Detail page analysis content . . . . .	7-15
Figure 8-1:	AppServer Profile Report — Average_Afternoon data . . . . .	8-8
Figure 8-2:	AppServer Profile Report — Bad_Afternoon data . . . . .	8-10
Figure 8-3:	AppServer Profile Report — Crisis_Report data . . . . .	8-11
Figure 8-4:	Broker Performance View for asbroker1 on host pcskye . . . . .	8-16

Figure 8–5: AS Broker Activity Status for asbroker1 ..... 8–17

Figure 8–6: AppServer asbroker1 log file ..... 8–18

Figure 8–7: Servers Performance View page for asbroker1 ..... 8–19

Figure 8–8: Total Servers CPU for asbroker1 ..... 8–20



## Tables

Table 2–1:	Performing OpenEdge server-related activities . . . . .	2–6
Table 2–2:	Sections of the OpenEdge main page . . . . .	2–10
Table 2–3:	Performance pages and their graphical content . . . . .	2–21
Table 2–4:	Graph properties and options for time-based graphs . . . . .	2–26
Table 3–1:	WebSpeed status details . . . . .	3–3
Table 3–2:	Additional WebSpeed information . . . . .	3–5
Table 3–3:	Process statistics section real-time operational data . . . . .	3–13
Table 3–4:	Agent pool data initial configuration field definitions . . . . .	3–15
Table 3–5:	Agents state field definitions . . . . .	3–16
Table 3–6:	Agent pool summary field definitions . . . . .	3–17
Table 3–7:	Broker connection workload details . . . . .	3–33
Table 3–8:	WebSpeed Broker performance-related graphs . . . . .	3–34
Table 3–9:	WebSpeed agents performance-related graphs . . . . .	3–36
Table 4–1:	AppServer status details . . . . .	4–3
Table 4–2:	Additional AppServer information . . . . .	4–5
Table 4–3:	Process statistics section real-time operational data . . . . .	4–12
Table 4–4:	Server pool data initial configuration field definitions . . . . .	4–14
Table 4–5:	Servers state field definitions . . . . .	4–15
Table 4–6:	Server pool summary field definitions . . . . .	4–16
Table 4–7:	AppServer broker connection workload details . . . . .	4–33
Table 4–8:	AppServer broker performance-related graphs . . . . .	4–34
Table 4–9:	AppServers performance-related graphs . . . . .	4–36
Table 5–1:	NameServer Status details . . . . .	5–3
Table 5–2:	Additional NameServer information . . . . .	5–4
Table 5–3:	Summary details on the Operational Status page . . . . .	5–18
Table 5–4:	NameServer details on the Operational Status page . . . . .	5–20
Table 5–5:	Properties details on the Static Configuration page . . . . .	5–22
Table 6–1:	Monitoring plan default values . . . . .	6–8
Table 6–2:	Resource status legend . . . . .	6–19
Table 6–3:	Examples of AppServer-related 4GL procedure entries . . . . .	6–25
Table 7–1:	Configuration Advisor details . . . . .	7–8
Table 7–2:	Tasks using the Configuration Advisor Calculations page . . . . .	7–12
Table 7–3:	Detail page fields and descriptions . . . . .	7–15
Table 8–1:	Initial Investigative Checklist . . . . .	8–6
Table 8–2:	Crisis review checklist . . . . .	8–13
Table 8–3:	References for more in-depth Fathom performance-related information . . . . .	8–22



---

# Preface

---

This Preface contains the following sections:

- [Purpose](#)
- [Audience](#)
- [Fathom Management and Progress or OpenEdge](#)
- [Organization](#)
- [Typographical conventions](#)

## Purpose

This guide details the concepts, features, and procedures that Progress ® Fathom™ Management Standard Edition supports to monitor and manage specific resources associated with these OpenEdge™ server products: AppServer™, WebSpeed® Transaction Server, and OpenEdge™ NameServer.

---

**Note:** The OpenEdge database is documented in its own manual. See the [Database Management Guide](#) for all details about this product.

---

## Audience

This manual is designed for system administrators, database administrators and any other personnel responsible for the administrative and daily activities associated with managing an OpenEdge-based application environment that uses Fathom Management.

## Fathom Management and Progress or OpenEdge

This version of Fathom Management runs against the following:

- OpenEdge 10.0B.
- Progress Version 9.1D and the 9.1D09 service pack.

For the sake of simplicity, the procedures and screen shots provided in this manual refer to the running of Fathom against OpenEdge Version 10.0B. However, unless indicated otherwise, the procedures are the same for both OpenEdge 10.0B and Progress Version 9.1D with the 9.1D09 service pack. For example, if a procedure refers to an OpenEdge database, the procedure applies to a Progress database as well.

# Organization

## Chapter 1, “Fathom Management Support for OpenEdge Servers”

Presents an overview of the features and functionality available in Fathom Management to support these OpenEdge server products: WebSpeed Transaction Server, AppServer, and NameServer.

## Chapter 2, “Getting Started”

Describes how to navigate the Fathom Management Console to access OpenEdge resource-related details.

## Chapter 3, “Managing WebSpeed Transaction Server Data”

Explains how to use the Fathom Management features and functionality related to WebSpeed Transaction Servers.

## Chapter 4, “Managing AppServer Data”

Explains how to use the Fathom Management features and functionality related to AppServers.

## Chapter 5, “Managing NameServer Data”

Explains how to use the Fathom Management features and functionality related to NameServers.

## Chapter 6, “Monitoring Plans and Rules for OpenEdge Server Resources”

Describes the settings and tasks involved in maintaining the OpenEdge server-specific resource monitors and performing related activities.

## Chapter 7, “Calculating Rule Threshold Settings Using the Configuration Advisor”



Describes how to use the Configuration Advisor feature to generate recommended threshold rule settings for specific WebSpeed and AppServer rules.

## Chapter 8, “Analyzing OpenEdge Application Performance”

Describes how you can use Fathom features to analyze OpenEdge server application performance.

## Typographical conventions

This manual uses the following typographical conventions:

Convention	Description
<b>Bold</b>	Bold typeface indicates commands or characters the user types, or the names of user interface elements.
<i>Italic</i>	Italic typeface indicates the title of a document, provides emphasis, or signifies new terms.
<b>SMALL, BOLD CAPITAL LETTERS</b>	Small, bold capital letters indicate OpenEdge™ key functions and generic keyboard keys; for example, <b>GET</b> and <b>CTRL</b> .
<b>KEY1-KEY2</b>	A hyphen between key names indicates a <i>simultaneous</i> key sequence: you press and hold down the first key while pressing the second key. For example, <b>CTRL-X</b> .
<b>KEY1 KEY2</b>	A space between key names indicates a <i>sequential</i> key sequence: you press and release the first key, then press another key. For example, <b>ESCAPE H</b> .
<b>Syntax:</b>	
Fixed width	A fixed-width font is used in syntax statements, code examples, and for system output and filenames.
<i>Fixed-width italics</i>	Fixed-width italics indicate variables in syntax statements.
<b><i>Fixed-width bold</i></b>	Fixed-width bold indicates variables with special emphasis.
UPPERCASE fixed width	Uppercase words are Progress® 4GL language keywords. Although these always are shown in uppercase, you can type them in either uppercase or lowercase in a procedure.
	This icon (three arrows) introduces a multi-step procedure.
	This icon (one arrow) introduces a single-step procedure.

---

# Fathom Management Support for OpenEdge Servers

---

This chapter provides an overview of Progress® Fathom™ Management support for these OpenEdge™ server products: AppServer™, WebSpeed® Transaction Server, and OpenEdge™ NameServer.

Topics in this chapter include:

- [OpenEdge servers overview](#)
- [OpenEdge server resource-related prerequisites](#)
- [Fathom features to support OpenEdge server resources](#)

---

**Note:** Throughout this guide, references to OpenEdge servers are commonly interchanged with these references: OpenEdge, OpenEdge server-related resources, and OpenEdge resources.

---

## OpenEdge servers overview

This section identifies and briefly describes each of the OpenEdge servers for which you can set up and use the Fathom Management application monitoring capabilities and related features. These servers include:

- [AppServer](#).
- [WebSpeed Transaction Server](#).
- [NameServer](#).

---

**Note:** The OpenEdge database is documented in its own manual. See the [Database Management Guide](#) for all details about this product.

---

### AppServer

The AppServer is an OpenEdge™ application that allows you to build and deploy complex distributed applications using the Progress 4GL. Each AppServer consists of an Application broker (also known as an AppServer™ Broker, or broker) and one or more Application servers. AppServers work with the AdminServer and an optional, integrated OpenEdge NameServer. Fathom Management supports discovering and monitoring AppServer brokers and managing activities associated with their respective servers from the Fathom Management console.

### WebSpeed Transaction Server

The WebSpeed product includes WebSpeed® Messengers, WebSpeed® Brokers, and WebSpeed® Agents. WebSpeed Transaction Servers work with the AdminServer and NameServer. The WebSpeed brokers launch WebSpeed Agents to drive your Web applications.

Fathom supports discovering and monitoring WebSpeed brokers and managing activities associated with their respective agents from the Fathom Management console.

---

**Note:** Fathom Management supports monitoring and managing the WebSpeed Transaction Server product. Throughout this guide, the WebSpeed Transaction Server is commonly referred to either the Transaction Server or WebSpeed.

---



## NameServer

The NameServer is an administrative component that can be integrated with the Transaction Server and AppServer. The NameServer works with a pool of brokers to identify and distribute client requests to register specific application services. For example, an AppServer broker can register Application Services with a NameServer; a WebSpeed broker can register WebSpeed Services that it provides with a NameServer. Also, a NameServer can connect a client request for a WebSpeed Service that is registered with the NameServer with an available WebSpeed broker. The NameServer can also provide location transparency.

Fathom supports discovering and monitoring NameServers. You can also manage activities associated with NameServers from the Fathom Management console.

---

**Note:** The NameServer can also be configured to work with other OpenEdge products such as OpenEdge Dataservers and SonicMQ® Adapter. However, these features are beyond the scope of this document.

---

## Managing broker resources

On systems that support shared processes, a *broker* is a main server process. It functions like a traffic director, handling client requests for specific resources that support the business logic associated with an application. A broker identifies and accounts for resource availability and consumption; it accomplishes these tasks by processing a pool of servers or agents and attempting to fulfill specific resource requests.

For example, an AppServer broker manages connection requests initiated by its clients for the business logic and processes located on an AppServer. In this context, a broker executes its tasks somewhat in isolation, only executing and performing according to its defined configuration properties and parameters.

Using Fathom you can optimize broker performance; the Fathom Management console supports viewing status details, controlling, monitoring, and managing broker components to ensure appropriate resources are available.

---

**Note:** The term Fathom as used throughout this guide refers to Fathom Management Standard Edition.

---

### **The ubroker.properties file**

The `ubroker.properties` file stores all the configuration definitions for each instance of the Transaction Server, the AppServer, and the NameServer. Each configuration definition contains environment variables, registry entries (for Windows), and property settings for each product instance. Fathom references and displays this configuration data as it relates to the performance data you can gather and analyze in Fathom.

While you can use Fathom Management to perform limited configuration tasks, you should continue to use Progress Explorer or the command line to customize configuration details stored in the `ubroker.properties` file.

### **Server and agent details**

In association with each broker, Fathom displays server and agent data. This information provides you additional performance data to better manage your connection workload. As necessary, you can add or trim servers or agents to maximize the use of your existing resources and respond to fluctuations in processing demands.

The OpenEdge server resource discovery process begins with the discovery of resources and the automatic creation of default monitoring plans for these resources. As part of this process, Fathom creates log file monitors not only for the primary local resources (AppServer broker, WebSpeed broker, or NameServer), but also for the server and agent resources associated with these resources. For example, Fathom creates an AppServer broker server log file for each local AppServer broker resource.

Log file monitors, in general, are information tools that can help you to analyze the data you can collect from within their associated files; these details can aid you in determining performance expectations and examine trends.

### **Managing NameServer resources**

Similar to Transaction Servers and AppServers, Fathom allows you to view status details for NameServers. This data helps you to effectively monitor and manage the services that NameServers provide.

## Fathom features to support OpenEdge server resources

The following Fathom features support WebSpeed, AppServer, and NameServer servers:

- Automatic discovery of each WebSpeed, AppServer, and NameServer resource that is locally defined. Specific configuration tasks are not required for these resources because Fathom recognizes the configuration data already established in the `ubroker.properties` file.
- Automatic discovery of each WebSpeed, AppServer, and NameServer resource that is remotely defined. However, remote monitoring requires some additional installation steps before this feature is available. See the [Installation and Configuration Guide](#) for details.
- OpenEdge server resources are integrated into the Fathom Management console and are accessible using Fathom features, functionality, and navigational conventions. This includes the use of the basic Fathom monitoring plan features, such as default and customizable schedules rules, alerts, and actions.
- Automatic discovery of each WebSpeed, AppServer, and NameServer resource that is locally defined. Specific configuration tasks are not required for these resources because Fathom recognizes the configuration data already established in the `ubroker.properties` file.
- Automatic discovery of each WebSpeed, AppServer, and NameServer resource that is remotely defined. However, remote monitoring requires some additional installation steps before this feature is available. See the [Installation and Configuration Guide](#) for details.
- Use of the Configuration Advisor feature for WebSpeed and AppServer broker resources. This feature helps you to determine optimum settings for threshold values used for defined rules. The Configuration Advisor suggests values by analyzing data stored in the FathomTrendDatabase.
- Fathom OpenEdge broker resources support the collection of statistical data for WebSpeed and AppServer brokers. This data can be used to generate Fathom-based reports and graphs.

## OpenEdge server resource-related prerequisites

This section highlights the criteria that must be met to enable Fathom to recognize and monitor OpenEdge server resources.

### Installation

A Fathom Management installation and configuration process must include a Transaction Server product and/or AppServer product, depending on the specific product resource monitoring capabilities you intend to use. To simply monitor OpenEdge resources, a database is not required. However, to trend and report on data generated by AppServer and WebSpeed broker resources, the FathomTrendDatabase must be installed.

For details about the Fathom Management installation and the related configuration process and procedures, see the *Installation and Configuration Guide*.

### Discovering and enabling local resources

Once the installation and configuration steps have been completed, Fathom automatically creates a Fathom resource monitor for each WebSpeed broker, AppServer broker, and NameServer instance that it detects; the specific resources depend on the actual OpenEdge server products installed. This discovery process occurs any time Fathom detects new OpenEdge resource instances. This process will initially take place after you install and configure Fathom, and any time new OpenEdge resources are introduced.

As part of this discovery process, Fathom enables each locally defined broker or NameServer instance and begins monitoring them right away. You can elect to disable any resources, implement data collection (for brokers only), and modify the default monitoring plan and rules as needed.

---

**Note:** Fathom Management runs as a managed service in the AdminServer. Therefore, a *local resource* is defined as a Fathom-recognized resource that is running in the AdminServer on the same machine where Fathom Management is installed.

---

## Discovering and enabling remote resources

If you have performed the necessary steps to monitor remote resources using Fathom Management, Fathom will also create a Fathom resource monitor for each WebSpeed broker, AppServer broker, and NameServer instance that it detects.

As in the discovery process for local resources, Fathom enables each remotely defined broker or NameServer instance and begins monitoring them right away. This discovery process occurs any time Fathom detects new OpenEdge resource instances. This process will initially take place after you install and configure Fathom, and any time new OpenEdge resources are introduced.

As needed, you can elect to disable any resources, implement data collection (for brokers only), or modify the default monitoring plan and rules.

---

**Note:** Fathom Management runs as a managed service in the AdminServer. Therefore, a *remote resource* is defined as a Fathom-recognized resource that runs in an AdminServer that is *not* running Fathom. This resource might be on a machine that is physically separate from the machine where Fathom Management is installed, or it could be a different instance of an AdminServer running on the same machine that Fathom is running on.

---

## Role authorization and OpenEdge tasks

Users with Administrator privileges can automatically perform all of the following OpenEdge tasks:

- Starting and stopping brokers.
- Adding and trimming agents.
- Adding and trimming servers.
- Stopping, or killing, processes.
- Editing and deleting WebSpeed, AppServer, and NameServer resources.
- Initiating OpenEdge rule threshold calculations using the Configuration Advisor.

Users with Operator privileges, the Fathom Administrator must grant explicit authorization to perform any of the tasks in the previous list.

For more detailed information about setting up these various OpenEdge-related options on the **Operator Role Customization** page, see the appropriate sections of the *[Installation and Configuration Guide](#)*.

---

## Getting Started

---

This chapter describes how to navigate the Fathom Management console to access OpenEdge resource-related details. Information presented in this chapter assumes that you have a working knowledge of the management console functionality described in the *Resource Monitoring Guide*.

Topics in this chapter include:

- [Fathom Management console and OpenEdge information](#)
- [Introducing the OpenEdge main page](#)
- [Accessing OpenEdge resource information](#)
- [Deleting OpenEdge resources](#)
- [Effects of an AdminServer warm start on Fathom](#)
- [Understanding OpenEdge server graphs](#)

## Fathom Management console and OpenEdge information

Menu bars available from the Fathom Management console allow you to access OpenEdge-related features and functionality.

This section highlights:

- [Fathom menu bars](#).
- [Using the main menu bar for OpenEdge server tasks](#).

### Fathom menu bars

The management console supports two menu bars: main menu and detail menu bar.

#### Main menu bar

The main menu bar always appears at the top of management console. Use the main menu bar to select from various menu categories. [Figure 2–1](#) shows the main menu bar.



**Figure 2–1: Main menu bar**

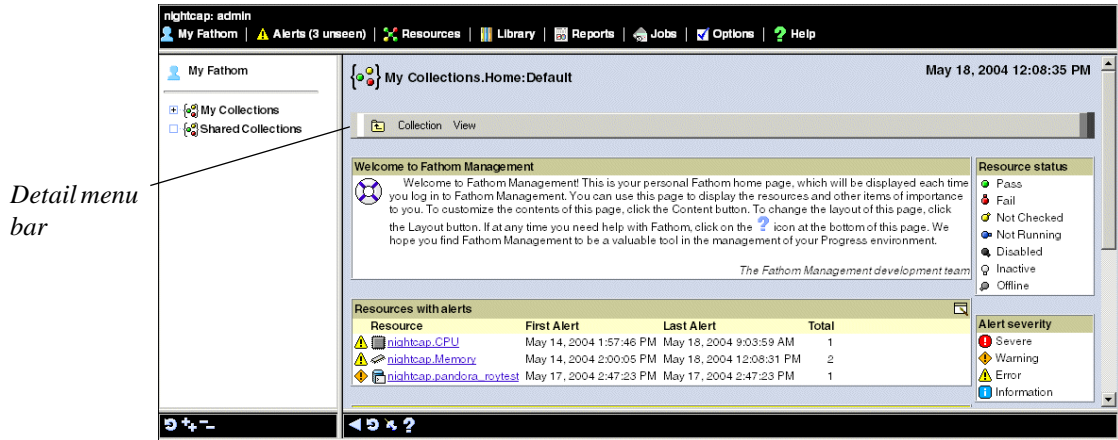
Clicking a menu category changes the content that displays in the list frame and in the detail frame to allow you to perform tasks associated with that category. Throughout this manual, all procedures describe how to perform tasks using this menu bar. See the [“Using the main menu bar for OpenEdge server tasks”](#) section on page 2–5 for more information about the OpenEdge tasks you can perform.



## Detail menu bar

The detail menu bar appears at the top of the detail frame of the Fathom Management console to support specific tasks.

For example, [Figure 2–2](#) shows the detail menu bar that appears on the **My Collections Home Default** page when you click **My Fathom** on the main menu bar.



**Figure 2–2: Collection views — detail menu bar**

The **Collection** and **View** detail menu options allow you to access specific activities associated with setting up and managing collections; these options supplement the other main menu bar options.

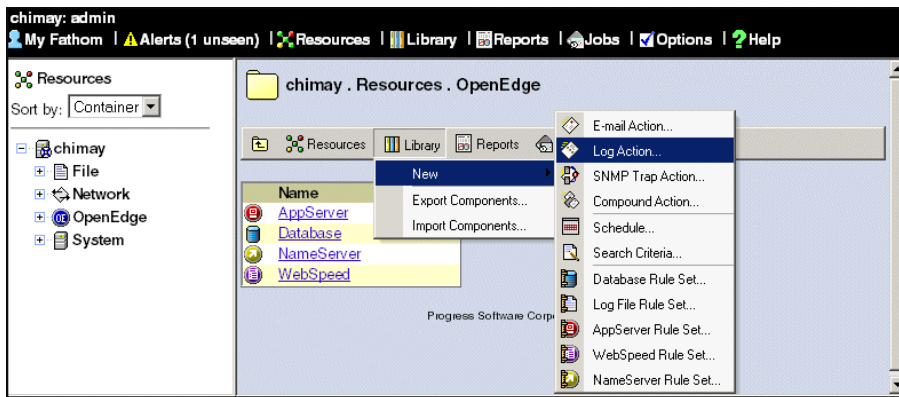
[Figure 2–3](#) shows another example of the detail menu bar. All the options on this menu bar display when you select a resource-, library-, reports-, or jobs-related category from the list frame. ([Figure 2–3](#) specifically shows the detail menu bar that appears on the detail frame when you click the **OpenEdge** resource category in the list frame.)



**Figure 2–3: Resource category view — detail menu bar**

The detail menu bar displayed in [Figure 2–3](#) supports many of the same options available from the main menu bar; due to its drop-down menu-driven design, the detail menu is an alternative way to access Fathom options more quickly.

[Figure 2–4](#) shows an example of the detail menu options that appear when you select **Library** → **New**.



**Figure 2–4:** Library→New selections from the detail menu bar

For more details about either of these menu bars and how to navigate in the Fathom Management console, see the appropriate section of the [Resource Monitoring Guide](#).

## Using the main menu bar for OpenEdge server tasks

[Table 2–1](#) describes how to use the main menu bar to perform OpenEdge server-related tasks. For a broader definition of each menu bar’s functional area and its associated activities, see the appropriate section of the *Resource Monitoring Guide*.

---

**Note:** You can also use the detail menu bar to perform many of the activities described in [Table 2–1](#). For information about the detail menu bar, see the “[Fathom menu bars](#)” section on page 2–2 and the *Resource Monitoring Guide*.

---

**Table 2–1: Performing OpenEdge server-related activities***(1 of 2)*

Select this menu bar category...	To perform these OpenEdge server-related activities...
<b>Resources</b>	<p>Display and update resource details for:</p> <ul style="list-style-type: none"><li>• WebSpeed brokers, and their associated agents and servers.</li><li>• AppServer brokers and their associated servers.</li><li>• NameServer instances and their associated Application Services.</li></ul> <p>When Fathom is installed, local and remote OpenEdge resources are automatically discovered as <b>Enabled</b> and initial status information reported. For more information about this topic, see the <a href="#">“Accessing OpenEdge resource information”</a> section on page 2–12.</p> <p>Additional, specific management and monitoring tasks can be performed using the OpenEdge main page. For more introductory information about the OpenEdge main page, see the <a href="#">“Introducing the OpenEdge main page”</a> section on page 2–8.</p>
<b>Library</b>	<p>Access and perform all library-based functions available for OpenEdge resources.</p> <p>See the <a href="#">“Working with rule sets”</a> section on page 6–29 for library options such as creating and deleting rule sets for specific OpenEdge resources.</p>

**Table 2–1: Performing OpenEdge server-related activities** (2 of 2)

Select this menu bar category...	To perform these OpenEdge server-related activities...
<b>Reports</b>	<p>Access and generate these OpenEdge server-related reports:</p> <ul style="list-style-type: none"><li>• AppServer Performance.</li><li>• AppServer Application Profiler.</li><li>• WebSpeed Performance.</li><li>• WebSpeed Application Profiler.</li></ul> <p>For details about each of these reports, see the appropriate sections of the <i>Reporting Guide</i>.</p>
<b>Options</b>	<p>Review and update authorization features related to OpenEdge resources.</p> <p>For a summary of OpenEdge authorization options, see the “<a href="#">Role authorization and OpenEdge tasks</a>” section on page 1–7 and the <i>Installation and Configuration Guide</i>.</p>

---

**Note:** This Fathom Management release does not support any OpenEdge server-specific job features. For details about the **Job** category and jobs, see the appropriate sections of the *Resource Monitoring Guide*.

---

## Introducing the OpenEdge main page

The OpenEdge main page is the central user interface of the management console. From here you can access information for each OpenEdge resource. Each AppServer broker, NameServer instance, WebSpeed broker has its own OpenEdge main page; each main page provides the controls, activities, and data associated with its specific resource.

---

**Note:** This guide references the OpenEdge main page when addressing functionality common to the OpenEdge resource-related main pages. However, when discussing functionality unique to a product, the **WebSpeed main** page, the **AppServer main** page, or **NameServer main** page are referenced, as appropriate.

---

See the “[Accessing OpenEdge resource information](#)” section on page 2–12 for details on how to access the OpenEdge main page.

## Main page format and content

Figure 2–5 shows a sample **WebSpeed main** page; its format and content sections are identical to the **AppServer main** page, and very similar to the **NameServer** page.



Figure 2–5: WebSpeed main page sample

Each OpenEdge main page follows the Fathom title page naming conventions. That is, the specific resource type, container name, and resource name appear in the upper-left corner of the main page. For example, in [Figure 2–5](#), the title **WebSpeed: chimay.wsbroker1** identifies the default **wsbroker1** broker discovered on the container, or host, **chimay** for a Transaction Server.

A *container* represents a named instance of an AdminServer that is either running Fathom or configured to be monitored by Fathom. There is a one-to-one relationship between the host name and container name, unless there are multiple AdminServers running Fathom on the same host.

[Table 2–2](#) identifies and provides a general description of the four sections on the OpenEdge main page.

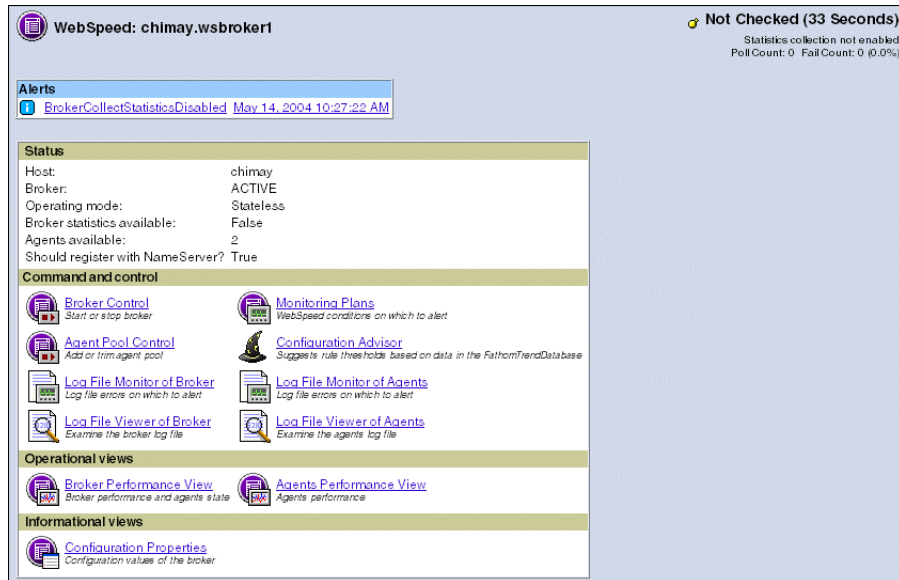
**Table 2–2: Sections of the OpenEdge main page**

<b>This section...</b>	<b>Identifies information you use to...</b>
<b>Status</b>	Review the current operational statistics for a specific OpenEdge resource.
<b>Command and control</b>	Perform various tasks associated with an OpenEdge resource, including: <ul style="list-style-type: none"><li>• Modifying the start and stop controls for a specific resource, or adding and/or trimming a resource's pool of available agents (WebSpeed) or servers (AppServer).</li><li>• Accessing and examining log file monitor and viewer details (for local resources and their associated agents or servers, only).</li><li>• Configuring monitoring plans and rules.</li><li>• Generating recommended rule threshold settings using the Configuration Advisor. (WebSpeed brokers and AppServer brokers polled rules, only.)</li></ul>
<b>Operational views</b>	Help analyze the performance of these OpenEdge resources: <ul style="list-style-type: none"><li>• AppServer brokers and their associated servers through the display of data and associated graphs.</li><li>• WebSpeed brokers and their associated agents through the display of data and associated graphs.</li><li>• Status details about a NameServer instance and its related Application Services.</li></ul>
<b>Informational views</b>	Review the static configuration values for a resource as they are defined in the <code>ubroker.properties</code> file.



## Polling and statistical details on the OpenEdge main page

As shown in [Figure 2–6](#), the page's upper-right corner displays summarized resource polling information pertinent to the currently displayed resource monitor. This section can also report broker resource status details.



**Figure 2–6: Broker statistics not available information**

[Figure 2–6](#) shows the additional line of information — **Statistics collection not enabled** — in the upper-right corner. This resource status detail displays when the **Broker statistics not available** field is not currently enabled (that is, it is set to **False**) on the **Broker Control** page. This message indicates that this resource is not currently collecting statistical data. Therefore, no trending, polling, or graphing can occur for this resource until the status of the **Broker statistics available** field is changed to **True**.

The **WebSpeed** and **AppServer main** pages also present the collection status information in the **Broker statistics available** field in the **Status** section.

---

**Note:** The Fathom internal information alert, **BrokerCollectStatisticsDisabled**, can also appear on the OpenEdge main page if the current setting of the **Broker statistics available** field is **False**.

---

For more information about the **Broker statistics available** field and specific OpenEdge resources, see the:

- “[Data collection details](#)” section on page 3–9, as it describes using this field with WebSpeed broker resources.
- “[Data collection details](#)” section on page 4–9, as it describes using this field with AppServer broker resources.

## Accessing OpenEdge resource information

From the management console, you can display OpenEdge resources and their associated data. There is a unique main page for each instance of an OpenEdge resource type — WebSpeed broker, AppServer broker, and NameServer instance.

This section presents procedures to access

- OpenEdge resource information from the list frame.
- A specific OpenEdge main page.

For more details about navigating the Fathom Management console, see the [Resource Monitoring Guide](#).

### Accessing OpenEdge resources from the list frame

This section describes the procedure to access OpenEdge resources from the list frame.



**To access OpenEdge resources from the list frame:**

1. Click **Resources** in the management console menu bar.

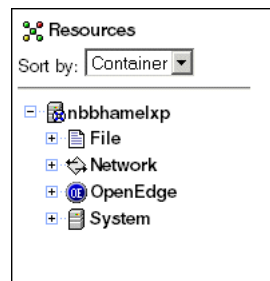
2. Click **Container** in the **Sort by** field on the list frame; at a minimum, all containers defined to Fathom appear in the list frame.
3. Perform one of the following tasks from the list frame:

Click the...	To...
Plus icon (+) associated with the container name whose OpenEdge resources you want to access	Display the <b>OpenEdge</b> subcategory in the list frame tree below the container's name. See the <a href="#">“Accessing an OpenEdge main page”</a> section on page 2–14.
Container's name	Display the <b>Container</b> page in the detail frame. The <b>OpenEdge</b> category displays in the resource category box at the bottom of this page. See the <a href="#">“Accessing an OpenEdge main page”</a> section on page 2–14.

Note these points:

- Local containers have a life preserver associated with the container's icon.
- The plus (+) and minus (-) icons in the list frame indicate whether Fathom has discovered resources for a container and individual resource categories. You can also click the plus and minus icons in the list frame to expand and collapse the contents of each levels. (An empty box indicates either a category has been fully expanded or it has no contents.)

For example, this figure shows the contents of the list frame after the plus sign associated with a local container has been clicked:



Note that the resource categories beneath the **nbbhamelp** local container are available for further expansion.

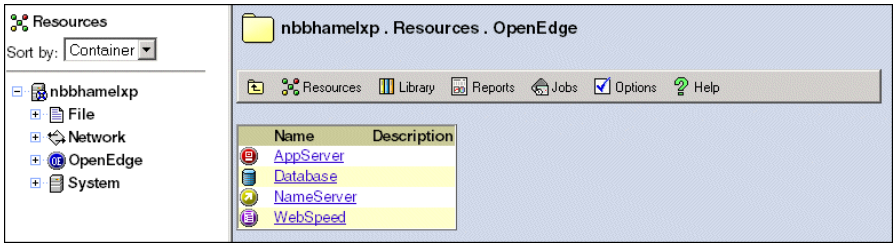
# Accessing an OpenEdge main page

This section presents the procedure to access an OpenEdge main page.



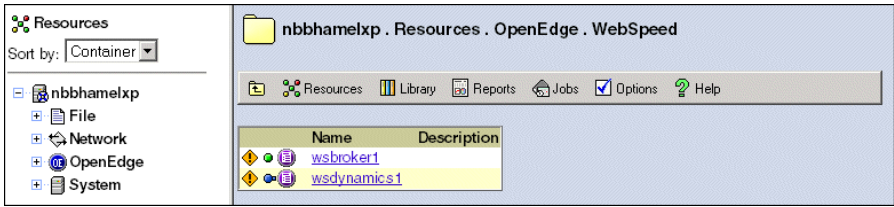
## To access an OpenEdge main page:

1. Click the word **OpenEdge** in the list frame. The available **OpenEdge** resources appear in the detail frame. For example, click **OpenEdge** in the list frame as shown:



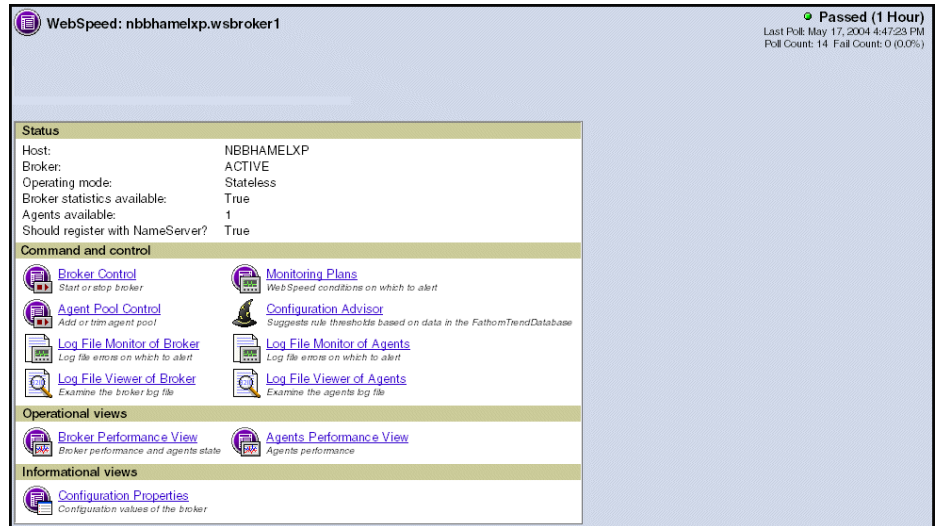
**Note:** If you were to click the container name in the list frame, you would then click the **OpenEdge** category that would appear in the **Resource** list at the bottom of the **Container** page in the detail frame.

2. Click one of the **OpenEdge** resource categories: **AppServer**, **Database**, **NameServer**, or **WebSpeed**. If you select **WebSpeed**, a list that identifies all of the associated **WebSpeed** resources displays in the detail frame appears:



In this example, Fathom discovered two WebSpeed resources: **wsbroker1** (the default broker) and **wsdynamics1**.

- Click on a name in the list that displays in the detail frame; the detailed OpenEdge main page for the associated broker displays in the detail frame:



This example shows the **WebSpeed main** page for the WebSpeed broker named **wsbroker1** on the container **nbbhamelp**.

---

**Note:** With the exception of the log file monitor and log file viewer links for remotely discovered OpenEdge resources, all links are automatically enabled for all resource types when they are discovered.

---

## Deleting OpenEdge resources

Note the following considerations before attempting to delete any OpenEdge resource:

- Each Transaction Server, AppServer, and NameServer resource that you can monitor and manage using the Fathom Management console has an associated configuration definition stored in the `ubroker.properties` file. Before you can delete any of these OpenEdge resources from Fathom, you must first delete this source definition from the `ubroker.properties` file.
- You cannot delete a remote OpenEdge resource when the container in which it resides is currently offline. The container must be back online before you can delete a remote resource of this kind.

- Before attempting to delete an OpenEdge resource, it must be stopped. Therefore, you must ensure that the resource is stopped in Fathom and in Progress Explorer. For details about how to stop an OpenEdge resource, see the appropriate chapter in this guide.
- Fathom cannot recognize specific requests, including resource deletions, while an AdminServer warm start process is occurring. For more details about initiating an AdminServer warm start and its implications for Fathom functionality, see the [“Effects of an AdminServer warm start on Fathom”](#) section on page 2–18.



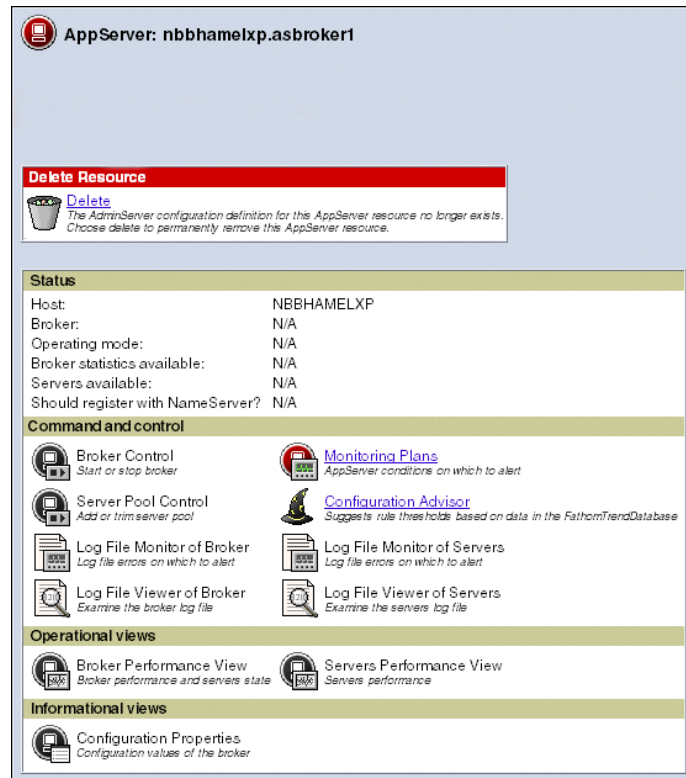
### To delete an OpenEdge resource:

1. Ensure that the resource you want to delete is stopped in Fathom and in the Progress Explorer.
2. Delete the WebSpeed, AppServer, or NameServer instance. This action removes the configuration data stored in the `ubroker.properties` file.

You can use either the Progress Explorer interface or manually configure the `ubroker.properties` file directly to perform this step.

3. Display the OpenEdge main page for the instance that you want to start. Refer to the [“Accessing OpenEdge resource information”](#) section on page 2–12 for the detailed steps.

The **Delete Resource** message displays above the OpenEdge main page as shown:



4. Click the **Delete** link. After you confirm your deletion, the **Fathom Resource** page appears.
5. Refresh the list frame; the resource instance that you deleted no longer appears in the list frame.

## Effects of an AdminServer warm start on Fathom

An AdminServer warm start is a user-initiated process that allows you to manually edit the `ubroker.properties` file while the AdminServer is running. Performing this type of activity is reserved for making small, simple changes to an individual resource's configuration properties stored in the `ubroker.properties` file.

Some benefits to performing a warm start include eliminating the:

- Use of the Progress Explorer tool to perform minimal configuration changes.
- Requirement to stop and restart the AdminServer to accommodate unique situations that might require you to manually edit the `ubroker.properties` file, minimizing Fathom downtime.

For more extensive changes to the `ubroker.properties` file, it is recommended that you use the Progress Explorer tool.

---

**Caution:** Due to the potential for file corruption and other possible unintended side effects of initiating a warm start, it is recommended that only advanced OpenEdge users who understand the concepts and activities presented in this section consider performing the warm start activity as described.

---

## Stages of a warm start

The general stages in an AdminServer warm start are:

1. An advanced user opens the `ubroker.properties` file in an editor and makes minimal configuration changes.
2. The user saves the changes to submit configuration property changes.
3. The AdminServer validates the configuration property changes.

Fathom cannot accept any other broker-related requests that users might try to initiate. This situation means that you, and other users logged in to Fathom at this time, might see links grayed out or disabled that are normally available.

4. Complete Fathom functionality is restored when the AdminServer completes the warm start. This includes the availability of all temporarily disabled links.



## Validating the property changes

You can also use the product-specific configuration utilities to validate the configuration property changes.

See the appropriate OpenEdge documentation for information about how to use these configuration utilities:

- **WSCONFIG** — To display and validate values associated with the property settings in the `ubroker.properties` file for a Transaction Server.
- **ASCONFIG** — To display and validate values associated with the property settings in the `ubroker.properties` file for an AppServer.
- **NSCONFIG** — To display and validate values associated with the property settings in the `ubroker.properties` file for a NameServer.

## Warm start implications

An AdminServer warm start is initiated when the configuration changes are saved. The following specific Fathom activities will be restricted until the AdminServer completes the warm start:

- Data for any OpenEdge resource will not be trended to the FathomTrendDatabase.
- Resource polling for any OpenEdge resource will not occur.
- OpenEdge resource controls will not be available.
- OpenEdge resource views will not display any data.
- OpenEdge log file viewers will not display any data.

Also, specific Fathom activities will be unpredictable until the warm start is completed. Unpredictable activities include:

- Fathom jobs might not be initiated or completed, as expected, due to a possible interruption in communications process.
- Alert generation might be unpredictable and therefore unreliable due to possible interruption in the communications process.

## Initiating an AdminServer warm start

This section describes the steps to initiate an AdminServer warm start.



### To initiate an AdminServer warm start:

1. Start the editor you intend to use to manually edit the configuration properties associated with a specific resource's `ubroker.properties` file.
2. Make the changes to the `ubroker.properties` file.
3. Save the changes to initiate the AdminServer warm start.

See the “[Warm start implications](#)” section on page 2–19 for details about Fathom feature restrictions and limitations during a warm start.

The AdminServer begins the warm start. See the “[AdminServer warm start process](#)” section on page 2–20.

## AdminServer warm start process

After you perform the steps described in the “[Initiating an AdminServer warm start](#)” section on page 2–20, the stages of the AdminServer process occur as follows:

1. The AdminServer begins the warm start:
  - a. The AdminServer validates the changes requested.
  - b. Fathom rejects any OpenEdge resource-related requests initiated.
  - c. Fathom displays a **Warm Start Began** internal alert.
2. The AdminServer finishes the warm start:
  - a. Fathom functionality restored to its pre-warm start state.
  - b. Fathom displays a **Warm Start Finished** internal alert.

## Understanding OpenEdge server graphs

Fathom displays OpenEdge server data in a graphical format for:

- WebSpeed and AppServer resources on **Performance View** pages.
- OpenEdge resource members on user-selected viewlets in collections.

### Graphs available on Performance View pages

OpenEdge resources that have defined monitoring plans can display certain data in graphical formats. [Table 2–3](#) identifies various OpenEdge resource types, the data that can appear in individual graphs, and the performance data page on which the graphs display:

**Table 2–3: Performance pages and their graphical content**

For this OpenEdge resource type...	Data addressing each of these topics...	Displays as an individual graph on this page...
WebSpeed brokers	- Broker Request Activity - Broker Activity Status - Client Connections	Broker Performance View
WebSpeed Agents	- Agent States - Total Agents CPU - Total Agents Memory	Agents Performance View
AppServer brokers	- Broker Request Activity - Broker Activity Status - Client Connections	Broker Performance View
AppServers	- Server States - Total Servers Memory	Servers Performance View

For an explanation of each graph’s content, see the performance view sections in [Chapter 3, “Managing WebSpeed Transaction Server Data”](#) and [Chapter 4, “Managing AppServer Data.”](#)

---

**Note:** The production of graphs is CPU-intensive. If you are monitoring CPU usage, an alert may fire when the graph is generated. To avoid the firing of such an alert, increase the number of failed polls after which Fathom throws an alert.

---

## Launching graph pinup pages

To launch a separate graph pinup page for any of the individual graphs identified in [Table 2–3](#), select the binoculars icon associated with that graph on its respective performance page.

[Figure 2–7](#) shows the binoculars icon.



**Figure 2–7: Binoculars icon**

As needed, you can change the displayed characteristics of the graph that displays in that pinup. See the “[Changing OpenEdge pinup graphical displays](#)” section on page 2–25 for details.

## Additional graph-related considerations

Depending on the browser in which you are viewing a graph, the graph type and its property settings, and the number of data points displayed, you can display pop-up content details from within the graph. Review pop-up content to inspect resource activity in greater detail.



To display pop-up content from within a graph, place the mouse over regions of the graph.

## Displaying OpenEdge viewlets on a collection view

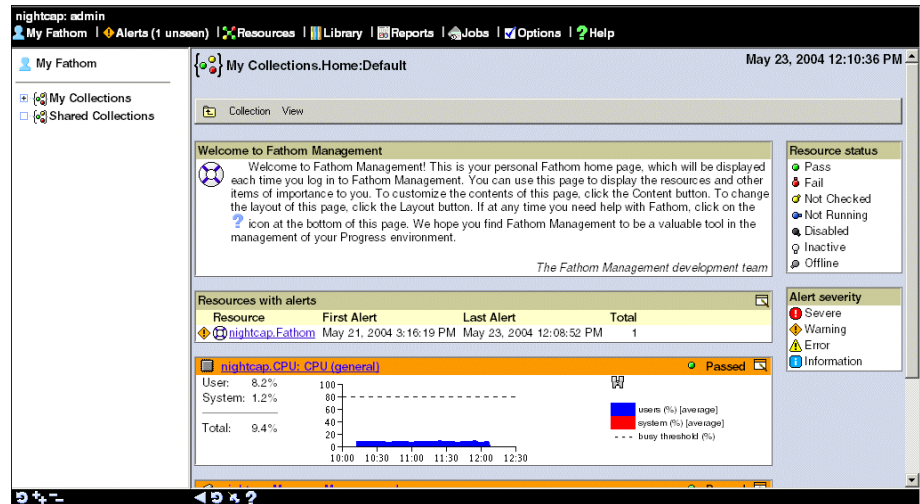
Among other activities, collections allow you to define and organize OpenEdge resource details and to prominently display these details in Fathom. Specifically, you can define OpenEdge resource viewlets to display resource information in a graphical form on a collections page. The second column entitled “Data addressing this topic” in [Table 2–3](#) identifies some of the graphs that the OpenEdge viewlets support.

You can define viewlets for any OpenEdge resource that is a member of a collection.




To access and select OpenEdge resource viewlets for display on a collections page:

1. Click **My Fathom** on the main menu bar. The **My Collections Default Home** page appears in the detail frame:



2. If you are updating the **My Collections Default Home** page, go to [Step 4](#). Otherwise, from the list frame, expand the collections category (**My Collections** or **Shared Collections**) that contains the collection page you want to update.
3. Click the collection. The collections page appears in the detail frame.

- 4. From the detail menu bar on the collections page, click **View** →**Customize View**→**View Content**. (The checkmark in the drop-down menu list identifies the currently active view.)
- 5. The **Edit My Collections** page for the collection appears:

 **Edit My Collections.**OpenEdge\_Server\_Collection:Default: Default  
Contains resources related to AppServers, WebSpeed, and NameServer resources

Save

Cancel

Properties

Name:

Default

Standard viewlets to show

☒ Collection members

☐ Active monitoring plans

☒ Running reports


☐ Alert severity legend

☒ Resources with alerts

☒ Running jobs

☐ Resource status legend

Resource viewlets to show

Resource	Viewlets
 nightcap.asbroker1	<div><div><input type="checkbox"/> AS Broker Request Activity</div><div><input type="checkbox"/> AS Client Connections</div><div><input type="checkbox"/> AS Total Servers CPU</div><div><input type="checkbox"/> AS Broker Queued Requests (percent)</div></div> <div><div><input type="checkbox"/> AS Broker Activity Status</div><div><input type="checkbox"/> AS Server States</div><div><input type="checkbox"/> AS Total Servers Memory</div><div><input type="checkbox"/> AS Broker Rejected Requests (percent)</div></div>

| nightcap.NS1 | ☐ nightcap.NS1 |

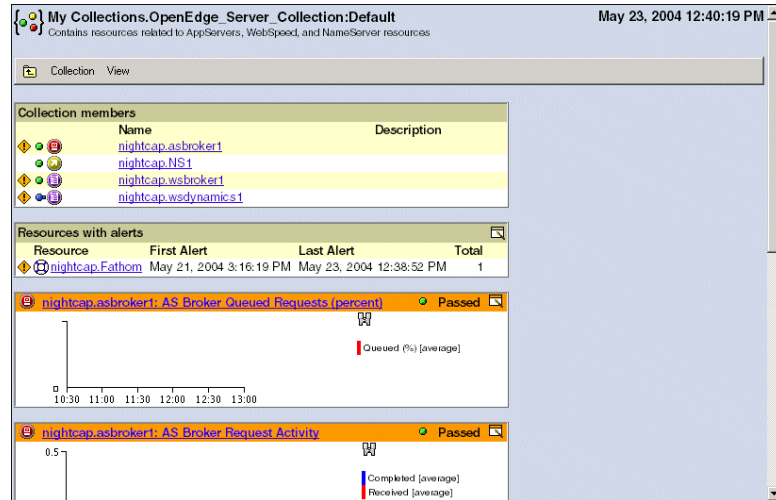
| nightcap.wsbroker1 | ☐ WS Broker Request Activity  ☐ WS Client Connections  ☐ WS Total Agents CPU  ☐ WS Broker Activity Status  ☐ WS Agent States  ☐ WS Total Agents Memory |

For each OpenEdge server resource type defined for the collection, the **Resource viewlets to show** section displays the associated viewlet options.

**Note:** The **Resource viewlets to show** section also supports these AppServer viewlets that are not available on the **AppServer Performance View** page: **AS Total Servers CPU**, **AS Broker Queued Requests (percent)**, and **AS Broker Rejected Requests (percent)**. The NameServer-related viewlet provides access to the NameServer instance’s OpenEdge main page; there are no graphs associated with NameServer resources.

- 6. Click in the box associated with a viewlet option to select it.

7. Click **Save**. The main view of the collections page redisplay with the selected viewlets. Use the scroll bars to view all items defined, as needed:



## Changing OpenEdge pinup graphical displays

You can modify a particular graph by displaying it as a pinup graph. A *pinup graph* is a graph that displays in a separate window and can have its appearance customized. For example, you can choose elements such as the graph's size, style, and how often it refreshes.



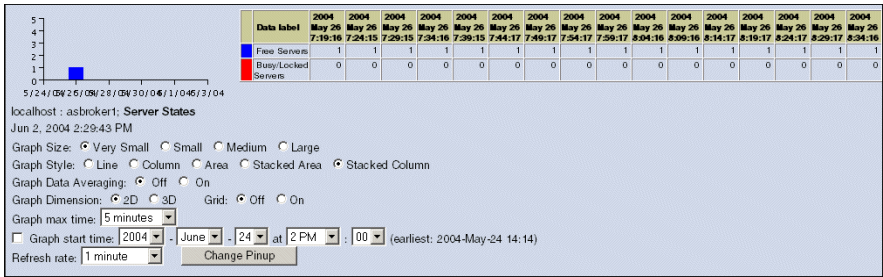
**To access the pinup graph to change the appearance of a graph:**

1. Click the binocular icon in the lower right of the graph whose view you want to modify:



2. The pinup graph window opens, displaying the graph.

3. Drag the lower-right corner of the window to expand the page, as shown in the following figure:



The data label at the top of the graph serves as the graph's legend.

4. From the pinup you can, depending on the graph, customize the graph properties described in [Table 2-4](#).

**Table 2-4:** Graph properties and options for time-based graphs (1 of 3)

Property	Options	Comments
Graph Size	Very small Small Medium Large	If you have a graph with small statistics, you can choose to have the pinup graph larger so you can better see its details.
Graph Style	Line Area Column Stacked Area Stacked Column	Depending on the kind of graph you are viewing in the pinup, you can change its style from one style to another recognized style.
Graph Data Averaging	Off On	Default is <b>Off</b> . If you select <b>On</b> , data displays as a weighted average for the time period set for the <b>Graph max time</b> option.  Setting this option to <b>On</b> reduces the number of data points displayed.
Graph Dimension	2D 3D	Changes the display from 2 dimensional to 3 dimensional.



**Table 2–4: Graph properties and options for time-based graphs** (2 of 3)

Property	Options	Comments
<b>Grid</b>	<b>On</b> <b>Off</b>	Default is <b>Off</b> .
<b>Graph max time</b>	<b>5 minutes</b> <b>10 minutes</b> <b>15 minutes</b> <b>30 minutes</b> <b>45 minutes</b> <b>1 hour</b> <b>2 hours...</b>	Controls how much time the graph spans. Note that this does not affect how often or how much data is collected for graphing.  However, the ranges for the specific value options from which you can select are governed by the settings you define for the <b>Graph cache</b> option. For details about the <b>Graph cache</b> option, see the appropriate section in the <i>Installation and Configuration Guide</i> .

**Table 2–4: Graph properties and options for time-based graphs** *(3 of 3)*

Property	Options	Comments
<b>Graph start time</b>	Click in checkbox.  Provide year/month/day/time start time settings.	Identifies the start date and time for the graph. The purpose of this start information is to help you drill deeper into the resource activity details recorded for a specific time frame.  To select this option, click in the checkbox on the side left of the field label. Select the year/month/day/time settings from the fields displayed on the right side of the field label.  To ensure a meaning graphing of data, review the <b>Graph cache</b> option and the <b>Graph max time</b> setting values as you determine the value for this start time setting.
<b>Refresh rate</b>	None 15 seconds 1 minute 2 minutes 3 minutes 4 minutes 5 minutes 10 minutes 15 minutes	The refresh rate is the rate at which the resource is checked to see if there is more information to put in the graph.  The refresh rate should not be less than the polling rate for the resource. For example, if you set the refresh rate to 1 minute and the polling rate is at 5 minutes, you do not get new graph data every minute; you get it only at the same rate as the polling occurs.

5. Click **Change Pinup** when you finish making your selections. The graph appears in the pinup with the new characteristics.

---

**Note:** You cannot save the pinup graph settings.

---

---

# Managing WebSpeed Transaction Server Data

---

This chapter presents Fathom Management features and functionality related to the Transaction Server.

Topics in this chapter include:

- [Overview](#)
- [Reviewing WebSpeed broker status](#)
- [Modifying WebSpeed control settings](#)
- [Accessing and reviewing WebSpeed-related log file data](#)
- [Using the WebSpeed log file viewers](#)
- [Examining WebSpeed-related Operational views](#)
- [Examining WebSpeed-related Informational views](#)

# Overview

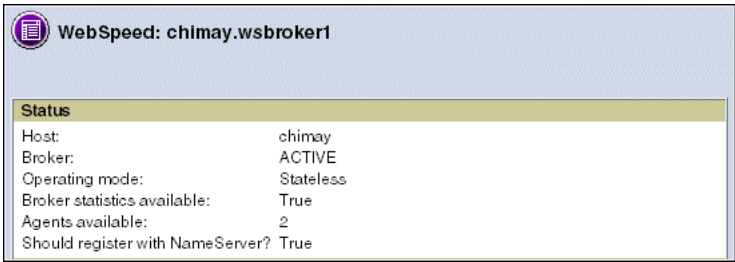
Fathom supports a variety of tasks that you can perform to manage a Transaction Server, including:

- Reviewing your current operating status and associated details.
- Modifying broker-related control settings, such as starting and stopping a broker, and adding or trimming agents.
- Accessing and viewing broker- and agent-specific data collected through log file resource monitors.
- Monitoring and managing WebSpeed brokers using monitoring plans and rules.
- Generating rule threshold values for rules using the Configuration Advisor.
- Working with OpenEdge resource-related data that is available through broker- and agent-specific information and operational views. WebSpeed information views provide data in both text and graph formats.

You must have appropriate Fathom role authorization to perform several of these tasks. See the [“Role authorization and OpenEdge tasks”](#) section on page 1–7 for details.

## Reviewing WebSpeed broker status

The **Status** section of the **WebSpeed main** page summarizes current operational details about the WebSpeed broker. [Figure 3–1](#) shows an example of the **Status** section. Note that the broker’s name displays in the page’s title area.



**Figure 3–1:     WebSpeed Status section sample**

Table 3–1 describes each of the WebSpeed broker details.

**Table 3–1: WebSpeed status details**

This field...	Displays the...
<b>Host</b>	Host machine's name.
<b>Broker</b>	<p>Running status of the broker. Possible values are:</p> <ul style="list-style-type: none"> <li><b>ACTIVE</b> (The broker is currently running.)</li> <li><b>Not Running</b> (The broker is not currently running.)</li> </ul> <p>The broker can also report <b>Starting</b> and <b>Shutting Down</b> values; however, depending on the speed of the machine on which your management console is running, you may not see these intermediary states.</p>
<b>Operating mode</b>	Operating mode of the broker. This mode determines how client requests are dispatched to individual agent processes running on the WebSpeed instance.
<b>Broker statistics available</b>	<p>Status of the broker as it relates to data collection. The possible states are <b>True</b> or <b>False</b>.</p> <p>See <a href="#">“Data collection details”</a> section on page 3–9 for more information about data collection.</p>
<b>Agents available</b>	<p>Number of agents running and available to fulfill a connection request from a client through this broker when the broker's status is <b>ACTIVE</b>.</p> <p>This value can change frequently, reporting the real-time changes in number of agents available.</p>
<b>Should register with NameServer?</b>	Status of <b>True</b> or <b>False</b> to indicate whether the broker resource is registered with a NameServer.

These points relate to the fields highlighted in Table 3–1:

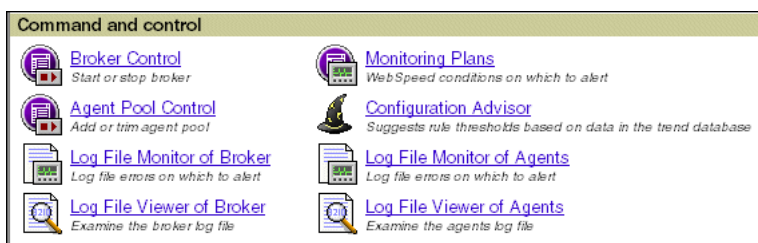
- Broker-related changes that you can make, using either the **Broker Control** or **Agent Pool Control** options in the **Command and control** section of the **WebSpeed main** page, can affect the broker and agent values that display in this **Status** section.
- The values that display in the WebSpeed-related **Status** section are obtained either from the `ubroker.properties` file or the current, real-time status of the broker (if it is running).

## Modifying WebSpeed control settings

The **Command and control** section of the **WebSpeed main** page allows you to:

- Start and stop a WebSpeed broker, and change its associated property settings.
- Add or trim the pool of available agents associated with the broker.
- Obtain and review WebSpeed-related data collected through broker- and agent-specific log files for which you can set up resource monitors.
- Monitor and manage WebSpeed brokers using monitoring plans and rules, including the option to use Configuration Advisor rule-recommended threshold settings.

Figure 3–2 shows an example of the **Command and control** section of the **WebSpeed main** page.



**Figure 3–2: Command and control section example**

---

**Note:** The information in this section presents functional descriptions and procedural details related to the **Broker Control** and **Agent Pool Control** pages.

---

[Table 3–2](#) identifies where you can find information about other functionality related to the **Command and control** section.

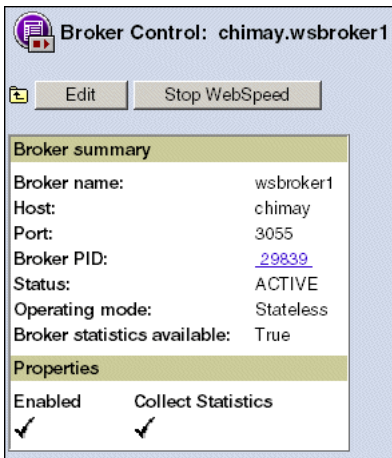
**Table 3–2: Additional WebSpeed information**

For WebSpeed-related details about...	See the...
Broker and agent pool log file monitors and viewers	<a href="#">“Accessing and reviewing WebSpeed-related log file data”</a> section on page 3–20.
Broker monitoring plans and rules	<a href="#">Chapter 6, “Monitoring Plans and Rules for OpenEdge Server Resources.”</a>
Broker rule sets	<a href="#">“Working with rule sets”</a> section on page 6–29.

## Broker Control page content

The **Broker Control** page summarizes details about a specific WebSpeed Broker resource. From this page, you can start and stop a WebSpeed broker, and change some broker-related properties, as your processing needs warrant.

Figure 3–3 shows an example of the **Broker Control** page.



**Figure 3–3: Broker Control page example**

---

**Note:** The values associated with the **Collect Statistics** property and the **Broker statistics available** field are interdependent. See the [“Data collection details”](#) section on page 3–9 for details.

---

The following details describe the two sections of the **Broker Control** page.

**Broker summary section**

The **Broker summary** section displays read-only values for these fields: the broker name, its host machine’s name, associated port number and process identification number (PID), the broker’s current status, operating mode, and whether the broker is currently set to collect broker-related statistical data.



Note the following additional details about these fields:

- The values shown in the **Broker name**, **Host** (machine name), **Port** (number), and **Operating mode** fields display values as they are defined in the `ubroker.properties` file.
- The **Broker PID** and **Status** fields reflect real-time values based on the broker's current status. The **Broker PID** is also a link to more broker process details. See the [“Viewing broker process details”](#) section on page 3–11 for more details.
- The **Broker statistics available** field also reflects a current, real-time value. However, the value displayed in this field depends on additional factors. See the [“Data collection details”](#) section on page 3–9 for more details.

### Properties section

The **Properties** section displays the status of two user-defined, broker-related properties, **Enabled** and **Collect Statistics**:

- The **Enabled** option indicates that this broker resource recognizes a monitoring plan and its associated rules when the broker resource is active.

During the discovery process, all WebSpeed brokers that Fathom discovers and lists in the list frame under the WebSpeed category are *enabled*, or set, by default. Once a broker is enabled, Fathom uses the Fathom-supplied default values to establish a monitoring plan and rules. (You can customize the plan and rules at any time.)

- The **Collect Statistics** option enables data collection to occur in the WebSpeed broker. Fathom uses this data to identify the broker's performance. If you do not select the **Collect Statistics** option for a specific broker (that is, **True** status), Fathom displays only non-statistical data such as **Status** and **PID** (pid number) on the various WebSpeed broker pages; polled rules are not evaluated and data is not trended.

The **Collect Statistics** value plays a central role in data collection. See the [“Data collection details”](#) section on page 3–9 for more details.

A check mark indicates that the individual property is set.

---

**Note:** To set the **Broker statistics available** option to a **True** status for a specific broker, you must enable the **Collect Statistics** option. See the procedure in the [“Data collection details”](#) section on page 3–9 for details.

---

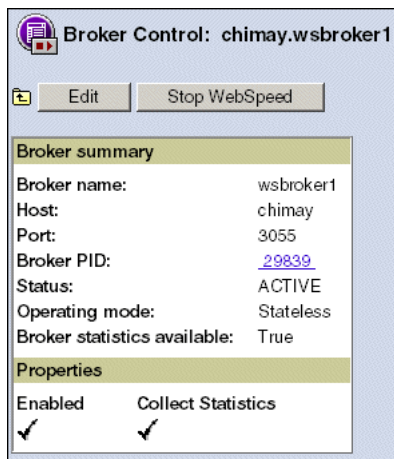
## Changing WebSpeed Broker controls

This section describes how to change WebSpeed broker controls.



**To start or stop the WebSpeed broker and to change its property settings:**

1. Display the **WebSpeed main** page for the broker you want to start. See the “[Accessing OpenEdge resource information](#)” section on page 2–12 for the detailed steps.
2. Click **Broker Control** in the **Command and control** section to display the **Broker Control** page as shown:



This table highlights the controls that you can change:

To...	Then...
<p>Change the current setting of the <b>Enabled</b> property.</p> <p><b>Note:</b> A checkmark displays to indicate that the <b>Enabled</b> property is set. To clear this option, click the checkmark in the box associated in the option; the checkmark is deleted to indicate that the option is no longer set.</p>	<p>Click <b>Edit</b>. Then select or deselect the <b>Enabled</b> property to add or remove the checkmark.</p> <p>You must also restart the WebSpeed broker so that the property change is recognized.</p>
<p>Change the current setting of the <b>Broker statistics available property</b> displayed in the <b>Broker Summary</b> section of the <b>Broker Control</b> page.</p>	<p>See the <b>“Data collection details”</b> section on page 3–9.</p>
<p>Toggle between stopping and starting the WebSpeed broker, depending on the current value displayed in the <b>Status</b> field.</p> <p>For example, if the broker status currently displays <b>ACTIVE</b>, the button label will read <b>Stop WebSpeed</b>. You can click this button to stop the WebSpeed broker; Fathom stops this broker and updates the value in the <b>Status</b> field to display <b>Not Running</b>.</p>	<p>Click <b>Stop WebSpeed</b> or <b>Start WebSpeed</b>.</p>
<p>Exit this page without changing any values and redisplay the <b>WebSpeed main</b> page.</p>	<p>Click <b>Cancel</b>.</p>

### Data collection details

Data collection ensures that broker-related performance statistics can be trended to the FathomTrendDatabase. Options and conditions available on the **Broker Control** page and the WebSpeed broker resource monitoring plan must be fulfilled to successfully implement data collection.

On the **Broker Control** page, these conditions include:

- Selecting the **Collect Statistics** check box to enter a check mark.
- Starting, or stopping and restarting the WebSpeed broker; you must explicitly perform this step on the **Broker Control** page to effect this change.

- The value **TRUE** displayed in the **Broker statistics available** field. (Fathom automatically updates this field when it detects that the **Collect Statistics** option is enabled after you have started, or stopped and re-started, the WebSpeed broker.)

On the WebSpeed broker resource monitoring plan, you must also check the **Trend Performance Data** option.

---

**Note:** You are not required to use trending with the data collection activity. However, without the **Trend Performance Data** option selected, you cannot trend data. Data trended to the FathomTrendDatabase is required for WebSpeed-related rule evaluation, graphical displays, and report generation.

---

For information about the **Trend Performance Data** option and monitoring plans for WebSpeed broker resources, see [Chapter 6, “Monitoring Plans and Rules for OpenEdge Server Resources.”](#)

---

**Note:** Using data collection might cause the Web Speed broker to exhibit some level of performance degradation, memory degradation, or both.

---



**To set the options to perform data collection in a WebSpeed broker:**

1. Check the current setting of the **Collect Statistics** field in the **Properties** section of the **Broker Control** page; it must be checked to indicate that this property is set.

If the **Collect Statistics** field is not checked, then click **Edit**. In the **Collect Statistics** field, click in the check box to type a check mark. Click **Save**.

2. Stop and restart the WebSpeed broker you want to update.

---

**Caution:** You cannot interactively change a running process. Therefore, you must restart the WebSpeed broker after you change the setting of the **Collect Statistics** field.

---

The **Broker statistics available** field in the **Broker summary** section will display **TRUE** if the broker re-started successfully. The **TRUE** value indicates that you have successfully set data collection and that broker statistical data is now available to be stored in the FathomTrendDatabase.

3. Check the current setting of the **Trend** field in the **Properties** section of the **Broker Control** page; it must be checked to indicate that this property is set. See the [“Default polling and trend values”](#) section on page 6–5 and the [“Default monitoring plan details”](#) section on page 6–8 for details about this field.

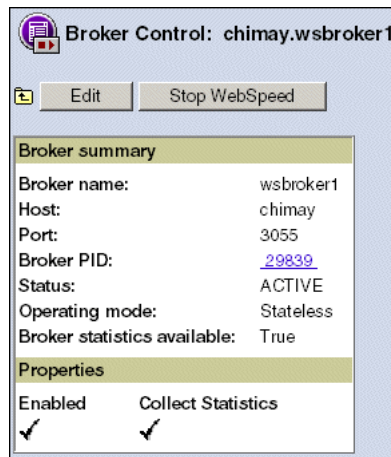
## Viewing broker process details

You can also access real-time details and statistics that provide you with “snapshot” information about an individual broker at the point you access this information from the **Broker Control** page. Review this information to help you determine a broker’s performance.

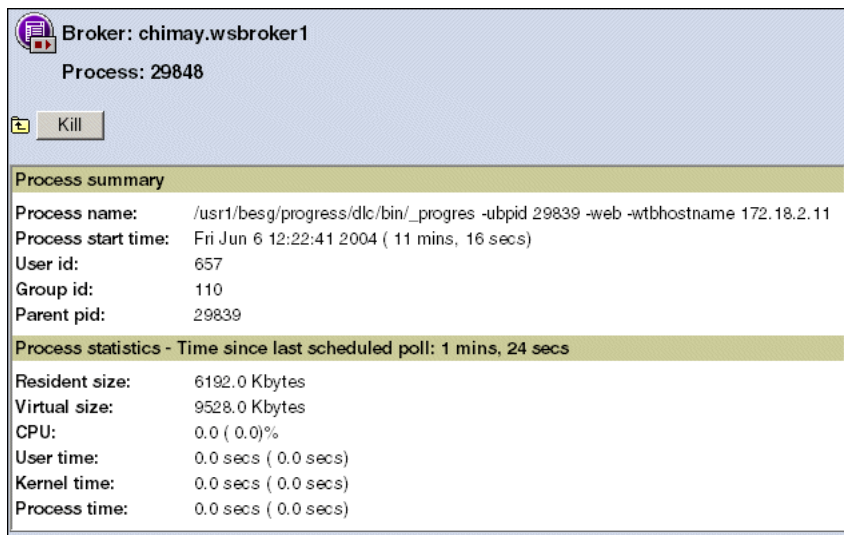


### To access broker processing details:


1. Display the **WebSpeed main** page for the WebSpeed broker you want to review. See the [“Accessing OpenEdge resource information”](#) section on page 2–12 for the detailed steps.
2. Click **Broker Control** in the **Command and control** section to display the **Broker Control** page as shown:



- Click the unique PID number associated with the **Broker PID** field to display a **Broker process** page. This page contains summary and real-time statistics about the broker as shown:



Broker: chimay.wsbroker1  
Process: 29848

 Kill

**Process summary**

Process name:	/usr1/besg/progress/dlc/bin/_progres -ubpid 29839 -web -wtbhostname 172.18.2.11
Process start time:	Fri Jun 6 12:22:41 2004 ( 11 mins, 16 secs)
User id:	657
Group id:	110
Parent pid:	29839

**Process statistics - Time since last scheduled poll: 1 mins, 24 secs**

Resident size:	6192.0 Kbytes
Virtual size:	9528.0 Kbytes
CPU:	0.0 ( 0.0)%
User time:	0.0 secs ( 0.0 secs)
Kernel time:	0.0 secs ( 0.0 secs)
Process time:	0.0 secs ( 0.0 secs)

The two sections that comprise the **Broker process** page present relevant information about the WebSpeed broker and its current operations:

- The **Process summary** section identifies the **Process name** and **Process start time**. **User id** and **Group id** values display when Unix-based data displays. The **Parent pid** identifies the process identifier number associated with the process that spawned this current process.
- The **Process statistics** section presents details about the broker's real-time operational status. Values presented without parentheses identify that the processing time determined since the last scheduled polling interval, as noted, has occurred. Values presented within parentheses have been calculated based on information obtained since the start of the process. [Table 3-3](#) identifies and describes these attributes.

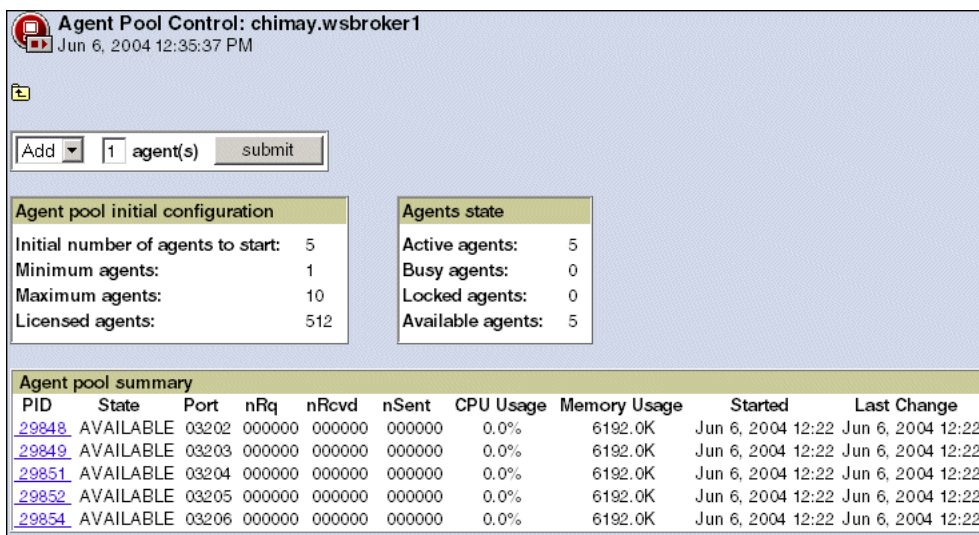
**Table 3–3: Process statistics section real-time operational data**

This field...	Identifies the...
<b>Resident size</b>	Physical size of the process as defined by host system.
<b>Virtual size</b>	Virtual size of the process as defined by host system.
<b>CPU%</b>	Percentage of time spent using the CPU in either the user or kernel mode since the last scheduled poll.
<b>User time</b>	Amount of CPU time spent in the user mode since the last scheduled poll.
<b>Kern time</b>	Amount of CPU time spent in the kernel mode since the last scheduled poll.
<b>Process time</b>	Sum of the values that display in the <b>User time</b> and <b>Kernel time</b> fields.
<b>Weighted CPU</b>	<p>Percentage of time spent using the CPU in either the user or kernel mode since the last scheduled poll divided by the number of CPU processors on the system.</p> <p>This value displays only when there is more than one CPU process on the system where the process is running.</p>

## Agent Pool Control page content

Figure 3–4 displays the **Agent Pool Control** page. It displays data relevant to your current WebSpeed workload and allows you to add or reduce the number of WebSpeed agents currently running.

For example, use this page to add agents when agent requests are high; you can add agents to the maximum number of agents that your license recognizes. Also, use this page to reduce the agent count during a lag in agent requests. Using the trim feature, you can reduce agents down to the **minimum agents** property setting.



**Figure 3–4: Agent Pool Control page example**

The **Agent Pool Control** page is comprised of these sections:

- An add/trim selection control that you use to specify which activity you want to perform. When you initiate a manual trim request, Fathom determines which agents to remove. See the [“Adding or trimming agents”](#) section on page 3–18 for detailed steps.
- Three distinct, agent-related data summary tables that allow you to review relevant agent-pool specific data quickly:
  - **Agent pool initial configuration.**
  - **Agents state.**
  - **Agent pool summary.**

The changes that you make through add/trim activities can affect the data displayed in these summary tables. The **Agent pool summary** also allows you to kill a specific agent process. See the [“Killing a WebSpeed agent process”](#) section on page 3–19 for the detailed steps.

The following information describes each section of the **Agent Pool Control** page. For the detailed procedures on how to add or trim agents, see the [“Adding or trimming agents”](#) section on page 3–18 for more information.



## Agent pool data initial configuration section

The **Agent pool initial configuration** section identifies WebSpeed broker configuration properties set in the `ubroker.properties` file. These values display in a read-only mode.

Table 3–4 identifies and describes each field that displays in the **Agent pool data initial configuration** section.

**Table 3–4: Agent pool data initial configuration field definitions**

This property field...	Identifies...
<b>Initial number of agents to start</b>	The value Fathom references when the WebSpeed broker starts agents. Depending on your license agreement and your strategy for setting up your configuration information in the <code>ubroker.properties</code> file, this value may be the same as the value displayed in the <b>Licensed to start</b> field.
<b>Minimum agents</b>	<p>The minimum number of agents that must be simultaneously running before the WebSpeed broker will start additional agents. The broker strives to maintain this specified minimum; if at any time the number of agents falls below the specified minimum, the broker will automatically start any additional agents needed to maintain this minimum.</p> <p>If you set a trim value that would require Fathom to trim the number of agents below the number specified for this field, Fathom displays a message.</p>
<b>Maximum agents</b>	The maximum number of WebSpeed processes that can be running simultaneously. Fathom will not fulfill add requests you initiate that exceed this specified maximum. Fathom will display a message to state this condition so that you can re-consider your request and, if necessary, initiate a new request.
<b>Licensed agents</b>	The number of agents that a broker can recognize and start as determined by your license agreement.

Agents state section

The **Agents state** section provides a snapshot of the total number of agents currently associated with a specific agent state. The state details related to agents and the number of agents reported reflect real-time data. This data can fluctuate due to changes in the agents’ workflow and changes you initiate using the add and trim feature.

Table 3–5 identifies and describes each field presented in the **Agents state** section.

Table 3–5: Agents state field definitions

This agent state...	Identifies the number of agents currently...
Active	Running.
Busy	Serving 4GL client requests.
Locked	Servicing a bound connection.
Available	Available to handle broker requests.

Agent pool summary section and the kill process option

The **Agent pool summary** section provides:

- Detailed data about each individual agent in the WebSpeed pool associated with a specific WebSpeed broker. Table 3–6 identifies and describes each field displayed in the **Agent pool summary** section.
- Access to:
  - More data about a specific agent.
  - A control to terminate, or kill, the agent process.

Use the **PID** field to access these features. Table 3–6 provides more information about **PID**.

**Table 3–6: Agent pool summary field definitions**

<b>This field...</b>	<b>Identifies the...</b>
<b>PID</b>	Process identifier for this agent. Click on the specific <b>PID</b> number to display a detail page that provides specific information about this agent process and, as necessary, kill the process. See the “ <a href="#">Killing a WebSpeed agent process</a> ” section on page 3–19 for more information.
<b>State</b>	Current execution state of the agent process.
<b>Port</b>	TCP/IP port number that the agent process uses.
<b>nRq (Number of Requests)</b>	Number of messages sent to the agent process.
<b>nRcvd</b>	Number of messages received by the agent process.
<b>nSent</b>	Number of requests sent by the agent process.
<b>CPU Usage</b>	Percentage of CPU user and system time consumed by a process.
<b>Memory Usage</b>	Amount of virtual memory (in Kbytes) consumed by a process.
<b>Started</b>	Time stamp that indicates when the agent process started. If the broker is restarted for any reason, the <b>PID</b> and the <b>Last Changed</b> value might change.
<b>Last Changed</b>	Time stamp that indicates when the agent process last changed execution state.

## Adding or trimming agents

This section identifies the steps to add and trim agents.



To initiate a WebSpeed agent add and trim request:

1. Display the **WebSpeed main** page for the broker you want to start. See the “[Accessing OpenEdge resource information](#)” section on page 2–12 for the detailed steps.
2. Click **Agent Pool Control** in the **Command and control** section to display the **Agent Pool Control** page as shown:

Agent Pool Control: chimay.wsbroker1  
Jun 6, 2004 12:35:37 PM

Add

1 agent(s)

submit

Agent pool initial configuration

Initial number of agents to start: 5  
Minimum agents: 1  
Maximum agents: 10  
Licensed agents: 512

Agents state

Active agents: 5  
Busy agents: 0  
Locked agents: 0  
Available agents: 5

Agent pool summary

PID	State	Port	nRq	nRcvd	nSent	CPU Usage	Memory Usage	Started	Last Change
<a href="#">29848</a>	AVAILABLE	03202	000000	000000	000000	0.0%	6192.0K	Jun 6, 2004 12:22	Jun 6, 2004 12:22
<a href="#">29849</a>	AVAILABLE	03203	000000	000000	000000	0.0%	6192.0K	Jun 6, 2004 12:22	Jun 6, 2004 12:22
<a href="#">29851</a>	AVAILABLE	03204	000000	000000	000000	0.0%	6192.0K	Jun 6, 2004 12:22	Jun 6, 2004 12:22
<a href="#">29852</a>	AVAILABLE	03205	000000	000000	000000	0.0%	6192.0K	Jun 6, 2004 12:22	Jun 6, 2004 12:22
<a href="#">29854</a>	AVAILABLE	03206	000000	000000	000000	0.0%	6192.0K	Jun 6, 2004 12:22	Jun 6, 2004 12:22

3. From the drop-down list box, select **Add** or **Trim**.
4. In the **agent(s)** field, enter the number of agents you want to add or trim. The value you enter must be a positive integer.

When you initiate an add or trim request, Fathom consults two sets of initial configuration to determine if and how it can honor either request type:

- The number of agents for which you are licensed.
- The broker property configuration settings stored in the `ubroker.properties` file.

See the “[Agent pool data initial configuration section](#)” section on page 3–15 for details about these initial configuration values.

5. Select **Submit**. Depending on the changes you make and Fathom’s capability to implement them, you might notice changes to the numeric values displayed in the **Agents State** table. See the “[Agents state section](#)” section on page 3–16 for more information.

---

**Note:** Any time you either add or trim WebSpeed agents, it is recommended that you refresh the management console to ensure that you are not viewing stale data.

---

## Killing a WebSpeed agent process

You might want to manually terminate an agent process when:

- An agent process hangs.
- You determine from the available data that an agent process is a runaway process.

The specific **PID** on the **Agent pool summary** section of the **Agent Pool Control** page allows you to access the page to kill the offending agent’s process.

---

**Note:** Fathom references the specific **PID** and its associated date and time start details to be sure of a process’ identify before it attempts to kill a process.

---



### To initiate a kill process:

1. Click the **PID** associated with the agent process you want to terminate. The specific WebSpeed agent process page appears, as shown:

Broker: vesta.wsbroker1  
Process: 28390

Kill

**Process summary**

Process name: /usr1/bsg/progress/dlc/bin/.proges -ulpid 42750 -web -withhostname 172.16.7.195 -withport 41304 -withamport 3502 -withmaxport 3502 -withname wsbroker1  
-withinstance 5 -ubprofile /usr1/bsg/progress/dlc/properties/ubroker.properties -logname wsbroker1 -p web/objects/web-disp.p -weblogger

Process start time: Tue May 27 13:13:51 2004 (3 hours, 7 mins, 57 secs)

User id: 657  
Group id: 110  
Parent pid: 42750

**Process statistics - Time since last scheduled poll: 3 mins, 20 secs**

Resident size: 5532.0 Kbytes  
Virtual size: 6160.0 Kbytes  
CPU: 0.0 (0.0)%  
User time: 0.0 secs (0.0 secs)  
Kernel time: 0.0 secs (0.0 secs)  
Process time: 0.0 secs (0.0 secs)

Note that the two sections on this page present relevant, summary information about this WebSpeed agent and its current operational status. See the [“Viewing broker process details”](#) section on page 3–11 for details about this data.

2. Click **Kill** to terminate this process. Fathom will prompt you once again to ensure that you want to terminate this process. Click **OK**.

Fathom displays a final status page that identifies the status of your kill request. Fathom displays one of the following messages:

- **Process number *xxxxx* has been terminated.**

This message indicates that the process was successfully killed. The PID number previously associated with this process is now available for the operating system to reassign.

- **Process number *xxxxx* cannot be killed at this time.**

This message indicates that the process could not be killed. In very rare instances, it is possible that you will not be successful in an attempt to kill a process. You can retry the kill process procedure; however, it is possible that the process will persist for a number of unknown reasons.

- **Process number *xxxxx* has been reused.**

Fathom has determined that the process PID number and associated time and date stamp do not match the values that the operating system has stored for this same process. Consequently, when you click **Kill**, the process cannot be destroyed.

3. Click **Cancel** at the top of the page to exit this page without terminating this process.

## Accessing and reviewing WebSpeed-related log file data

Fathom supports monitoring log files and their associated viewers for these WebSpeed resources:

- An individual WebSpeed broker.
- The agents associated with the broker.

Log files can store a tremendous amount of data. Therefore, monitoring and analyzing data collected within these files might help you to better determine performance expectations and examine trends related to brokers and agents.

This section presents information related to both types of WebSpeed log file monitors. However, only the procedures specific to a WebSpeed broker log file monitor and its associated viewer are presented. These same procedures will work with a WebSpeed agent log file monitor. For more general information about Fathom log file monitor features and functionality, see the [Resource Monitoring Guide](#).

---

**Note:** Log file monitors are not available for either remote WebSpeed brokers or their associated agents.

---

## Getting started with log files for WebSpeed resources

For each local WebSpeed broker that Fathom Management discovers, Fathom supports monitoring its two associated log file monitors. Fathom provides a log file resource monitor for the WebSpeed broker itself and another for its associated agents. Each of these log file monitors has its own log file monitoring capabilities.

WebSpeed log file resource monitors are not enabled until the WebSpeed broker for which the resource monitors were created is started. When the log file monitor first starts monitoring either a WebSpeed broker or agents, it always starts at the end of the log file.

### Naming conventions

Fathom prepends the broker's name to the name of the broker and agent log file monitors and viewers. For example, Fathom generates the following log file monitor and associated viewer names for a WebSpeed broker instance named **wsbroker1** and the container named **vesta**:

- **Broker-related log file names** — Displays **vesta.wsbroker1BrokerLogFileMonitor** and **vesta.wsbroker1 Broker Log File Contents**.
- **Agent-related log file names** — Displays **vesta.wsbroker1AgentLogFileMonitor** and **vesta.wsbroker1 AgentLog File Contents**.

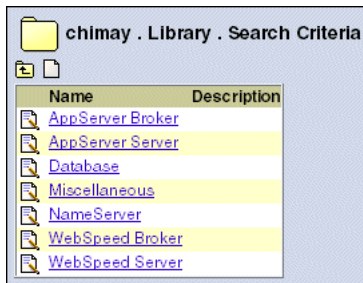
You cannot change these names.

## Characteristics of WebSpeed resource log file monitors

Data that you can capture and view using the WebSpeed resources log file monitors and viewers help you to:

- Ensure the integrity of these log files by monitoring files for errors and allowing you to define actions that trigger when errors occur.
- Use predefined WebSpeed-related search criteria, or create your own, to run against the data in these log files. Fathom predefines search criteria to support the broker's and agents' log file monitors.

Figure 3–5 shows an example of the **Search Criteria** subcategories, including the **WebSpeed Broker** and **WebSpeed Agent** links to the predefined search criteria. (Note that the **WebSpeed Server** link identified in Figure 3–5 is actually the **WebSpeed Agent** link.)



**Figure 3–5: WebSpeed-related search criteria**

You can create and maintain the search criteria for each of the WebSpeed resources in two locations:

- At the WebSpeed resource local file monitor instance level. The search text and type are not shareable at this level. See the “[Customizing a WebSpeed broker log file monitor](#)” section on page 3–25 for details.
- At the Fathom Component Library level under the appropriate WebSpeed subcategory. The search text and type are shareable at this level. See the “[Working with rule sets](#)” section on page 6–29 for details.

Specifically, the predefined search criteria provide:

- Detailed data about the recorded operations of a WebSpeed broker or agents.
- A means for you to extract the detailed data.



## WebSpeed log file monitor default values

Once a WebSpeed broker is enabled, Fathom creates log file monitors for any discovered brokers and their associated agents using several default values. You can modify only the default description. However, you have several options regarding the **Search Criteria** you can use for the log file monitor. See the “[Customizing a WebSpeed broker log file monitor](#)” section on page 3–25 for details.

The default values are as follows:

- The WebSpeed default log file monitor is disabled until the agent is first started.
- The **Bookmark** is set to **Last Line**, and it is unique.
- The **On First Poll** property is set to **Search From End**.

For detailed information about the Bookmark feature and **On First Poll** property as they relate to log file monitors in general, see the [Resource Monitoring Guide](#).

### File Resource Defaults page

Fathom also supports a polling interval default value for the WebSpeed broker log file monitor and the WebSpeed agent log file monitor.



**To display or update a polling interval default value:**

1. Click **Resources** on the menu bar.
2. Click **Resource Monitor Defaults** → **File Resource Defaults**.
3. Scroll down the **File Resource Defaults** page to display the **WebSpeed Broker Log File Monitor** and the **WebSpeed Agent Log File Monitor** entries.

You can revert back to the original Fathom-supplied default value set for the **Polling Interval** field at any time by clicking **Restore Fathom Defaults**.

## Reviewing predefined log file monitor search criteria

Each log file provides predefined search criteria that address common WebSpeed broker- or agents-related events. You can use these searches as defined, or copy and customize them.

---

**Note:** It is recommended that you do not edit or delete the predefined criteria.

---



**To review predefined log file monitor search criteria:**

1. Select **Library** from the menu bar.
2. Click the plus (+) icon next to **Search Criteria** in the list frame to expand this category.
3. Click either **WebSpeed Broker** or **WebSpeed Agent** in the list frame. A list of predefined search criteria related to the category that you selected appears in the detail frame. For example, the following screen shows the list of the **WebSpeed Server** default search criteria:

The screenshot shows a web application window titled 'chlmay . Library . Search Criteria . WebSpeed Server'. The window has a menu bar with 'Resources', 'Library', 'Reports', 'Jobs', 'Options', and 'Help'. Below the menu bar is a table with two columns: 'Name' and 'Description'. The table lists six search criteria, each with a small icon to its left.

Name	Description
<a href="#">Fatal_Memory_Error</a>	Catches the following error: Fatal memory error.
<a href="#">Fatal_Transport_Failure</a>	Catches the following error: Fatal transport failure (error_code).
<a href="#">Insufficient_Stack_Space</a>	Catches the following error: Insufficient stack space. Increase the -s startup parameter. (number of bytes) bytes are needed, only (number of bytes) are available.
<a href="#">Transport_Resources_Unavailable</a>	Catches the following error: Transport resources unavailable.
<a href="#">Unexpected_Transport_Error</a>	Catches the following error: Unexpected transport error received (error_code).
<a href="#">Unknown_Transport_Error</a>	Catches the following error: Unknown transport error received (error_code).

---

**Note:** You can also create your own search criteria to address a particular WebSpeed error for which you want to monitor a WebSpeed broker or agent. See the [“Customizing a WebSpeed broker log file monitor”](#) section on page 3–25 for details.

---

## Customizing a WebSpeed broker log file monitor

This procedure describes how to customize a WebSpeed broker log file monitor. Use these same general steps to customize a WebSpeed agent log file monitor.



### To customize a WebSpeed log file monitor:

1. Navigate to the **WebSpeed main** page specific to your broker, using the procedure detailed in the [“Accessing OpenEdge resource information”](#) section on page 2–12.
2. Click **Log File Monitor of Broker** on the **WebSpeed main** page. The log file monitor summary monitoring page for the WebSpeed broker you selected appears:

**Log File: chimay.wsbroker1BrokerLogFileMonitor**  
Broker Log File Monitor for: wsbroker1

Passed (3 Hours)  
Last Poll: May 20, 2:34 PM  
Last Failure: May 20, 11:24 AM  
Poll Count: 44 Fail Count: 6 (13.6%)

Edit Log File Viewer

Name	Poll	Alerts
Default Schedule Plan	5 mins	✓

**Monitoring plans**

Name	Status	Severity
WS_Broker_LogFileMonitor_RuleSet	---	---
NameServer.Consistency_Error	Passed	
NameServer.IOException	Passed	
NameServer.NOT_started	Passed	
NameServer.keepAlive_stopped	Passed	
Broker_Error_Mag_Received	Passed	
Cannot_Start_Server	Passed	
Connection_Refused	Passed	
Disconnecting_Client	Passed	
Disconnecting_Client_2	Passed	
Error_Loading_Prop_File	Passed	
Fatal_Error	Passed	
IOException_Client_Response	Passed	
IOException_Message_From_Server	Passed	
Invalid_Action_For_State	Passed	
Invalid_Admin_Response	Passed	
Invalid_State	Passed	
Invalid_State_2	Passed	
MsgFormatException	Passed	
No_Servers_Available	Passed	
ServerIPCException	Passed	
Server_Exec_Error	Passed	
Unimplemented_State	Passed	

Add Plan

3. Customize or view the contents of a WebSpeed broker log file monitor as follows:

To...	Select...
Add an existing monitoring plan to this resource monitor.	<b>Add Plan.</b>
Modify a specific plan.	<b>Edit</b> associated with that plan.
Change the description of this log file monitor.	<b>Edit</b> at the top of the page.
View the contents of the log file monitor.	<b>Log File Viewer</b> at the top of the page.

---

**Note:** Fathom prevents the assignment of schedules that share days or times that overlap. For example, if you have a **Default\_Schedule** set up for a resource monitor, you cannot set up an additional plan because the **Default\_Schedule** is defined for 7 days a week, 24 hours a day. You must modify or remove the **Default\_Schedule** to set up additional plans.

---

4. To add individual rules, click **Edit** within the monitoring plans section to display the edit page for the log file monitor.
  - a. Click **Add Rule** under the **Rules selected for this plan** section of the broker monitoring plan page. Proceed as follows:

To...	Then...
Use a WebSpeed broker rule already defined in the library	<p>Select:</p> <ul style="list-style-type: none"> <li>• <b>WebSpeed Broker</b> from the drop-down list associated with the <b>Choose Criteria Category</b>.</li> <li>• The appropriate value from each drop-down list associated <b>Choose Search Criteria</b>.</li> </ul>
Create a new WebSpeed broker rule	<p>Click <b>Create Criterion</b> to display the <b>Create Search Criterion</b> page. Complete this page as follows:</p> <ul style="list-style-type: none"> <li>• Enter values in the required fields: <b>Name</b> (identifies the name of the search criteria you are creating) and <b>Search Text</b> (identifies the information you are looking for in the log).</li> <li>• Review the default option <b>Use Existing Category</b>; it indicates that the new rule will be stored in an existing group. Select the <b>WebSpeed Broker</b> category from those that display in the drop-down list associated with the <b>Use Existing Category</b> option.</li> <li>• Click <b>Save</b>. The <b>Rules Properties</b> section of the <b>Log File Rule</b> page redisplay. The values you defined and selected to create a rule on the <b>Create Search Criterion</b> page are now available on the <b>Log File Rule</b> page. The <b>Choose Criteria Category</b> drop-down list displays the category in which you elected to store the new rule. The <b>Choose Search Category</b> drop-down list displays the name you entered in the <b>Name</b> field on the <b>Create Search Criterion</b> page.</li> </ul>

- b. Select the appropriate values from the **Severity** and **On Alert Action Perform** fields to complete the alert severity and action definition, respectively, that you want to associate with this rule.
  - c. To add another individual rule, repeat [Step a](#) and [Step b](#).
- 5. Click **Select Rule Sets** to create a new log file rule or choose from existing rule sets to add to the monitoring plan.

If you choose **Select Rule Sets**, you can choose from a list of predefined rule sets to add to the monitoring plan.
- 6. Click the detail page for the parent icon (the file folder with the up arrow on it) to redisplay this WebSpeed broker's monitoring plan page with the rules section updated with the new rules.

For more information about editing search criteria for rules, see the appropriate sections of the [Resource Monitoring Guide](#).

---

**Note:** You can copy the default WebSpeed log file rule set, but you cannot delete it.

---

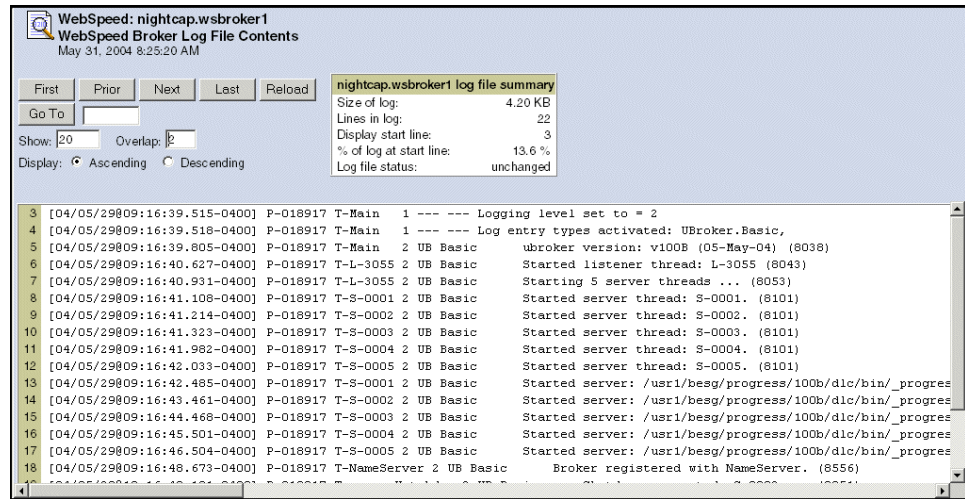
## Using the WebSpeed log file viewers

To view the contents of each WebSpeed log file, access the viewer associated with each individual log file.

The log file viewer allows you to examine the contents of a WebSpeed-related log file through an HTML interface. You can access these log file viewers from two locations:

- Click the link in the **Command and control** section of the WebSpeed main page; click **Log File Viewer of Broker** to display the broker's file contents or click **Log File Viewer of Agents** to display the agents' file contents.
- Click the **Log File Viewer** button that appears at the top of the log file monitor summary monitoring page.

Figure 3–6 shows an example of the WebSpeed broker log file viewer with the contents of a WebSpeed broker log file displayed.



**Figure 3–6: WebSpeed broker log file viewer example**

The following information will help you use the WebSpeed log file viewer:

- Use the **Show** field to control how many WebSpeed log file entries display at one time. The number entered into the **Show** field cannot be less than 10.
- Use the **Overlap** field to control how many entries are repeated from screen to screen.

---

**Note:** The value in the **Overlap** field cannot be more than the number in the **Show** field minus one. For example, if you show 30 entries, you can overlap only 29 or fewer of them.

---

- Click **Reload** after changing the values in the **Show** field or **Overlap** field. Note that Fathom will prompt you to click **Reload**. The warning message that reads *changed, reloaded needed*, flashes in the **File log status** field in the **Log file summary** section of this page.

If you do not reload, the viewer displays the previous values.

- Click **Go To** to control which numbered entry in the log file the viewer begins its display with. For example, a value of 10 entered into the **Go To** field will begin the display from the tenth log file entry.

---

**Note:** You must click **Go To** after entering a value in the **Go To** field, or the viewer will not update its display.

---

- The default display of entries is in ascending order; choose **Descending** to change the display. Note that the **Show** field dictates the number of entries shown, whether they display in ascending or descending order.
- Click **First** to display the first  $x$  entries, where  $x$  is the value in the **Show** field.
- Click **Prior** to display the previous  $x$  entries, where  $x$  is the value in the **Show** field.
- Click **Next** to display the next  $x$  entries, where  $x$  is the value in the **Show** field.
- Click **Last** to display the last  $x$  entries, where  $x$  is the value in the **Show** field.
- To view additional log file entries without changing your current starting log file entry, leave the **Go To** field blank, change the value in the **Show** field, and click **Reload**.

## Refreshing log file data

Periodically refresh log file data. Select the **Refresh** page icon from the status bar for either the list or detail frame to repaint an existing page. You can also set a default value that Fathom uses to automatically refresh the management console.



To set a default value that Fathom uses to automatically refresh the management console, select **Options** → **User Preferences** → **Automatically Refresh Pages**.

Refresh data to avoid the following situations:

- Fathom considers a viewer that has been inactive for more than four hours “stale.” Once a viewer becomes stale, Fathom releases ninety-five percent of any memory it holds. If you try to use a stale viewer, Fathom automatically reloads the file. Because additional resource activity might have occurred during the viewer’s inactivity, the reloaded log file view might not match the previous log file view of that resource.



- Fathom considers a viewer that has been inactive for forty-eight hours “dead.” Once a viewer dies, Fathom releases all of its memory. To return to the log file displayed in a dead view, you need to renavigate to it, even if you pinned up the view or saved a link to it before the viewer died.

## Examining WebSpeed-related Operational views

The **WebSpeed main** page provides an **Operational views** section that allows you to access and review data related to the performance of a:

- Specific WebSpeed broker.
- Pool of agents associated with a specific broker.

Data for both the WebSpeed broker and the broker’s agent pool can display in text and graph formats.

---

**Note:** The graphs associated with the WebSpeed **Operational views** display only when the **Broker statistics available** field on the **Broker Control** page displays a **True** status. See the “[Data collection details](#)” section on page 3–9 for details.

---

Figure 3–7 shows the **Operational views** section of the **WebSpeed main** page.



**Figure 3–7: WebSpeed Operational views section example**

The following sections present detailed information about how to access and review each of these views.

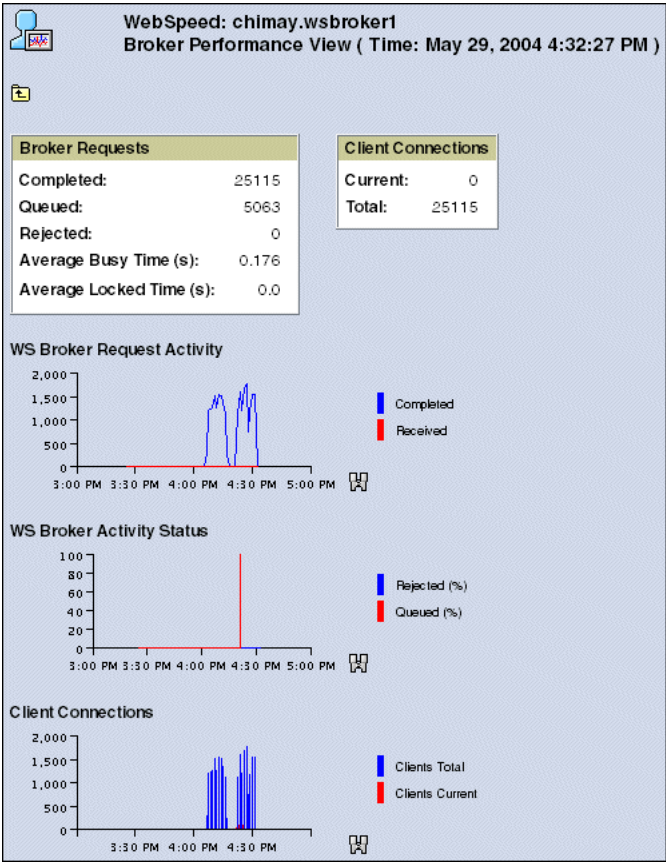
## Accessing and reviewing the Broker Performance View

The WebSpeed **Operational views** section allows you to display information about the WebSpeed broker’s performance and the state of the broker’s associated agents. Review this data frequently; these details can help you make informed decisions about your use of the broker and agent pool controls as discussed earlier in this chapter.



To display and review this information:

- 1. Display the **WebSpeed main** page for the WebSpeed broker instance that you want to review. See the “[Accessing OpenEdge resource information](#)” section on page 2–12 for the detailed procedure.
- 2. Click **Broker Performance View** in the **Operational Views** section. Fathom can display the **WebSpeed Broker Performance View** page (which can be comprised of data summary sections and graphs as shown):



Data summary sections

The summarized display-only text data on this page is comprised of two sections:

- **Broker Requests** — Details about the broker’s connection workload as identified in [Table 3–7](#).

**Table 3–7: Broker connection workload details**

This field...	Identifies the...
<b>Completed</b>	Number of broker requests fulfilled.
<b>Queued</b>	Number of broker requests to be processed.
<b>Rejected</b>	Number of broker requests that could not be processed.
<b>Average Busy Time (ms)</b>	Average amount of time that the broker is busy servicing requests (expressed in milliseconds).
<b>Average Locked Time (ms)</b>	Average amount of time that the broker is locked (expressed in milliseconds).

- **Client connections** — Identifies the number of client connections that the broker is currently handling. Also, the total number of client connections this broker has processed since the broker started.

### Graphs presentation section

The graphs presentation section of the **Broker Performance View** contains three graphs: **WS Broker Request**, **WS Broker Activity Status**, and **Client Connections**. Provided that the conditions for data collection are set and the **Trend** option is selected, the graphically displayed data associated with each of these graphs complements the summarized text data that displays on the **WebSpeed Broker Performance View** page. See the [“Data collection details”](#) section on page 3–9 for details.

One display format for these graphs, as shown on the **Broker Performance View** page, is a line graph. This format measures how a particular broker-related activity has changed over a period of time. [Table 3–8](#) identifies and briefly describes each of these graphs.

**Table 3–8:      WebSpeed Broker performance-related graphs**

This WebSpeed Broker graph...	Displays two lines...
<b>WS Broker Request</b>	Of broker-related performance data over a specified time period. The blue line identifies the number of requests that the broker has completed; the red line identifies the number of requests that this same broker has received in this time period.
<b>WS Broker Activity Status</b>	Of broker-related performance data over a specified time period. The blue line identifies the percent of requests that the broker has rejected, up to and including the last poll Fathom has completed for this broker resource. The red line identifies the percent of requests in the queue waiting for the broker, up to and including the last poll completed.
<b>Client Connections</b>	<p>Of client connections related to this broker over a specified time period.</p> <p>The blue line identifies the client total number of connections requested; the red line identifies the number of clients currently connected to this broker.</p> <p><b>Note:</b> It is possible for this graph to accurately show that the number of current connections is higher than the total number of connections. The <b>Clients Total</b> reflects only new connections over the specified time period. In contrast, the <b>Clients Current</b> reflects <i>all</i> current connections, both newly connected and those that might still be connected from a previous polling period, in place when the graph is displayed.</p>

See the “[Changing OpenEdge pinup graphical displays](#)” section on page 2–25 for details about changing the data appearance of graphs.

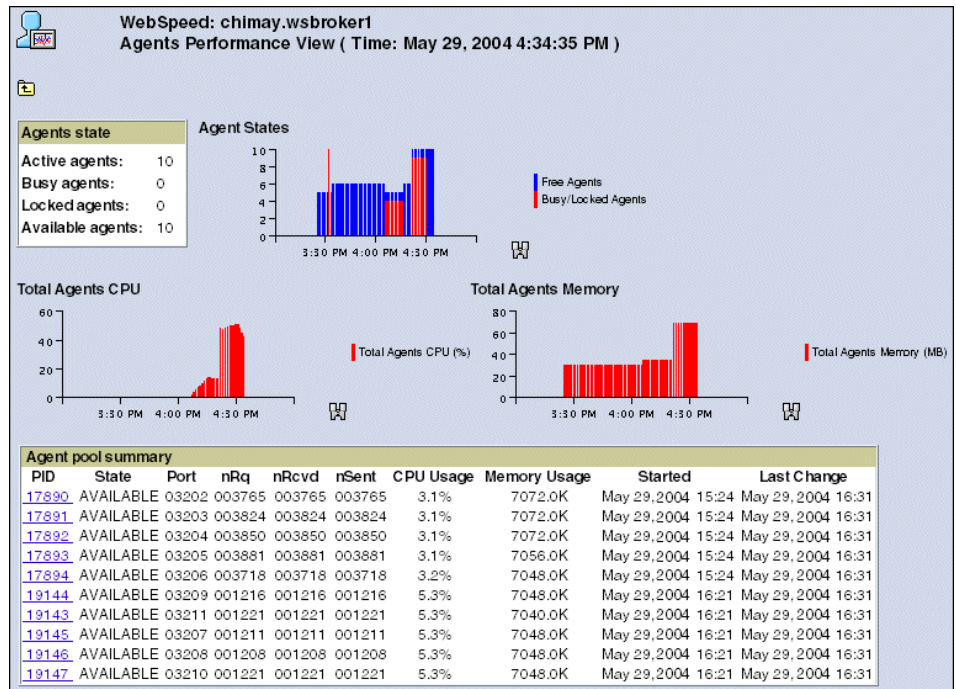
## Accessing and reviewing the Agent Performance View

The WebSpeed **Operational views** section allows you to display information about agents' status.



To display and review agents' status information:

1. Display the **WebSpeed main** page for the WebSpeed resource that you want to review. See the “[Accessing OpenEdge resource information](#)” section on page 2–12 for the detailed procedure.
2. Click **Agent Performance View** in the **Operational views** section to display the **WebSpeed Agents Performance View** page which is comprised of data summary sections and graphs as shown in this sample page:



## Data summary sections

The summarized display-only text data on this page is comprised of two sections:

- **Agents state** — Displays the four possible states of the agents that are currently associated with this WebSpeed broker: **Active**, **Busy**, **Locked**, and **Available**. See [Table 3–5](#) for a definition of each of these states.
- **Agent pool summary** — Displays detailed data about each individual agent in the WebSpeed agent pool that is associated with a specific WebSpeed broker. See [Table 3–6](#) for a description of each field that displays in the **Agent pool summary** section. You also have access to additional data about a specific agent and a control that allows you to kill a process. See the “[Killing a WebSpeed agent process](#)” section on page 3–19 for detailed steps.

## Graphs presentation section

The graphs presentation section of the **Agents Performance View** contains three graphs: **Agent States**, **Total Agents CPU**, and **Total Agents Memory**. Provided that the options for data collection are set and the **Trend** option is selected, the graphically displayed data complements the summarized text data that displays on the **WebSpeed Agents Performance View** page. See the “[Data collection details](#)” section on page 3–9 for details.

One display format for these graphs, as previously shown on the **Broker Performance View** page, is a line graph. This format measures how a particular broker-related activity has changed over a period of time. [Table 3–9](#) identifies and briefly describes each of these graphs.

**Table 3–9: WebSpeed agents performance-related graphs**

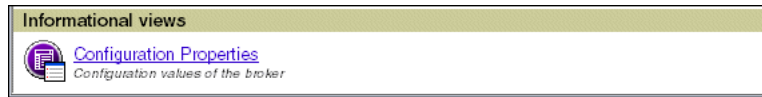
This WebSpeed agents graph...	Displays...
<b>Agent States</b>	Two lines of agents-related performance data over a specified time period. The blue line identifies the number of free agents; the red line identifies the number of busy/locked agents during this same time period.
<b>Total Agents CPU</b>	One line of agents-related performance data over a specified time period. This single data line indicates the total percent of the agents’ CPU usage.
<b>Total Agents Memory</b>	One line of agents-related performance data over a specified time period. This single data line indicates the total percent of the agents’ Memory consumption.

See the “[Changing OpenEdge pinup graphical displays](#)” section on page 2–25 for details about changing the data appearance of graphs.

## Examining WebSpeed-related Informational views

The **WebSpeed main** page provides an **Informational views** section that allows you to access and review data related to the WebSpeed broker’s configuration properties. The values that display originate from the `ubroker.properties` file.

[Figure 3–8](#) shows an example of the **Informational views** section of the **WebSpeed main** page.



**Figure 3–8: WebSpeed Informational views section example**

The following information presents detailed information about how to access and review detailed information associated with the **Configuration Properties** view.



### To display and review **Configuration Properties** view information:

1. Display the **WebSpeed main** page for the WebSpeed broker instance that you want to review. See the “[Accessing OpenEdge resource information](#)” section on page 2–12 for the detailed procedure.

2. Click **Configuration Properties** in the **Informational views** section to display the **Configuration Properties** page:



3. Review the values that display. Note that the properties list is quite long. Scroll to see more of the properties and their associated values.



---

## Managing AppServer Data

---

This chapter presents Fathom Management features and functionality related to the Progress AppServer.

Topics in this chapter include:

- [Overview](#)
- [Reviewing AppServer broker status](#)
- [Modifying AppServer control settings](#)
- [Accessing and reviewing AppServer-related log file data](#)
- [Using the AppServer log file viewers](#)
- [Examining AppServer-related Operational views](#)
- [Examining AppServer-related Informational views](#)

# Overview

Fathom supports a variety of tasks that you can perform to manage an AppServer, including:

- Reviewing your current operating status and associated details.
- Modifying broker-related control settings, such as starting and stopping a broker, and adding or trimming servers.
- Accessing and viewing broker- and server-specific data collected through log files.
- Monitoring and managing AppServer brokers using monitoring plans and rules.
- Generating threshold values for rules using the Configuration Advisor.
- Working with OpenEdge resource-related data that is available through broker- and server-specific information and operational views. AppServer information views provide data in both text and graph formats.

You must have appropriate Fathom role authorization to perform several of these tasks. See the [“Role authorization and OpenEdge tasks”](#) section on page 1–7 for details.

## Reviewing AppServer broker status

The **Status** section of the **AppServer main** page summarizes current operational details about the AppServer broker. [Figure 4–1](#) shows an example of the Status section. Note that the broker’s name displays in the page’s title area.



**Figure 4–1:** AppServer Status section sample

Table 4–1 describes each of these AppServer broker details.

**Table 4–1: AppServer status details**

This field...	Displays the...
<b>Host</b>	Host machine’s name.
<b>Broker</b>	<p>The running status of the broker. Possible values are:</p> <ul style="list-style-type: none"> <li>• <b>ACTIVE</b> (The broker is currently running.)</li> <li>• <b>Not Running</b> (The broker is not currently running.)</li> </ul> <p>The broker can also report <b>Starting</b> and <b>Shutting Down</b> values; however, depending on the speed of the machine on which your management console is running, you may not see these intermediary states.</p>
<b>Operating mode</b>	<p>Operating mode of the broker. This mode determines how client requests are dispatched to individual Application Server processes running on the AppServer instance.</p> <p>One of four possible modes can be reported: <b>Stateless</b>, <b>State-free</b>, <b>State-aware</b>, or <b>State-reset</b>.</p>
<b>Broker statistics available</b>	<p>Status of the broker as it relates to data collection. The possible states are <b>True</b> or <b>False</b>.</p> <p>See the “<a href="#">Data collection details</a>” section on page 4–9 for more information about data collection.</p>
<b>Servers available</b>	<p>Number of AppServers running and available to fulfill a connection request from a client to an AppServer through this broker when the broker’s status is <b>ACTIVE</b>.</p> <p>This value can change frequently, reporting the real-time changes in number of servers available.</p>
<b>Should register with NameServer?</b>	Status of <b>True</b> or <b>False</b> to indicate whether the broker resource is registered with a NameServer.

These points relate to the fields highlighted in [Table 4–1](#):

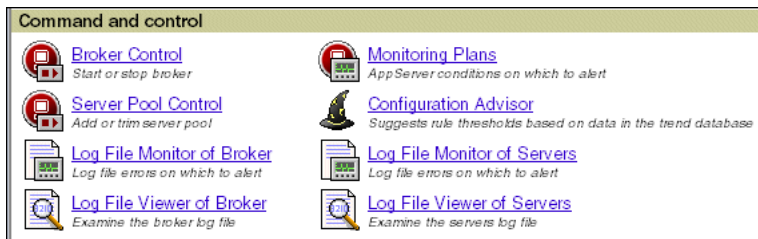
- Broker-related changes that you can make, using either the **Broker Control** or **Server Pool Control** options in the **Command and control** section of the **AppServer main** page, can affect the broker and server values that display in this **Status** section.
- The values that display in the AppServer **Status** section are obtained either from the `ubroker.properties` file or the current, real-time status of the broker (if it is running).

## Modifying AppServer control settings

The **Command and control** section of the **AppServer main** page for an AppServer broker allows you to:

- Start and stop and AppServer broker, and change its associated property settings.
- Add or trim the pool of available AppServers associated with the broker.
- Obtain and review AppServer-related data collected through broker- and server-specific log files associated with this instance.
- Monitor and manage AppServer brokers using monitoring plans and rules, including the option to use Configuration Advisor-recommended settings.

[Figure 4–2](#) shows an example of the **Command and control** section of the **AppServer main** page.



**Figure 4–2: Command and control section example**

---

**Note:** The information in this section presents functional descriptions and procedural details related to the **Broker Control** and **Server Pool Control** pages.

---

Table 4–2 identifies where you can find information about other functionality related to the AppServer **Command and control** section.

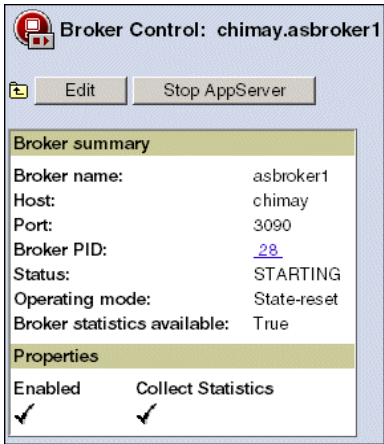
**Table 4–2: Additional AppServer information**

For AppServer-related details about...	See the...
Broker and server pool log file monitors and viewers, respectively	<a href="#">“Accessing and reviewing AppServer-related log file data”</a> section on page 4–20.
Broker monitoring plans and rules	<a href="#">Chapter 6, “Monitoring Plans and Rules for OpenEdge Server Resources.”</a>
Broker rule sets	<a href="#">“Working with rule sets”</a> section on page 6–29.

## Broker Control page content

The **Broker Control** page summarizes details about a specific AppServer broker resource. From this page, you can start and stop an AppServer broker, and change some broker-related properties, as your processing needs warrant.

Figure 4–3 shows an example of the **Broker Control** page.



**Figure 4–3: Broker Control page example**

The following details describe the two sections of the **Broker Control** page.

---

**Note:** The values associated with the **Collect Statistics** property and the **Broker statistics available** field are interdependent. See the procedure in the [“Data collection details”](#) section on page 4–9 for details.

---

### Broker summary section

The **Broker summary** section displays read-only values for these fields: the broker name, its host machine’s name, associated port number and process identification number (PID), the broker’s current status, operating mode, and whether the broker is currently set to collect broker-related statistical data.

Note the following additional details about these fields:

- The values shown in the **Broker name**, **Host** (machine name), **Port** (number), and **Operating mode** fields display values as they are defined in the `ubroker.properties` file.
- The **Broker PID** and **Status** fields reflect real-time values based on the broker’s current status. The **Broker PID** is also a link to more broker process details. See the [“Viewing broker process details”](#) section on page 4–10 for more details.
- The **Broker statistics available** field also reflects a current, real-time value. However, the value displayed in this field depends on additional factors. See the [“Data collection details”](#) section on page 4–9 for more details.

### Properties section

The **Properties** section displays the status of two user-defined, broker-related properties, **Enabled** and **Collect Statistics**:

- The **Enabled** option indicates that this broker resource recognizes a monitoring plan and its associated rules when the broker resource is active.

During the discovery process, all AppServer brokers that Fathom discovers and lists in the list frame under the AppServer category are *enabled* by default. Once a broker is enabled, Fathom uses the Fathom-supplied default values to establish a monitoring plan and rules. (You can customize the plan and rules at any time.)

- The **Collect Statistics** option enables data collection to occur in the AppServer broker. Fathom uses this data to identify the broker’s performance. If you do not select the **Collect Statistics** option for a specific broker (that is, True status), Fathom displays only non-statistical data such as **Status** and **PID** (pid number) on the various AppServer broker pages; polled rules are not evaluated and data is not trended.

The **Collect Statistics** value plays a central role in data collection. See the “[Data collection details](#)” section on page 4–9 for more details.

A check mark associated with a property indicates that the property is set.

**Note:** To set the **Broker statistics available** option to a **True** status for a specific broker, you must enable the **Collect Statistics** option. See the procedure in the “[Data collection details](#)” section on page 4–9 for details.

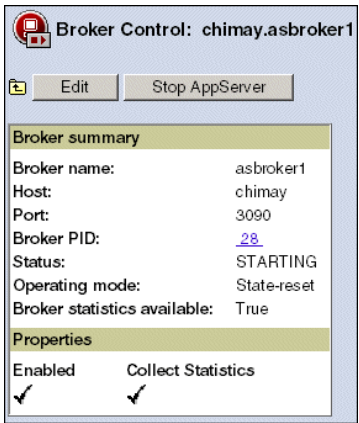
## Changing AppServer Broker controls

This section describes how to change AppServer broker controls.



**To start or stop the AppServer Broker and to change its property settings:**

1. Display the **AppServer main** page for the broker you want to start. See the “[Accessing OpenEdge resource information](#)” section on page 2–12 for the detailed steps.
2. Click **Broker Control** in the **Command and control** section to display the **Broker Control** page as shown:



This table highlights the controls that you can change:

To...	Then...
<p>Change the current setting of the <b>Enabled</b> property.</p> <p>(A checkmark displays to indicate that the <b>Enabled</b> property is set. To clear this option, click the checkmark in the box associated in the option; the checkmark is deleted to indicate that the option is no longer set.)</p>	<p>Click <b>Edit</b>. Then select or deselect the <b>Enabled</b> property to add or remove the checkmark.</p> <p>You must also restart the AppServer broker so that the property change is recognized.</p>
<p>Change the current setting of the <b>Broker statistics available</b> property displayed in the <b>Broker Summary</b> section of the <b>Broker Control</b> page.</p>	<p>See the “<a href="#">Data collection details</a>” section on page 4–9.</p>
<p>Toggle between stopping and starting the AppServer broker, depending on the current value displayed in the <b>Status</b> field.</p> <p>For example, if the broker status currently displays <b>ACTIVE</b>, the button label will read <b>Stop AppServer</b>. You can click this button to stop the AppServer broker; Fathom stops this broker and updates the value in the <b>Status</b> field to display <b>Not Running</b>.</p>	<p>Click <b>Stop AppServer</b> or <b>Start AppServer</b>.</p>
<p>Exit this page without changing any values.</p> <p>The <b>AppServer main</b> page redisplay in the management console.</p>	<p>Click <b>Cancel</b>.</p>



## Data collection details

Data collection ensures that broker-related performance statistics can be trended to the FathomTrendDatabase. Options and conditions available on the **Broker Control** page and the AppServer broker resource monitoring plan must be fulfilled to successfully implement data collection.

On the **Broker Control** page, these conditions include:

- Selecting the **Collect Statistics** check box to enter a check mark.
- Starting, or stopping and restarting the AppServer broker; you must explicitly perform this step on the **Broker Control** page to effect this change.
- The value **TRUE** displayed in the **Broker statistics available** field. (Fathom automatically updates this field when it detects that the **Collect Statistics** option is enabled after you have started, or stopped and re-started, the AppServer broker.)

On the AppServer broker resource monitoring plan, you must also check the **Trend Performance Data** option.

---

**Note:** You are not required to use trending with the data collection activity. However, without the **Trend Performance Data** option selected, you cannot trend data. Data trended to the FathomTrendDatabase is required for AppServer-related rule evaluation, graphical displays, and report generation.

---

For information about the **Trend Performance Data** option and monitoring plans for AppServer broker resources, see [Chapter 6, “Monitoring Plans and Rules for OpenEdge Server Resources.”](#)

---

**Note:** Using data collection might cause the AppServer broker to exhibit some level of performance degradation, memory degradation, or both.

---



### To set the options to perform data collection in an AppServer broker:

1. Check the current status of the **Collect Statistics** field in the **Properties** section of the **Broker Control** page; it must be checked to indicate that this property is set.

If the **Collect Statistics** field is not checked, then click **Edit**. In the **Collect Statistics** field, click in the check box; a check mark appears. Click **Save**.

2. Stop and restart the AppServer broker you want to update.

---

**Caution:** You cannot interactively change a running process. Therefore, you must restart the AppServer broker after you change the setting of the **Collect Statistics** field.

---

The **Broker statistics available** field in the **Broker summary** section will display **TRUE** if the broker re-started successfully. The **TRUE** value indicates that you have successfully set data collection and that broker statistical data is now available to be stored in the FathomTrendDatabase.

3. Check the current setting of the **Trend** field in the **Properties** section of the **Broker Control** page; it must be checked to indicate that this property is set. See the [“Default polling and trend values”](#) section on page 6–5 and the [“Default monitoring plan details”](#) section on page 6–8 for details about this field.

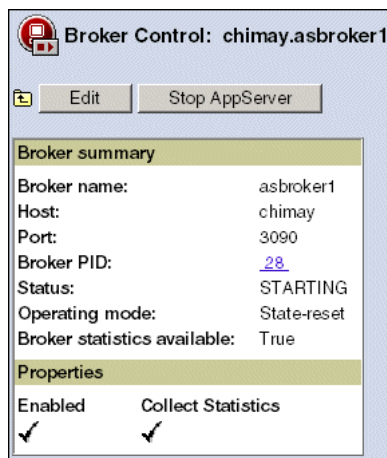
## Viewing broker process details

You can also access real-time details and statistics that provide you with snapshot information about an individual broker at the point you access this information from the **Broker Control** page. Review this information to help you determine a broker’s performance.

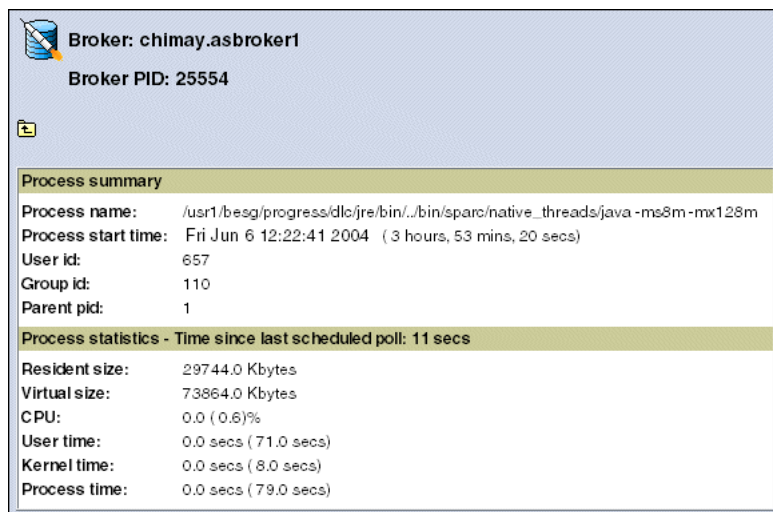


### To access broker processing details:

1. Display the **AppServer main** page for the AppServer broker you want to review. See the [“Accessing OpenEdge resource information”](#) section on page 2–12 for the detailed steps.
2. Click **Broker Control** in the **Command and control** section to display the **Broker Control** page as shown:



3. Click the unique PID number associated with the **Broker PID** field to display a **Broker process** page. This page contains summary and real-time statistics about the broker as shown:



The two sections that comprise the broker process page present relevant information about the AppServer broker and its current operations:

- The **Process summary** section identifies the **Process name** and **Process start time**. **User id** and **Group id** values display when Unix-based data displays. The **Parent pid** identifies the process identifier number associated with the process that spawned this current process.
- The **Process statistics** section presents details about the broker’s real-time operational status. Values presented without parentheses identify that the processing time determined since the last scheduled polling interval, as noted, has occurred. Values presented within parentheses have been calculated based on information obtained since the start of the process.

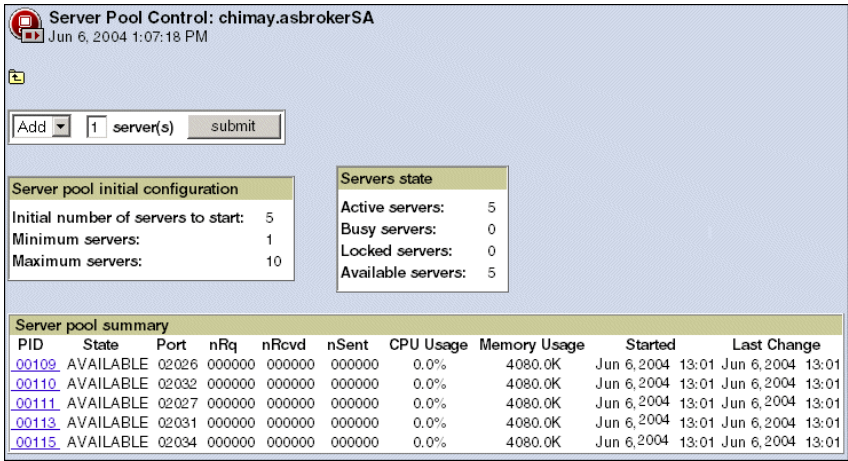
Table 4–3 identifies and describes the attributes presented in the **Process statistics** section.

**Table 4–3: Process statistics section real-time operational data**

This field...	Identifies the...
<b>Resident size</b>	Physical size of the process as defined by host system.
<b>Virtual size</b>	Virtual size of the process as defined by host system.
<b>CPU%</b>	Percentage of time spent using the CPU in either the user or kernel mode since the last scheduled poll.
<b>User time</b>	Amount of CPU time spent in the user mode since the last scheduled poll.
<b>Kernel time</b>	Amount of CPU time spent in the kernel mode since the last scheduled poll.
<b>Process time</b>	Sum of the values that display in the <b>User time</b> and <b>Kernel time</b> fields.
<b>Weighted CPU</b>	Percentage of time spent using the CPU in either the user or kernel mode since the last scheduled poll divided by the number of CPU processors on the system.  This value displays only when there is more than one CPU process on the system where the process is running.

## Server Pool Control page content

The **Server Pool Control** page, as shown in [Figure 4-4](#), displays data relevant to your current AppServer workload, and allows you add or reduce the number of AppServers currently running.



**Figure 4-4: Server Pool Control page example**

For example, use this page to add agents when agent requests are high; you can add agents to the maximum number of agents that your license recognizes. Also, use this page to reduce the agent count during a lag in agent requests. Using the trim feature, you can reduce agents down to the **minimum agents** property setting.

The **Server Pool Control** page is comprised of these sections:

- An add/trim selection control that you use to specify which activity you want to perform. When you initiate a manual trim request, Fathom determines which server(s) to actually remove. See the [“Adding or trimming AppServers”](#) section on page 4-17 for detailed steps.
- Three distinct, server-related data summary tables that allow you to review relevant AppServer-pool specific data quickly:
  - **Server pool initial configuration.**
  - **Servers state.**
  - **Server pool summary.**

The changes that you make through add/trim activities can affect the data displayed in these summary tables. The **Server pool summary** also allows you to kill a specific server process. See the [“Killing an AppServer process”](#) section on page 4–18 for the detailed steps.

The following information describes each section of the **Server Pool Control** page. For the detailed procedures on how to add or trim servers, see the [“Adding or trimming AppServers”](#) section on page 4–17 for more information.

### Server pool data initial configuration section

The **Server pool initial configuration** section identifies AppServer broker configuration properties set in the `ubroker.properties` file. These values display in a read-only mode.

[Table 4–4](#) identifies and describes each field that displays in the **Server pool data initial configuration** section.

**Table 4–4: Server pool data initial configuration field definitions**

This displayed property field...	Identifies...
<b>Initial number of servers to start</b>	The value Fathom references when the AppServer broker starts AppServers.
<b>Minimum servers</b>	<p>The minimum number of AppServers that must be simultaneously running before the AppServer broker will start additional servers. The broker strives to maintain this specified minimum; if at any time the number of servers falls below the specified minimum, the broker will automatically start any additional servers needed to maintain this minimum.</p> <p>If you set a trim value that would require Fathom to trim the number of servers below the number specified for this field, Fathom displays a message.</p>
<b>Maximum servers</b>	The maximum number of AppServer processes that can be running simultaneously. Fathom will not fulfill add requests you initiate that exceed this specified maximum. Fathom will display a message to state this condition so that you can re-consider your request and, if necessary, initiate a new request.

## Servers state section

The **Servers state** section provides a snapshot of the total number of AppServers currently associated with a specific server state. The state details related to agents and the number of agents reported reflect real-time data. This data can fluctuate due to changes in the AppServers' workflow and changes you initiate using the add and trim option.

[Table 4–5](#) identifies and describes each field presented in the **Serves state** section.

**Table 4–5: Servers state field definitions**

This server state...	Identifies the number of AppServers currently...
<b>Active</b>	Running.
<b>Busy</b>	Serving 4GL client requests.
<b>Locked</b>	Servicing a bound connection. (This state applies to a stateless AppServer.)
<b>Available</b>	Available to handle broker requests.

## Server pool summary section and the kill process option

The **Server pool summary** section provides:

- Detailed data about each individual server in the AppServer pool associated with a specific AppServer broker. [Table 4–6](#) identifies and describes each field displayed in the **Server pool summary** section.
- Access to:
  - More data about a specific agent.
  - A control to terminate, or kill, the agent process.

Use the **PID** field to access these features. [Table 4–6](#) provides more information about **PID**.

**Table 4–6: Server pool summary field definitions**

<b>This field identifies...</b>	<b>Identifies the...</b>
<b>PID</b>	Process identifier for this Appserver. Click on the specific PID number to display a detail page that provides specific information about this server process and, as necessary, kill the process. See the <a href="#">“Killing an AppServer process”</a> section on page 4–18 for more information.
<b>State</b>	Current execution state of the AppServer process.
<b>Port</b>	TCP/IP port number that the AppServer process uses.
<b>nRq (Number of Requests)</b>	Number of messages sent to the AppServer process.
<b>nRcvd</b>	Number of messages received by the AppServer process.
<b>nSent</b>	Number of requests sent by the AppServer process.
<b>CPU Usage</b>	Percentage of CPU user and system time consumed by a process.
<b>Memory Usage</b>	Amount of virtual memory (in Kbytes) consumed by a process.
<b>Started</b>	Time stamp that indicates when the AppServer process started. If the broker is restarted for any reason, the <b>PID</b> and the <b>Last Changed</b> value might change.
<b>Last Changed</b>	Time stamp that indicates when the AppServer process last changed execution state




## Adding or trimming AppServers

This section describes how to add and trim AppServers.



To initiate an AppSever add and trim request:

1. Display the **AppServer main** page for the broker you want to start. See the [“Accessing OpenEdge resource information”](#) section on page 2–12 for the detailed steps.
2. Click **Server Pool Control** in the **Command and control** section to display the **Server Pool Control** page as shown:

 **Server Pool Control: chimay.asbrokerSA**  
Jun 6, 2004 1:07:18 PM

Add

1 server(s)

submit

Server pool initial configuration

Initial number of servers to start: 5

Minimum servers: 1

Maximum servers: 10

Servers state

Active servers: 5

Busy servers: 0

Locked servers: 0

Available servers: 5

Server pool summary

PID	State	Port	nRq	nRcvd	nSent	CPU Usage	Memory Usage	Started	Last Change
<a href="#">00109</a>	AVAILABLE	02026	000000	000000	000000	0.0%	4080.0K	Jun 6, 2004 13:01	Jun 6, 2004 13:01
<a href="#">00110</a>	AVAILABLE	02032	000000	000000	000000	0.0%	4080.0K	Jun 6, 2004 13:01	Jun 6, 2004 13:01
<a href="#">00111</a>	AVAILABLE	02027	000000	000000	000000	0.0%	4080.0K	Jun 6, 2004 13:01	Jun 6, 2004 13:01
<a href="#">00113</a>	AVAILABLE	02031	000000	000000	000000	0.0%	4080.0K	Jun 6, 2004 13:01	Jun 6, 2004 13:01
<a href="#">00115</a>	AVAILABLE	02034	000000	000000	000000	0.0%	4080.0K	Jun 6, 2004 13:01	Jun 6, 2004 13:01

3. From the drop-down list box, select **Add** or **Trim**.
4. In the **server(s)** field, enter the number of servers you want to add or trim. The value you enter must be a positive integer.

When you initiate an add or trim request, Fathom consults two sets of initial configuration to determine if and how it can honor either request type:

- The number of AppServers for which you are licensed.
- The broker property configuration settings stored in the `ubroker.properties` file.

See the [“Server pool data initial configuration section”](#) section on page 4–14 for details about these initial configuration values.

5. Select **Submit**. Depending on the changes you make and Fathom's capability to implement them, you might notice changes to the numeric values displayed in the **Servers state** table. See the "[Servers state section](#)" section on page 4–15 for more information.

---

**Note:** Any time you either add or trim AppServers, it is recommended that you refresh the management console to ensure that you are not viewing stale data.

---

## Killing an AppServer process

You might want to manually terminate an agent process when:

- An agent process hangs.
- You determine from the available data that an agent process is a runaway process.

The specific **PID** on the **Agent pool summary** section of the **Server Pool Control** page allows you to access the page to kill the offending agent's process.

---


**Note:** Fathom references the specific **PID** and its associated date and time start details to be sure of a process' identify before it attempts to kill a process.


---



### To initiate a kill process:

1. Click **PID** associated with the server process you want to terminate. The specific AppServer process page displays as shown:


**Broker: chimay.asbrokerSA**  
**Process: 00113**


**Kill**

Process summary	
Process name:	/usr1/besg/progress/dlc/bin/_proapsv -ubpid 48 -Ms 1 -logname asbrokerSA -logfi
Process start time:	Fri Jun 6 13:01:28 2004 ( 7 mins, 34 secs)
User id:	657
Group id:	110
Parent pid:	48
Process statistics - Time since last scheduled poll: 7 mins, 34 secs	
Resident size:	4080.0 Kbytes
Virtual size:	9040.0 Kbytes
CPU:	0.0 ( 0.0)%
User time:	0.0 secs ( 0.0 secs)
Kernel time:	0.0 secs ( 0.0 secs)
Process time:	0.0 secs ( 0.0 secs)

Note that the two sections on this page present relevant, summary information about this AppServer agent and its current operational status. See the [“Viewing broker process details”](#) section on page 4–10 for details about this data.

2. Click **Kill** to terminate this process. Fathom will prompt you once again to ensure that you want to terminate this process. Click **OK**.

Fathom displays a final status page that identifies the status of your kill request. Fathom displays one of the following messages:

- **Process number xxxxx has been terminated.**

This message indicates that the process was successfully killed. The PID number previously associated with this process is now available for the operating system to re-assign.

- **Process number xxxxx cannot be killed at this time.**

This message indicates that the process could not be killed. In very rare instances, it's possible that you will not be successful in an attempt to kill a process. You can retry the kill process procedure; however, it is possible that the process will persist for any number of unknown reasons.

- **Process number xxxxx has been reused.**

Fathom has determined that the process PID number and associated time and date stamp do not match the values that the operating system has stored for this same process. Consequently, when you click **Kill**, the process cannot be destroyed.

3. Click **Cancel** at the top of the page to exit this page without terminating this process.

## Accessing and reviewing AppServer-related log file data

Fathom supports log file monitors and associated, dedicated viewers for the following AppServer resources:

- An individual AppServer broker.
- The AppServers associated with the broker.

Log files can store a tremendous amount of data. Therefore, monitoring and analyzing data collected within these files might help you to better determine performance expectations and examine trends related to brokers and AppServers.

This section presents information related to both types of AppServer log file monitors. However, only the procedures specific to an AppServer broker log file monitor and its associated viewer are presented. These same procedures will work with an AppServer agent log file monitor. For more general information about Fathom log file monitor features and functionality, see the [Resource Monitoring Guide](#).

---

**Note:** Log file monitors are not available for either remote AppServer brokers or their associated AppServers.

---

## Getting started with log files for AppServer resources

For each local AppServer broker that Fathom Management discovers, Fathom supports monitoring its two associated log file monitors. Fathom provides a log file resource monitor for the AppServer broker itself and another for its associated AppServer server. Each of these log file monitors has its own log file monitoring capabilities.

The AppServer log file resource monitors are not enabled until the AppServer for which the resource monitors were created is started. When the log file monitor first starts monitoring either an AppServer broker or AppServers, it always starts at the end of the log file.

### Naming conventions

Fathom prepends the broker's name to the name of the broker and server log file monitors and viewers. For example, Fathom generates the following log file monitor and associated viewer names for an AppServer broker instance named **asbroker50** and the container named **vesta**:

- **Broker-related log file names** — Displays **vesta.asbroker1BrokerLogFileMonitor** and **vesta.asbroker1 Broker Log File Contents**.
- **AppServer-related log file names** — Displays **vesta.asbroker1ServerLogFileMonitor** and **vesta.asbroker1 Server Log File Contents**.

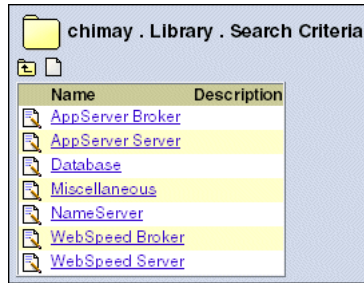
You cannot change these names.

## Characteristics of AppServer resource log file monitors

Data that you can capture and view using the AppServer resources log file monitors and viewers can help you to:

- Ensure the integrity of these log files by monitoring files for errors and allowing you to define actions that trigger when errors occur.
- Use predefined AppServer-related search criteria, or create your own, to run against the data in these log files. Fathom predefines search criteria to support the broker and servers log file monitors.

Figure 4–5 shows an example of the Search Criteria subcategories, including the **AppServer Broker** and **AppServer Server** links to the predefined search criteria.



**Figure 4–5: AppServer-related search criteria**

You can create and maintain the search criteria for each of the AppServer resources in two locations:

- At the AppServer resource local file monitor instance level. The search text and type are not shareable at this level. See the [“Customizing an AppServer Broker log file monitor”](#) section on page 4–25 for details.
- At the Fathom Component Library level under the AppServer subcategory. The search text and type are shareable at this level. See the [“Working with rule sets”](#) section on page 6–29 for details.

Specifically, the predefined search criteria provide:

- Detailed data about the recorded operations of an AppServer broker or AppServers.
- A means by which you can extract detailed data.

## AppServer log file monitor default values

Once an AppServer is enabled, Fathom creates log file monitors for any discovered brokers and their associated AppServers, using several default values. Of all the default AppServer log file monitor properties, you can modify only its description. However, you have several options regarding the Search Criteria you can use for the log file monitor. See the [“Customizing an AppServer Broker log file monitor”](#) section on page 4–25 for details.

The default values are as follows:

- The AppServer default log file monitor is disabled until the AppServer is first started.
- The **Bookmark** is set to **Last Line**, and it is unique.
- The **On First Poll** property is set to **Search From End**.

For detailed information about the Bookmark feature and On **First Poll** property as they relate to log file monitors in general, see the [Resource Monitoring Guide](#).

### File Resource Defaults page

Fathom also supports a polling interval default value for the AppServer broker log file monitor and the AppServer agent log file monitor.



**To display or update a polling interval default value:**

1. Click **Resources** on the menu bar.
2. Click **Resource Monitor Defaults** → **File Resource Defaults**.
3. Scroll down the **File Resource Defaults** page to display the **AppServer Broker Log File Monitor** and the **AppServer Agent Log File Monitor** entries.

You can revert back to the original Fathom-supplied default value set for the **Polling Interval** field at any time by clicking **Restore Fathom Defaults**.

## Reviewing predefined log file monitor search criteria

Each log file provides predefined search criteria that address common AppServer broker- or AppServers-related events. Use these searches as defined, or copy and customize them. Review the predefined search criteria before you customize an AppServer log file monitor.

---

**Note:** It is recommended that you do not edit or delete the predefined criteria.

---



### To review predefined log file monitor search criteria:

1. Select **Library** from the menu bar.
2. Click the plus (+) icon next to **Search Criteria** in the list frame to expand this category.
3. Click either **AppServer Broker** or **AppServer Server**. A list of predefined search criteria related to the category that you selected appears in the detail frame. The following screen shows a partial list of the **AppServer Broker** default search criteria:

Name	Description
<a href="#">Broker_Error_Msg_Received</a>	Catches the following error: Admin error response received: (adminErrorMsg).
<a href="#">Cannot_Start_Server</a>	Catches the following error: ERROR: cannot start server.
<a href="#">Connection_Refused</a>	Catches the following error: ERROR: connection refused : maximum number of client connections has been reached.
<a href="#">Disconnecting_Client</a>	Catches the following error: Error (errMsg) ... disconnecting client.
<a href="#">Error_Loading_Prop_File</a>	Catches the following error: Error loading properties file (propertiesFileName) : (ProException_string) : (ProExceptionMsg_string).
<a href="#">Fatal_Error</a>	Catches the following error: FATAL ERROR : (rspoode) (errMsg).
<a href="#">Invalid_Action_For_State</a>	Catches the following error: Client FSM Error : Invalid ACTION for STATE : state= (state) (descState) event= (event) (descEvent) action= (action) nextState= (state) (descState).
<a href="#">Invalid_Admin_Response</a>	Catches the following error: Invalid admin response received: (adminErrorMsg).
<a href="#">Invalid_State</a>	Catches the following error: FSM Error : Invalid State : state= (currentState).
<a href="#">IOException_Client_Response</a>	Catches the following error: IOException while sending clientRsp : (IOException_string).
<a href="#">IOException_Message_From_Server</a>	Catches the following error: IOException reading message from server : (IOException_string)
<a href="#">MsgFormatException</a>	Catches the following error: MsgFormatException received on client connection : (MsgFormatException_string) (MsgFormatExceptionMsg_string).
<a href="#">NameServer_Consistency_Error</a>	Catches the following error: Internal Consistency Error on NameServer (host):(portNum) : (InternalConsistencyException).
<a href="#">NameServer_IOException</a>	Catches the following error: IOException on startup of NameServer (hostName):(portNum) : (IOException_string).
<a href="#">NameServer_keepAlive_stopped</a>	Catches the following error: NameServer keepAlive to NameServer (host):(portNum) : stopped.
<a href="#">NameServer_NOT_started</a>	Catches the following error: NameServer NOT started for services (appServices).
<a href="#">No_Servers_Available</a>	Catches the following error: ERROR: No servers available.

---

**Note:** You can also create your own search criteria to address a particular AppServer error for which you want to monitor an AppServer. See the “[Customizing an AppServer Broker log file monitor](#)” section on page 4–25 for details.

---



## Customizing an AppServer Broker log file monitor

The following procedure describes how to customize an AppServer broker log file monitor. Use these same general steps to customize a log file monitor for AppServers.



**To customize a log file monitor:**

1. Navigate to the **AppServer main** page specific to your broker, using the procedure detailed in the [“Accessing OpenEdge resource information”](#) section on page 2–12.
2. Click **Log File Monitor of Broker** on the **AppServer main** page. The log file monitor summary monitoring page for the AppServer broker you selected appears:

Log File: chimay.asbroker1BrokerLogFileMonitor  
Broker Log File Monitor for: asbroker1

Not Checked (20 Minutes)  
Last Poll: Jun. 6, 1:11 PM  
Poll Count: 5 Fail Count: 0 (0.0%)

Edit Log File Viewer

Monitoring plans

Name	Poll	Alerts	
Default_Schedule_Plan	5 mins	✓	Edit

Rule Summary

Name	Status	Severity
AS_Broker_LogFileMonitor_RuleSet	---	---
Broker_Error_Msg_Received	Passed	
Cannot_Start_Server	Passed	
Connection_Refused	Passed	
Disconnecting_Client	Passed	
Disconnecting_Client_2	Passed	
Error_Loading_Prop_File	Passed	
Fatal_Error	Passed	
Invalid_Action_For_State	Passed	
Invalid_Admin_Response	Passed	
Invalid_State	Passed	
Invalid_State_2	Passed	
IOException_Client_Response	Passed	
IOException_Message_From_Server	Passed	
MsgFormatException	Passed	
No_Servers_Available	Passed	
Server_Exec_Error	Passed	
ServletPCEException	Passed	
Unimplemented_State	Passed	
NameServer_Consistency_Error	Passed	
NameServer_IOException	Passed	
NameServer_keepAlive_stopped	Passed	
NameServer_NOT_started	Passed	

Add Plan

3. Customize or view the contents of an AppServer broker log file monitor as follows:

To...	Select...
Add an existing monitoring plan to this resource monitor	<b>Add Plan.</b>
Modify a specific plan	<b>Edit</b> associated with that plan.
Change the description of this log file monitor	<b>Edit</b> at the top of the page.
View the contents of the log file monitor	<b>Log File Viewer</b> at the top of the page.

---

**Note:** Fathom prevents the assignment of schedules that share days or times that overlap. For example, if you have a **Default\_Schedule** set up for a resource monitor, you cannot set up an additional plan because the **Default\_Schedule** is defined for 7 days a week, 24 hours a day. You must modify or remove the **Default\_Schedule** to set up additional plans.

---

4. To add individual rules, click **Edit** within the monitoring plans section to display the edit page for the log file monitor.
  - a. Click **Add Rule** under the **Rules selected for this plan** section of the broker monitoring plan page. Proceed as follows:

To...	Then...
Use an AppServer broker rule already defined in the library	<p>Select:</p> <ul style="list-style-type: none"> <li>• <b>AppServer Broker</b> from the drop-down list associated with the <b>Choose Criteria Category</b>.</li> <li>• The appropriate value from each drop-down list associated <b>Choose Search Criteria</b>.</li> </ul>
Create a new AppServer broker rule	<p>Click <b>Create Criterion</b> to display the <b>Create Search Criterion</b> page. Complete this page as follows:</p> <ul style="list-style-type: none"> <li>• Enter values in the required fields: <b>Name</b> (identifies the name of the search criteria you are creating) and <b>Search Text</b> (identifies the information you are looking for in the log).</li> <li>• Review the default option <b>Use Existing Category</b>; it indicates that the new rule will be stored in an existing group. Select the <b>AppServer Broker</b> category from those that display in the drop-down list associated with the <b>Use Existing Category</b> option.</li> <li>• Click <b>Save</b>. The <b>Rules Properties</b> section of the <b>Log File Rule</b> page redisplay. The values you defined and selected to create a rule on the <b>Create Search Criterion</b> page are now available on the <b>Log File Rule</b> page. The <b>Choose Criteria Category</b> drop-down list displays the category in which you elected to store the new rule. The <b>Choose Search Category</b> drop-down list displays the name you entered in the <b>Name</b> field on the <b>Create Search Criterion</b> page.</li> </ul>

- b. Select the appropriate values from the **Severity** and **On Alert Action Perform** fields to complete the alert severity and action definition, respectively, that you want to associate with this rule.
  - c. To add another individual rule, repeat [Step a](#) and [Step b](#).
- 5. Click **Select Rule Sets** to create a new log file rule or choose from existing rule sets to add to the monitoring plan.

If you choose **Select Rule Sets**, you can choose from a list of predefined rule sets to add to the monitoring plan.
- 6. Click the detail page for the parent icon (the file folder with the up arrow on it) to redisplay this AppServer broker's monitoring plan page with the rules section updated with the new rules.

For more information about editing search criteria for rules, see the appropriate sections of the *[Resource Monitoring Guide](#)*.

---

**Note:** You can copy the default AppServer log file rule set, but you cannot delete it.

---

## Using the AppServer log file viewers

To view the contents of each AppServer log file, access the viewer associated with each individual log file.

The log file viewer allows you to examine the contents of an AppServer-related log file through an HTML interface. You can access these log file viewers from two locations:

- Click the link in the **Command and control** section of the **AppServer main** page; click **Log File Viewer of Broker** to display the broker's file contents and click **Log File Viewer of Servers** to display the AppServers' file contents.
- Click the **Log File Viewer** button that appears at the top of the log file monitor summary monitoring page.

Figure 4–6 shows an example of the AppServer broker log file viewer with the contents of a AppServer broker log file displayed.

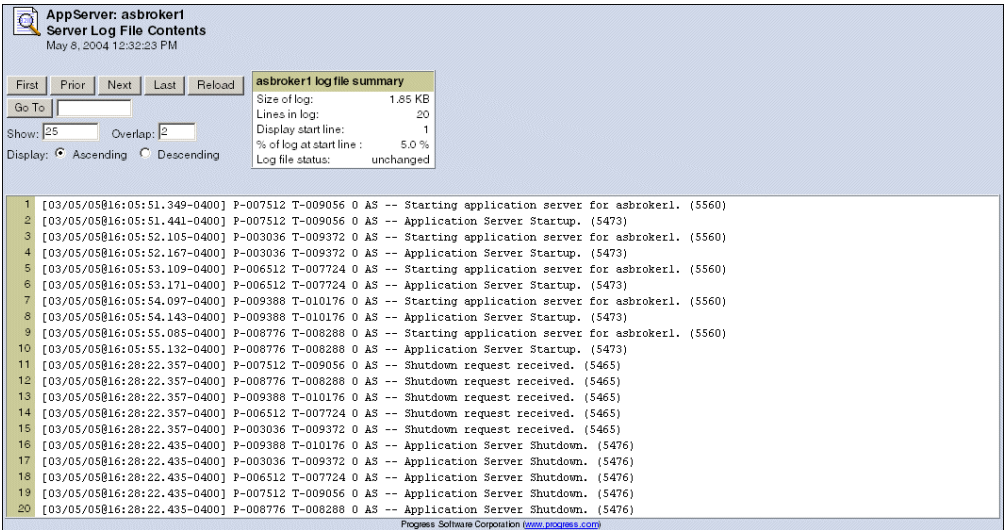


Figure 4–6: AppServer Broker log file viewer example

The following information will help you use the AppServer log file viewer:

- Use the **Show** field to control how many AppServer log file entries display at one time. The number entered into the **Show** field cannot be less than 10.
- Use the **Overlap** field to control how many entries are repeated from screen to screen.

**Note:** The value in the **Overlap** field cannot be more than the number in the **Show** field minus one. For example, if you show 30 entries, you can overlap only 29 or fewer of them.

- Click **Reload** after changing the values in the **Show** field or **Overlap** field. Note that Fathom will prompt you to click **Reload**. The warning message that reads *changed, reloaded needed*, flashes in the **File log status** field in the Log file summary section of this page.

If you do not reload, the viewer displays the previous values.

- Click **Go To** to control which numbered entry in the log file the viewer begins its display with. For example, a value of 10 entered into the **Go To** field will begin the display from the tenth log file entry.

---

**Note:** You must click **Go To** after entering a value in the **Go To** field, or the viewer will not update its display.

---

- The default display of entries is in ascending order; choose **Descending** to change the display. Note that the **Show** field dictates the number of entries shown, whether they display in ascending or descending order.
- Click **First** to display the first  $x$  entries, where  $x$  is the value in the **Show** field.
- Click **Prior** to display the previous  $x$  entries, where  $x$  is the value in the **Show** field.
- Click **Next** to display the next  $x$  entries, where  $x$  is the value in the **Show** field.
- Click **Last** to display the last  $x$  entries, where  $x$  is the value in the **Show** field.
- To view additional log file entries without changing your current starting log file entry, leave the **Go To** field blank, change the value in the **Show** field, and click **Reload**.

## Refreshing log file data

Periodically refresh log file data. Select the **Refresh** page icon from the status bar for either the list or detail frame to repaint an existing page. You can also set a default value that Fathom uses to automatically refresh the management console.



To set a default value that Fathom uses to automatically refresh the management console, select **Options** → **User Preferences** → **Automatically Refresh Pages**.

Refresh data to avoid the following situations:

- Fathom considers a viewer that has been inactive for more than four hours “stale.” Once a viewer becomes stale, Fathom releases ninety-five percent of any memory it holds. If you try to use a stale viewer, Fathom automatically reloads the file. Because additional resource activity might have occurred during the viewer’s inactivity, the reloaded log file view might not match the previous log file view of that resource.

- Fathom considers a viewer that has been inactive for forty-eight hours “dead.” Once a viewer dies, Fathom releases all of its memory. To return to the log file displayed in a dead view, you need to renavigate to it, even if you pinned up the view or saved a link to it before the viewer died.

## Examining AppServer-related Operational views

The **AppServer main** page provides an **Operational views** section that allows you to access and review data related to the performance of a:

- Specific AppServer broker.
- Pool of AppServers associated with a specific broker.

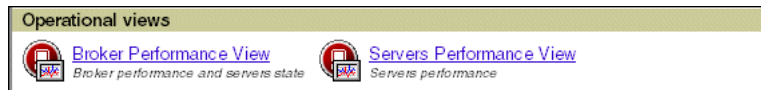
Data for both the broker and the broker’s AppServer pool can display in text and graph formats.

---

**Note:** The graphs associated with the AppServer Operational views only display when the **Broker statistics available** field on the **Broker Control** page displays a **True** status. See the “[Data collection details](#)” section on page 4–9 for details.

---

Figure 4–7 shows the **Operational views** section of the **AppServer main** page.



**Figure 4–7: Operational views section example**

The following information presents detailed information about how to access and review detailed information associated with each of these views.

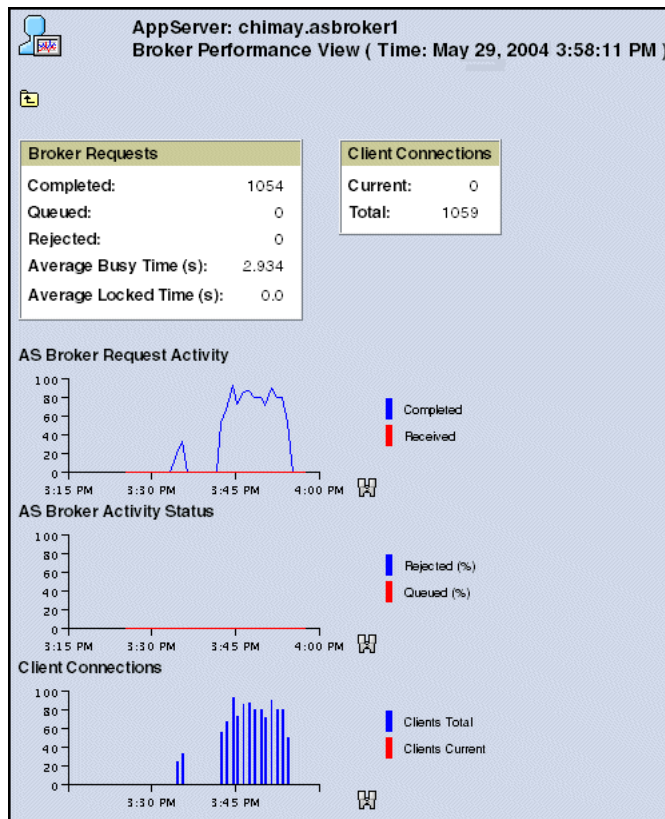
## Accessing and reviewing the Broker Performance View

The AppServer **Operational views** section allows you to display information about the AppServer broker's performance and the state of the broker's associated servers. Review this data frequently; these details can help you make informed decisions about your use of the broker and server pool controls as discussed earlier in this chapter.



**To display and review AppServer Operational views information:**

1. Display the **AppServer main** page for the AppServer broker instance that you want to review. See the [“Accessing OpenEdge resource information”](#) section on page 2–12 for the detailed procedure.
2. Click in the **Operational views** section to display the **AppServer Broker Performance View** page. This page is comprised of data summary sections and graphs as shown:





## Data summary sections

The summarized display-only text data on this page is comprised of two sections: **Broker Requests** and **Client Connections**. Data in both of these text boxes is determined when the page is initialized or refreshed:

- **Broker Requests** — Details about the AppServer broker's connection workload as identified in [Table 4–7](#).

**Table 4–7: AppServer broker connection workload details**

This field...	Displays a value to indicate the...
<b>Completed</b>	Number of broker requests fulfilled.
<b>Queued</b>	Number of broker requests to be processed.
<b>Rejected</b>	Number of broker requests that could not be processed.
<b>Average Busy Time (ms)</b>	Average amount of time that the broker is busy servicing requests (expressed in milliseconds).
<b>Average Locked Time (ms)</b>	Average amount of time that the broker is locked (expressed in milliseconds).

- **Client Connections** — Identifies the number of client connections that the broker is currently handling. Also, the total number of client connections this broker has processed since the broker started.

## Graphs presentation section

The graphs presentation section of the **Broker Performance View** contains three graphs: **AS Broker Request**, **AS Broker Activity Status**, and **Client Connections**. Provided that data collection is set and the **Trend** option is selected, the graphically displayed data complements the summarized text data that displays on the **AppServer Broker Performance View** page. See the [“Data collection details”](#) section on page 4–9 for details.

One display format for these graphs, as previously shown on the **Broker Performance View** page, is a line graph. This format measures how a particular broker-related activity has changed over a period of time. [Table 4–8](#) identifies and briefly describes each of these graphs.

**Table 4–8: AppServer broker performance-related graphs**

This AppServer broker graph...	Displays two lines...
<b>AS Broker Request</b>	<p>Of broker-related performance data over a specified time period.</p> <p>The blue line identifies the number of requests that the broker has completed since the last poll; the red line identifies the number of requests that this same broker has received in this time period.</p>
<b>AS Broker Activity Status</b>	<p>Of broker-related performance data over a specified time period. The blue line identifies the percent of requests that the broker has rejected, up to and including the last poll Fathom has completed for this broker resource. The red line identifies the percent of requests in the queue waiting for the broker, up to and including the last poll completed.</p>
<b>Client Connections</b>	<p>Of client connections related to this broker over a specified time period. For example, data displayed might be related to the last polling activity.</p> <p>The blue line identifies the client total number of connections requested; the red line identifies the number of clients currently connected to this broker.</p> <p><b>Note:</b> It is possible for this graph to accurately show that the number of current connections is higher than the total number of connections. The <b>Clients Total</b> reflects only new connections over the specified time period. In contrast, the <b>Clients Current</b> reflects <i>all</i> current connections, both newly connected and those that might still be connected from a previous polling period, in place when the graph is displayed.</p>

See the “[Changing OpenEdge pinup graphical displays](#)” section on page 2–25 for details about changing the data appearance of graphs.

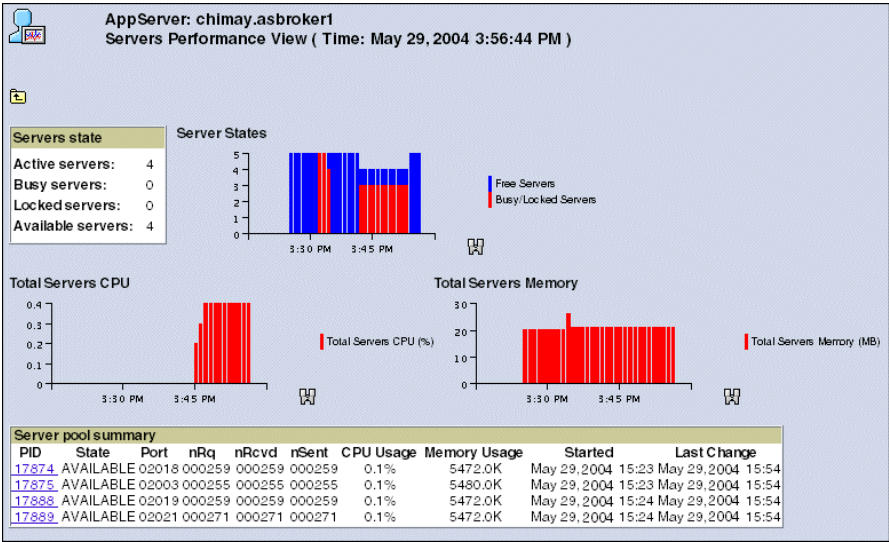
## Accessing and reviewing the Server Performance View

The AppServer **Operational views** section allows you to display information about servers' status.



To display and review servers' status information:

1. Display the **AppServer main** page for the AppServer instance that you want to review. See the “[Accessing OpenEdge resource information](#)” section on page 2–12 for the detailed procedure.
2. Click **Server Performance View** in the **Operational views** section to display the **AppServer Servers Performance View** page:



### Data summary section

This display-only view is comprised of two sections:

- **Servers state** — Displays the four possible states of the servers that are currently associated with this AppServer broker: **Active**, **Busy**, **Locked**, and **Available**. See [Table 4–5](#) earlier in this chapter for a definition of each of these states.

- **Server pool summary** — Displays detailed data about each individual server in the AppServer pool associated with a specific AppServer broker. See [Table 4–6](#) for a description of each field that displays in the Server pool summary section. You also have access to additional data about a specific AppServer and a control that allows you to kill a server process. See the “[Killing an AppServer process](#)” section on page 4–18 for the detailed steps.

**Graphs summary section**

The graphs presentation section of the **Servers Performance View** contains three graphs: **Server States**, **Total Servers CPU**, and **Total Servers Memory**. Provided that data collection is set and the **Trend** option is selected, the graphically displayed data displays to complement the summarized text data that displays on the **AppServer Servers Performance View** page. See the “[Data collection details](#)” section on page 4–9 for details.

One display format for these graphs, as previously shown on the **Servers Performance View** page, is a line graph. This format measures how a particular broker-related activity has changed over a period of time. [Table 4–9](#) identifies and briefly describes each of these graphs.

**Table 4–9: AppServers performance-related graphs**

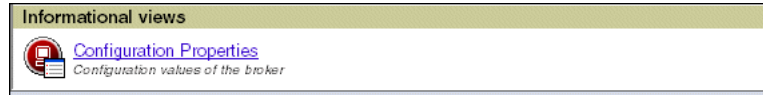
This AppServers graph...	Displays...
<b>Server States</b>	Two lines of servers-related performance data over a specified time period. The blue line identifies the number of free servers; the red line identifies the number of busy/locked servers during this same time period.
<b>Total Servers CPU</b>	One line of servers-related performance data over a specified time period. This single data line indicates the total percent of the servers’ CPU usage.
<b>Total Servers Memory</b>	One line of servers-related performance data over a specified time period. This single data line indicates the total percent of the servers’ Memory consumption.

See the “[Changing OpenEdge pinup graphical displays](#)” section on page 2–25 for details about changing the data appearance of graphs.

## Examining AppServer-related Informational views

The **AppServer main** page provides an Informational views section that allows you to access and review data related to the AppServer broker's configuration properties. The values that display originate from the `ubroker.properties` file.

Figure 4–8 shows an example of the **Informational views** section of the **AppServer main** page.

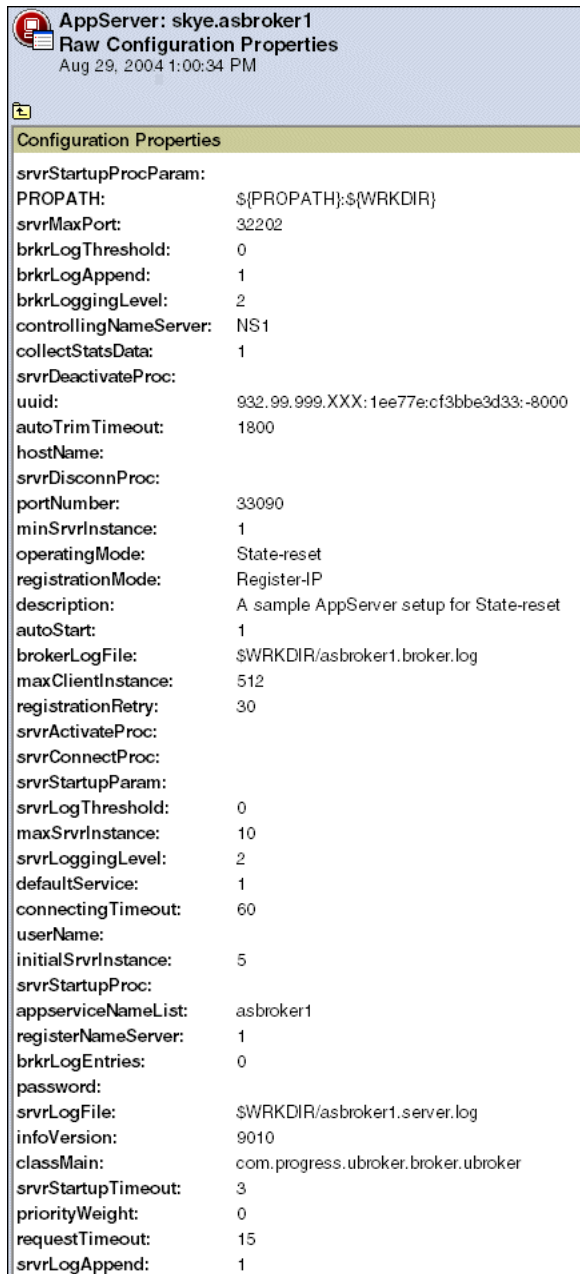


**Figure 4–8: Informational views section example**



**To display and review Configuration Properties view details:**

1. Display the **AppServer main** page for the AppServer broker instance that you want to review. See the “[Accessing OpenEdge resource information](#)” section on page 2–12 for the detailed procedure.
2. Click **Configuration Properties** in the **Informational views** section to display the **Configuration Properties** page as shown in the following example:



3. Review the values that display. Note that the properties list is quite long. You might need to scroll to see more of the properties and their associated values.

---

## Managing NameServer Data

---

This chapter presents Fathom Management features and functionality related to NameServers.

Topics in this chapter include:

- [Overview](#)
- [Reviewing NameServer status](#)
- [Modifying NameServer control settings](#)
- [Accessing and reviewing NameServer-related log file data](#)
- [Using the NameServer log file viewer](#)
- [Examining NameServer Operational and Informational views](#)

## Overview

Fathom supports a variety of tasks you can perform to manage a specific NameServer, including:

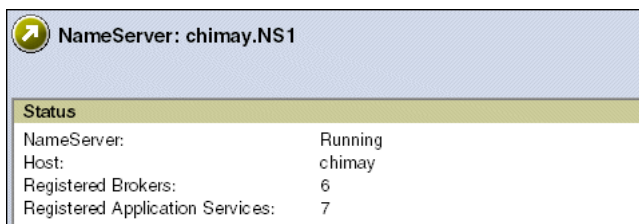
- Reviewing your current operating status and associated details.
- Reviewing property settings associated with a NameServer.
- Accessing and viewing data collected in a NameServer log file monitor.
- Working with NameServer resource-related details available through informational and operational views.
- Monitoring and managing the NameServer using monitoring plans and rules.

With the exception of monitoring procedures, this chapter presents detailed information about each task identified in this list. See [Chapter 6, “Monitoring Plans and Rules for OpenEdge Server Resources,”](#) for details about monitoring plans and rules.

You must have appropriate Fathom role authorization to perform several of these tasks. For more information, see the [“Role authorization and OpenEdge tasks”](#) section on page 1–7.

## Reviewing NameServer status

The **Status** section of the **NameServer main** page summarizes current operational details about the NameServer. [Figure 5–1](#) shows an example of the **Status** section. Note that the NameServer’s name displays in the page’s title area.



**Figure 5–1: NameServer Status section sample**



Table 5–1 describes each of these NameServer-related details.

**Table 5–1: NameServer Status details**

This field...	Displays the...
<b>NameServer</b>	Running status of the NameServer. Possible values are: <ul style="list-style-type: none"> <li><b>Running.</b></li> <li><b>Not running.</b></li> </ul>
<b>Host</b>	Host machine's name.
<b>Registered Brokers</b>	Number of brokers currently registered with the NameServer.
<b>Registered Application Services</b>	Number of Application Services (that is, WebSpeed and AppServer) that are registered with the NameServer.

## Modifying NameServer control settings

The **Command and control** section of the **NameServer main** page allows you to:

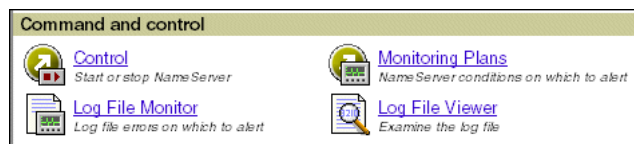
- Start and stop a specific NameServer instance, and enable or disable the monitoring of it.
- Obtain and review data collected through a NameServer log file associated with the instance.
- Monitor and manage a NameServer instance using monitoring plans and rules.

---

**Note:** The NameServer does not use the Configuration Advisor feature because the NameServer does not collect and trend data.

---

Figure 5–2 shows an example of the **Command and control** section of the **NameServer main** page.



**Figure 5–2: Command and control section example**

The information in this section presents functional descriptions and procedural details related to the **NameServer Control** page.

Table 5–2 identifies where you can find information about other functionality related to the NameServer **Command and control** section.

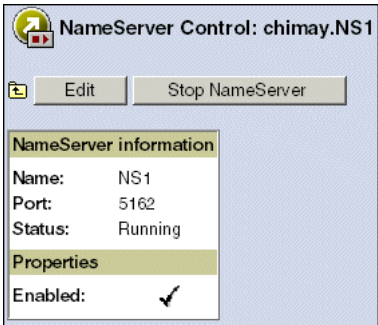
**Table 5–2: Additional NameServer information**

For NameServer-related details about...	See the...
Log file monitors and viewers	<a href="#">“Accessing and reviewing NameServer-related log file data”</a> section on page 5–6.
Monitoring plans and rules	<a href="#">Chapter 6, “Monitoring Plans and Rules for OpenEdge Server Resources.”</a>
Rule sets	<a href="#">“Working with rule sets”</a> section on page 6–29.

## NameServer Control page content

The **NameServer Control** page summarizes details about a specific NameServer resource. From this page, you can start and stop a NameServer instance, and change the **Enabled** option.

Figure 5–3 shows an example of the **NameServer Control** page.



**Figure 5–3: NameServer Control page example**

The following details describe the two sections of the **NameServer Control** page.

## NameServer information section

The **NameServer information** section displays read-only values for these fields: the **Name** (NameServer's name), **Port** (number), and the **Status**. The NameServer's name and port number are defined in the `ubroker.properties` file; the **Status** field details reflect real-time values based on the NameServer's current operating status.

## Properties section

The **Properties** section displays the state of the **Enabled** option. The **Enabled** option indicates that this resource recognizes a monitoring plan and its associated rules when the broker resource is active.

During the discovery process, all NameServers that Fathom discovers and identifies in the list frame under the category are *enabled* by default. (A check mark indicates that the **Enabled** option is set.) Once you enable a NameServer resource, Fathom uses the Fathom-supplied default values to establish a monitoring plan and rules. (You can customize the plan and rules at any time.)

## Changing NameServer controls

This section describes how to change NameServer controls.



**To start or stop the NameServer, or change the Enabled property setting:**

1. Display the **NameServer main** page for the NameServer instance that you want to start. See the “[Accessing OpenEdge resource information](#)” section on page 2–12 for the detailed steps.
2. Click **Control** in the **Command and control** section to display the **NameServer Control** page:



This table highlights the controls that you can now change:

To...	Click this button...
Change the current setting for the <b>Enabled</b> property. To enable the NameServer, click in the box associated with that specific option; a checkmark displays to indicate that the property is now set. To disable the NameServer, click the existing checkmark in the box associated in the option; the checkmark is deleted to indicate that the option is no longer set.	<b>Edit.</b> Then select or clear the Enable property.
Toggle between stopping and starting the NameServer, depending on the current value displayed in the <b>Status</b> field.  For example, if the status currently displays <b>Running</b> , the button label will read <b>Stop NameServer</b> . You can click this button to stop the NameServer. Fathom stops the and updates the value in the <b>Status</b> field to now read <b>Not Running</b> .	<b>Stop or Start.</b>
Exit this page without changing any values. The <b>NameServer main</b> page redisplay in the management console.	<b>Cancel.</b>

## Accessing and reviewing NameServer-related log file data

You can access and view log file data generated for each locally defined NameServer instance. Log files can store a tremendous amount of data. Therefore, monitoring and analyzing data collected within these files might help you to better determine NameServer performance expectations and examine trends.

---

**Note:** Log file resource monitoring cannot be performed for remote NameServers.

---

This section presents information and procedures specific to a log file monitor and viewer. For more general information about Fathom log file monitor features and functionality, see the [Resource Monitoring Guide](#).

## Getting started with NameServer log files

For each local NameServer instance that Fathom Management discovers, Fathom supports monitoring its associated log file monitor. By default, the log file monitor's name uses the same name as the NameServer's name. Fathom also provides a log file viewer for each NameServer log file monitor to help you quickly access and review this data.

The NameServer log file monitor is not enabled until the NameServer for which the resource monitor was created is enabled. When the NameServer log file monitor first starts monitoring, it always starts at the end of the log file.

## Characteristics of a NameServer log file monitor

Data that you can capture and view using NameServer log file monitors and viewers help you to:

- Ensure the integrity of NameServer log files by monitoring files for errors and allowing you to define actions that trigger when errors occur.
- Use predefined NameServer-related search criteria, or create your own, to run against the data in a NameServer file. You can create and maintain the search criteria in two locations:
  - At the NameServer local file monitor instance level. The search text and type are not shareable at this level.
  - At the Fathom Component Library level under the NameServer subcategory. The search text and type are shareable at this level.

Specifically, the predefined search criteria provide:

- Detailed data about the recorded operations of a NameServer.
- A means for you to extract the detailed data.

## NameServer log file monitor default values

Once a NameServer is enabled, Fathom creates a NameServer log file monitor, using several default values, for that NameServer resources. Of the default NameServer log file monitor properties, you can modify only its description. However, you have several options regarding the Search Criteria you can use for a NameServer log file monitor. See the [“Customizing a NameServer log file monitor”](#) section on page 5–9 for more details about this topic.

The default values are as follows:

- The NameServer default log file monitor is enabled and disabled along with the NameServer instance.
- The **Bookmark** is set to **Last Line**, and it is unique.
- The **On First Poll** property is set to **Search From End**.

For detailed information about the Bookmark feature and **On First Poll** property as they relate to log file monitors in general, see the appropriate section in the *Resource Monitoring Guide*.

### File Resource Defaults page



**To display or update a polling interval default value:**

1. Click **Resources** on the menu bar.
2. Click **Resource Monitor Defaults** → **File Resource Defaults**.
3. Scroll down the **File Resource Defaults** page to display the current value set in the **Polling Interval** field for the NameServer **Log File Monitor** entry.

To revert to the original Fathom-supplied default value, click **Restore Fathom Defaults**.

## Reviewing predefined log file monitor search criteria

The NameServer log file monitor provides predefined search criteria that address common NameServer events. Use these searches as defined, or copy and customize them. Review this information before you customize a NameServer log file monitor.

---

**Note:** It is recommended that you do not edit or delete the predefined criteria.

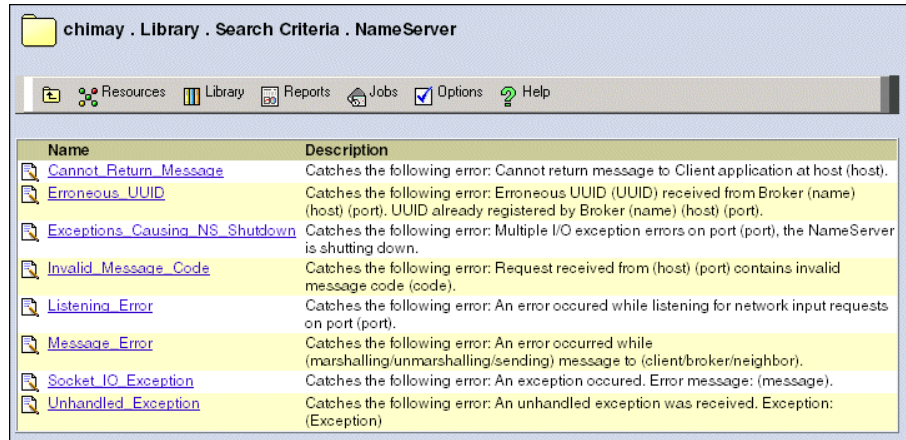
---



**To access predefined search criteria for a NameServer log file monitor:**

1. Select **Library** from the menu bar.
2. Click the plus (+) icon next to the **Search Criteria** in the list frame to expand this category.

3. Click **NameServer**. A list of predefined NameServer search criteria appears in the detail frame:



The screenshot shows a software window titled "chimay . Library . Search Criteria . NameServer". It has a menu bar with "Resources", "Library", "Reports", "Jobs", "Options", and "Help". Below the menu bar is a table with two columns: "Name" and "Description". The table lists seven predefined search criteria for NameServer errors.

Name	Description
<a href="#">Cannot Return Message</a>	Catches the following error: Cannot return message to Client application at host (host).
<a href="#">Erroneous UUID</a>	Catches the following error: Erroneous UUID (UUID) received from Broker (name) (host) (port). UUID already registered by Broker (name) (host) (port).
<a href="#">Exceptions Causing NS Shutdown</a>	Catches the following error: Multiple I/O exception errors on port (port), the NameServer is shutting down.
<a href="#">Invalid Message Code</a>	Catches the following error: Request received from (host) (port) contains invalid message code (code).
<a href="#">Listening Error</a>	Catches the following error: An error occurred while listening for network input requests on port (port).
<a href="#">Message Error</a>	Catches the following error: An error occurred while (marshalling/unmarshalling/sending) message to (client/broker/neighbor).
<a href="#">Socket I/O Exception</a>	Catches the following error: An exception occurred. Error message: (message).
<a href="#">Unhandled Exception</a>	Catches the following error: An unhandled exception was received. Exception: (Exception)

**Note:** You can also create your own search criteria to address a particular NameServer error for which you want to monitor a NameServer. For additional information, see the “Customizing a NameServer log file monitor” section on page 5–9.

## Customizing a NameServer log file monitor

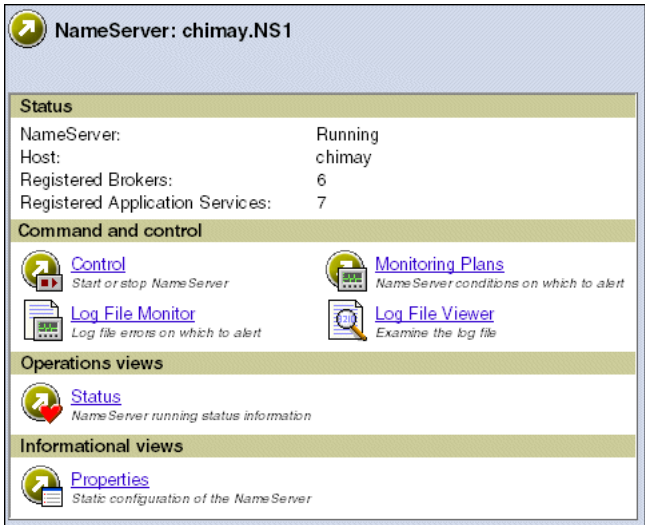
This section describes how to customize a NameServer log file monitor.



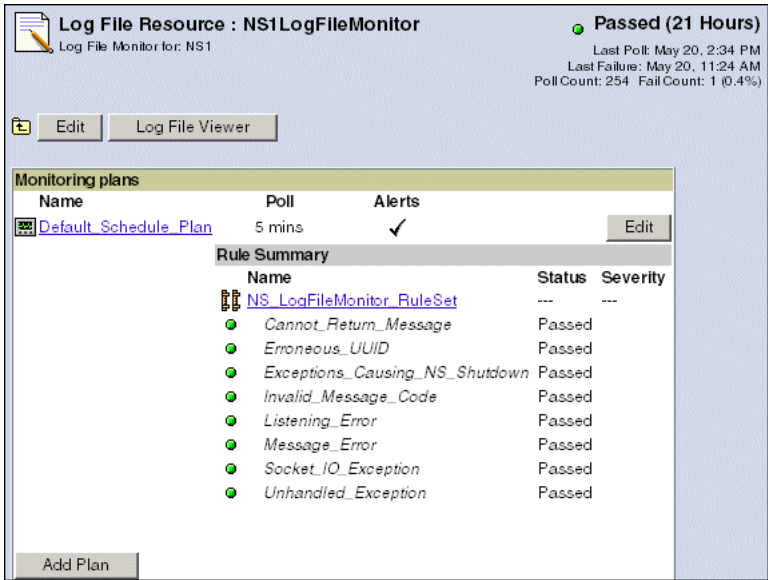
### To customize a NameServer log file monitor:

1. Click **Resources** in the menu bar. The main resource type categories appear in the list frame.
2. Click **OpenEdge** in the list frame; the **OpenEdge details** page for the OpenEdge category appears in the detail frame.
3. Click **NameServer** to display a list of NameServers.

4. Click the name of the **NameServer** for which you want to access information. The **NameServer main** page that you selected appears in the detail frame:



5. Click **Log File Monitor**. The log file monitor summary monitoring page for the NameServer instance that you selected appears:





6. Customize or view the contents of a NameServer log file monitor as follows:

To...	Select...
Add an existing monitoring plan to this resource monitor	<b>Add Plan.</b>
Modify a specific plan	<b>Edit</b> associated with that plan.
Change the description of this log file monitor	<b>Edit</b> at the top of the page.
View the contents of the log file monitor	<b>Log File Viewer</b> at the top of the page.

---

**Note:** Fathom prevents the assignment of schedules that share days or times that overlap. For example, if you have a **Default\_Schedule** set up for a resource monitor, you cannot set up an additional plan because the **Default\_Schedule** is defined for 7 days a week, 24 hours a day. You must modify or remove the **Default\_Schedule** to set up additional plans.

---

7. To add individual rules, click **Edit** within the monitoring plans section to display the edit page for the NameServer log file monitor. To add rule sets to this plan, perform [Step 8](#) later in this procedure.

- a. Click **Add Rule** under the **Rules selected for this plan** section of the NameServer monitoring plan page. Proceed as follows:

To...	Then...
Use a NameServer rule already defined in the library	<p>Select the:</p> <ul style="list-style-type: none"><li>• <b>NameServer</b> from the drop-down list associated with the <b>Choose Criteria Category</b>.</li><li>• The appropriate value from each drop-down list associated <b>Choose Search Criteria</b>.</li></ul>
Create a new NameServer rule	<p>Click <b>Create Criterion</b> to display the <b>Create Search Criterion</b> page. Complete this page as follows:</p> <ul style="list-style-type: none"><li>• Enter values in the required fields: <b>Name</b> (identifies the name of the search criteria you are creating) and <b>Search Text</b> (identifies the information you are looking for in the log).</li><li>• Review the default option <b>Use Existing Category</b>; it indicates that the new rule will be stored in an existing group. Select the <b>NameServer</b> category from those that display in the drop-down list associated with the <b>Use Existing Category</b> option.</li><li>• Click <b>Save</b>. The <b>Rules Properties</b> section of the <b>Log File Rule</b> page redisplays. The values you defined and selected to create a rule on the <b>Create Search Criterion</b> page are now available on the <b>Log File Rule</b> page. The <b>Choose Criteria Category</b> drop-down list displays the category in which you elected to store the new rule. The <b>Choose Search Category</b> drop-down list displays the name you entered in the <b>Name</b> field on the <b>Create Search Criterion</b> page.</li></ul>

- b.** Select the appropriate values from the **Severity** and **On Alert Action Perform** fields to complete the alert severity and action definition, respectively, that you want to associate with this rule.
  - c.** To add another individual rule, repeat [Step a](#) and [Step b](#).
- 8.** Click **Select Rule Sets** to create a new log file rule or choose from existing rule sets to add to the monitoring plan.

If you choose **Select Rule Sets**, you can choose from a list of predefined rule sets to add to the monitoring plan.
- 9.** Click the detail page for the parent icon (the file folder with the up arrow on it) to redisplay this NameServer's Monitoring plan page with the rules section updated with the new rules.

For more information about editing search criteria for rules, see the appropriate sections of the [Resource Monitoring Guide](#).

---

**Note:** You can copy the default NameServer log file rule set, but you cannot delete it.

---

## Using the NameServer log file viewer

The NameServer log file viewer allows you to examine the contents of a log file through an HTML interface. You can access the log file viewer from two locations:

- Click the **Log File Viewer** link in the **Command and control** section of the **NameServer main** page.
- Click the **Log File Viewer** button that appears at the top of the **NameServer Log Monitor** page.

Figure 5–4 shows an example of the NameServer log file viewer with the contents of a NameServer log file displayed.

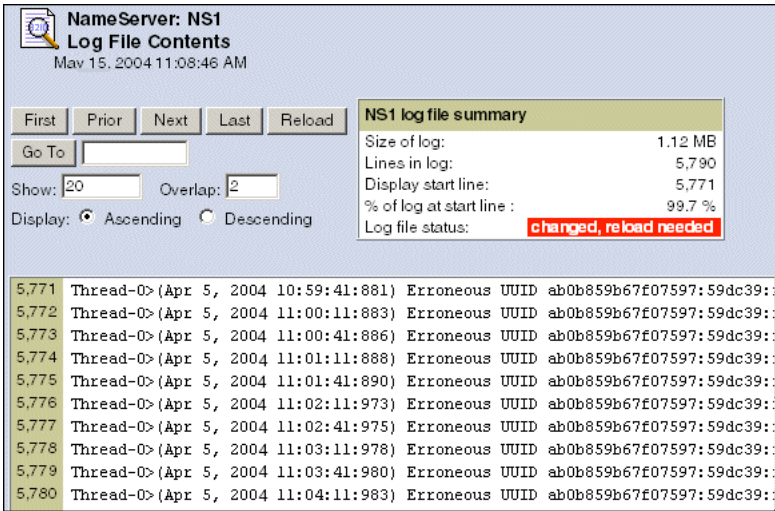


Figure 5–4: NameServer log file viewer example

The following information will help you use the NameServer log file viewer:

- Use the **Show** field to control how many log file entries display at one time. The number entered into the **Show** field cannot be less than 10.
- Use the **Overlap** field to control how many entries are repeated from screen to screen.

---

**Note:** The value in the **Overlap** field cannot be more than the number in the **Show** field minus one. For example, if you show 30 entries, you can overlap only 29 or fewer of them.

---

- Click **Reload** after changing the values in the **Show** field or **Overlap** field. Note that Fathom will prompt you to click **Reload**. The warning message that reads *changed, reload needed*, flashes in the **File log status** field in the **Log file summary** section of this page.

If you do not reload, the viewer displays the previous values.

- Click **Go To** to control which numbered entry in the log file the viewer begins its display with. For example, a value of 10 entered into the **Go To** field will begin the display from the tenth log file entry.

---

**Note:** You must click **Go To** after entering a value in the **Go To** field, or the viewer will not update its display.

---

- The default display of entries is in ascending order; choose **Descending** to change the display. Note that the **Show** field dictates the number of entries shown, whether they display in ascending or descending order.
- Click **First** to display the first  $x$  entries, where  $x$  is the value in the **Show** field.
- Click **Prior** to display the previous  $x$  entries, where  $x$  is the value in the **Show** field.
- Click **Next** to display the next  $x$  entries, where  $x$  is the value in the **Show** field.
- Click **Last** to display the last  $x$  entries, where  $x$  is the value in the **Show** field.
- To view additional log file entries without changing your current starting log file entry, leave the **Go To** field blank, change the value in the **Show** field, and click **Reload**.

## Refreshing log file data

Periodically refresh log file data. Select the **Refresh** page icon from the status bar for either the list or detail frame to repaint an existing page. You can also set a default value that Fathom uses to automatically refresh the management console.



To set a default value that Fathom uses to automatically refresh the management console, select **Options** → **User Preferences** → **Automatically Refresh Pages**.

Refresh data to avoid the following situations:

- Fathom considers a viewer that has been inactive for more than four hours “stale.” Once a viewer becomes stale, Fathom releases ninety-five percent of any memory it holds. If you try to use a stale viewer, Fathom automatically reloads the file. Because additional resource activity might have occurred during the viewer’s inactivity, the reloaded log file view might not match the previous log file view of that resource.
- Fathom considers a viewer that has been inactive for forty-eight hours “dead.” Once a viewer dies, Fathom releases all of its memory. To return to the log file displayed in a dead view, you need to renavigate to it, even if you pinned up the view or saved a link to it before the viewer died.

## Examining NameServer Operational and Informational views

The **NameServer main** page provides two sections that provide access to NameServer-relevant operating details. These sections are:

- Operational views
- Informational views

The following information presents details about accessing and reviewing detailed information associated with each of these sections.

### Accessing and reviewing Operational views


The NameServer **Operational views** display information about the NameServer’s current running status.



**To display and review runtime information about the NameServer instance:**

1. Display the **NameServer main** page for the instance that you want to start. See the [“Accessing OpenEdge resource information”](#) section on page 2–12 for the detailed procedure.

2. Click **Status** in the **Operational Views** section to display the **Operational Status** page:

 <b>NameServer: nightcap.NS1</b> <b>Operational Status</b> May 29, 2004 9:33:04 AM			
<b>Summary</b> Host: nightcap Port: 5162 Timeout: 30 Start time: Wed May 26 11:05:54 EDT 2004 Number of AppServices: 3 Number of Brokers: 2 Total client requests received: 0 Total client requests rejected: 0			
<b>AppService: WS.wsbroker1</b> Requests Received: 0 Broker: WS.wsbroker1   UUID: 932.99.000.ZZZ:1ee77e:cf3bbe04fd-8000 Host: nightcap/172.18.2.89   Port: 3055 Weight: 0   Timeout: 30 Requests Directed: 0			
<b>AppService: AS.asbroker1</b> Requests Received: 0 Broker: AS.asbroker1   UUID: 932.99.999.XXX:1ee77e:cf3bbe3d33~8000 Host: nightcap/172.18.2.89   Port: 3090 Weight: 0   Timeout: 30			

This view is comprised of a single **Summary** section that displays at the top of the view, followed by multiple **AppService** sections, one for each registered Application Service (AppService) that this NameServer recognizes.

## Operational Views content examination

In general, the summary section data pertains to the resource as a whole; each Application Services detail section focuses primarily on data for an individually registered broker. The total values displayed in the **Summary** section are derived by adding the unique values that display in each unique **AppService** section. However, there are some situations in which the request-related counts between these sections might not correlate. See the description of the **Total client requests received** and **Total client requests rejected** fields in [Table 5–3](#) for details, and the **Requests Received** and **Requests Directed** fields in [Table 5–4](#) for details.

[Table 5–3](#) briefly describes each of these fields displayed on the **Summary** section.

## Summary section

In the **Summary** section, the **Number of Brokers** field displays a total of all the brokers currently registered with a specific resource. The number of unique brokers identified in each of the separate **AppService** detail sections equals the number displayed in the **Number of Brokers** field.

Table 5–3 identifies and describes all the **Summary** fields and their display-only details. Most of the values that display on this page are originally defined in the `ubroker.properties` file.

**Table 5–3: Summary details on the Operational Status page** (1 of 2)

This field...	Displays the...
<b>Host</b>	NameServer's host machine name.
<b>Port</b>	Number of the UDP Port that the NameServer uses to listen for client connection requests and registration messages from AppServers, and Transaction Servers.
<b>Timeout</b>	<p>A value, in seconds, that indicates how often the NameServer checks for WebSpeed or AppServer broker instances that have timed out.</p> <p>When a WebSpeed or AppServer broker instance registers with a NameServer, the instance indicates how often it will send “keep-alive” messages by setting a registration retry value (a property setting in the <code>ubroker.properties</code> file). Once a NameServer determines that it has not received a “keep-alive” message from a broker instance within the broker's registration retry time, the NameServer automatically unregisters the instance.</p>
<b>Start time</b>	Date and time stamp when the NameServer started. Any time the NameServer is restarted, this field will be updated to display the NameServer's most recent start time.
<b>Number of AppServices</b>	Number of Application Services associated with this NameServer instance. The count associated with this display field matches the number of Application Services listed in the detailed <b>AppService</b> sections in this view.
<b>Number of Brokers</b>	<p>Number of brokers in the broker pool that are currently registered with this NameServer instance, directing client connection requests to a requested Application Service.</p> <p>A broker might register more than one Application Service with a NameServer instance. Therefore, it might appear several times in the <b>AppServices</b> detail section. However, the broker will only count as 1 towards the total number of brokers recorded in this field.</p>



**Table 5–3: Summary details on the Operational Status page** (2 of 2)

This field...	Displays the...
<b>Total client requests received</b>	Total number of client requests received by the NameServer since it was started. Any time the NameServer is restarted, this field will be reset to display a request total relative to the NameServer's most recent start time.
<b>Total client requests rejected</b>	<p>Total number of times that a client requested a broker for an Application Service that the NameServer had no knowledge of and therefore a client could not be matched up with a registered broker. This value identifies real-time client requests.</p> <p>Data related to any requests that the NameServer passes to NameServer Neighbors with which it typically works are not captured in this total.</p> <p>Any time the NameServer is restarted, this field will be reset to display a count relative to the NameServer's most recent start time.</p>

### Application Services detail section

For each Application Service (**AppService**) currently identified to the NameServer, there is a unique table of displayed values that appears on the **Operational Status** page. [Table 5–4](#) briefly describes each of these fields. Also, note these additional points about the relationship of these fields as they relate to each other and to data presented in the **Summary** section:

- An individually registered broker can support multiple Application Services. Therefore, you might see several **AppService** detail sections associated with a NameServer instance, but only a small total number reported in the **Number of Brokers** field in the **Summary** section.
- In an **Application Services** detail section, the value displayed in the **Requests Received** field reflects a total number of requests for this service. However, each broker identified as supporting a client request to an Application Service maintains its own individual **Requests Directed** total.
- If an active broker goes down or is unavailable, any **AppService** detail sections uniquely associated with that broker will no longer display. If the broker re-registers, its total **Requests Directed** count will be reset to zero.

Table 5–4 identifies and describes the fields and their display-only details that appear in each AppService section of the **Operational Status** page. Many of these values originate from the configuration settings stored in the `ubroker.properties` file.

**Table 5–4: NameServer details on the Operational Status page**

This field...	Displays the...
<b>Requests Received</b>	Number of client requests received for this Application Service. This count is maintained when one or more brokers are registered to support the Application Service. The count is reset when the Application Service is first identified to the NameServer.
<b>Broker</b>	Name of the broker that is capable of fulfilling the connection between the requesting client and the Application Service.  When more than one broker is servicing the same Application Service, each new broker's data is appended to the individual <b>AppService</b> details section.
<b>Host</b>	Broker's host machine name and numeric address.
<b>Weight</b>	Priority weight assigned to the Unified Broker instance for the purpose of load balancing.
<b>Requests Directed</b>	Total number of client connection requests for the Application Service as directed by the NameServer to the broker instance.  This count is maintained while the broker remains registered. If the broker is stopped or timesout, the broker's count is reset to zero when the broker next runs. There is a separate counter for each Application Service that the broker supports.
<b>UUID</b>	Unique number for the Unified Broker instance.
<b>Port</b>	TCP/IP port number that the broker listens on to pick up client connection requests.
<b>Timeout</b>	Amount of time, in seconds, that elapses between the "keep alive" messages that the broker sends to the resource as part of a broker's registration retry entry process.

## Accessing and reviewing Informational views

The **Properties** link in the **Informational views** section allows you to access static configuration details about a specific NameServer instance.



To access and review **Properties** details:

1. Display the **NameServer main** page for the instance that you want to start. See the [“Accessing OpenEdge resource information”](#) section on page 2–12 for the detailed steps.
2. Click **Properties** in the **Informational views** section to display the **Static Configuration** page:

NameServer: chimay.NS1 Static Configuration		
Properties		
Location	NameServer location:	local
	Host name:	chimay
	Port number:	5162
General	Working directory:	/usr1/besg/progress/wrk
	Broker keep alive timeout:	30
	Autostart:	1
Logging	Server log filename:	/usr1/besg/progress/wrk/NS1.ns.log
	Logging level:	2
	Append to log file:	1
Advanced	Neighboring NameServers:	
Environment	MYENV	MYVAR

This view is comprised of a single **Properties** section that displays fields and values previously defined in the `ubroker.properties` file. These values are derived at startup. [Table 5–5](#) describes the contents of this section.

**Table 5–5: Properties details on the Static Configuration page** (1 of 2)

This field...	Displays the...
<b>Location</b>	<p>Specific values that pertain to these fields:</p> <ul style="list-style-type: none"><li>• <b>NameServer location</b> — Indicates whether the NameServer is <i>local</i> or <i>remote</i>. A local service identifies a NameServer instance that runs locally on the selected host. A remote service runs remotely on a network machine that is separate from the selected host.</li><li>• <b>Host name</b> — Identifies the name of the host machine.</li><li>• <b>Port number</b> — Identifies the number of the UDP port that the NameServer uses to listen for client connection requests and registration messages from AppServers and Transaction Servers.</li></ul>
<b>General</b>	<p>Specific values that pertain to these fields:</p> <ul style="list-style-type: none"><li>• <b>Working directory</b> — Identifies the NameServer working directory, including the pathname.</li><li>• <b>Broker keep alive timeout</b> — Identifies a value, in seconds, that indicates how often the NameServer should check for Unified Broker instances that have timed out.  When a Unified Broker instance registers with a NameServer, the instance indicates how often it will send “keep-alive” messages by setting a registration retry value. Once a NameServer determines that it has not received a “keep-alive” message from a Unified Broker instance within the broker's registration retry time, the NameServer automatically unregisters the instance.</li><li>• <b>Autostart</b> — Indicates whether the NameServer will start automatically when the controlling AdminServer starts. If the value 1 displays, Autostart option is set. If the value zero displays, then the Autostart option is not set.</li></ul>

**Table 5–5: Properties details on the Static Configuration page** *(2 of 2)*

This field...	Displays the...
<b>Logging</b>	<p>Specific values that pertain to these fields:</p> <ul style="list-style-type: none"><li>• <b>Server log filename</b> — Identifies the NameServer log filename, including the pathname.</li><li>• <b>Logging level</b> — Displays one of three possible values to specify the amount of information to be written to the server log: Error only, Terse, or Verbose.</li><li>• <b>Append to log file</b> — Indicates if a new NameServer log file will be created when the NameServer is started. A 1 indicates that log entries will be appended to the existing NameServer log file.</li></ul>
<b>Advanced</b>	<p>Specific value that pertains to this optional field:</p> <ul style="list-style-type: none"><li>• <b>Neighboring NameServers</b> — Identifies a list of selected NameServers to which this NameServer can forward connection requests for Application Services that are not registered with it (that is, the Application Service name is unknown).</li></ul>
<b>Environment</b>	<p>Specific NameServer environment variables that are defined for the process in which the NameServer executes.</p>



---

## Monitoring Plans and Rules for OpenEdge Server Resources

---

You use Fathom's monitoring capabilities to monitor OpenEdge resources as you do other Fathom resource types. This chapter highlights OpenEdge-specific resource monitoring details, tasks, and related activities.

Specifically, this chapter presents:

- [Fathom resource monitoring overview](#)
- [Default polling and trend values](#)
- [General rule conventions](#)
- [Maintaining monitoring plans](#)
- [Understanding and using resource monitor rules](#)
- [Working with rule sets](#)

For an in-depth discussion of the Fathom resource monitoring feature and the various components that comprise a resource monitoring plan, see the *Resource Monitoring Guide*. For complete details about alerts, see the *Alerts Guide and Reference*.

## Fathom resource monitoring overview

Fathom uses active monitoring plans and their associated rules to support many fundamental resource-related features, including:

- Data trending
- Rule evaluation
- Data analysis
- Alert notification

When Fathom discovers any of the OpenEdge resource types, it automatically creates a resource monitoring plan. The default values Fathom provides includes a default name for the resource, and default values for each individual resource's monitoring plan and its associated rule set.

For example, if Fathom discovers a NameServer resource whose server name is **NS2**, then it creates a monitoring plan called the **NS2 monitoring plan** and immediately associates the default NameServer rule set with **NS2**. (Since Fathom supports changing any Fathom resource monitoring plan, you can edit or modify any OpenEdge monitoring plan and rules, setting your own values at any time.)

Other resource types that Fathom can recognize—database, system, network, and file resources—also require monitoring plans and rules with which these standard resource monitoring activities can occur. Within the Fathom Management framework, all Fathom resources share standardized ways to perform various operations and a common terminology with which to reference these resource activities.

Review the resource monitoring features, and key terms and concepts about resource monitoring highlighted in this section; this information will help orient you to the basics of Fathom resource monitoring. Then, perform the procedures outlined in the “[Maintaining monitoring plans](#)” section on page 6–13 and the “[Understanding and using resource monitor rules](#)” section on page 6–20 to actually apply these concepts to work with the WebSpeed brokers, AppServer brokers, and NameServer resources.



## Key terms and definitions

This section highlights some important terms and concepts to help you immediately begin working with OpenEdge resource monitoring plans and rules. For more detailed information about this terminology, see the [Resource Monitoring Guide](#).

Fathom resource monitoring terms include:

- **Resource** — A specific component of your configuration such as a WebSpeed broker, AppServer broker, or NameServer instance. Other Fathom resources include CPU, memory, system, network, and OpenEdge database.
- **Resource monitoring** — Criteria set up to monitor a resource's performance. As necessary, you can adjust the criteria according to your specific performance expectations.
- **Monitor** — As specifically addressed in this guide, the combination of an OpenEdge resource, schedules, and rules. You can monitor any of these OpenEdge server-related resources:
  - AppServer brokers.
  - AppServer broker log files.
  - AppServer server log files.
  - WebSpeed brokers.
  - WebSpeed broker log files.
  - WebSpeed agent log files.
  - NameServers.
  - NameServer log files.

A *schedule* defines a block of time in Fathom (for example, weekdays), and a *rule* (for example, the ReregisteredBroker rule) determines how a resource's performance is judged. For example, the AgentMemoryUsageHigh rule determines when the memory usage of a WebSpeed agent process exceeds the specified threshold.

- **Rules** — The resource monitoring component that Fathom checks to verify whether a resource complies with its performance criteria. Rule values, or settings, can be established using Fathom-supplied default values or user-defined values. Also, WebSpeed and AppServer brokers can optionally use calculated, resource-specific baseline rule values as determined by the Configuration Advisor.

Rules are broken when a resource is not in compliance with the rule-based criteria that you set up; Fathom generates alerts in the management console to alert you to this fact.

- **Rule Set** — A combination of rules.
- **Defaults and default values** — Values that are predefined in Fathom in one location but can apply in another location. Resource monitoring plans contain several default values. Some of the more general, common defaults pertaining to resource monitors include default schedule, default alerts, and actions. These defaults help expedite the setup tasks associated with configuring a monitoring plan. Also, there are default values associated with a given resource type. These types of default values include polling intervals and rule sets. See the [“Default polling and trend values”](#) section on page 6–5 for details.
- **Schedule** — Defines the block of time when a set of monitoring rules is active for a resource. When you add a monitoring plan to a resource, you specify the schedule to indicate when the monitoring plan will be active. Fathom supports using, modifying, and copying predefined schedules to help you define schedules quickly. However, you can also create new schedules to suit your operating needs.
- **Alerts** — Notifications that some specified activity has occurred regarding an actively monitored resource. Alerts can occur to indicate a real or potential problem exists, such as a rule violation, or that a typical or interesting activity regarding a resource has occurred.
- **Actions** — Activities that are triggered in response to alerts. For example, you might specify that you receive an e-mail if a WebSpeed agent is trimmed.
- **Resource monitoring plan** — A plan that defines a block of time during which a specific resource is to be monitored and identifies the rules that are to be checked during the specified time frame. All resources you create in Fathom must have one or more monitoring plans before Fathom can monitor the resource. Monitoring plans provide you with access to data that is immediately usable in indicating performance failures, giving you an opportunity to improve performance and report on trends gathered over a period of time specified by you.

- **Baseline value** — As used in this guide, a number that serves as the base for calculating a set of possible threshold settings based on your system’s past activity for a specific rule. The Configuration Advisor determines a baseline value as part of its data analysis process to calculate recommended rule threshold settings for specific WebSpeed and AppServer rules. See [Chapter 7, “Calculating Rule Threshold Settings Using the Configuration Advisor.”](#)

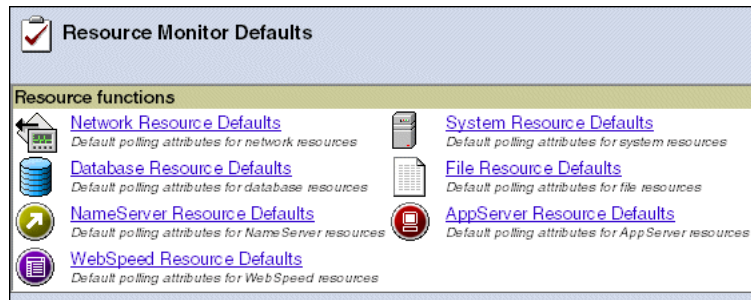
## Default polling and trend values

During the OpenEdge server discovery process, Fathom creates a resource monitoring plan for each AppServer broker, WebSpeed broker, and NameServer instance that it discovers. At that time, each resource inherits and displays default polling and trending values as defined for that specific resource type on the **Resource Monitor Defaults** page.

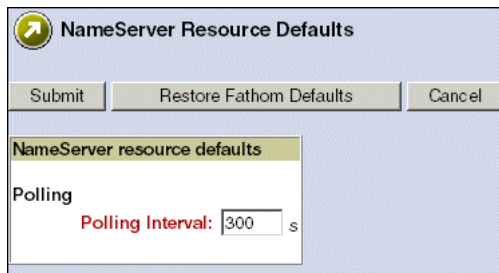


To display OpenEdge resource-specific default values:

1. Select **Resources** from the menu bar. On the **Fathom Resources** page, click **Resource Monitor Defaults**. The **Resource Monitor Defaults** page appears:

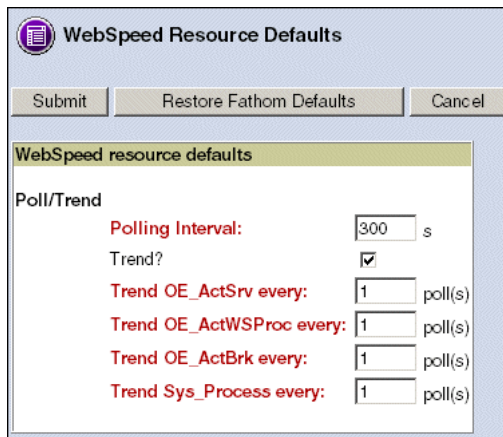


2. Click the link associated with the specific resource default values you want to review: **NameServer Resource Defaults**, **WebSpeed Resource Defaults**, or **AppServer Resource Defaults**. The associated **OpenEdge Resource Defaults** page appears as described:
  - a. If you click **NameServer Resource Defaults**, the **NameServer Resource Defaults** page appears:



The screenshot shows the "NameServer Resource Defaults" dialog box. It has a title bar with a green circular icon and the text "NameServer Resource Defaults". Below the title bar are three buttons: "Submit", "Restore Fathom Defaults", and "Cancel". The main content area has a yellow header "NameServer resource defaults". Underneath, the word "Polling" is displayed. Below "Polling" is the text "Polling Interval:" followed by a text input field containing "300" and a small "s" for seconds.

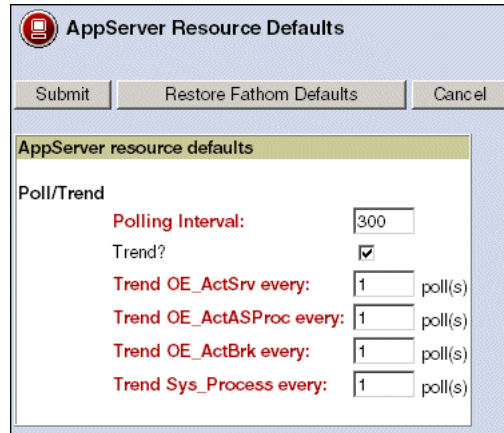
- b. If you click **WebSpeed Resource Defaults**, the **WebSpeed Resource Defaults** page appears:



The screenshot shows the "WebSpeed Resource Defaults" dialog box. It has a title bar with a purple circular icon and the text "WebSpeed Resource Defaults". Below the title bar are three buttons: "Submit", "Restore Fathom Defaults", and "Cancel". The main content area has a yellow header "WebSpeed resource defaults". Underneath, the text "Poll/Trend" is displayed. Below "Poll/Trend" are several settings:

- "Polling Interval:" followed by a text input field containing "300" and a small "s" for seconds.
- "Trend?" followed by a checked checkbox.
- "Trend OE\_ActSrv every:" followed by a text input field containing "1" and the text "poll(s)".
- "Trend OE\_ActWSPProc every:" followed by a text input field containing "1" and the text "poll(s)".
- "Trend OE\_ActBrk every:" followed by a text input field containing "1" and the text "poll(s)".
- "Trend Sys\_Process every:" followed by a text input field containing "1" and the text "poll(s)".

- c. If you click **AppServer Resource Defaults**, the **AppServer Resource Defaults** page appears:



The screenshot shows a web-based configuration window titled "AppServer Resource Defaults". At the top, there are three buttons: "Submit", "Restore Fathom Defaults", and "Cancel". Below the buttons is a section header "AppServer resource defaults". Under this header, the "Poll/Trend" section is visible. It contains the following settings:

- Polling Interval:** A text input field containing the value "300".
- Trend?** A checkbox that is checked.
- Trend OE\_ActSrv every:** A text input field containing "1" followed by "poll(s)".
- Trend OE\_ActASProc every:** A text input field containing "1" followed by "poll(s)".
- Trend OE\_ActBrk every:** A text input field containing "1" followed by "poll(s)".
- Trend Sys\_Process every:** A text input field containing "1" followed by "poll(s)".

---

**Note:** OpenEdge-specific resource log file monitor defaults are also set on the **File Resource Defaults** page. For information about the AppServer-, WebSpeed-, and NameServer-specific log file monitors, see the appropriate chapter in this guide.

---

3. Change the default values, as necessary. Individual resources created from these categories inherit the updated default values. However, you can still override values for individual resources.

Note that you can revert back to the original Fathom-supplied default values at any time by clicking **Restore Fathom Defaults**.

### Trend default values for WebSpeed and AppServer brokers

Data for rule evaluation, graphical displays, and reports is not available unless brokers are configured to collect and trend data to the FathomTrendDatabase and to poll.

Before you can use either data collection or the Configuration Advisor feature successfully, you must set up these options:

- **Trend**
- **Polling**

**Note:** See the “[Data collection details](#)” section on page 3–9 for details about how to implement data collection with WebSpeed brokers. See the “[Data collection details](#)” section on page 4–9 for details about how to use data collection with AppServer brokers. See [Chapter 7, “Calculating Rule Threshold Settings Using the Configuration Advisor](#)” for details about data collection and polled rules with WebSpeed and AppServer brokers.

## Default monitoring plan details

Using default values helps you standardize and simplify your resource monitoring tasks. You can begin using many of the features of Fathom resource monitoring right away.

This section:

- Identifies each resource monitoring plan’s fields and associated default values that are common to all OpenEdge resource types.
- Provides an example of each OpenEdge default monitoring plan.

## Monitoring plan default values

[Table 6–1](#) identifies and describes the common monitoring plan default values that the OpenEdge resource types use. A default value defined as **Enabled**, **Selected**, or **True** indicates that a check mark is associated with that field to indicate that the option is set.

**Table 6–1: Monitoring plan default values** (1 of 2)

Field	Default value	Description
Schedule	Default Schedule Plan	Identifies the system-defined 24/7 default schedule used when the plan is active. This default plan is the same for all Fathom resources.
Poll	5 minutes	Identifies the polling cycle, which is the frequency at which the resource’s rules are checked, set up for each individual Fathom resource monitor.

**Table 6–1: Monitoring plan default values***(2 of 2)*

Field	Default value	Description
<b>Enabled</b>	<b>Selected</b> (Check mark displayed.)	Specifies whether the OpenEdge resource is enabled (started). When any OpenEdge server resource is discovered, Fathom automatically creates it with the Enabled property selected (check mark displayed).
<b>Alerts</b>	<b>Enabled</b>	Indicates whether alerts are active and will be generated when the plan is active.
<b>Trend</b>	<b>True</b> (This value is set to on.)	Indicates whether the statistical data monitored while the plan is active will (or will not) be stored to the FathomTrendDatabase.  The <b>Trend Performance Data</b> field value is involved in setting data collection.
<b>Rule Summary</b>	Default rule set for the specific OpenEdge resource	There is a default rule set for each type of OpenEdge resource. The following list identifies each rule in the respective default rule sets: <ul style="list-style-type: none"> <li>• AppServer brokers — <b>Broker Abnormal Shutdown.</b></li> <li>• WebSpeed brokers — <b>Broker Abnormal Shutdown.</b></li> <li>• NameServers — <b>Abnormal Shutdown, Broker Timeout, and Duplicate Broker UUID.</b></li> </ul>

## Default Schedule details

Fathom provides one default monitoring plan, entitled the **Default\_Schedule\_Plan**, per OpenEdge resource. However, when you update a monitoring plan, you can add different plans to monitor different resource activities.

---

**Note:** Fathom prevents the assignment of schedules that share overlapping time periods. For example, if you have a Default Schedule set up for a resource monitor, you cannot set up an additional plan because the **Default\_Schedule** is defined for 7 days a week, 24 hours a day. You must modify the **Default\_Schedule** or remove it from the plan in order to add any other plans.

---

Each OpenEdge resource that Fathom discovers will automatically have its own default monitoring plan and associated rule set established. You can change these default values at any time using the standard resource monitoring procedures.

## Default values in the Rule Summary

All monitoring plans also includes a Rule Summary. The *Rule Summary* is a list of the rules and rule sets that are applied to the particular monitoring plan. Fathom automatically applies the default rule set associated with a specific OpenEdge resource to a plan. [Table 6–1](#) also identifies the rules that comprise each of these specific rule sets.



## OpenEdge default monitoring plan examples

This section shows some of the key components of a monitoring plan as they appear on each of the NameServer, AppServer, and WebSpeed resource’s summary monitoring page. The purpose of these examples is to show the default values that are automatically applied when a resource is discovered, highlighted by the default plan and associated default rule set.

### NameServer default monitoring plan example

Figure 6–1 depicts a NameServer default monitoring plan and associated default rules example. It shows the default plan and rule set for a NameServer named **NS1**.

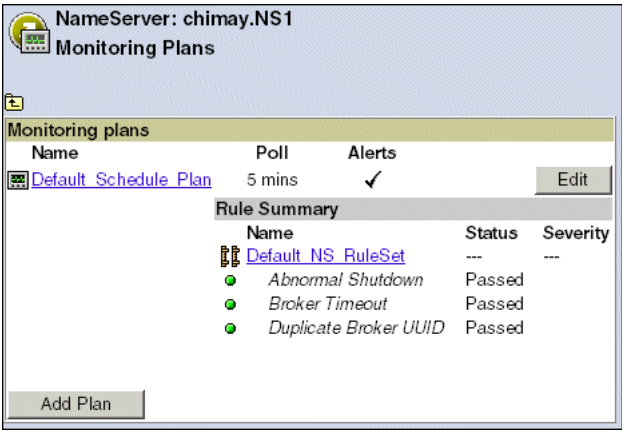


Figure 6–1: NameServer instance default monitoring example

AppServer default monitoring plan example

Figure 6–2 shows another default monitoring plan and associated default rules example. It shows the default AppServer broker-related plan and rule set. With the exception of the unique rule sets, the details displayed are identical between the AppServer and WebSpeed broker resources.

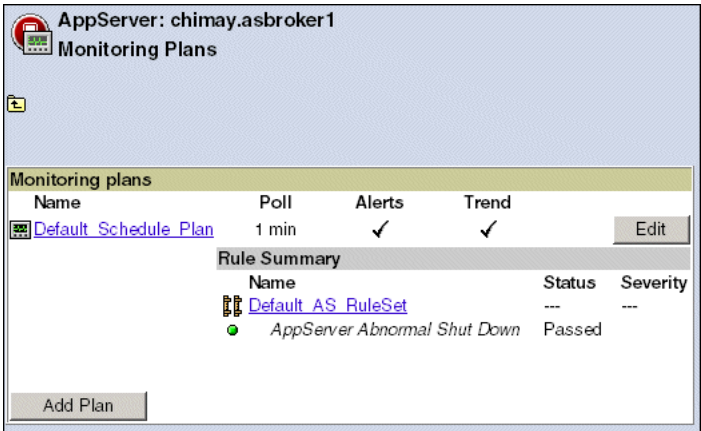


Figure 6–2: AppServer broker default monitoring plan example

WebSpeed default monitoring plan example

Figure 6–3 shows another default monitoring plan and associated default rules example. It shows the default WebSpeed broker-related plan and rule set. With the exception of the unique rule sets, the details displayed are identical between the AppServer and WebSpeed broker resources.

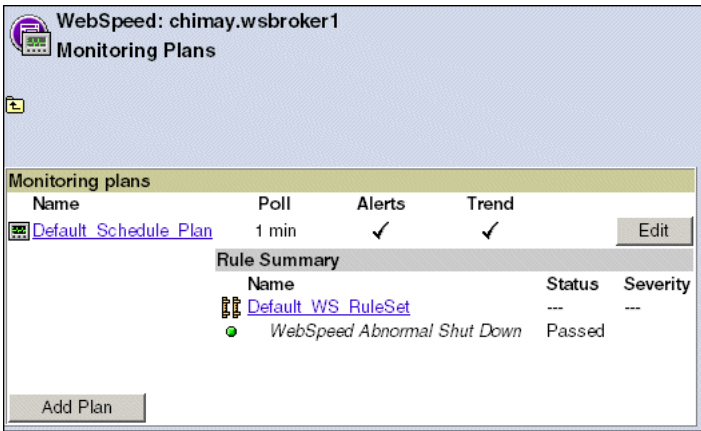


Figure 6–3: WebSpeed broker default monitoring plan example

## Maintaining monitoring plans

The basic tasks you use to create resource monitoring plans for any Fathom resource have been standardized. In the case of OpenEdge resources, these tasks share an even greater sense of commonality due to the similarity in these resource types and their monitoring goals.

This section describes how to access and update a default monitoring plan and its associated rules, using the AppServer broker resource as the example. These same steps can be applied to WebSpeed broker and a NameServer instance; any unique requirements for a given OpenEdge resource type are noted.

For additional details about rules and rules sets, see the [“Understanding and using resource monitor rules”](#) section on page 6–20, and the [“Working with rule sets”](#) section on page 6–29.

## Updating monitoring plans

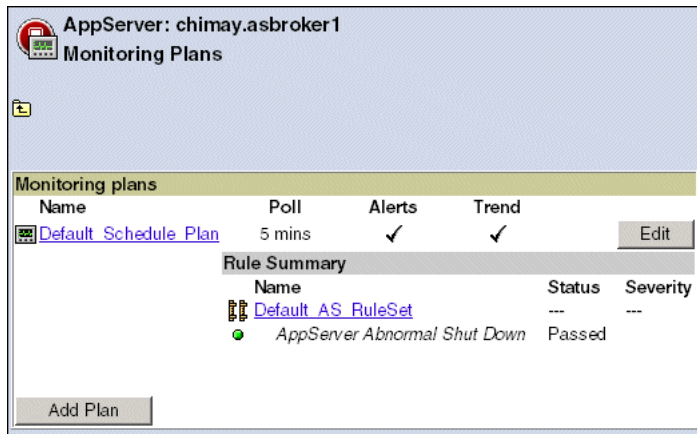
This section describes how to access and update a monitoring plan and associated rules using the AppServer broker resource as the example. These same steps can be applied to WebSpeed broker and a NameServer instance; any unique requirements for a given OpenEdge resource type are noted.



### To update an OpenEdge resource monitoring plan:

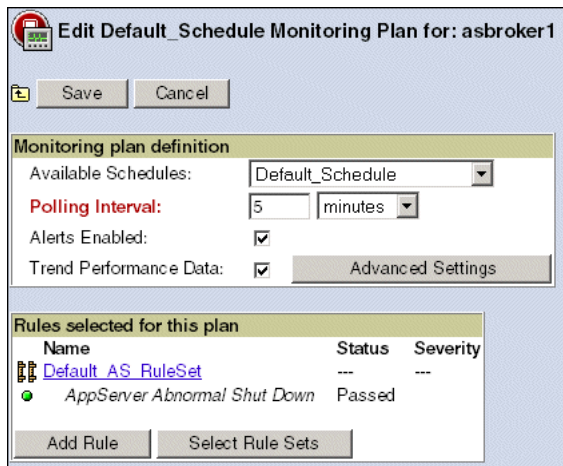
1. Click **Resources** in the menu bar.
2. In the list frame, navigate to the specific OpenEdge resource you want to display. See the [“Accessing OpenEdge resource information”](#) section on page 2–12. In this example, the AppServer broker resource, **asbroker1**, is selected.

- Click **Monitoring Plans** in the **Command and control** section of the OpenEdge main page. In this example, the **Monitoring Plans** page for AppServer broker resource **asbroker1** appears:



- Select the specific schedule associated with the plan that you want to update. Click **Edit** associated with that specific plan.

The figure below shows the page that appears in edit mode when the **Default Schedule Plan** is selected:



5. Update the monitoring plan values, as needed:
  - a. Change current values in these fields: **Available Schedules**, **Polling Interval**, or **Alerts Enabled**. (Available to all OpenEdge resources.)
  - b. Change the value in the **Trend Performance Data** option. (Available to WebSpeed and AppServer broker resources only.) For details, go to [Step Note](#).
  - c. Click **Advanced Settings**. (Available to WebSpeed and AppServer broker resources only.) For details, go to [Step Note](#).
  - d. Select a specific rule or rule set to add, update, or remove from this plan. For details, go to [Step 6](#).


---

**Note:** For WebSpeed and AppServer broker resources, you can also change the setting of the **Trend Performance Data** option. However, note that this option is required to ensure that data gathered using data collection is trended to the FathomTrendDatabase. For WebSpeed broker-related details, see the [“Properties section”](#) section on page 3–7. For AppServer broker-related details, see the [“Properties section”](#) section on page 4–6.

---

- e. Change the setting of the **Trend Performance Data** option. However, note that this option is required to ensure that data gathered using data collection is trended to the FathomTrendDatabase. For WebSpeed broker-related details, see the [“Properties section”](#) section on page 3–7. For AppServer broker-related details, see the [“Properties section”](#) section on page 4–6.

- f. Click **Advanced Settings** to see all trend value settings, as shown in this example:

**Edit Default\_Schedule Monitoring Plan for: asbroker1**

Save

Cancel

Monitoring plan definition

Available Schedules:

Default\_Schedule

Polling Interval:

5

minutes

Alerts Enabled:

☒

Trend Performance Data:

☒

Hide Advanced Settings

Trend OE\_ActSrv every:

1

poll(s)

Trend OE\_ActASProc every:

1

poll(s)

Trend OE\_ActBrk every:

1

poll(s)



Trend Sys\_Process every:

1

poll(s)

Reset Trend Defaults

Rules selected for this plan

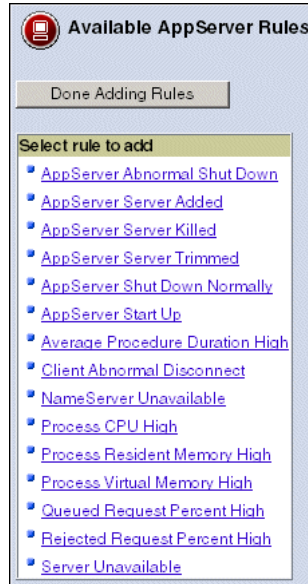
Name	Status	Severity
 <a href="#">Default_AS_RuleSet</a>	---	---
 AppServer Abnormal Shut Down	Passed	

Add Rule

Select Rule Sets

6. Click the individual rule to display details about that rule, including alert severity, action to perform upon the firing of the alert, and a brief description of the rule.

For example, click **Add Rule** in the rules selected for this plan. The **Available Rules** page for rules that are unique to the OpenEdge resource displays. In this example, the rules associated with an Appserver broker resource appear:



The **Available Rules** page is a dynamic list; it displays only those rules not yet applied to a given monitoring plan.

---

**Note:** The step to select rules for each OpenEdge resource is the same. However, each OpenEdge Resource has a unique **Available Rules** page. For more information about each set of rules, see the “[Understanding and using resource monitor rules](#)” section on page 6–20.

---

7. Click the rule you want to add. If you select **Process CPU High**, for example, the detailed rule information shown in the figure below displays. Note the rule's description on the bottom of the rule page:

**Rule: Process CPU High**  
Resource: asbroker1 Schedule: Default Schedule

Save Cancel

Threshold: 80.0 percent  
Alert severity: Error  
Throw alert after: 1 failed poll(s)  
On alert perform action: None  
Clear alert after: 0 successful poll(s)  
On clear perform action: None

**Rule description**  
A measure of how much CPU a single process is using.

8. Update any unique values you want to define for this instance of the rule. Note that using this procedure as a guide, none of the steps in this procedure required you to enter values for these fields. Although these fields serve different purposes, they all can display default values.

Also, this rule, as defined, is associated only with this particular plan. When you update another plan with the same rule, you can select values that are appropriate for that particular plan.

The **Threshold** field associated with this page indicates the actual rule criterion. For details about rules, see the [“Understanding and using resource monitor rules”](#) section on page 6–20. The remaining fields on this page are alert- and action-related fields. For details, see the [Alerts Guide and Reference](#).

9. Click **Save**. The **Available Rules** page reappears. Repeat [Step 7](#) and [Step 8](#) for each additional rule you want to apply to this plan. After you add and define the criteria for each rule you want to add, click **Done Adding Rules** on the **Available Rules** page.
10. Click **Select Rule Sets** in the rules section of the page to choose the rule sets you want to add to the monitoring plan. Fathom displays the default rule set for the resource type you are updating, and any additional rule sets created using the **Fathom Component Library**. See the [“Working with rule sets”](#) section on page 6–29 for details.
11. Click **Save**. The updated monitoring plan appears in the monitoring plan definition on the top of the **Monitoring Plan summary** page.



## General rule conventions

For each rule, the following details are provided:

- A colored dot, preceding the rule name, that indicates the status associated with each rule. See [Table 6–2](#) for a description of each status.
- The alert severity for each rule if the rule fails.
- The action to take place when the alert fires.

See the *Resource Monitoring Guide* for more resource status information.

**Table 6–2: Resource status legend**

(1 of 2)

This status...	Has a color indicator of...	To identify that...
<b>Pass</b>	Green	The resource monitor is currently working.
<b>Fail</b>	Red	<p>The most recent test involving the associated resource failed. For some resources, such as network, this includes statuses such as tardy, timeout, unreachable, and so forth.</p> <p>Check the <b>Alert Summary</b> page or the specific monitor for possible alert details.</p> <p>This status can also identify an internal error that prevents the resource from being monitored.</p>
<b>Not Running</b>	Blue	This resource is currently not running. This status is particularly informative as it applies to resources such as the OpenEdge databases and servers that must be operating before you can monitor them.
<b>Not Checked</b>	Yellow	The resource monitor's status is currently unknown. For example, if system startup has just occurred, it is possible that the resource has not yet been polled.

**Table 6–2: Resource status legend** (2 of 2)

This status...	Has a color indicator of...	To identify that...
Disabled	Dark Gray	The resource monitor has been disabled and is not currently monitoring a resource.
Inactive	White	There is no active monitoring plan.
Offline	Light Gray	The resource is currently offline.

## Understanding and using resource monitor rules

The concept of a rule as it applies to OpenEdge resource monitors is identical to that expressed by the specific rules for other resource monitor types: a rule is the resource monitoring component that Fathom checks to verify whether a resource complies with an expected performance criteria. Certain rules specific to WebSpeed and AppServer, as this section identifies, can also use the Configuration Advisor to generate intelligent threshold values based on an analysis of data collected for a given rule.

In addition to the rules identified as default rule sets in the [“Updating monitoring plans”](#) section on page 6–13, you can also choose from different individual resource-specific rules and define them for a monitoring plan.

### Common rule characteristics

The following characteristics are common to all rules, regardless of their individual resource type:

- Only the rules that are not already part of the monitoring plan appear in each resource types **Available Rules list**.
- When you select any of the rules available on the specific available rules list, the particular criteria associated with each rule appear.
- You can modify the default values associated with each individual rule.

- To display the rules available for each OpenEdge resource type, click **Rules on the monitoring** page when it is displayed in edit mode. See the procedure in the “[Maintaining monitoring plans](#)” section on page 6–13 for the details of this task.
- If a rule is part of a monitoring plan and a member of a rule set, the individual rule definition supersedes the rule in the rule set.

As with all Fathom resource monitoring rules, if the Alert-related options are enabled for an OpenEdge monitoring plan, any rule violation causes an alert to trigger. See the [Alerts Guide and Reference](#) for detailed information about Fathom alert types and rules, and specific definitions about the alert feature’s relationship with each rule.

The following sections identify each of these unique rules, and provide additional details about rules and rule processing for each resource type.

## NameServer rules

Figure 6–4 shows the **Available Rules** page as it displays the complete list of NameServer rules.



**Figure 6–4:** Available NameServer Rules page

## AppServer rules

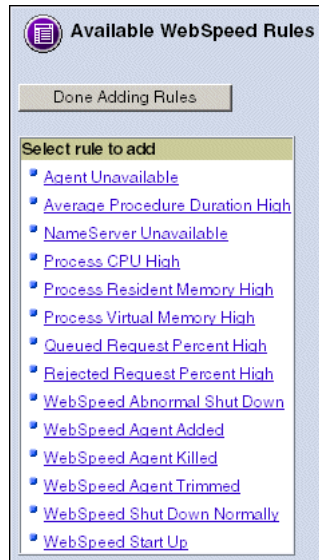
Figure 6–5 shows the **Available Rules** page as it displays the complete list of AppServer broker rules.



**Figure 6–5:** Available AppServer Rules page

## WebSpeed rules

Figure 6–6 shows the **Available Rules** page as it displays the complete list of WebSpeed broker rules.



**Figure 6–6:** Available Webspeed Rules page

## Average Procedure Duration High rule

The WebSpeed and AppServer lists of available rules include **Average Procedure Duration High**. This rule measures the average duration of a 4GL procedure run by a server, or agent, process. This average is calculated based on the polling interval set for the resource, not the average for the lifetime of the broker.

### Calculating the average duration for a procedure

The average is determined by the sum of time noted for a procedure name to run divided by the total number of times the procedure ran. The data used to determine this average is collected during a polling interval. This calculated result is then compared to the threshold defined for the procedure name.

Since this calculation determines an average based on data collected for each procedure, an individual spike will not necessarily skew the average. The rule's algorithm is designed to eliminate these spike conditions, minimizing unnecessary alerts.

---

**Note:** The **Procedure Duration High rule** only measures the execution time of the Progress 4GL procedure from the server's, or agent's, viewpoint. The time measure does not include network and client processing overhead.

---

### Accessing the Average Procedure Duration High rule page

The **Procedure Duration High rule** page is accessible from the **Available Rules** page. On this page, you specify the specific procedures you want to measure, setting the average duration threshold in milliseconds. You can also set alert and action criteria.

### Supplying data for Progress 4GL procedures and WebSpeed Transaction Servers

For Progress 4GL procedures related to the Transaction Server, you must reference the CGI environment variable as defined in the `Value` of `PATH_INFO` on the URL. Enter this string in the **Procedure** field to identify the name of a Progress 4GL procedure, entering one procedure on one line. These procedures will generally be file types such as `.p`, `.w`, or `.html`. The following example shows the type of information required to measure a WebSpeed procedure using the following URL:

```
http://hostname/scripts/cgiip.exe/src/web/exmaples/status.p
```

The procedure name that is executed is the `PROPATH` relative name `src/web/examples/status.p`. This is the value of the CGI environment variable `PATH_INFO`.

## Supplying data for 4GL procedures and AppServers

Appserver 4GL procedures run on the 4GL RUN statement based on an AppServer connection handle. The procedures on the RUN statement can reference PROPATH relative directories, unqualified procedure names, internal procedures, and user-defined functions. To measure the duration of specific AppServer procedures, enter the procedure name in the **Average Duration High Rule** page exactly as it is referenced on the RUN statement.

Table 6–3 identifies and describes three examples.

**Table 6–3: Examples of AppServer-related 4GL procedure entries**

This 4GL procedure entry...	Runs a procedure that...
RUN myOrders.p ON SERVER hAppSrv	Is PROPATH relative. This entry refers to a procedure that is located in a directory or procedure library specified on PROPATH.
RUN myApp/myAccounts.p ON Server hAppSrv	Is PROPATH relative. This entry refers to a procedure that is located in the subdirectory called myApp that is relative to PROPATH.
RUN processOrder IN hProc	Shows the execution of an internal procedure.

To measure any of the example procedures noted in Table 6–3 using the **Average Procedure Duration High** rule, you must enter the procedure name exactly as it appears on the RUN statement.

## Rejected Request Percent High rule

The WebSpeed and AppServer lists of available rules include the **Rejected Request Percent High** rule. This rule highlights the percentage of client requests rejected during a polling interval set for either a WebSpeed agent or AppServer Server. It determines violations based on the number of initiated requests that exceed the defined threshold setting. You can review this information to determine processing bottlenecks or tuning problems. Based on your processing needs, you can then adjust your threshold setting to help minimize the impact of these problems from preventing client requests from being serviced.

### Determining the percentage of rejected requests

The percentage of rejected requests for either a WebSpeed resource or an AppServer resource is determined by a formula that compares data from the previous poll period with data from the most current poll period; the result is always calculated on a per-poll-period basis.

This rule subtracts the number of requests rejected during the current poll from the number of requests rejected during the previous poll to determine the number of new rejected requests for the current poll period. Also, this rule subtracts the number of requests received during the current poll from the number of requests rejected during the previous poll to determine the number of new received requests for the current poll. Then, the number of requests rejected is divided by the number of requests received to determine the percentage of requests rejected during this poll period.

### Accessing the Rejected Request Percent High rule page

The **Rejected Request Percent High** rule page is accessible from the **Available Rules** page. On this page, you specify the threshold value, in percent form, that identifies the number of rejected client requests during the polling interval that you consider acceptable. Any rejected requests that exceed this value will cause the alert and action criteria that you also set on this page to be triggered.

## Queued Request Percent High rule

The WebSpeed and AppServer lists of available rules include the **Queued Request Percent High**. This rule highlights the percentage of client requests queued during a polling interval set for either a WebSpeed agent or AppServer server. This rule determines violations based on the number of queued requests that exceed the defined threshold setting. You can review this information to determine processing bottlenecks or tuning problems. Based on your processing needs, you can then adjust your threshold setting to help minimize the impact of these problems.

### Determining the percentage of queued requests

The percentage of queued requests for either a WebSpeed agent or an AppServer server is determined by a formula that compares data from the previous poll period with data from the most current poll period; this data is always calculated on a per-poll-period basis.



This rule subtracts the number of requests queued during the current poll from the number of requests queued during the previous poll to determine the number of new queued requests for the current poll period. Also, this rule subtracts the number of requests completed during the current poll from the number of requests completed during the previous poll to determine the number of new completed requests for the current poll. Then, the number of requests queued is divided by the number of requests completed to determine the percentage of requests completed during this poll period.

### Accessing the Queued Request Percent High rule page

The **Queued Request Percent High** rule page is accessible from the **Available Rules** page. On this page, you specify the threshold value, in percent form, that identifies the number of queued client requests during the polling interval that you consider acceptable. Any queued requests that cause the percentage to exceed this value will cause the alert and action criteria that you also set on this page to be triggered.

## Agent (Server) Unavailable rule

The list of available rules includes the following:

- **Agent Unavailable** rule for a Transaction Server — This rule monitors an agent's processing state to determine the agent's availability to service requests.
- **Server Unavailable** rule for an AppServer — This rule measures a server's processing state to determine the server's availability to service requests.

For either an agent or a server, this condition can indicate a failed, hung, or runaway process.

---

**Note:** Unlike other Fathom rules, the **WS\_Agent Unavailable** rule and the **WS\_Server Unavailable** rule monitor the state of either an agent or server, rather than the data each resource collects.

---

## Accessing the Agent (Server) Unavailable page

The **Agent Unavailable** page is accessible from the **WebSpeed Available Rules** page; the **Server Unavailable** page is accessible from the **AppServer Available Rules** page. On each page, you specify an integer to identify the threshold number of polls at which point you want to be alerted that the agent (or server) has been unavailable. You can also set other alert and action criteria.

### WebSpeed agent example

A user, initiating a customer order query in WebSpeed through a browser, accidentally enters a date range for one year (requesting the processing of 52 weeks' worth of data records) rather than the date range for one week (requesting 1 week worth of data records). The user, expecting a quick display of a results set, is unaware that the agent is tied up for an unknown period of time attempting to process more than 2,000,000 records associated with the year. Impatient due to the time he is waiting, the user begins clicking the **Submit** button over and over, hoping for some indication that the job has been submitted and the results set is ready for display.

Unknowingly, each time the user clicks the **Submit** button it causes the allocation of a new agent to service the request. This allocation might initiate the spawning of a new agent process. While this is occurring, the existing agents, processing the previous query requests, are unaware that the connection to the requesting client's browser page has been lost. These agents continue to consume resources as they process a request with no destination. If the request is long running, as defined by this example, the agent(s) are unavailable to service new client requests, potentially impacting application performance and throughput. This performance degradation can easily be compounded by the drain that these agents place on other resources such as CPU, memory, and databases.

As this example illustrates, you can use the **Agent Unavailable** rule as designed to help call attention to potential processing difficulties as soon as possible, and to thwart the potential for performance problems to escalate.

### AppServer server example

An AppServer Server might be stuck in an unavailable state due to either a startup fault or an application-level fault. The **Server Unavailable** rule is designed to alert you to a server that is unavailable due to these types of potential situations.

---

**Note:** This rule and its implications as described apply only to stateless and statefree implementations of an AppServer. This rule does not apply to state aware or state reset implementations.

---

## Working with rule sets

A *rule set* is a set of rules that you associate with one or more resources through a monitoring plan. These rule sets are stored by resource type in the **Fathom Component Library**. These links allow you to create OpenEdge-related rule sets:

- **Create AppServer Rule Set.**
- **Create NameServer Rule Set.**
- **Create WebSpeed Rule Set.**



To display the **Fathom Component Library** page where these OpenEdge links appear, click **Library**.

---

**Note:** The **Log File Rule Set** link on the **Fathom Component Library** page allows you to create rule sets to be shared among all log file resource monitors.

---

Rule sets provide a way for you to manage many broker resource types by sharing rule definitions. In this way, you create a common set of rules that you can associate with multiple resource instances.

Each rule set you create is stored in the **Fathom Component Library**, making the rule set available for reuse as well as use by other resource monitors within a given resource type.

You can also add individual rules to a monitoring plan, whether or not the rules are part of any rule set. Note that if you include a rule in a monitoring plan's rule set and then add the same rule again with modifications, the rule in the rule set is overridden by the rule with the modifications.

Fathom provides a default rule set for each OpenEdge resource type as it does for other resource types. For example, when an AppServer broker resource is added to Fathom, a default monitoring plan, with a default rule set, is assigned to it. Fathom also provides this default rule set assignment for WebSpeed and NameServer resources.

## Characteristics of and benefits associated with rule sets

Rule sets allow you to do the following:

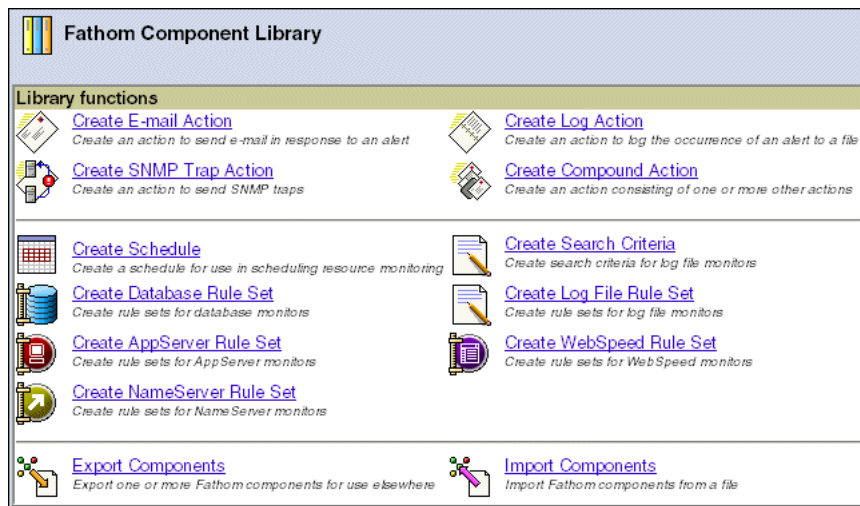
- Associate the rule set with a monitoring plan. The polled rules in the sets are evaluated when the monitoring plan is active and the resource is polled. (Asynchronous rules trigger immediately when these rules are violated.)
- Use an updated rule set. If you associate a rule set with a monitoring plan and you later update the rule set, the updated rule set is then used by the monitoring plan.
- Share the same rule set among several resource instances, such as all NameServers using the same common rule sets.
- Associate zero, one, or more rule sets with a broker monitoring plan.
- Override one or more rules defined in any rule set used by a monitoring plan.

The following procedure describes how to create a rule set, using the creation of a NameServer rule set as an example.

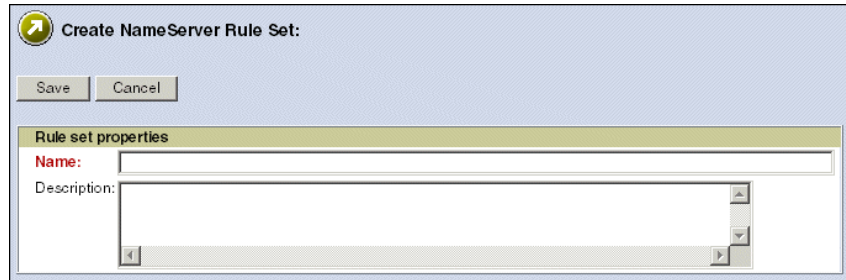


**To create a NameServer rule set:**

1. From the management console menu bar, click **Library**. The **Fathom Component Library** page appears:



2. Click **Create NameServer Rule Set**. The **Create NameServer Rule Set** page appears:



3. In the **Name** field, enter the name of the rule set (no spaces allowed).
4. In the **Description** field, enter a brief description of the rule set.
5. Click **Save**. The **NameServer Rule Set** page appears:



Note the following about rule sets:

- The rule set is now listed under the **Fathom Component Library** list frame under **Rule Sets** → **NameServer**, in this instance. (Rules sets associated with WebSpeed rules and AppServer rules are listed in the same **Rule Sets** category, but under the specific **WebSpeed** and **AppServer** rule set-related subcategory.)
- Once you create a rule set, you can edit, copy, or delete it.
- If you add a rule or a rule set to an existing rule set, the change affects all resources using the rule set.

## Editing a rule set

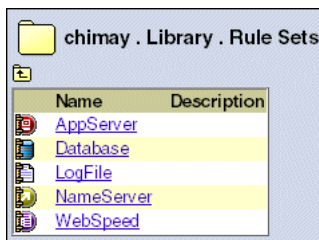
You can edit any rule set that you have created.



### To edit a rule set that you have created:

1. From the specific resource type's **Rule Set** page, click **Edit** to change the name or description of the rule set.
2. From either the **Rule Set** page or the **Edit Rule Set** page, you can also click **Add Rule** to add a rule to the rule set.

Note that you can access the list of existing rule sets at any time from the **Fathom Component Library** list frame. For example, click **Rule Sets** from those categories that display in the list frame. [Figure 6–7](#) shows the **Rule Sets** subcategories that display in the detail frame.



**Figure 6–7:** Accessing rule sets from the detail frame

## Copying a rule set

You can copy a rule set and make whatever modifications you want. You must, at a minimum, be sure to give the copy a unique name.



### To copy an AppServer rule set:

1. From the **AppServer Rule Set** page, click **Copy**. The **Copy AppServer Rule Set** page appears.
2. Provide a new name for the copy and (optionally) change the description.
3. Click **Save**.

You can now add one or more rules to the copy from either the **Copy AppServer Rule Set** page or the **AppServer Rule Set** page.

Note that you can access the list of existing AppServer rule sets at any time from the **Fathom Component Library** list frame. Click **Rule Sets**, and then click **AppServer**.

## Deleting a rule set

You can delete a rule set as long as it is not currently associated with any resource monitoring plans.



To delete a rule set from the **Rule Set** page, click **Delete**. Click **OK** to confirm the deletion.

---

**Note:** You can access the list of existing rule sets at any time for the **Fathom Component Library** list frame. For example, click **Rule Sets**, then click **AppServer**.

---

## Adding rule sets that have one or more rules in common

If you have multiple rule sets associated with a monitoring plan and you edit one of the rule sets, evaluation of only the first occurrence of any identically named rules takes place when the resource is polled. Which occurrence is considered “first” is determined by the alphabetic order of the rule set.

## Associating a rule set with a monitoring plan

You create a rule set for a specific OpenEdge resource to associate and use it with one or more monitoring plans. Once you establish the association, the rule set is active for the resource whenever the monitoring plan is active.

The following procedure shows how to associated a rule set with a monitoring plan by associating an AppServer rule set with a plan.



**To associate an AppServer rule set with a broker monitoring plan:**

1. Click **Resources** in the management console menu bar. See the “[Accessing OpenEdge resource information](#)” section on page 2–12 for the detailed procedures.
2. Click **Monitoring Plan** on the **AppServer main** page associated with the broker you selected. When the **Monitoring Plans** page appears, click the monitoring plan you want to update.
3. Click **Edit**. The **Edit Monitoring Plan** page appears.
4. Under **Rules selected for this plan**, click **Select Rule Sets**. A list of available rule sets appears. If a rule’s check box is selected, that rule set is already associated with the monitoring plan.
5. Select one or more rule sets you want to associate with the plan. If you want to review the rule set before you select it, click it. The rule set detail page opens.
6. Click **Save** when you finish. The monitoring plan is updated, and the **Edit Monitoring Plan** page reappears.



---

# Calculating Rule Threshold Settings Using the Configuration Advisor

---

This chapter describes how to use the Configuration Advisor to generate recommended threshold rule settings tailored for your system.

Topics in this chapter include:

- [Overview](#)
- [Setting rules-related criteria](#)
- [Understanding the recommended threshold settings](#)
- [Determining the effectiveness of your selections](#)

See the *Database Management Guide* for Configuration Advisor details related to databases.  
See the *Resource Monitoring Guide* for Configuration Advisor details related to CPU, Disk, and FileSystem resources.

## Overview

The Configuration Advisor is a Fathom Management feature that helps you determine optimum threshold settings for specific polled rules that you use. An alternative to using Fathom-supplied default values or values that you might arbitrarily set, the Configuration Advisor recommends threshold settings based on a representative sampling of historical data stored in the FathomTrendDatabase.

---

**Note:** You must have administrator privileges to use the Configurator Advisor.

---

The Configuration Advisor analyzes a rule's past performance for a specified period of time and, based on this data, calculates a baseline value. A *baseline value* is a number that serves as the base for calculating a set of possible threshold settings based on your system's past activity for a specific rule.

You then compare the existing rule threshold value with the recommended options to determine how to set the rule's threshold. When you select one of the recommended settings, Fathom will use this setting the next time the rule is evaluated.

Because these recommendations are based on a representative sampling of data from the FathomTrendDatabase, when you apply a recommend rule threshold setting the alerts triggered as a result of rule violations provide a more meaningful indication of your resource's performance.

---

**Note:** Depending on such factors as the time Fathom requires to retrieve, evaluate, and generate baseline values, Fathom's resources could be dedicated to this task for an unknown period of time. Consider planning a period of time in which you experiment with this feature to become familiar with its benefits and processing requirements.

---

## Rule details

The Configuration Advisor calculates recommended rule threshold settings for rules associated with a variety of Fathom resources. This section highlights the WebSpeed broker and AppServer broker rules. See the [Database Management Guide](#) for details about the database rules. See the [Resource Monitoring Guide](#) for details about using the Configuration Advisor with a disk, CPU, or file system resource.

The Configuration Advisor recognizes these WebSpeed broker and AppServer broker polled rules as candidates to process:

- Queued Request Percent High.
- Rejected Request Percent High.
- Process CPU High.
- Process Resident Memory High.
- Process Virtual Memory High.

For the Configuration Advisor to effectively analyze data for these polled rules, each rule must collect and trend data on every poll. Regardless of the time interval set for the polling, you must maintain a one-to-one relationship between trending and polling data.

Also, the options to implement data collection for a broker resource must have been set (checked). For details about data collection and the WebSpeed broker, see the [“Data collection details”](#) section on page 3–9. For details about data collection and the AppServer broker, see the [“Data collection details”](#) section on page 4–9.

### Rule-related considerations

Note these points concerning rule processing:

- A polled rule must be currently associated with a defined monitoring plan for it to be a candidate for the Configuration Advisor’s data analysis process.
- All rules associated with a given OpenEdge resource are individually evaluated against the rule-specific data retrieved from the FathomTrendDatase for the period of time you define.
- The Configuration Advisor *evaluates individual rules in a rule set*. Therefore, updating a rule with a recommended setting changes the value that a rule uses if the rule is part of a rule set. Because rule sets are shared among resources of a given resource type, this value change might adversely effect other resources that are using this rule set.

## Data analysis and recommended values overview

The goal of the Configuration Advisor's data analysis process is to determine a set, or range, of meaningful threshold values for a specific rule as used by your resources based on several factors.

### User-supplied criteria

Figure 7–1 shows an example of the initial **Configuration Advisor** page. In this example, the resource is an AppServer broker, **asbroker1**.

**Configuration Advisor**  
skye.asbroker1

Submit Cancel

For rule threshold calculations to take place, data must be retrieved from the FathomTrendDatabase. Please select the start and end date range to use for calculating the thresholds.

Start Date: 2004 August 14

End date: 2004 August 21

Choose time period to analyze :

☐ Sun ☒ Mon ☒ Tue ☒ Wed ☒ Thu ☒ Fri ☐ Sat

☒ Full day (24 hours)

☐ From: 9 : 00 AM To: 5 : 00 PM

Select rules (for analysis) :

☒ Queued Request Percent High

☒ Rejected Request Percent High

☒ Process CPU High

**Figure 7–1: Configuration Advisor page example**

On the **Configuration Advisor** page, you specify these values:

- A particular period of time, such as a week, in which data about a given rule is gathered and stored in the FathomTrendDatabase. Consider using the Fathom-supplied default values associated with a rule to establish this setting.

- A time frame that defines a representative period in which a rule is generally active or being used; this time frame is the period against which you want to calculate your baseline value. To gather this data with a high degree of accuracy, you will want to select a period of time in which your resources are most active. *Active* means when your resources are engaged, performing read, write, and update activities to your system.

It is recommended that you use the Fathom-supplied Configuration Advisor-related default values for a set period of time (for example, one week) to capture data to the FathomTrendDatabase for a rule. This initial step will provide you sufficient data with which to perform the comparison.

---

**Note:** Your monitoring plan schedules are not necessarily the best choice for a time frame. A schedule defines a period of time in which rules are in effect; it does not necessarily focus on time periods in which your resource usage is highest. For example, you might use the 24x7 monitoring plan schedule to constantly monitor your system, but want to select Monday through Friday from 8 a.m. to 6 p.m. to calculate your baseline settings.

---

- The rule or rules for which you want to determine recommended values.

The Configuration Advisor reviews monitoring plans defined for a resource, looking for polled rules that can be calculated by the Configuration Advisor. If any of these rules are present, it displays them as preselected (as identified by a check mark). Deselect any rules for which you do not want recommended values to be calculated by removing the check mark next to the rule. When you deselect the check mark, the Configuration Advisor does not calculate the rule threshold setting for that rule.

## The Configuration Advisor's data analysis process

When you submit the completed **Configuration Advisor** page, the Configuration Advisor extracts individual rule-related data from the FathomTrendDatabase. Based on the availability of a minimum requirement of 32 valid data samples per rule to be calculated for the designated date range, the Configuration Advisor determines a baseline value.

A *valid data sample* is a data sample that is determined to be not a null value (any whole number that is not zero). For example, the Rejected Request Percent High rule is determined when the quantity of rejected requests is divided by the quantity of received requests. The result must be a non-zero, whole number.

This baseline value is used to calculate the recommended ranges. A data sample of 32 identifies a statistically meaningful representative portion of a rule's performance data as stored in the FathomTrendDatabase. This sampling provides sufficient data from which the Configuration Advisor can determine a baseline value and subsequently perform a successful analysis of each rule's data.

## An individual rule's definition

Based on the availability of the values and data, the Configuration Advisor generates a set of recommended values, or settings, for each rule processed. As necessary, the range of values is adjusted to ensure that the rules do not violate the minimum or maximum allowable values for the rule.

## Calculated recommended threshold rules

Once the calculation process is completed for each rule, the Configuration Advisor presents its results on the **Configuration Advisor calculations** page. Your initial criteria and each rule and the associated monitoring plans for which the rule applies, are displayed. [Figure 7-2](#) shows the page that appears after you have submitted your initial **Configuration Advisor** page.

Configuration Advisor  
skye.asbroker1

Update Selected Rules Cancel

The following time period was used for analysis  
Start Date: 2004/7/14  
End Date: 2004/7/21  
Days: Sun Mon Tue Wed Thu Fri Sat  
Hours: From: 9:00 To: 17:00

Default Schedule Select:  
All  
None

Rule	Recommend Values	Update	Current Threshold	Detail
Process CPU High	3.1 [0]* percent	<input checked="" type="checkbox"/>	80.0	Detail
Queued Request Percent High	4.0 [0]* Percent queued	<input checked="" type="checkbox"/>	20.0	Detail
Rejected Request Percent High	4.0 [0]* Percent rejected	<input checked="" type="checkbox"/>	5.0	Detail

**Figure 7-2: Configuration Advisor recommend thresholds page**

The **The following time period was used for analysis** section of this page summarizes the values defined on the initial **Configuration Advisor** page. These values are displayed here to remind you about the time period criteria you set.

The **Rule** section contains all the rule-related calculated data. For each rule that is successfully processed, the range of recommended results appears in the **Recommended Values** drop-down list; each rule row also displays the current rule setting for each rule as defined for each individual monitoring plan. You can select a recommended rule threshold setting and existing monitoring plan, or plans, to which you want the range to apply.

The recommended settings are expressed in a mathematical expression consistent with the rule threshold's unit of measure. [Figure 7-2](#) shows that the unit of measure for **Queued Request Percent High** is **percent queued** and **Rejected Request Pernt High** is **percent rejected**. The unit of measure for **Process CPU High** is **percent**.

---

**Note:** As you compare the existing and recommended values, you can elect to change none, some, or all values for a rule and each individual monitoring plan.

---

Until you click **Update Selected Rules**, Fathom does not apply any of your selections.

If the data analysis calculation for a rule was unsuccessful, the Configuration Advisor cannot define a range and an **Insufficient data for analysis** message displays in the **Recommended Values** drop-down list field. For example, if a data sample for the defined time period is not equal to or greater than 32, this message displays because there are not enough data samples available from which the Configuration Advisor can suggest a meaningful recommendation.

## Generating and applying threshold rule settings

Table 7–1 highlights the major topics and procedures in the remaining sections of this chapter to help you to use the Configuration Advisor.

**Table 7–1: Configuration Advisor details**

For information about...	See the...
The procedure to set user-supplied criteria used in the data analysis process	<a href="#">“Setting rules-related criteria”</a> section on page 7–9.
Understanding the Configuration Advisor’s recommended settings, including evaluating and applying these settings	<a href="#">“Understanding the recommended threshold settings”</a> section on page 7–12.
Reviewing your selections	<a href="#">“Determining the effectiveness of your selections”</a> section on page 7–17.



## Setting rules-related criteria

Once you have completed your specific rule analysis, you have the necessary information to use the Configuration Advisor. The following procedure shows how to use the Configuration Advisor to calculate AppServer threshold values. Use these same steps to calculate WebSpeed threshold values, substituting the WebSpeed-specific rules and data for those shown in the procedure.



### To initiate the Configuration Advisor:

1. Display the **AppServer main** page for the broker for which you intend to run the Configuration Advisor. Refer to the [“Accessing OpenEdge resource information”](#) section on page 2–12 for the detailed steps.
2. Click **Configuration Advisor** in the **Command and control** section to display the **Configuration Advisor** page as shown:

**Configuration Advisor**  
skye.asbroker1

Submit Cancel

For rule threshold calculations to take place, data must be retrieved from the FathomTrendDatabase. Please select the start and end date range to use for calculating the thresholds.

Start Date: 2004 April 15

End date: 2004 August 21

**Choose time period to analyze :**

☒ Sun ☒ Mon ☒ Tue ☒ Wed ☒ Thu ☒ Fri ☒ Sat

☐ Full day (24 hours)

☒ From: 9 : 00 AM To: 5 : 00 PM

**Select rules (for analysis) :**

☒ Queued Request Percent High

☒ Rejected Request Percent High

☒ Process CPU High

☒ Process Resident Memory High

☒ Process Virtual Memory High

3. In the **Start Date** and **End Date** fields, define a date range that Fathom will use to collect data from the FathomTrendDatabase. (The default data range, as shown in the previous sample, is one week.)

Keep these points in mind:

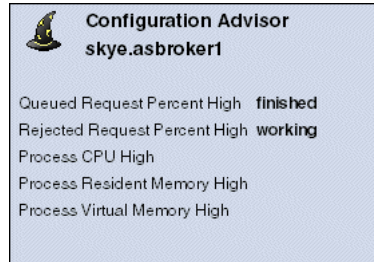
- A polled rule must currently be associated with a monitoring plan for it to be a candidate for the Configuration Advisor to process.
  - Trending must have been set to **TRUE** for a candidate rule for the time period time you specify. This requirement ensures that data was trended to the TrendFathomDatabase for this rule.
  - The options to implement data collection for a broker resource for which you want to determine recommended rule threshold settings must have been set (checked).
  - All rules associated with a given OpenEdge resource are individually evaluated against the rule-specific data retrieved from the FathomTrendDatabase for the period of time you define.
4. In the **Choose time period to analyze** section, identify the time frame that defines a representative period of time in which the rules are generally active, or being used. This time frame is the period against which you want Fathom to calculate your baseline activity. (The default time period, as shown in [Step 2](#), is Monday through Friday, 9:00 AM to 5 PM.)
  5. In the **Select rules (for analysis)** section, click the polled rules that you want the Configuration Advisor to use to calculate threshold settings.

Only those polled rules that are currently defined in existing monitoring plans for a broker resource can be candidates for the Configuration Advisor to process; the Configuration Advisor displays these rules in this section with a check mark associated with the rule to indicate that the Configuration Advisor will calculate new settings. (In the previous page sample, the Configuration Advisor determined that there are three rules that are associated with this broker resource's existing monitoring plans. The Configuration Advisor will attempt to provide recommended values for these rules.)

This requirement ensures that data was trended to the FathomTrendDatabase for this rule.

6. Click **Submit**.

As the Configuration Advisor attempts to calculate the rules threshold settings, the following information displays, reporting the progress of each calculation it is performing, as shown:



Depending upon the criteria that you set on the initial **Configuration Advisor** page, the number of rules you selected, and other factors such as your machine's speed, this calculation process could take some time.

---


**Note:** Once you click **Submit**, you can elect to go to another page and perform some other action. You can return to the Configuration Advisor at some later point to check status and/or result details.

---

When all calculations have been completed and reported, the Configuration Advisor displays the calculated results. See the [“Understanding the recommended threshold settings”](#) section on page 7–12 for details.

# Understanding the recommended threshold settings

Figure 7–3 shows the data calculation page that appears after the Configuration Advisor has applied the criteria you submitted to calculate the threshold settings.



Configuration Advisor  
skye.asbroker1

Update Selected Rules

Cancel

The following time period was used for analysis

Start Date: 2004/4/15

End Date: 2004/8/21

Days:  
Sun Mon Tue Wed Thu Fri Sat

Hours:  
From: 9:00 To: 17:00

Rule	Recommend Values	Weekdays		Weekends		
		Select:	Select:	Select:	Select:	
		All	All	All	All	
		None	None	None	None	
		Update	Current Threshold	Update	Current Threshold	
Process CPU High	3.1 [0]* percent	<input checked="" type="checkbox"/>	80.0	<input checked="" type="checkbox"/>	80.0	<div>Detail</div>
Process Resident Memory High	3256.6 [0]* kilobytes	<input checked="" type="checkbox"/>	10000.0	<input checked="" type="checkbox"/>		<div>Detail</div>
Process Virtual Memory High	6650.5 [0]* kilobytes	<input checked="" type="checkbox"/>	100.0	<input checked="" type="checkbox"/>	100.0	<div>Detail</div>
Queued Request Percent High	4.0 [0]* Percent queued	<input checked="" type="checkbox"/>	20.0	<input checked="" type="checkbox"/>		<div>Detail</div>
Rejected Request Percent High	4.0 [0]* Percent rejected	<input checked="" type="checkbox"/>	5.0	<input checked="" type="checkbox"/>	5.0	<div>Detail</div>

Figure 7–3: Configuration Advisor recommended values

This page has multiple purposes. Table 7–2 identifies the key tasks you can perform from this page and their associated procedures.

Table 7–2: Tasks using the Configuration Advisor Calculations page (1 of 2)

To...	See the...
Review and evaluate the recommended threshold settings calculated for each rule processed.	<a href="#">“Evaluating recommended settings”</a> section on page 7–13.
Display and review the specific details about each individual rule’s analysis.	<a href="#">“Evaluating recommended settings”</a> section on page 7–13.

**Table 7–2: Tasks using the Configuration Advisor Calculations page** (2 of 2)

To...	See the...
Compare current threshold settings defined for each of the rules processed with the recommended threshold calculations.	<a href="#">“Comparing and selecting threshold settings”</a> section on page 7–16.
Update the threshold values for the rules and the specific schedules that you have selected.	<a href="#">“Submitting your threshold setting selections”</a> section on page 7–17.

## Evaluating recommended settings

As [Figure 7–3](#) shows, each rule the Configuration Advisor has analyzed appears as an individual line item in the **Rule** section. Associated with each rule is a **Recommended Values** drop-down list; it can contain one of the following entries:

- Numeric values that identify the recommended rule threshold settings. This list can contain up to seven different numeric items; collectively, these values comprise the range of recommended threshold settings.
- An **Insufficient data for analysis** message. The Configuration Advisor displays this message when the criteria to perform the data analysis successfully are not met. See the [“Setting rules-related criteria”](#) section on page 7–9 for details.

Reviewing recommended values

The Configuration Advisor displays a range of possible values from which to select.

For example, [Figure 7-4](#) shows the full range of seven recommended values for the **Process CPU High** rule. Note that the Configuration Advisor’s primary (default) recommendation displays in the **Recommended Values** field with an asterisk.

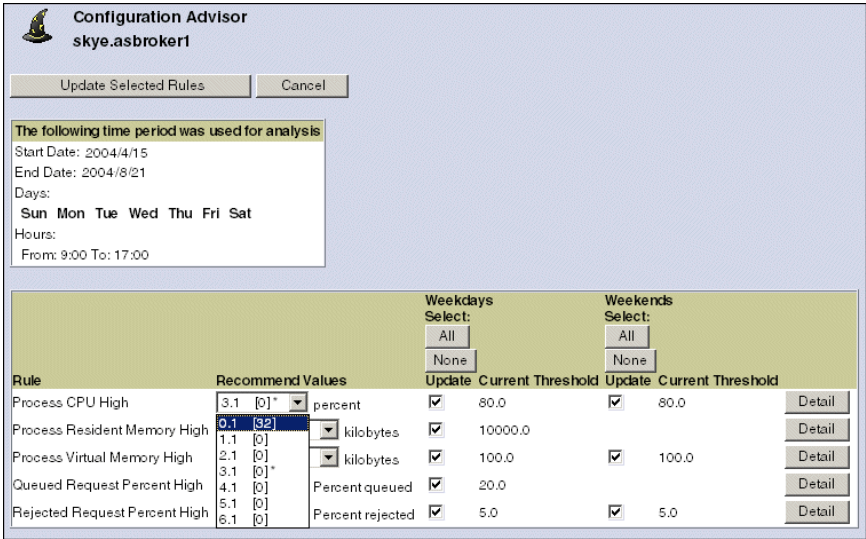


Figure 7-4: Recommended Values field content

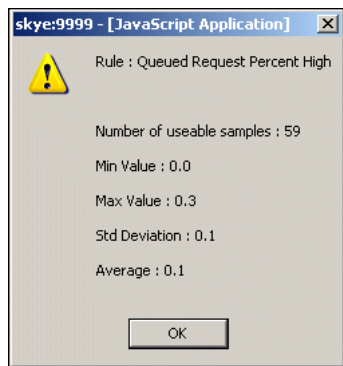
Each recommended value is expressed as a set of two numbers. The left-most number specifies the recommended threshold setting; the right-most number, displayed in brackets, identifies the number of times the threshold value set at the associated setting would be exceeded and an alert would be fired. The asterisked number indicates the Configuration Advisor’s primary recommendation. As you review the recommended threshold settings, note the rule behavior and alert notification frequency you want to establish for a resource.

Using the Detail button

Each rule row has an associated **Detail** button.

- To display details about a rule’s analysis, click **Detail** for a row. The **Detail** page appears:

Figure 7–5 shows an example of a Detail page, displaying the data used to evaluate the **Queued Request Percent High** rule.



**Figure 7–5: Detail page analysis content**

Figure 7–5 shows an example of a **Detail** page, displaying the data used to evaluate the **Queued Request Percent High** rule.

Table 7–3 describes the contents of the **Detail** page.

**Table 7–3: Detail page fields and descriptions**

This field.....	Describes the...
<b>Number of useable samples</b>	Number of data samples extracted from the FathomTrendDatabase.
<b>Min Value</b>	Minimum value derived from the data set.
<b>Max Value</b>	Maximum value derived from the data set.
<b>Std Deviation</b>	Root mean squared deviation.
<b>Average</b>	Average value derived from the data set.

**Note:** The **Detail** page for a rule for which there is insufficient data for analysis identifies the number of samples found; this number is always lower than the minimum of 32 data samples required. Review this data to help you decide if you need to expand your time period to try to capture more samples and re-run the Configuration Advisor for a given rule.

# Comparing and selecting threshold settings


By default, the Configuration Advisor assumes that you are going to select and submit one of the recommended threshold settings; it provides a check mark in the **Update** field for each rule row. However, you have options concerning the selection process. As you compare the existing and recommended values, you can elect to change none, some, or all values for a rule and each individual monitoring plan.

Use the following procedure to compare the current rule setting with the recommended threshold settings and to update each schedule with your specific selections. Perform this comparison to help you further determine your final selection.



## To compare and select threshold settings:

- For a specific rule row, note the value that displays in the **Current Threshold** field under a specific schedule. For example, note the values that appear in the **Current Threshold** field for the following two monitoring plans: **Weekdays** and **Weekends**:



Configuration Advisor  
skye.asbroker1

Update Selected Rules

Cancel

The following time period was used for analysis

Start Date: 2004/4/15

End Date: 2004/8/21

Days:  
Sun Mon Tue Wed Thu Fri Sat

Hours:  
From: 9:00 To: 17:00

Rule	Recommend Values	Weekdays		Weekends		Detail		
		Select:	Select:	Select:	Select:			
		All	All	All	All			
		None	None	None	None			
		Update	Current Threshold	Update	Current Threshold			
Process CPU High	3.1 [0]* percent	<input checked="" type="checkbox"/>	80.0	<input checked="" type="checkbox"/>	80.0	<div>Detail</div>		
Process Resident Memory High	3256.6 [0]* kilobytes	<input checked="" type="checkbox"/>	10000.0	<input checked="" type="checkbox"/>	100.0	<div>Detail</div>		
Process Virtual Memory High	6650.5 [0]* kilobytes	<input checked="" type="checkbox"/>	100.0	<input checked="" type="checkbox"/>	100.0	<div>Detail</div>		
Queued Request Percent High	4.0 [0]* Percent queued	<input checked="" type="checkbox"/>	20.0	<input checked="" type="checkbox"/>	5.0	<div>Detail</div>		
Rejected Request Percent High	4.0 [0]* Percent rejected	<input checked="" type="checkbox"/>	5.0	<input checked="" type="checkbox"/>	5.0	<div>Detail</div>		

- Click **Recommended Values** to display the range of recommended values for the associated rule.
- Compare the possible **Recommended Values** that display with the value in the **Current Threshold** field. As you determine the best threshold rule setting, keep your goals for this rule in mind. Also, consider any additional selection criteria in mind as you compare the various values. See the [“Additional selection criteria”](#) section on page 7–17 for details.



4. Repeat [Step 1](#) through [Step 3](#) for each rule and its associated monitoring plan.

Also, if you know that you are going to accept or deselect the recommended threshold settings for a schedule, you can use these options:

- a. Click **All** to select all of the recommended threshold settings for a monitoring plan.
- b. Click **None** to deselect all of the recommended threshold settings for a monitoring plan.

### **Additional selection criteria**

The following list identifies more criteria you might consider for selecting one value and not another:

- How often you want alerts generated.
- Factors unique to your resource's performance that you want to consider in making your selection.
- Your knowledge of your system's operational needs and goals.

## **Submitting your threshold setting selections**

When you click **Update Selected Rules**, Fathom applies all of your selections at the same time. There is no undo option associated with this group submission. To reset any values back to a previously defined setting, you must access the resource's monitoring plan, displaying the individual rule and override the current value that displays.

## **Determining the effectiveness of your selections**

The most effective way to determine if your threshold adjustments are serving your needs is to review your alert notifications. Strive for a threshold setting that is consistent with your resource and business needs. If you find you are receiving alerts too frequently or too infrequently to suit your operational needs, you might want to further refine your threshold settings.



---

# Analyzing OpenEdge Application Performance

---

This chapter describes how you can use Fathom Management to analyze OpenEdge server application performance.

Topics in this chapter include:

- [Overview](#)
- [Investigating application performance issues](#)
- [Fathom in the workplace](#)
- [Planning an application performance review](#)
- [Responding to an application crisis](#)
- [For more Fathom assessment information](#)

## Overview

System administrators deal with a variety of situations that threaten the performance, and even the availability, of a production system. Small resource issues can become bigger issues if left unaddressed; larger resource problems can threaten the health of the system, jeopardizing critical business operations.

To track and respond to resource situations, system administrators need the right data from which to determine corrective action. Regardless of the type of problem that might occur, each situation requires investigation and a solid recovery plan based on valid data. With the right data, a system administrator can determine options and plan short or long term strategies and solutions. Every strategy should include a solid recovery plan.

This chapter describes:

- [Investigating application performance issues.](#)

This section provides a model for administrators to use.

- [Fathom in the workplace.](#)

Using the fictitious company XYZ Corporation, information in this section provides background for the performance scenarios that follow. Both scenarios use the AppServer as a key component.

- [Planning an application performance review.](#)

This sample scenario shows how one administrator's pro-active work practices using Fathom reports helps him to uncover clues about application performance changes and degradation.

- [Responding to an application crisis.](#)

This sample scenario highlights how the use of various Fathom features can help administrators quickly analyze and respond to a system or application problem.

---

**Note:** These scenarios are intentionally limited in scope. They are intended to help you understand some of the general principles by which Fathom features can be used to investigate and troubleshoot. Keep in mind that elements such as your company's application and database designs will potentially play a larger role in performance issues than is described in these fictitious circumstances.

---

## Investigating application performance issues

With the aid of Fathom Management, you can follow a simple process to identify, understand, and address performance issues. This process involves:

- Understanding your business requirements and reviewing them periodically.

It is essential to have a thorough knowledge of your business needs, work practices, and acceptable and unacceptable trade-offs. Armed with this fundamental understanding, you can use Fathom-supplied data to proactively anticipate and plan for change, minimizing the effects of system problems on your business operations.

- Defining your problem or goal clearly.

Given your business and work practices, ask:

- What problems do you want to anticipate or eliminate?
- What performance goals would you like to achieve?

Whatever the problem you want to minimize or eliminate, or the performance goal you want to achieve, define it in a straightforward, concise manner.

- Reviewing Fathom-supplied data to investigate and analyze your problem or goal.

Use your problem definition to review Fathom-generated information to better understand your problem. Through a process of elimination, you can evaluate the data and identify potential components that maybe contributing to a given problem.

- Documenting the steps you perform to address your issues, and test all documented options that you generate.

Not all problems or performance issues can be resolved immediately. Maintain a log of issues and a checklist of areas investigated to solve a given problem. Review them periodically and you may be able to improve on your original solution over time.

## Fathom in the workplace

This section demonstrates the process of investigating application performance issues using a fictitious company, XYZ Corporation. At XYZ, the administrator has installed and configured Fathom Management.

## Implementing business requirements with Fathom Management at XYZ Corporation

The XYZ Corporation's system administrator has customized his Fathom resource monitoring capabilities and frequently consults his system's data as monitored by Fathom. For example, this administrator:

- Set up the **Trend performance data** option for all monitored resources, including AppServer brokers. This feature helps him review real time and historical data available for reports, most noteworthy in this case the **Performance** and **Profile** reports.
- Establish rules from the **Library** menu option as default rules for all AppServer broker resources for their performance criteria value: **Average Procedure Duration High, Queued Request Percent High, Rejected Request Percent High, and Agent (Server) Unavailable**. Establishing these rules, with threshold values that are unique to this system environment, is key because of the heavy network- and AppServer-related processing demands. He also sets up alert and actions for each of these rules.
- Consults the **Broker Performance View** and **Servers Performance View** for AppServer broker and server performance statistics frequently throughout the work day for a real-time picture of broker and server activity levels.
- Set up his AppServer brokers and servers on his **Customized My Collections Home** page, along with other vital system operations such as memory and CPU consumption, so he can quickly reference this data. Among other standard viewlets, he displays **resources running with alerts, active monitoring plans, and running reports viewlet options**. Also, he monitors all those viewlets related to his AppServer brokers.
- Reviews his **System Activity** report frequently throughout the work day as it displays real-time system performance and resource usage details.
- Reviews his **Database Summary** report frequently throughout the work day as it displays real-time system performance and resource usage details.
- Consults the AppServer-related log files for which he has set up monitors: AppServer broker log file and the AppServer servers log file.

He regularly reviews these pieces of data as they can provide him clues about his system's application performance.

## Consulting Fathom Management documentation

This administrator also frequently references the information in the Fathom Management documentation set. Among the ten books in this set that he can access online in pdf format, he frequently consults manuals such as the *Resource Monitoring Guide*, the *Alerts Guide and Reference*, and the *Reporting Guide*; much of the knowledge that he demonstrates in setting up, monitoring, and managing resources, and customizing his AppServer Profile report was gained through his using these manuals. He also reviews the *Database Management Guide*.

## Planning an application performance review

As a matter of good practice, the system administrator at XYZ Corporation is always on the watch for ways to improve his application's performance. With a high volume of data entry taking place between 9 a.m. and 6 p.m. on the system he supports, and most of the procedures distributed and run remotely on an AppServer, his users expect a consistently high level of application performance and availability. He has learned over time how to deliver system availability that is consistent with this goal.

He has come to learn that his application's performance depends on the effectiveness of four key elements:

- The application's integrity.
- The application's efficiency.
- The database and servers responsiveness.
- The network's responsiveness.

Of course, he might consider other technological elements, but these three remain of primary concern. He is most concerned with Fathom performance indicators that relate to these elements so that he can act on any potential performance issue before it affects his users and their ability to perform their jobs.

## Problem definition

Over the last two weeks, data entry personnel at XYZ Corporation have been mentioning some slight but noteworthy delays in performing routine updates to records on the company’s production system. One day, an update process might go fine, but the next day a similar transaction might take 30 to 40 seconds longer to complete. From a user’s perspective, this delay is an annoying problem.

From a system administrator’s perspective, it’s a bit of a mystery. He consults his system’s problem log, only to find that it has been several months since there has been an application or system problem of this kind reported. This new performance issue concerns him because any indication of a performance weakness could become a real performance problem if he doesn’t determine the source of the problem as soon as he can.

## Initial investigation

The first question he asks is: “What’s changed in my production environment that is causing poor performance?” To begin solving this performance problem, he starts to list the possibilities.

Table 8–1 lists the possibilities the system administrator considers. Note the blank column in the table. As he reviews each possibility, he can use this table as a checklist. He will check the items he decides to consider further.

**Table 8–1: Initial Investigative Checklist** (1 of 2)

√	Access and review...	As these topics relate to these questions...
	High-level performance indicators.	Have users been complaining about other performance issues that might be related to this performance problem?  Are any background processes running during these offending times that could be causing program delays?
	Hardware and/or software component changes.	Have there been any changes to the hardware or software installations that might have impacted the application’s performance?  For example, has a new disk been added, or a software upgrade been performed in the time frame in which problems have been noticed and reported?



**Table 8–1: Initial Investigative Checklist**

(2 of 2)

√	Access and review...	As these topics relate to these questions...
	Possible workload changes.	Is it possible that some or all of the application inefficiencies noted are related to the number of users working on the application, causing the delays as noted?
	Data details in the log files such as the database logs, AppServer log files, customized log files and so forth.	Are there any details in the log file data from the time period in which the application was performing poorly that might indicate an application performance problem?
	The database performance for possible database issues.	Does the database need to be tuned? A tuning effort of this kind can provide significant pay off in performance if it's found to be a contributing factor.
	Data from the FathomTrendDatabase from the troublesome time period.	By running reports at different time periods, are there any patterns in the data or reported application responsiveness that match experiences that the users have reported?

## Drilling deeper into Fathom-supplied data

In addition to the considerations noted in [Table 8–1](#), the system administrator reviews the data he gathers weekly through the AppServer Application Profile report. When his company installed and started to use Fathom, he began using the predefined report template feature to run report instances on a weekly basis; this report's data provides him with a high-level picture of his application's health. The [Reporting Guide](#) helped him to set up and use the AppServer Profile report and all the other Fathom-generated reports.

### Looking at the AppServer Profile report

The purpose of the AppServer Profile report is to provide details about procedures run by a broker. The data captured by this report can include these elements:

- How many times a specific procedure ran.
- The average and maximum durations of each request.
- The number of successful requests.

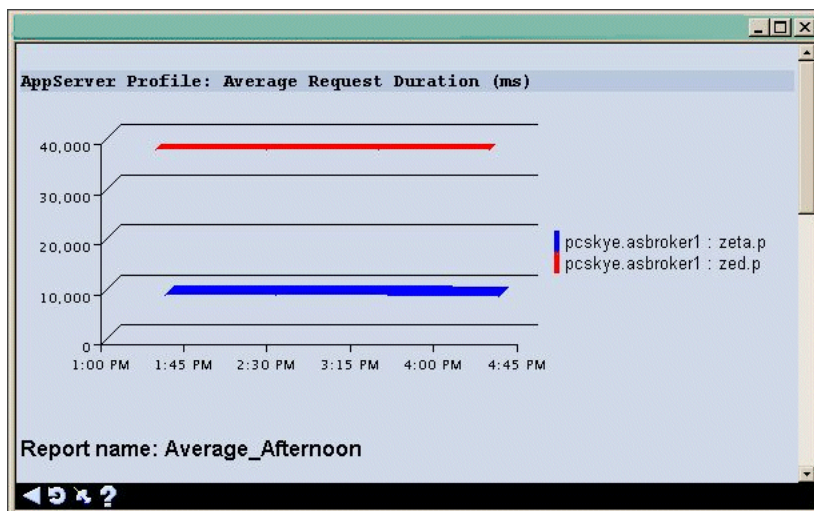
- The number of errors.
- The number of times each request quit and stopped.

In this instance, the administrator has customized his AppServer Profile report. As shown in the graphical data displayed in [Figure 8–1](#), this AppServer Profile report presents information about the average time it takes for two different procedures to run on his AppServer. Reviewing and routinely comparing reports from different time periods provides this administrator more insight into his AppServer’s performance.

### Finding performance-related clues in the AppServer Profile report

This administrator knows that reviewing performance details about two of his Progress 4GL procedures might provide him performance clues. Performance issues related to these high-level rate procedures—**zeta.p** and **zed.p**—might impact his application’s performance.

[Figure 8–1](#) displays typical AppServer workload-related data that is consistent with an average weekday afternoon at the XYZ Corporation:



**Figure 8–1:** AppServer Profile Report — Average\_Afternoon data

---

**Note:** In the figures presented in this section, the colors in the graphs are only intended to distinguish one procedure from the other.

---

The AppServer Profile report that appears in [Figure 8–1](#) is set up to:

- Capture the average time that it takes two individual Progress 4GL procedures—**zeta.p** and **zed.p**—to run during the system’s peak operational time.

By selecting the Average Procedure Duration High rule on the AppServer’s monitoring plan and identifying a polling interval threshold for it, the administrator can monitor the AppServer’s performance and behavior based on values that are significant to his performance expectations. For details about this rule, see the “[Average Procedure Duration High rule](#)” section on page 6–24. For details about monitoring plans, see [Chapter 6](#), “Monitoring Plans and Rules for OpenEdge Server Resources.”

- Display this data in a graphical mode in a browser.

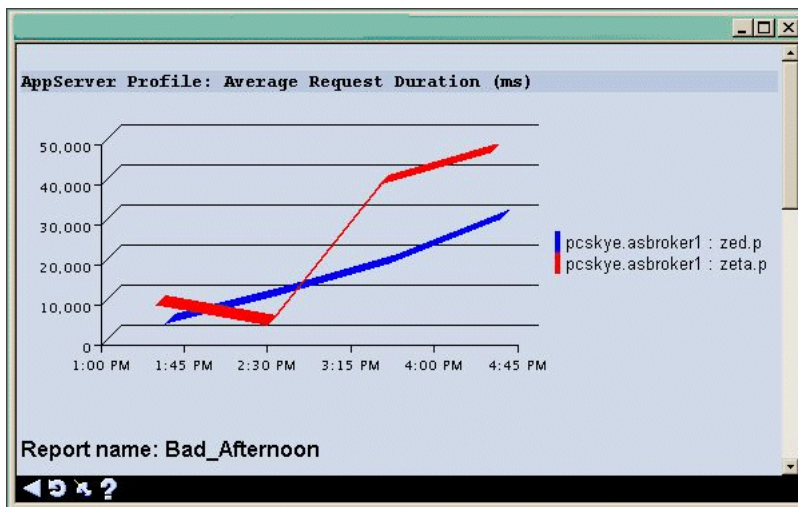
---

**Note:** In the figures presented in this section, the colors in the graphs are only intended to distinguish one procedure from the other.

---

These two procedures, **zeta.p** and **zed.p**, are among the procedures that the AppServer broker, **asbroker1**, is currently running. This is the kind of normal, predictable AppServer procedure processing that a system administrator likes to see; resources are being used and consumed, but not overly taxed so that his users’ and his company’s business needs are being well met.

This administrator compares his report data results from previous weeks to the data results that appear in [Figure 8–2](#); the fact that the procedure **zed.p** is hovering at the defined threshold use of **40,000** indicates that there is likely an otherwise hidden performance issue for him to investigate.



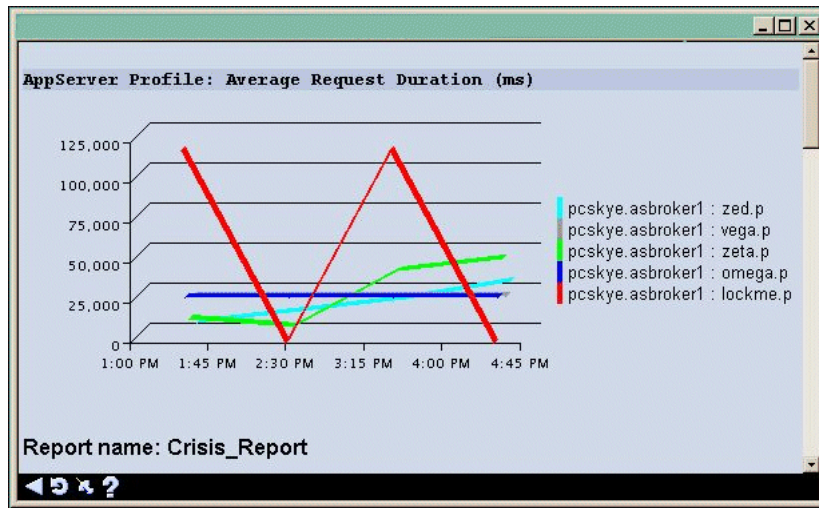
**Figure 8–2: AppServer Profile Report — Bad\_Afternoon data**

The same type of average request duration data that displays in [Figure 8–2](#) tells a very different story about another workday afternoon at XYZ Corporation. By comparing the generated data in [Figure 8–1](#) with the generated data for the same procedures and associated brokers in [Figure 8–2](#), this administrator can see that the slow growth in the average time it takes to complete a process requested by either the **zeta.p** or **zed.p** does cause problems if left on this current growth rate. As [Figure 8–2](#) shows, these procedures are either exceeding, or trending toward the possibility of exceeding, the threshold of 40,000. Given the data as reported in the **Bad\_Afternoon report**, the administrator could begin to make some notes about the application's response to pass along to the company's programmers so that they can consider changes to re-balance the work load.

This administrator's routine review and comparison of the data presented in [Figure 8–1](#) and [Figure 8–2](#) have helped him to thwart a potential application crisis. This problem detection points to where the administrator's code review with developers or system engineers should begin.

## Using report data to minimize an impending application performance crisis

Figure 8–3 shows the type of data the system administrator could have faced had he not been as diligent in his routine review and investigation of Fathom report data.



**Figure 8–3: AppServer Profile Report — Crisis\_Report data**

Assuming the same 40,000 threshold for all of the procedures listed in Figure 8–3, it is very apparent that this work day afternoon has reached crisis proportions. Not only are the procedures **zed.p** and **zeta.p** exceeding the threshold, the **lockme.p** procedure is definitely causing even more mayhem at approximately 1:30 PM and again at 3:30 PM.

## Testing and documenting your potential solutions

XYZ Corporation is fortunate to have hired this administrator; he is well seasoned in keeping a log of application and system problems, and consistently records his actions to correct difficulties.

To monitor this particular situation to ensure that the problem has been resolved satisfactorily, he:

- Works with the company's application group to ensure that they receive the time and records needed to address the application's performance problem.
- Monitors the impact of the fix closely to ensure that it did correct the problem and not introduce any other application or system difficulties.
- Interviews his application users to ensure that they experience an improvement in their application throughput.
- Documents the problem and his efforts to correct the problem so that it will be available for future reference.

## Responding to an application crisis

Despite all the best plans, an application crisis can occur. By employing various features and functionality of Fathom Management, a system administrator can arm himself with some fundamental informational tools. These tools help to provide immediate data to understand and address a crisis.

This section describes another problem that the system administrator from XYZ Corporation must face.

---

**Note:** The Fathom features outlined in the [“Implementing business requirements with Fathom Management at XYZ Corporation”](#) section on page 8–4 also apply to this sample scenario.

---

## Problem definition

The XYZ Corporation's system administrator was having a routine, mid-week work day. Normal system processing was occurring as the system was humming along just fine. Response time was good, and his users were very pleased. There's always room for improvement, but at the moment, the monitoring results showed no cause for concern.

Unexpectedly, however, the system's performance began to decline rapidly. The system administrator's phone began to ring off the hook with end-user calls. The complaints were all the same: transactions were not going through, and data entry tasks could not be completed. Even simple look-up activities failed.

## Initial investigation

In an application crisis situation of this type, this administrator can leverage Fathom-supplied information to alert him to immediate problems and provide him data related to the crisis.

Table 8–2 lists the possibilities the system administrator considers. Note the blank column in the table. As he reviews each possibility, he can use this table as a checklist. He will check the items he decides to consider further.

**Table 8–2: Crisis review checklist**

(1 of 2)

√	Access and review...	To...
	Alert and other data indicators you have set up to monitor and display data on the collections page.	Quickly examine issues that might be the crux of this dramatic change in performance.  As previously noted in the <a href="#">“Implementing business requirements with Fathom Management at XYZ Corporation”</a> section on page 8–4, this administrator has several indicators set up, including viewlets related to his AppServer brokers.
	Data details in log files such as the database log files, AppServer log files, customized log files and so forth.	Examine log file data from the time period that the crisis initially occurred.  Determine if there is any noteworthy, relevant information in error logs related to the crisis situation.

**Table 8–2: Crisis review checklist** (2 of 2)

√	Access and review...	To...
	Network- and server-related data details, using TCP resource monitors he previously set up	Determine the status and response time, if any, for mail, FTP, and Web Servers that might be running on the network.
	Network-related data details, using Packet Internet Grouper (PING) (ICMP) resource monitors he previously set up	Determine if network resources are available.
	Server-related data details	Determine if AppServer server details and/or AppServer broker details are of help in problem determination.

While quickly scanning the checklist, the system administrator kept thinking about what his frantic users said about the performance issue: nothing was working. He felt that there was possibly a network problem to resolve, but where was the source? Since most of the transactions related to the procedures that were not currently functioning ran on a remote AppServer, the administrator decided to follow this thought to a specific investigative path.

## Drilling deeper into Fathom-supplied data

As the checklist items in [Table 8–2](#) indicate, the administrator needs quick access to performance data. In a crisis situation such as this one, he needs to know that he has accurate, timely information available to determine, resolve, and learn from the problem situation to minimize—if not eliminate—such a crisis of this kind from reoccurring.

### Accessing and examining AppServer data

The administrator accesses his OpenEdge resources in the Fathom Management console, browsing to the AppServer resources. His network uses only one AppServer, so he can immediately click on either of the AppServer **Operational views** data—the **Server Performance View** or **Broker Performance View**.

---

**Note:** For detailed procedures on setting up and accessing AppServer resources, including the AppServer **Operational views**, see [Chapter 4, “Managing AppServer Data.”](#)

---



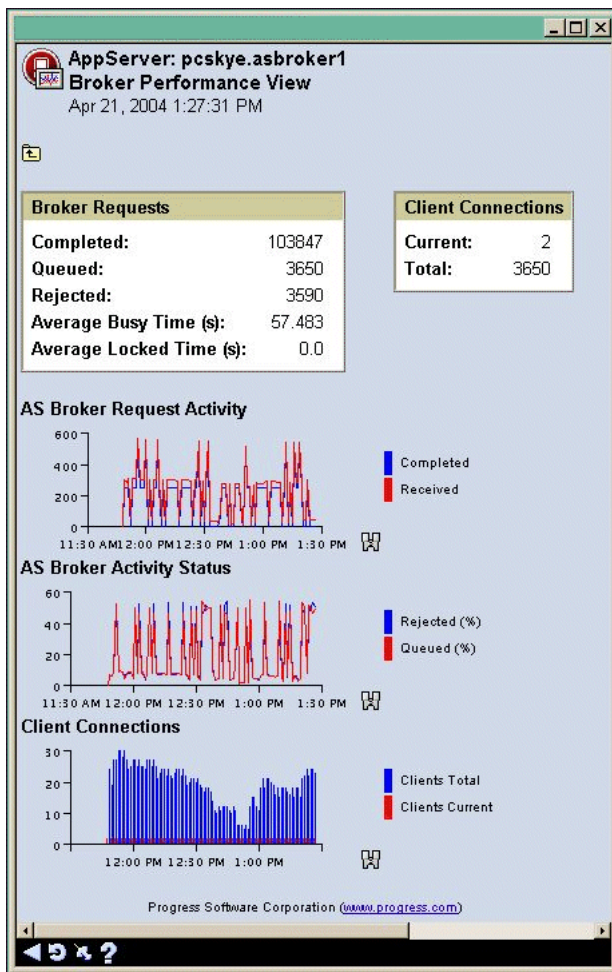
Scanning alert detail on his collection page and also on individual resources he displays in the list frame, he notices that there are no new alerts displayed.

He then accesses the **Database** page and scans for relevant information in the **Operational views** and **Informational views** sections. Finding no clues related to his issues, he next displays the **Server Performance View** details; however, the server state and server pool summary details that display in this view are not helpful. In this situation, he considers where he'd find the most valuable information, and he clicks on the **Broker Performance View**. [Figure 8–4](#) shows the data that appears in the **Broker Performance View** for the **asbroker**.

---

**Note:** In the figures presented in this section, the colors in the graphs are only intended to distinguish one data element from another.

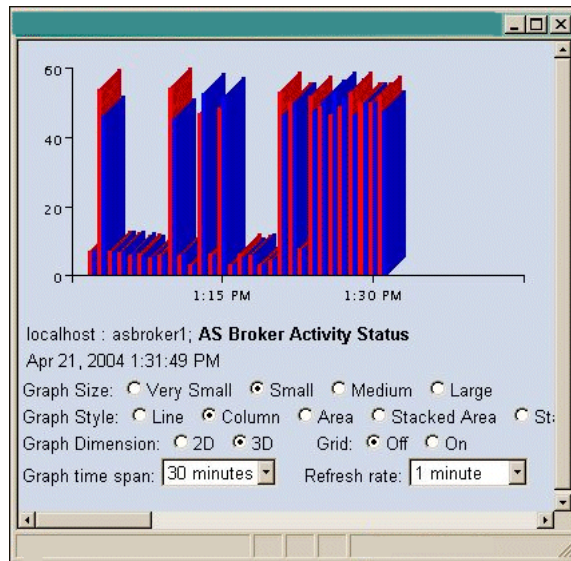
---



**Figure 8–4: Broker Performance View for asbroker1 on host pcskye**

He quickly scans the summarized data in the **Broker Requests**, noting the fact that the total of **Queued** requests is almost the same as the total number of **Rejected** requests. At this point, the administrator knows that there's a problem in this area, but still needs to do more research. From his previous use of the data on the **Broker Performance View** page, he knows that the **AS Broker Activity Status** graph is a representation of the **Queued** and **Rejected** values noted in **Broker Requests**.

He clicks the binocular icon associated with the **AS Broker Activity Status** and the AS Broker Activity Status pinup appears, as shown in [Figure 8–5](#).

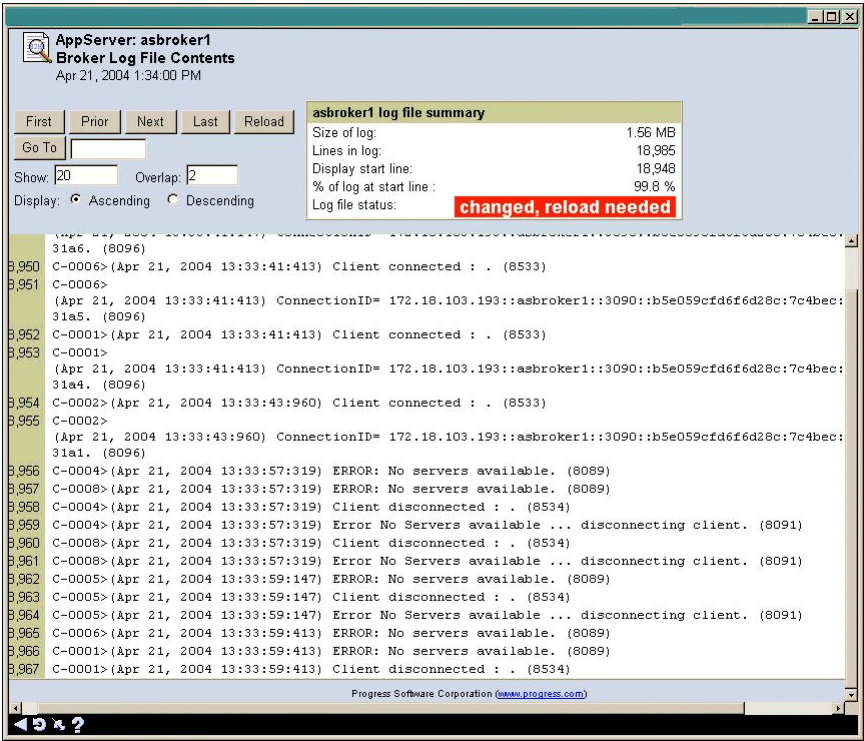


**Figure 8–5: AS Broker Activity Status for asbroker1**

The pinup graph in [Figure 8–5](#) has a much smaller time frame for the data, and the data confirms the very poor performance noted on the main **Broker Performance View** page. In fact, the number of rejected requests really is as high as the number of queued requests. What happened at the time frame indicated on the **AS Broker Activity Status** to cause this dramatic situation?

The administrator now decides to access the asbroker1’s log file, hoping to find more evidence of these same difficulties. Note the several **No Servers available** and the **Clients disconnected** error messages in the log as shown in [Figure 8–6](#).

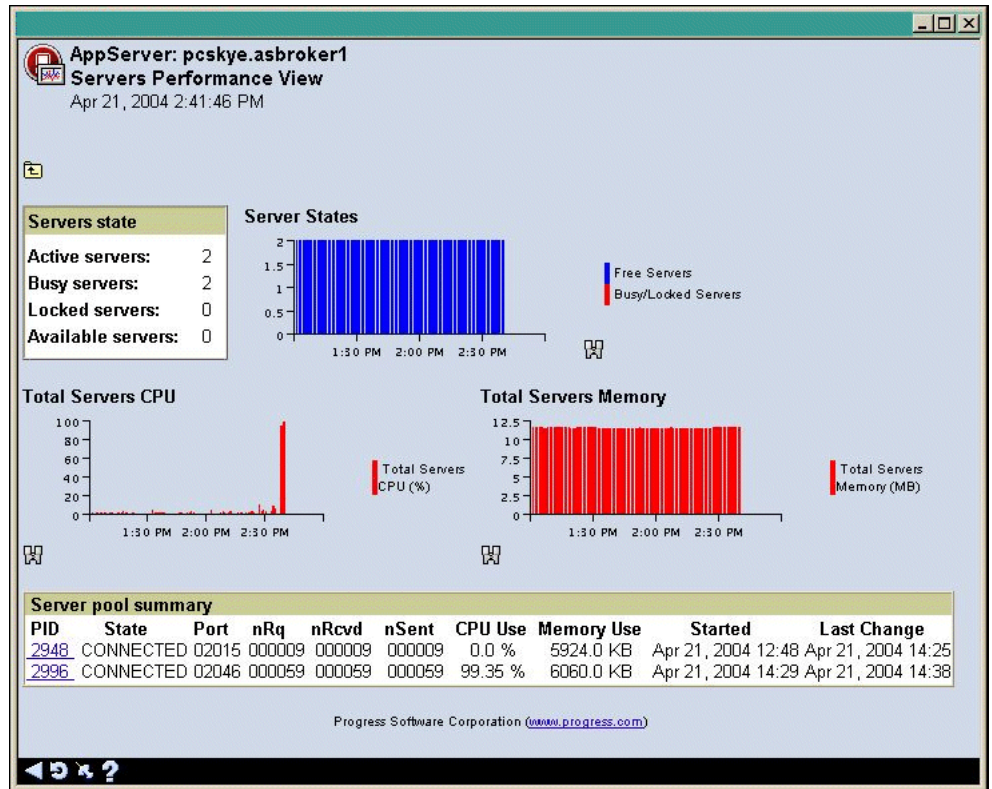
**Note:** For the information that the administrator references about accessing the AppServer log file, see the “[Accessing and reviewing AppServer-related log file data](#)” section on page 4–20.



**Figure 8–6: AppServer asbroker1 log file**

At approximately the same time as the number of rejected requests was starting to approach the total number of queued requests as shown in [Figure 8–4](#), the error log reports that the servers are not available and that connected clients are being disconnected.

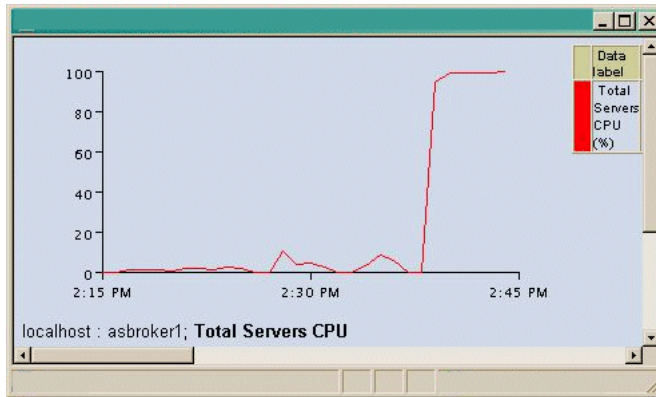
It now makes sense to the administrator as he redisplay the **Servers Performance View** page; all his investigative activities have confirmed that a runaway AppServer process has brought down his network, leaving his users unable to perform their application transaction-related tasks.



**Figure 8–7: Servers Performance View page for asbroker1**

Note the suspicious data in the **CPU Use** column as shown on the **Servers Performance View** page in [Figure 8–7](#). This information further indicates that no CPU consumption is occurring for the servers.

Again, by clicking the binocular icon, the administrator can display this data in a pinup, as shown in [Figure 8–8](#).



**Figure 8–8: Total Servers CPU for asbroker1**

By clicking on **PID 2996** as shown in [Figure 8–7](#), the administrator can display the specific **PID** process id number that is the problem process. By clicking the **Kill** button on the **Broker process** page, he can terminate this process, ending his network and application difficulties.

## Testing and documenting your potential solutions

The administrator puts two plans in place to monitor this particular situation.

### Adding new Fathom monitoring plans

The administrator determines that he has a few additional setup options and controls he might consider implementing. Using Fathom, he:

- Adds a monitoring plan and rule for the CPU on the asbroker process so that the system will alert him should processing not go according to expectations.
- Adds a system level CPU monitor and associated rule also to alert him to unacceptable asbroker processing.



### Gathering more data

Even though the immediate crisis has been resolved, the administrator's primary goal is to try to prevent it from reoccurring. The following lists identifies other ways to explore whether the crisis was a one-time occurrence or a problem that will happen again. The administrator:

- Looks at a larger historical time period in Fathom Management using reports. Report data might show other instances in which the process has gone runaway and what activities occurred to correct the problem.
- Reviews what has changed on the system to determine if a recent change has caused the issue.
- Checks the issues and answers available in the KnowledgeBase (KBase) section of the Knowledge Center available by accessing <http://www.progress.com>.
- Documents the problem and his efforts to correct it so that the information will be available for future reference.

## For more Fathom assessment information

This chapter demonstrates how to use data generated by Fathom features to understand and resolve application issues in a business environment. By exploring performance issues related to a fictitious XYZ Corporation, this chapter helps you to recognize opportunities to use Fathom in similar ways at your company site.

In this chapter, a fictitious system administrator uses a simple investigative process to guide him as he analyzes Fathom-supplied data inputs to resolve two different performance problems. These sample scenarios highlight the benefits that this company obtains from employing Fathom Management. Other investigative aids such as checklists and record keeping activities are presented to further illustrate how to maintain an audit trail of resolution activities.

The application performance topic is a large one. [Table 8–3](#) identifies more information within the Fathom Management documentation set that complements and more fully explores Fathom Management performance-related topics.

**Table 8–3:      References for more in-depth Fathom performance-related information**

For best practices details about...	Refer to...
Configuring the deployment, maintenance, tuning and debugging of OpenEdge Server-based applications, and more in-depth discussions about Fathom features to help you maintain, tune, and monitor OpenEdge servers	<i>OpenEdge Revealed: Achieving Server control with Fathom Management</i> by Doug Merrett.
Performance tuning, installation options, and some troubleshooting hints and tips for maintaining your OpenEdge-based application with Fathom Management	<i>OpenEdge Revealed: Mastering the OpenEdge Database with Fathom Management</i> by Adam Backman.



---

# Index

---

## A

AdminServer warm start  
    affects on Fathom features and  
        functionality 2–18 to 2–20  
    implications 2–19  
    stages 2–18

Agent Unavailable rule 6–27

Application crisis  
    responding to 8–12 to 8–21  
    review checklist 8–13

Application performance  
    AppServer Operational views 8–14  
    AppServer Profile report 8–7  
    assessment process 8–3  
    consulting Fathom Management  
        documentation 8–5, 8–22  
    crisis example 8–6 to 8–21  
    implementing Fathom business  
        requirements 8–3  
    planned review example 8–5 to 8–12

Application performance issues  
    crisis review checklist 8–13  
    examples using a fictitious company 8–3  
        to 8–21  
    general strategy 8–3  
    investigative checklist 8–6

Application review

    planned response 8–5 to 8–12

## AppServer

*See* AppServer server log file monitor  
    *See* AppServer broker log file monitor  
    AppServer broker status details 4–2  
    AppServer Informational views 4–37  
    AppServer Operational views  
        how to access and review the Broker  
            Performance View 4–32 to 4–34  
        how to access and review the Server  
            Performance View 4–35 to 4–36  
        overview 4–31  
        using to troubleshoot application  
            performance issues 8–14 to 8–18  
    Broker Control page content 4–5  
    how to add or trim AppServers 4–17  
    how to change AppServer broker controls  
        4–7 to 4–10  
    how to kill an AppServer process 4–15,  
        4–18 to 4–20  
    how to view broker process details 4–10  
        to 4–12  
    modifying AppServer control settings  
        4–4  
    overview 4–2  
    Server Pool Control page content 4–13 to  
        4–16

AppServer broker log file monitor  
    *See* AppServer log file viewer  
    accessing and reviewing log file data  
        4–20

- characteristics 4–21
- considerations 4–24
- default values 4–22
- how to customize 4–25 to 4–28
- naming conventions 4–21

AppServer log file viewer

- how to use 4–28 to 4–31
- using to troubleshoot application performance issues 8–18

AppServer Profile report

- using to troubleshoot application performance issues 8–7

AppServer server log file monitor

- See* AppServer log file viewer
- accessing and reviewing log file data 4–20
- characteristics 4–21
- considerations 4–24
- default values 4–22
- how to customize 4–25 to 4–28
- naming conventions 4–21

asbrokerlogfileview 4–20

Average Procedure Duration High rule  
6–24, 6–27

## B

Baseline value *See* Configuration Advisor

## C

Collection views

- defining OpenEdge viewlets 2–22 to 2–25

Configuration Advisor

- and data collection 7–3
- determining the effectiveness of your selections 7–17
- how to evaluate the recommended settings 7–13

- how to set rules-related criteria 7–9 to 7–11
- how to submit your selections 7–17
- overview 7–2
- process details
  - a rule's definition 7–6
  - data analysis 7–6
  - recommended threshold rules 7–7
  - user-supplied criteria 7–4
- rules
  - additional considerations 7–3
  - associated with specific WebSpeed broker rules and AppServer broker rules 7–2
- understanding the recommended threshold settings 7–12 to 7–15
- using the Detail button 7–14

## D

Data collection

- and the Configuration Advisor 7–10
- and trending data 3–10, 4–9
- fields related to
  - Broker statistics available 3–10, 4–9
  - Collect Statistics 3–9, 4–9
  - Trend Performance Data 3–10, 4–9
- options and conditions
  - AppServer brokers 4–9
  - WebSpeed brokers 3–9
- procedure to set for a WebSpeed broker 3–10
- procedure to set for an AppServer broker 4–9
- starting, or stopping and restarting AppServer brokers 4–9
- starting, or stopping and restarting WebSpeed brokers 3–9

Default monitoring plan details 6–8 to 6–10

- examples 6–11 to 6–12

Default polling and trend values

- how to review 6–5 to 6–8

Detail menu bar 2–3

**I**

Implementing Fathom business requirements  
example data 8–4

**M**

Main menu bar 2–2  
performing OpenEdge server-related tasks 2–5

Menu bars  
detail menu 2–3  
main menu 2–2

Monitoring plans  
how to maintain 6–13 to 6–18

**N**

NameServer  
how to change NameServer controls 5–5  
modifying NameServer control settings 5–3  
NameServer Control page content 5–4  
NameServer status details 5–2  
overview 5–2

NameServer Informational views  
how to access and review Properties details 5–21 to 5–23

NameServer log file monitor  
*See* NameServer log file viewer  
accessing and reviewing log file data 5–6  
characteristics 5–7  
considerations 5–8  
default values 5–7  
how to customize 5–9 to 5–13

NameServer log file viewer  
how to use 5–13 to 5–15

NameServer Operational views  
how to access and review Status details 5–16 to 5–20

**O**

OpenEdge graphs, viewlets, and pinups  
2–21 to 2–28

OpenEdge main page 2–8  
format and content 2–8  
polling and statistical details 2–11  
section details 2–10

OpenEdge resources  
accessing from the list frame 2–12 to 2–15  
deleting 2–15

OpenEdge servers 1–2  
Fathom features 1–5  
Fathom Management console supported activities  
using the main menu bar 2–5  
overview 1–2 to 1–4  
prerequisites 1–6 to 1–8

**P**

Pinup charts 2–25 to 2–28

**Q**

Queued Request Percent High rule 6–26

**R**

Rejected Request Percent High rule 6–25

Resource monitoring  
overview 6–2 to 6–5  
terminology review 6–3 to 6–5

Rule details  
AppServer rules 6–22  
common rule characteristics 6–20  
general rule conventions 6–19  
NameServer rules 6–21  
WebSpeed rules 6–23

Rule set

- associating with a monitoring plan 6–33
- characteristics of and benefits associated with 6–30
- evaluating rule sets that have one or more rules in common 6–33
- how to copy 6–32
- how to create 6–30
- how to delete 6–33
- how to edit 6–32
- overview 6–29

## S

Schedules

- Default\_Schedule 3–26, 4–26, 5–11

Server Unavailable rule 6–27

## T

Transaction Server

- See WebSpeed Transaction Server

## U

ubroker.properties file 1–4

## W

WebSpeed log file monitor

- See WebSpeed log file viewer
- accessing and reviewing log file data

3–20

- characteristics 3–22
- considerations 3–24
- default values 3–23
- how to customize 3–25 to 3–28

WebSpeed log file viewer

- See WebSpeed log file monitor
- additional considerations 3–30, 4–30, 5–15
- how to use 3–28 to 3–30

WebSpeed Transaction Server

- See WebSpeed log file monitor
- Agent Pool Control page content 3–13 to 3–17
- Broker Control page content 3–5
- how to add or trim agents 3–18
- how to change WebSpeed broker controls 3–8 to 3–11
- how to kill a WebSpeed agent process 3–16, 3–19 to 3–20
- how to view broker process details 3–11 to 3–13
- modifying WebSpeed control settings 3–4
- naming conventions 3–21
- overview 3–2
- WebSpeed broker status details 3–2
- WebSpeed Informational views 3–37
- WebSpeed Operational views
  - how to access and review the Agent Performance View 3–35 to 3–37
  - how to access and review the Broker Performance View 3–31 to 3–34
- overview 3–31