



OpenEdge Management: Servers 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.

A (and design), Actional, Actional (and design), Affinities Server, Allegrix, Allegrix (and design), Apama, Business Empowerment, ClientBuilder, ClientSoft, ClientSoft (and Design), Clientsoft.com, DataDirect (and design), DataDirect Connect, DataDirect Connect64, DataDirect Connect OLE DB, DataDirect Technologies, DataDirect XQuery, DataXtend, Dynamic Routing Architecture, EasyAsk, EdgeXtend, Empowerment Center, eXcelon, Fathom, IntelliStream, Neon, Neon New Era of Networks, O (and design), ObjectStore, OpenEdge, PDF, PeerDirect, Persistence, Persistence (and design), POSSENET, Powered by Progress, PowerTier, ProCare, Progress, Progress DataXtend, Progress Dynamics, Progress Business Empowerment, Progress Empowerment Center, Progress Empowerment Program, Progress Fast Track, Progress OpenEdge, Progress Profiles, Progress Results, Progress Software Developers Network, ProVision, PS Select, SequeLink, Shadow, ShadowDirect, Shadow Interface, Shadow Web Interface, ShadowWeb Server, Shadow TLS, SOAPStation, Sonic ESB, SonicMQ, Sonic Orchestration Server, Sonic Software (and design), SonicSynergy, SpeedScript, Stylus Studio, Technical Empowerment, Voice of Experience, WebSpeed, and Your Software, Our Technology—Experience the Connection are registered trademarks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and/or other countries. AccelEvent, Apama Dashboard Studio, Apama Event Manager, Apama Event Modeler, Apama Event Store, AppsAlive, AppServer, ASPen, ASP-in-a-Box, BusinessEdge, Cache-Forward, DataDirect Spy, DataDirect SupportLink, DataDirect XML Converters, Future Proof, Ghost Agents, GVAC, Looking Glass, ObjectCache, ObjectStore Inspector, ObjectStore Performance Expert, Pantero, POSSE, ProDataSet, Progress ESP Event Manager, Progress ESP Event Modeler, Progress Event Engine, Progress RFID, PSE Pro, SectorAlliance, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Sonic, Sonic Business Integration Suite, Sonic Process Manager, Sonic Collaboration Server, Sonic Continuous Availability Architecture, Sonic Database Service, Sonic Workbench, Sonic XML Server, The Brains Behind BAM, 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. Vermont Views is a registered trademark of Vermont Creative Software in the U.S. and other countries. IBM is a registered trademark of IBM Corporation. JMX and JMX-based marks and 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.

OpenEdge Management Third Party Acknowledgements:

OpenEdge 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), Copyright © 1999-2002 The Apache Software Foundation. All rights reserved (Xerces), Copyright © 1999-2000 The Apache Software Foundation. All rights reserved (Xerces), 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. 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.

OpenEdge 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.

OpenEdge Management includes software Copyright © 1994-2006 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.

OpenEdge Management includes software Copyright 1994-2000 Sun Microsystems, Inc. All Rights Reserved. -Neither the name of or trademarks of Sun may be used to endorse or promote products including or derived from the Java Software technology without specific prior written permission; and Redistributions of source or binary code must contain the above copyright notice, this notice and the following disclaimers: This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN MICROSYSTEMS, INC. AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN MICROSYSTEMS, INC. OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN MICROSYSTEMS, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

OpenEdge Management includes Sonic and ObjectStore software which 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 djacobson@modelobjects.com.

OpenEdge Management includes Sonic software which 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.

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

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

OpenEdge Management includes Sonic and ObjectStore software which include the JMX Technology from Sun Microsystems, Inc. Use and Distribution is subject to the Sun Community Source License available at <http://sun.com/software/communitysource>.

OpenEdge Management includes software Copyright © 2001-2003 James House. 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. This product uses and includes within its distribution, software developed by the Apache Software Foundation (<http://www.apache.org/>).

OpenEdge Management includes code licensed from RSA Security, Inc. Some portions licensed from IBM are available at <http://oss.software.ibm.com/icu4j/>.

OpenEdge Management includes Sonic software, which includes code licensed from Mort Bay Consulting Pty. Ltd. The Jetty Package is Copyright © 1998 Mort Bay Consulting Pty. Ltd. (Australia) and others.



February 2008

Last updated with new content: Release 3.1B

Product Code: 4496; R3.1C

For the latest documentation updates see the [OpenEdge Product Documentation category](http://www.psdn.com/library/kbcategory.jspa?categoryID=129) on PSDN (<http://www.psdn.com/library/kbcategory.jspa?categoryID=129>).

Contents

Preface	Preface-1
1. OpenEdge Management Support for OpenEdge Servers	1-1
OpenEdge servers overview	1-2
AppServer	1-2
WebSpeed Transaction Server	1-2
NameServer	1-3
Managing broker resources	1-3
The ubroker.properties file	1-3
Server and agent details	1-4
Managing NameServer resources	1-4
Features supporting OpenEdge server resources	1-4
OpenEdge server monitoring prerequisites	1-5
Installation	1-5
Discovering and enabling local resources	1-5
Discovering and enabling remote resources	1-6
Role authorization and OpenEdge tasks	1-6
2. Getting Started	2-1
OpenEdge Management console	2-2
OpenEdge Management menu bars	2-2
Using the main menu bar for OpenEdge server tasks	2-4
Introducing the OpenEdge main page	2-6
Main page format and content	2-6
Accessing OpenEdge resource information	2-9
Accessing OpenEdge resources from the list frame	2-9
Accessing an OpenEdge main page	2-10
Deleting OpenEdge resources	2-11
Effects of an AdminServer warm start on OpenEdge Management	2-13
Stages of a warm start	2-13
Warm start implications	2-14
Initiating an AdminServer warm start	2-15
AdminServer warm start process	2-15

Understanding OpenEdge server graphs	2-15
Graphs available on Performance View pages	2-16
Displaying OpenEdge viewlets on a collection view.	2-17
Changing OpenEdge pinup graphical displays	2-19
3. Managing WebSpeed Transaction Server Data	3-1
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	3-6
Viewing broker process details.	3-9
Agent Pool Control page content	3-11
Adding or trimming agents	3-14
Killing a WebSpeed agent process	3-16
Accessing and reviewing WebSpeed-related log file data	3-17
Getting started with log files for WebSpeed resources	3-17
Characteristics of WebSpeed resource log file monitors	3-18
WebSpeed log file monitor default values	3-19
Reviewing predefined log file monitor search criteria	3-20
Customizing a WebSpeed broker log file monitor	3-20
Using the WebSpeed log file viewers	3-23
Refreshing log file data	3-25
Examining WebSpeed-related Operational views	3-25
Accessing and reviewing the Broker Performance View	3-26
Accessing and reviewing the Agent Performance View	3-29
Examining WebSpeed-related Informational views	3-30
4. Managing AppServer Data	4-1
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	4-6
Viewing broker process details	4-9
Server Pool Control page content	4-11
Adding or trimming AppServers	4-14
Killing an AppServer process	4-15
Accessing and reviewing AppServer-related log file data	4-17
Getting started with log files for AppServer resources	4-17
Characteristics of AppServer resource log file monitors	4-18
AppServer log file monitor default values	4-19
Reviewing predefined log file monitor search criteria	4-20
Customizing an AppServer Broker log file monitor	4-21
Using the AppServer log file viewers	4-23
Refreshing log file data.	4-24
Examining AppServer-related Operational views	4-25
Accessing and reviewing the Broker Performance View	4-26
Accessing and reviewing the Server Performance View	4-28
Examining AppServer-related Informational views	4-30

5. Managing NameServer Data	5-1
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	5-6
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	5-12
Refreshing log file data	5-13
Examining NameServer Operational and Informational views	5-14
Accessing and reviewing Operational views	5-14
Accessing and reviewing Informational views	5-18
6. Monitoring Plans and Rules for OpenEdge Server Resources	6-1
OpenEdge Management resource monitoring overview	6-2
Key terms and definitions	6-2
Default polling and trend values	6-4
Default monitoring plan details	6-7
Monitoring plan default values	6-7
OpenEdge default monitoring plan examples	6-8
Maintaining monitoring plans	6-9
Updating monitoring plans	6-10
General rule conventions	6-13
Understanding and using resource monitor rules	6-14
Common rule characteristics	6-15
NameServer rules	6-15
AppServer rules	6-16
WebSpeed rules	6-16
Average Procedure Duration High rule	6-17
Rejected Request Percent High rule	6-18
Queued Request Percent High rule	6-19
Agent (Server) Unavailable rule	6-19
Working with rule sets	6-21
Benefits of using rule sets	6-21
Editing a rule set	6-23
Copying a rule set	6-24
Deleting a rule set	6-24
Adding rule sets that have one or more rules in common	6-24
Associating a rule set with a monitoring plan	6-25
7. Calculating Rule Threshold Settings Using the Configuration Advisor	7-1
Overview	7-2
Rule details	7-2
Data analysis and recommended values overview	7-3
Generating and applying threshold rule settings	7-7
Setting rules-related criteria	7-7
Understanding the recommended threshold settings	7-10
Evaluating recommended settings	7-11
Comparing and selecting threshold settings	7-13
Submitting your threshold setting selections	7-14
Determining the effectiveness of your selections	7-14

8. Analyzing OpenEdge Application Performance.....	8-1
Overview	8-2
Investigating application performance issues	8-3
OpenEdge Management in the workplace	8-3
OpenEdge Management at XYZ Corporation	8-4
Consulting OpenEdge Management documentation	8-4
Planning an application performance review	8-5
Problem definition.	8-5
Initial investigation	8-6
Drilling deeper into OpenEdge Management-supplied data	8-7
Testing and documenting your potential solutions	8-10
Responding to an application crisis	8-11
Problem definition.	8-11
Initial investigation	8-11
Drilling deeper into OpenEdge Management-supplied data	8-13
Testing and documenting your potential solutions	8-18
For more information	8-19
Index	Index-1

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 example	2-6
Figure 2-6:	Broker statistics not available information	2-8
Figure 2-7:	Binoculars icon	2-16
Figure 3-1:	Status section of the WebSpeed main page	3-2
Figure 3-2:	Command and control section of the WebSpeed main page	3-4
Figure 3-3:	Broker Control section of the WebSpeed main page	3-5
Figure 3-4:	Agent Pool Control page example	3-11
Figure 3-5:	WebSpeed-related search criteria	3-18
Figure 3-6:	WebSpeed broker log file viewer example	3-23
Figure 3-7:	WebSpeed Operational views section example	3-25
Figure 3-8:	WebSpeed Informational views section example	3-30
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	4-11
Figure 4-5:	AppServer-related search criteria	4-18
Figure 4-6:	AppServer Broker log file viewer example	4-23
Figure 4-7:	Operational views section example	4-25
Figure 4-8:	Informational views section example	4-30
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	5-12
Figure 6-1:	NameServer instance default monitoring example	6-8
Figure 6-2:	AppServer broker default monitoring plan example	6-9
Figure 6-3:	WebSpeed broker default monitoring plan example	6-9
Figure 6-4:	Available NameServer Rules page	6-15
Figure 6-5:	Available AppServer Rules page	6-16
Figure 6-6:	Available WebSpeed Rules page	6-16
Figure 6-7:	Accessing rule sets from the detail frame	6-23
Figure 7-1:	Configuration Advisor page example	7-4
Figure 7-2:	Configuration Advisor recommended thresholds	7-6
Figure 7-3:	Configuration Advisor recommended values	7-10
Figure 7-4:	Recommended Values field content	7-11
Figure 7-5:	Detail page analysis content	7-12
Figure 8-1:	AppServer Profile Report for Average_Afternoon data	8-8
Figure 8-2:	AppServer Profile Report for Bad_Afternoon data	8-9
Figure 8-3:	AppServer Profile Report for Crisis_Report data	8-10
Figure 8-4:	Broker Performance View for asbroker1	8-14
Figure 8-5:	AS Broker Activity Status for asbroker1	8-15
Figure 8-6:	AppServer asbroker1 log file	8-16
Figure 8-7:	Servers Performance View page for asbroker1	8-17
Figure 8-8:	Total Servers CPU for asbroker1	8-17

Tables

Table 2–1:	Performing OpenEdge server-related activities	2–5
Table 2–2:	Sections of the OpenEdge main page	2–7
Table 2–3:	Performance pages and their graphical content	2–16
Table 2–4:	Graph properties and options for time-based graphs	2–19
Table 3–1:	WebSpeed status details	3–3
Table 3–2:	Additional WebSpeed information	3–4
Table 3–3:	Process statistics section real-time operational data	3–10
Table 3–4:	Agent pool initial configuration field definitions	3–12
Table 3–5:	Agents state field definitions	3–13
Table 3–6:	Agent pool summary field definitions	3–14
Table 3–7:	Broker connection workload details	3–27
Table 3–8:	WebSpeed Broker performance-related graphs	3–28
Table 3–9:	WebSpeed agents performance-related graphs	3–30
Table 4–1:	AppServer status details	4–3
Table 4–2:	Additional AppServer information	4–4
Table 4–3:	Process statistics section real-time operational data	4–10
Table 4–4:	Server pool data initial configuration field definitions	4–12
Table 4–5:	Servers state field definitions	4–13
Table 4–6:	Server pool summary field definitions	4–13
Table 4–7:	AppServer broker connection workload details	4–27
Table 4–8:	AppServer broker performance-related graphs	4–27
Table 4–9:	AppServers performance-related graphs	4–30
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	5–15
Table 5–4:	NameServer details on the Operational Status page	5–17
Table 5–5:	Properties details on the Static Configuration page	5–19
Table 6–1:	Monitoring plan default values	6–7
Table 6–2:	Resource status legend	6–14
Table 6–3:	Examples of AppServer-related ABL procedure entries	6–18
Table 7–1:	Configuration Advisor details	7–7
Table 7–2:	Tasks using the Configuration Advisor Calculations page	7–10
Table 7–3:	Detail page fields and descriptions	7–12
Table 8–1:	Initial investigative checklist	8–6
Table 8–2:	Crisis review checklist	8–12
Table 8–3:	Additional OpenEdge Management performance-related information	8–19

Preface

This Preface contains the following sections:

- [Purpose](#)
- [Audience](#)
- [Organization](#)
- [Using this manual](#)
- [Typographical conventions](#)
- [Examples of syntax descriptions](#)
- [OpenEdge messages](#)

Purpose

This guide details the concepts, features, and procedures that OpenEdge® Management 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 [OpenEdge Management: 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 OpenEdge Management.

Organization

[Chapter 1, “OpenEdge Management Support for OpenEdge Servers”](#)

Presents an overview of the OpenEdge Management features that support the WebSpeed Transaction Server, the AppServer, and the NameServer.

[Chapter 2, “Getting Started”](#)

Describes how to navigate the OpenEdge Management console.

[Chapter 3, “Managing WebSpeed Transaction Server Data”](#)

Explains how to use OpenEdge Management features with WebSpeed Transaction Servers.

[Chapter 4, “Managing AppServer Data”](#)

Explains how to use the OpenEdge Management features with AppServers.

[Chapter 5, “Managing NameServer Data”](#)

Explains how to use the OpenEdge Management features with NameServers.

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

Describes how to set up OpenEdge Management monitoring plans and rules for OpenEdge server resources.

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

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

Chapter 8, “Analyzing OpenEdge Application Performance”

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

Using this manual

OpenEdge provides a special purpose programming language for building business applications. In the documentation, the formal name for this language is *ABL (Advanced Business Language)*. With few exceptions, all keywords of the language appear in all UPPERCASE, using a font that is appropriate to the context. All other alphabetic language content appears in mixed case.

References to ABL compiler and run-time features

ABL is both a compiled and interpreted language that executes in a run-time engine that the documentation refers to as the *ABL Virtual Machine (AVM)*. When documentation refers to ABL source code compilation, it specifies *ABL* or *the compiler* as the actor that manages compile-time features of the language. When documentation refers to run-time behavior in an executing ABL program, it specifies *the AVM* as the actor that manages the specified run-time behavior in the program.

For example, these sentences refer to the ABL compiler’s allowance for parameter passing and the AVM’s possible response to that parameter passing at run time: “ABL allows you to pass a dynamic temp-table handle as a static temp-table parameter of a method. However, if at run time the passed dynamic temp-table schema does not match the schema of the static temp-table parameter, the AVM raises an error.” The following sentence refers to run-time actions that the AVM can perform using a particular ABL feature: “The ABL socket object handle allows the AVM to connect with other ABL and non-ABL sessions using TCP/IP sockets.”

References to ABL data types

ABL provides built-in data types, pre-defined class data types, and user-defined class data types. References to built-in data types follow these rules:



- Like most other keywords, references to specific built-in data types appear in all UPPERCASE, using a font that is appropriate to the context. No uppercase reference ever includes or implies any data type other than itself.
- Wherever *integer* appears, this is a reference to the INTEGER or INT64 data type.
- Wherever *decimal* appears, this is a reference to the DECIMAL data type.
- Wherever *numeric* appears, this is a reference to the INTEGER, INT64, or DECIMAL data type.

References to pre-defined class data types appear in mixed case with initial caps, for example, `Progress.Lang.Object`. References to user-defined class data types appear in mixed case, as specified for a given application example.

Typographical conventions

This manual uses the following typographical conventions:

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

Convention	Description
	This icon (three arrows) introduces a multi-step procedure.
	This icon (one arrow) introduces a single-step procedure.
Period (.) or colon (:)	All statements except DO, FOR, FUNCTION, PROCEDURE, and REPEAT end with a period. DO, FOR, FUNCTION, PROCEDURE, and REPEAT statements can end with either a period or a colon.
[]	Large brackets indicate the items within them are optional.
[]	Small brackets are part of the ABL.
{ }	Large braces indicate the items within them are required. They are used to simplify complex syntax diagrams.
{ }	Small braces are part of the ABL. For example, a called external procedure must use braces when referencing arguments passed by a calling procedure.
	A vertical bar indicates a choice.
...	Ellipses indicate repetition: you can choose one or more of the preceding items.

Examples of syntax descriptions

In this example, `ACCUM` is a keyword, and *aggregate* and *expression* are variables:

Syntax

```
ACCUM aggregate expression
```

FOR is one of the statements that can end with either a period or a colon, as in this example:

```
FOR EACH Customer:
  DISPLAY Name.
END.
```

In this example, `STREAM stream`, `UNLESS-HIDDEN`, and `NO-ERROR` are optional:

Syntax

```
DISPLAY [ STREAM stream ] [ UNLESS-HIDDEN ] [ NO-ERROR ]
```

In this example, the outer (small) brackets are part of the language, and the inner (large) brackets denote an optional item:

Syntax

```
INITIAL [ constant [ , constant ] ]
```

A called external procedure must use braces when referencing compile-time arguments passed by a calling procedure, as shown in this example:

Syntax

```
{ &argument-name }
```

In this example, EACH, FIRST, and LAST are optional, but you can choose only one of them:

Syntax

```
PRESELECT [ EACH | FIRST | LAST ] record-phrase
```

In this example, you must include two expressions, and optionally you can include more. Multiple expressions are separated by commas:

Syntax

```
MAXIMUM ( expression , expression [ , expression ] . . . )
```

In this example, you must specify MESSAGE and at least one *expression* or SKIP [(*n*)], and any number of additional *expression* or SKIP [(*n*)] is allowed:

Syntax

```
MESSAGE { expression | SKIP [ ( n ) ] } . . .
```

In this example, you must specify { *include-file*, then optionally any number of *argument* or &*argument-name* = "*argument-value*", and then terminate with }:

Syntax

```
{ include-file  
  [ argument | &argument-name = "argument-value" ] . . . }
```

Long syntax descriptions split across lines

Some syntax descriptions are too long to fit on one line. When syntax descriptions are split across multiple lines, groups of optional and groups of required items are kept together in the required order.

In this example, WITH is followed by six optional items:

Syntax

```
WITH [ ACCUM max-length ] [ expression DOWN ]
    [ CENTERED ] [ n COLUMNS ] [ SIDE-LABELS ]
    [ STREAM-IO ]
```

Complex syntax descriptions with both required and optional elements

Some syntax descriptions are too complex to distinguish required and optional elements by bracketing only the optional elements. For such syntax, the descriptions include both braces (for required elements) and brackets (for optional elements).

In this example, ASSIGN requires either one or more *field* entries or one *record*. Options available with *field* or *record* are grouped with braces and brackets:

Syntax

```
ASSIGN { [ FRAME frame ] { field [ = expression ] }
        [ WHEN expression ] } ...
      | { record [ EXCEPT field ... ] }
```

OpenEdge messages

OpenEdge displays several types of messages to inform you of routine and unusual occurrences:

- **Execution messages** inform you of errors encountered while OpenEdge is running a procedure; for example, if OpenEdge cannot find a record with a specified index field value.
- **Compile messages** inform you of errors found while OpenEdge is reading and analyzing a procedure before running it; for example, if a procedure references a table name that is not defined in the database.
- **Startup messages** inform you of unusual conditions detected while OpenEdge is getting ready to execute; for example, if you entered an invalid startup parameter.

After displaying a message, OpenEdge proceeds in one of several ways:

- Continues execution, subject to the error-processing actions that you specify or that are assumed as part of the procedure. This is the most common action taken after execution messages.
- Returns to the Procedure Editor, so you can correct an error in a procedure. This is the usual action taken after compiler messages.
- Halts processing of a procedure and returns immediately to the Procedure Editor. This does not happen often.
- Terminates the current session.

OpenEdge messages end with a message number in parentheses. In this example, the message number is 200:

```
** Unknown table name table. (200)
```

If you encounter an error that terminates OpenEdge, note the message number before restarting.

Obtaining more information about OpenEdge messages

In Windows platforms, use OpenEdge online help to obtain more information about OpenEdge messages. Many OpenEdge tools include the following Help menu options to provide information about messages:

- Choose **Help**→**Recent Messages** to display detailed descriptions of the most recent OpenEdge message and all other messages returned in the current session.
- Choose **Help**→**Messages** and then type the message number to display a description of a specific OpenEdge message.
- In the Procedure Editor, press the **HELP** key or **F1**.

On UNIX platforms, use the OpenEdge `pro` command to start a single-user mode character OpenEdge client session and view a brief description of a message by providing its number.



To use the `pro` command to obtain a message description by message number:

1. Start the Procedure Editor:

```
OpenEdge-install-dir/bin/pro
```

2. Press **F3** to access the menu bar, then choose **Help**→**Messages**.
3. Type the message number and press **ENTER**. Details about that message number appear.
4. Press **F4** to close the message, press **F3** to access the Procedure Editor menu, and choose **File**→**Exit**.

OpenEdge Management Support for OpenEdge Servers

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

Topics in this chapter include:

- [OpenEdge servers overview](#)
- [Features supporting OpenEdge server resources](#)
- [OpenEdge server monitoring prerequisites](#)

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

You can set up and use the OpenEdge Management application monitoring capabilities and related features for the following servers:

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

Note: OpenEdge database monitoring is documented in its own manual. See the [OpenEdge Management: Database Management Guide](#) for details.

AppServer

The AppServer is an OpenEdge application that allows you to build and deploy complex distributed applications using ABL. 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. OpenEdge Management supports discovering and monitoring AppServer brokers and managing activities associated with their respective servers from the OpenEdge 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.

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

Note: OpenEdge Management supports monitoring and managing the WebSpeed Transaction Server product. Throughout this guide, the WebSpeed Transaction Server is commonly referred to as 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.

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

Note: The NameServer can also be configured to work with other OpenEdge products such as OpenEdge DataServers and the OpenEdge Adapter for SonicMQ®. For more information, see the relevant OpenEdge product documentation.

Managing broker resources

On systems that support shared processes, a *broker* is a main server process. A broker 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. The broker 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 OpenEdge Management you can optimize broker performance. The OpenEdge Management console supports viewing status details, and controlling, monitoring, and managing broker components to ensure appropriate resources are available.

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 (in Windows), and property settings for each product instance. OpenEdge Management references and displays this configuration data as it relates to the performance data you can gather and analyze in OpenEdge Management.

While you can use OpenEdge 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, OpenEdge Management displays server and agent data. This information provides you additional performance data to better manage your connection workload. 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, OpenEdge Management 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, OpenEdge Management 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 help you determine performance expectations and examine trends.

Managing NameServer resources

OpenEdge Management allows you to view status details for NameServers. This data helps you to monitor and manage effectively the services that NameServers provide.

Features supporting OpenEdge server resources

The following OpenEdge Management 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 OpenEdge Management 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 [OpenEdge Management: Installation and Configuration Guide](#) for details.
- OpenEdge server resources are integrated into the OpenEdge Management console and are accessible using OpenEdge Management features, functionality, and navigational conventions. This includes the use of the basic OpenEdge Management monitoring plan features, such as default and customizable schedules rules, alerts, and actions.
- 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 OpenEdge Management Trend Database.
- OpenEdge Management broker resources support the collection of statistical data for WebSpeed and AppServer brokers. This data can be used to generate OpenEdge Management-based reports and graphs.

OpenEdge server monitoring prerequisites

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

Installation

An OpenEdge 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. Trending is not required in order monitor OpenEdge resources. In order to trend and run reports, however, a trend database must be configured either locally or remotely against another OpenEdge Management installation.

For details about the OpenEdge Management installation and the related configuration process and procedures, see the [OpenEdge Management: Installation and Configuration Guide](#).

Discovering and enabling local resources

Once you complete the installation and configuration steps, OpenEdge Management automatically creates an OpenEdge Management 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 OpenEdge Management detects new OpenEdge resource instances. This process will initially take place after you install and configure OpenEdge Management, and any time new OpenEdge resources are introduced.

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

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

Discovering and enabling remote resources

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

As in the discovery process for local resources, OpenEdge Management enables each remotely defined broker or NameServer instance and begins monitoring them right away. This discovery process occurs any time OpenEdge Management detects new OpenEdge resource instances. This process will initially take place after you install and configure OpenEdge Management, 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: OpenEdge Management runs as a managed service in the AdminServer. Therefore, a *remote resource* is defined as an OpenEdge Management-recognized resource that runs in an AdminServer that is **not** running OpenEdge Management. This resource might be on a machine that is physically separate from the machine where OpenEdge Management is installed, or it could be a different instance of an AdminServer running on the same machine that OpenEdge Management 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.

For users with Operator privileges, the OpenEdge Management 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, see the appropriate sections of the *OpenEdge Management: Installation and Configuration Guide*.

Getting Started

This chapter describes how to navigate the OpenEdge 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 *OpenEdge Management: Resource Monitoring Guide*.

Topics in this chapter include:

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

OpenEdge Management console

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

This section highlights:

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

OpenEdge Management menu bars

The management console supports two menu bars: the main menu bar and the 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 2–4 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 OpenEdge Management console.

Figure 2–2 shows the detail menu bar that appears on the **My Collections Home Default** page when you click **My Dashboard** on the main menu bar.

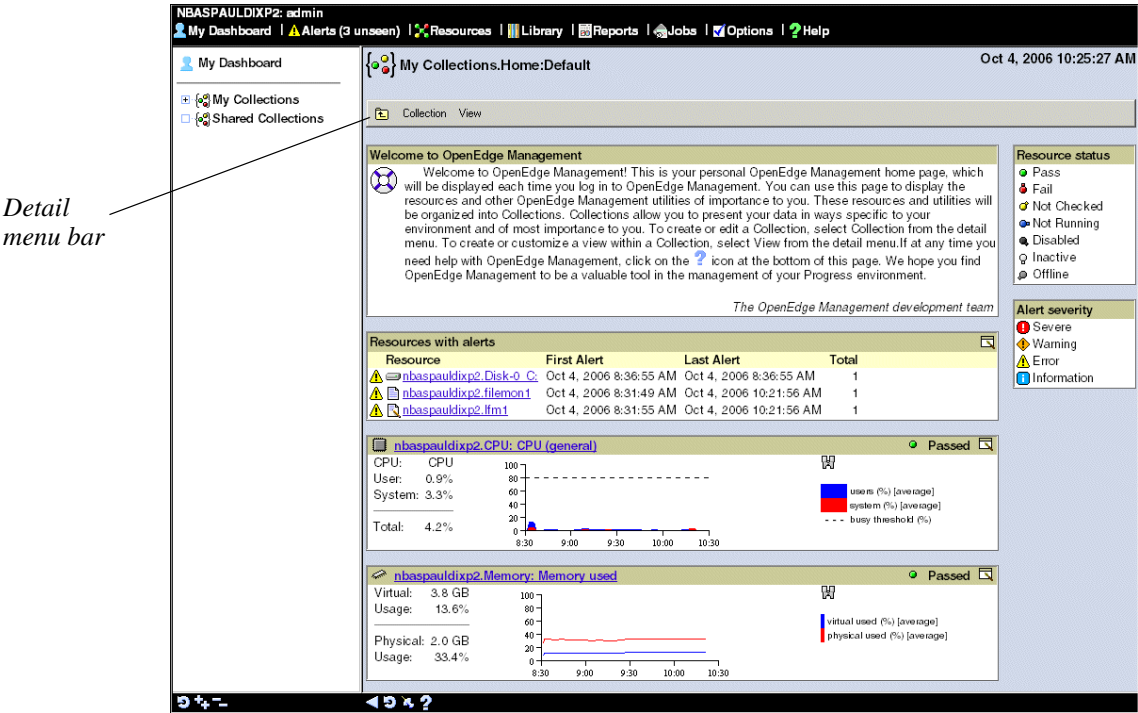


Figure 2–2: Collection views—detail menu bar

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.)

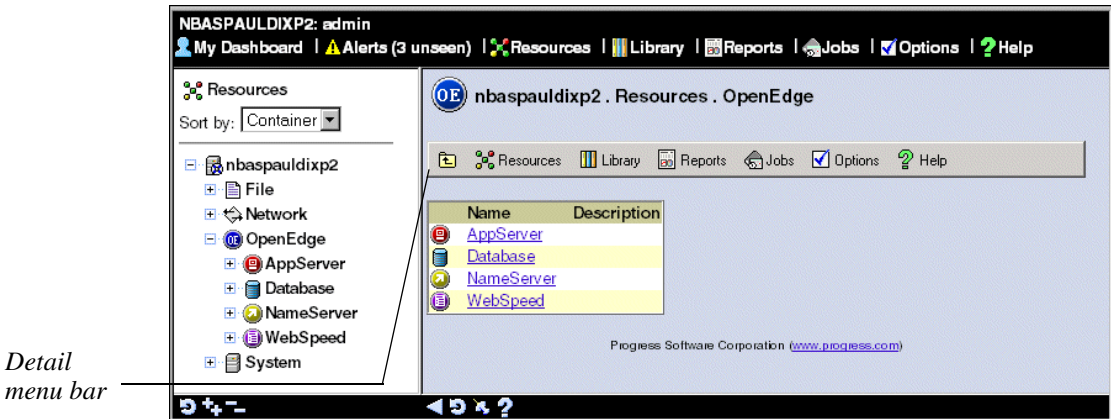


Figure 2–3: Resource category view—detail menu bar

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 OpenEdge Management options more quickly.

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

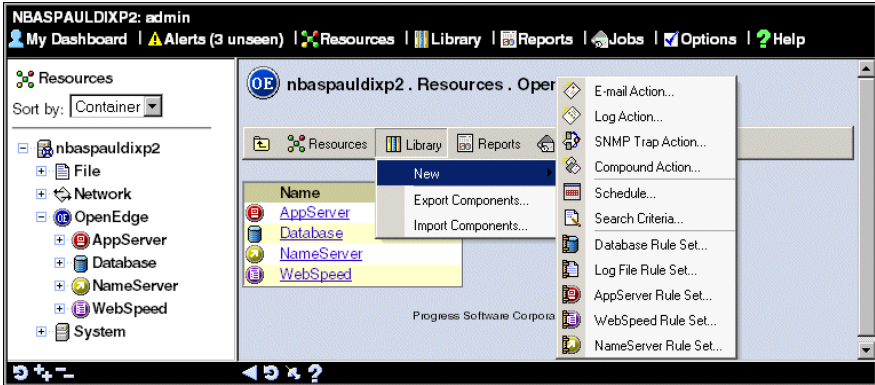


Figure 2–4: Library—New selections from the detail menu bar

For more details about these menu bars and how to navigate through the OpenEdge Management console, see the appropriate section of the [OpenEdge Management: 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 [OpenEdge Management: 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 “[OpenEdge Management menu bars](#)” section on page 2–2 and the [OpenEdge Management: Resource Monitoring Guide](#).

Table 2–1: Performing OpenEdge server-related activities

Select this menu bar category . . .	To perform these OpenEdge server-related activities . . .
Resources	<p>Display and update resource details for:</p> <ul style="list-style-type: none"> • WebSpeed brokers and their associated agents and servers. • AppServer brokers and their associated servers. • NameServer instances and their associated Application Services. <p>When OpenEdge Management is installed, local and remote OpenEdge resources are automatically discovered as Enabled and initial status information is reported. For more information about this topic, see the “Accessing OpenEdge resource information” section on page 2–9.</p> <p>Additional, specific management and monitoring tasks can be performed using the OpenEdge main page. For more information, see the “Introducing the OpenEdge main page” section on page 2–6.</p>
Library	<p>Access and perform all library-based functions available for OpenEdge resources.</p> <p>See the “Working with rule sets” section on page 6–21 for library options such as creating and deleting rule sets for specific OpenEdge resources.</p>
Reports	<p>Access and generate these OpenEdge server-related reports:</p> <ul style="list-style-type: none"> • AppServer Performance. • AppServer Application Profiler. • WebSpeed Performance. • WebSpeed Application Profiler. <p>For details about each of these reports, see the appropriate sections of the OpenEdge Management: Reporting Guide.</p>
Options	<p>Review and update authorization features related to OpenEdge resources.</p> <p>For a summary of OpenEdge authorization options, see the “Role authorization and OpenEdge tasks” section on page 1–6 and the OpenEdge Management: Installation and Configuration Guide.</p>

Note: This OpenEdge 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 [OpenEdge Management: Resource Monitoring Guide](#).

Introducing the OpenEdge main page

The OpenEdge main page is the central user interface of the management console. From the main page you can access information for each OpenEdge resource. Each AppServer broker, NameServer instance, and WebSpeed broker has its own OpenEdge main page; and 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 the **NameServer main** page are referenced, as appropriate.

See the “[Accessing OpenEdge resource information](#)” section on page 2–9 for details on how to access the OpenEdge main page.

Main page format and content

Figure 2–5 shows an example of a **WebSpeed main** page. The format and content sections are identical to the **AppServer main** page, and very similar to the **NameServer** page.

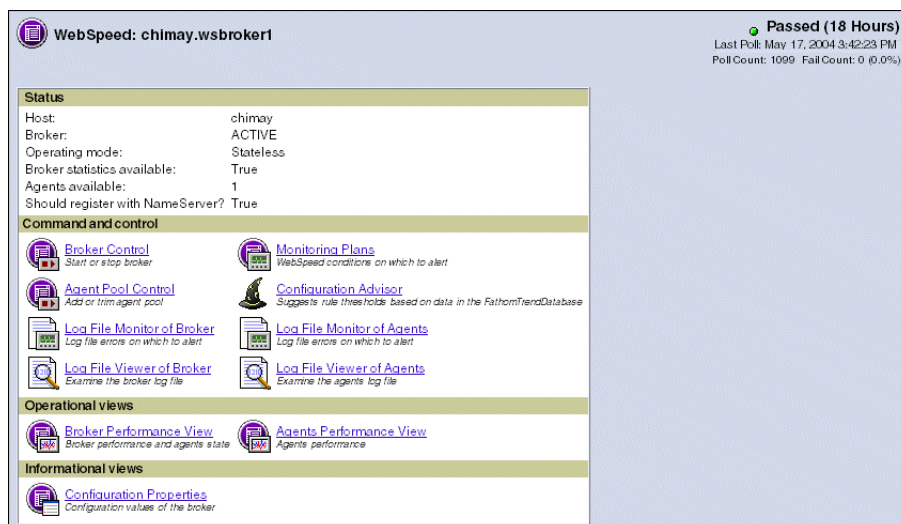


Figure 2–5: WebSpeed main page example

Each OpenEdge main page follows the OpenEdge Management 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 (host) **chimay** as a Transaction Server.

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

This section . . .	Identifies information you use to . . .
Status	Review the current operational statistics for a specific OpenEdge resource.
Command and control	<p>Perform various tasks associated with an OpenEdge resource, including:</p> <ul style="list-style-type: none"> • 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). • Accessing and examining log file monitor and viewer details (for local resources and their associated agents or servers only). • Configuring monitoring plans and rules. • Generating recommended rule threshold settings using the Configuration Advisor. (WebSpeed brokers and AppServer brokers polled rules only.)
Operational views	<p>Help analyze the performance of these OpenEdge resources:</p> <ul style="list-style-type: none"> • AppServer brokers and their associated servers through the display of data and associated graphs. • WebSpeed brokers and their associated agents through the display of data and associated graphs. • Status details about a NameServer instance and its related Application Services.
Informational views	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 upper-right corner of the page 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 OpenEdge Management 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 3–8, as it describes using this field with WebSpeed broker resources.
- The “[Data collection details](#)” section on page 4–8, 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 describes how to access:

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

For more details about navigating the OpenEdge Management console, see the [OpenEdge Management: 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:

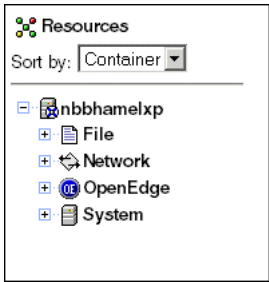
1. Click **Resources** in the management console menu bar.
2. Click **Container** in the **Sort by** field on the list frame. At a minimum, all containers defined to OpenEdge Management appear in the list frame.
3. From the list frame, perform one of the tasks in the following table:

Click the . . .	To . . .
Plus icon (+) associated with the container name whose OpenEdge resources you want to access.	Display the OpenEdge subcategory in the list frame tree below the container's name. See the " Accessing an OpenEdge main page " section on page 2–10.
Container's name.	Display the Container page in the detail frame. The OpenEdge category displays in the resource category box at the bottom of this page. See the " Accessing an OpenEdge main page " section on page 2–10.

Note the following points:

- Local containers have a life preserver associated with the container's icon.
- The plus (+) and minus (-) icons in the list frame indicate whether OpenEdge Management 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.)

The following example 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.

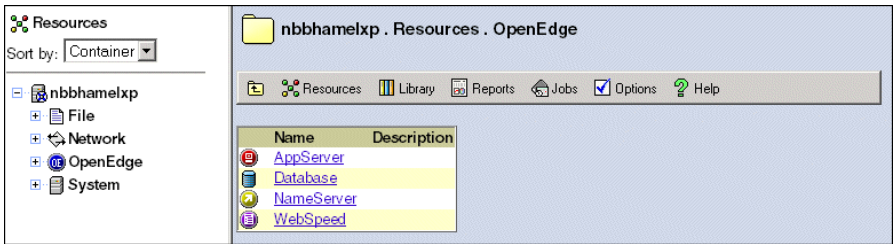
Accessing an OpenEdge main page

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



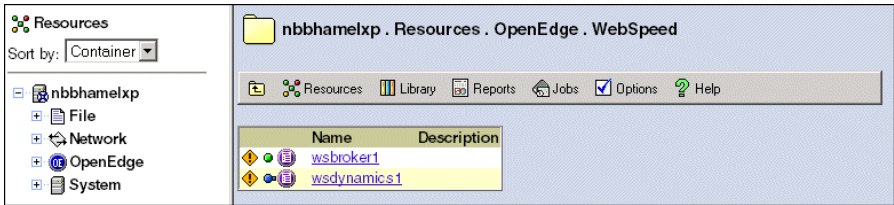
To access an OpenEdge main page:

1. Click the word **OpenEdge** in the list frame. The available **OpenEdge** resources appear in the detail frame:



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 appears:



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

3. Click a name from the list 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 OpenEdge Management console has an associated configuration definition stored in the `ubroker.properties` file. Before you can delete any of these OpenEdge resources from OpenEdge Management, 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, you must stop it. Be sure that the resource is stopped in both OpenEdge Management and Progress Explorer. For details about how to stop an OpenEdge resource, see the appropriate chapter in this guide.
- OpenEdge Management 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 OpenEdge Management functionality, see the “Effects of an AdminServer warm start on OpenEdge Management” section on page 2–13.



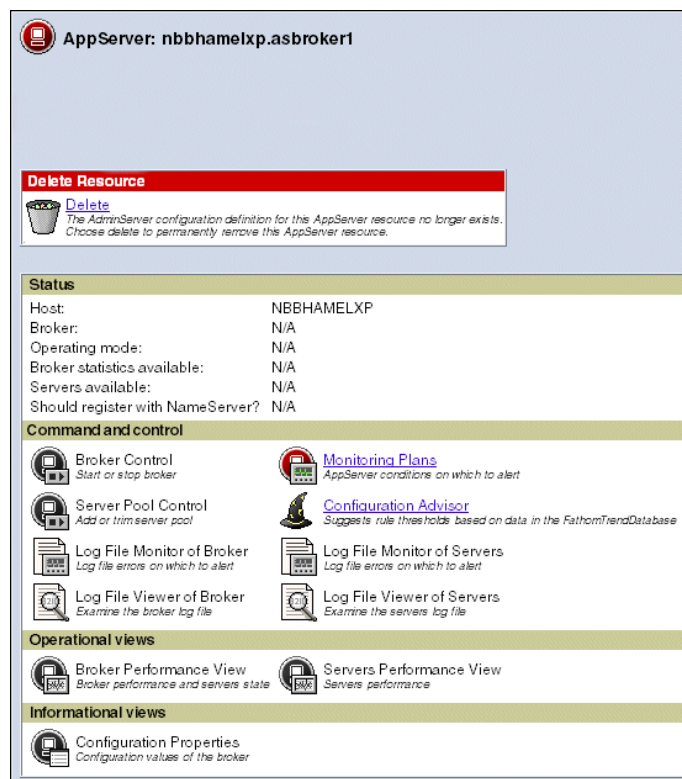
To delete an OpenEdge resource:

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

You can either use 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 2–9 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 **OpenEdge Management 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 OpenEdge Management

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.

Benefits to performing a warm start include:

- Eliminating the use of the Progress Explorer tool to perform minimal configuration changes.
- Eliminating the requirement to stop and restart the AdminServer to accommodate unique situations in which you must manually edit the `ubroker.properties` file. This minimizes any OpenEdge Management 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 property changes.
2. The user saves the changes.
3. The AdminServer validates the changes.

OpenEdge Management cannot accept any other broker-related requests that users might try to initiate. This situation means that you, and other users logged in to OpenEdge Management at this time, might see as unavailable links that are normally available.

4. Complete OpenEdge Management 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** — Displays and validates values associated with the property settings in the `ubroker.properties` file for a Transaction Server.
- **ASCONFIG** — Displays and validates values associated with the property settings in the `ubroker.properties` file for an AppServer.
- **NSCONFIG** — Displays and validates 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 OpenEdge Management activities will be restricted until the AdminServer completes the warm start:

- Data for any OpenEdge resource will not be trended to the OpenEdge Management Trend Database.
- 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, the outcome of specific OpenEdge Management activities will be unpredictable until the warm start is completed. These activities include:

- OpenEdge Management jobs might not be initiated or completed, as expected, due to a possible interruption in the 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 how to initiate an AdminServer warm start. For details about OpenEdge Management feature restrictions and limitations during a warm start, see the [“Warm start implications”](#) section on page 2–14.



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. The AdminServer begins the warm start. See the [“AdminServer warm start process”](#) section on page 2–15.

AdminServer warm start process

After you perform the steps described in the [“Initiating an AdminServer warm start”](#) section on page 2–15, the stages of the AdminServer process occur, as follows:

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

Understanding OpenEdge server graphs

OpenEdge Management 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	<ul style="list-style-type: none"> Broker Request Activity Broker Activity Status Client Connections 	Broker Performance View
WebSpeed Agents	<ul style="list-style-type: none"> Agent States Total Agents CPU Total Agents Memory 	Agents Performance View
AppServer brokers	<ul style="list-style-type: none"> Broker Request Activity Broker Activity Status Client Connections 	Broker Performance View
AppServers	<ul style="list-style-type: none"> 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 OpenEdge Management 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 appears in the pinup. See the “[Changing OpenEdge pinup graphical displays](#)” section on page 2–19 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 the 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.

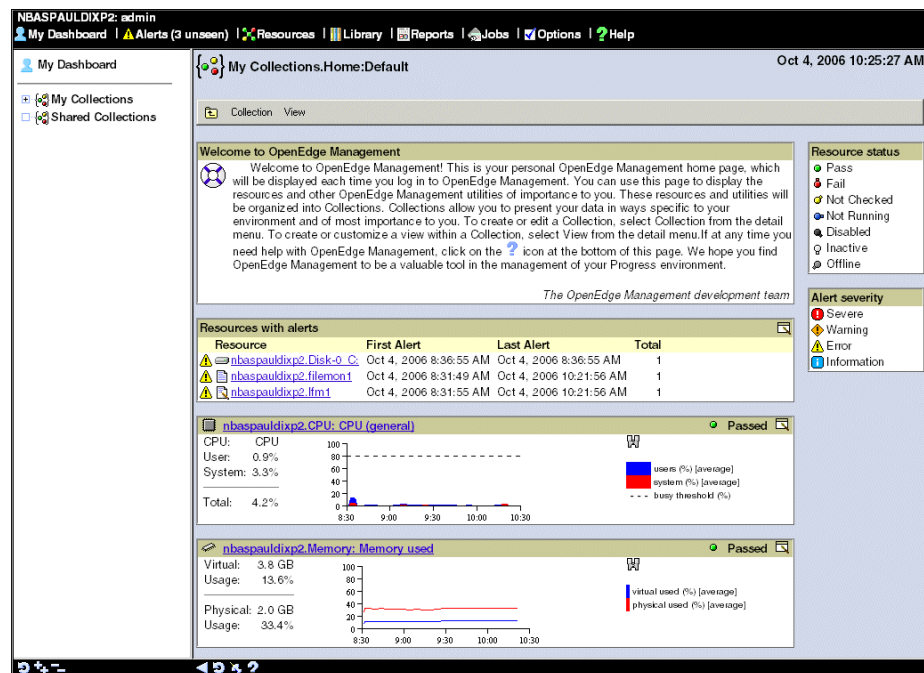
Displaying OpenEdge viewlets on a collection view

Collections allow you to define and organize OpenEdge resource details and to prominently display these details in OpenEdge Management. Specifically, you can define OpenEdge resource viewlets to display resource information in a graphical form on a collections page. In [Table 2–3](#), the second column titled “Data addressing this topic” 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 Dashboard** on the main menu bar. The **My Collections.Home:Default** page appears in the detail frame:



2. If you are updating the **My Collections.Home:Default** 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.)

The **Edit My Collections** page for the collection appears:

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.

5. Click the box associated with a viewlet option to select it.
6. Click **Save**. The main view of the collections page redisplay with the selected viewlets. Use the scroll bar to view all items you defined, as shown in the following example:

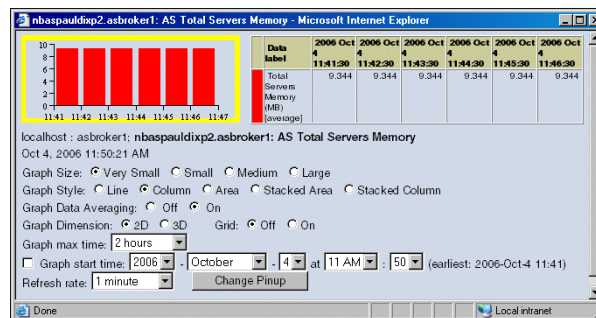
Changing OpenEdge pinup graphical displays

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 whose appearance you can customize. 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. The pinup graph window opens, displaying the graph.
2. Drag the lower-right corner of the window. The page expands, as shown:



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

3. 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 2)

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 Off . If you select On , data displays as a weighted average for the time period set for the Graph max time option. Setting this option to On reduces the number of data points displayed.
Graph Dimension	2D 3D	Changes the display from 2-dimensional to 3-dimensional.
Grid	On Off	Default is Off .

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

Property	Options	Comments
Graph max time	A number of options, from 5 minutes to 2 days .	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 Graph cache option. For details, see the appropriate section in the <i>OpenEdge Management: Installation and Configuration Guide</i> .
Graph start time	Click in checkbox. Provide year/month/day/time start time settings.	<p>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.</p> <p>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.</p> <p>To ensure a meaning graphing of data, review the Graph cache option and the Graph max time setting values as you determine the value for this start time setting.</p>
Refresh rate	None 15 seconds 30 seconds 1 minute 2 minutes 3 minutes 4 minutes 5 minutes 10 minutes 15 minutes	<p>The refresh rate is the rate at which the resource is checked to see if there is more information to put in the graph.</p> <p>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.</p>

- 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 OpenEdge Management features and functionality related to the Transaction Server, as described in the following sections:

- [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

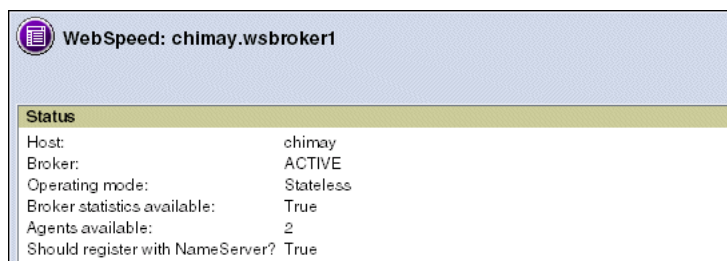
OpenEdge Management supports a variety of tasks that you can perform to manage a Transaction Server, including:

- 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 OpenEdge Management role authorization to perform several of these tasks. See the “[Role authorization and OpenEdge tasks](#)” section on page 1–6 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.

The screenshot shows a web interface for 'WebSpeed: chimay.wsbroker1'. Below the title bar is a 'Status' section with a table of operational details.

Status	
Host:	chimay
Broker:	ACTIVE
Operating mode:	Stateless
Broker statistics available:	True
Agents available:	2
Should register with NameServer?	True

Figure 3–1: Status section of the WebSpeed main page

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

Table 3–1: WebSpeed status details

This field . . .	Displays . . .
Host	The host machine's name.
Broker	<p>The running status of the broker. Possible values are:</p> <ul style="list-style-type: none"> ACTIVE (The broker is currently running.) Not Running (The broker is not currently running.) <p>The broker can also report Starting and Shutting Down values. However, depending on the speed of the machine on which your management console is running, you may not see these intermediary states.</p>
Operating mode	The operating mode of the broker. This mode determines how client requests are dispatched to individual agent processes running on the WebSpeed instance.
Broker statistics available	<p>The status of the broker as it relates to data collection. The possible states are True or False.</p> <p>See the “Data collection details” section on page 3–8 for more information.</p>
Agents available	<p>The number of agents running and available to fulfill a connection request from a client through this broker when the broker's status is ACTIVE.</p> <p>This value can change frequently, reporting the real-time changes in number of agents available.</p>
Should register with NameServer	The status of True or False to indicate whether the broker resource is registered with a NameServer.

These points relate to the fields listed in the first column 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 **Status** section are obtained from either 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.

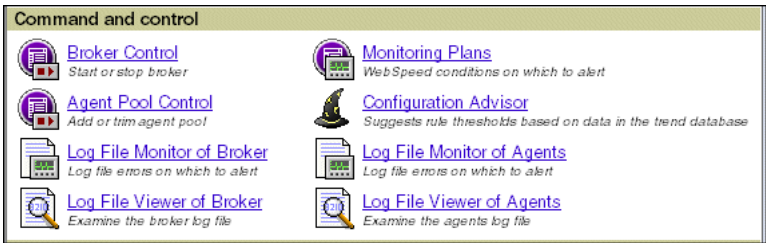


Figure 3–2: Command and control section of the WebSpeed main page

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 . . .
Broker and agent pool log file monitors and viewers.	The “ Accessing and reviewing WebSpeed-related log file data ” section on page 3–17.
Broker monitoring plans and rules.	Chapter 6, “ Monitoring Plans and Rules for OpenEdge Server Resources .”
Broker rule sets.	The “ Working with rule sets ” section on page 6–21.

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 needed.

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

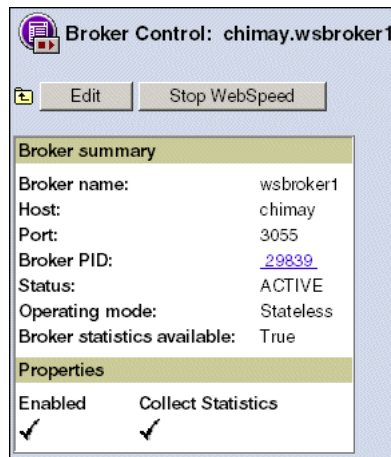


Figure 3–3: Broker Control section of the WebSpeed main page

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 3–8 for details.

The **Broker Control** page consists of two distinct sections:

- [Broker summary section](#)
- [Properties section](#)

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 3–9 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 3–8 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 OpenEdge Management discovers and lists in the list frame under the WebSpeed category are *enabled*, or set, by default. Once a broker is enabled, OpenEdge Management uses the OpenEdge Management-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. OpenEdge Management 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), OpenEdge Management 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 3–8 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 3–8.

Changing WebSpeed Broker controls

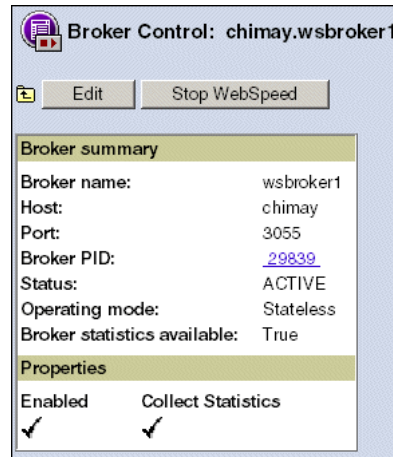
This section describes how to change WebSpeed broker controls.



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 2–9 for the detailed steps.

2. Click **Broker Control** in the **Command and control** section to display the **Broker Control** page, as shown:



The following table highlights the controls that you can change:

To . . .	Then . . .
<p>Change the current setting of the Enabled property.</p> <p>Note: A checkmark displays to indicate that the Enabled 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 Edit. Then select or deselect the Enabled 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 Broker statistics available property displayed in the Broker Summary section of the Broker Control page.</p>	<p>See the “Data collection details” section on page 3–8.</p>
<p>Toggle between stopping and starting the WebSpeed broker, depending on the current value displayed in the Status field.</p> <p>For example, if the broker status currently displays ACTIVE, the button label will read Stop WebSpeed. You can click this button to stop the WebSpeed broker. OpenEdge Management stops this broker and updates the value in the Status field to display Not Running.</p>	<p>Click Stop WebSpeed or Start WebSpeed.</p>
<p>Exit this page without changing any values and redisplay the WebSpeed main page.</p>	<p>Click Cancel.</p>

Data collection details

Data collection ensures that broker-related performance statistics can be trended to the OpenEdge Management Trend Database. 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. (OpenEdge Management automatically updates this field when it detects that the **Collect Statistics** option is enabled after you have started, or stopped and restarted, 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 OpenEdge Management Trend Database 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. Review 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, click **Edit**. Select the **Collect Statistics** option. 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 restarted 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 OpenEdge Management Trend Database.

3. Check the current setting of the **Trend** field in the **Properties** section of the **Broker Control** page. The field must be checked to indicate that this property is set. See the [“Default polling and trend values”](#) section on page 6–4 and the [“Default monitoring plan details”](#) section on page 6–7 for details about this field.

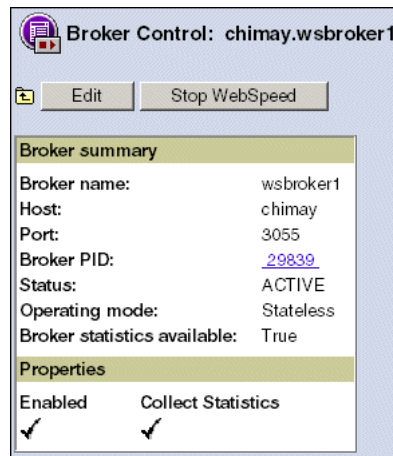
Viewing broker process details

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.

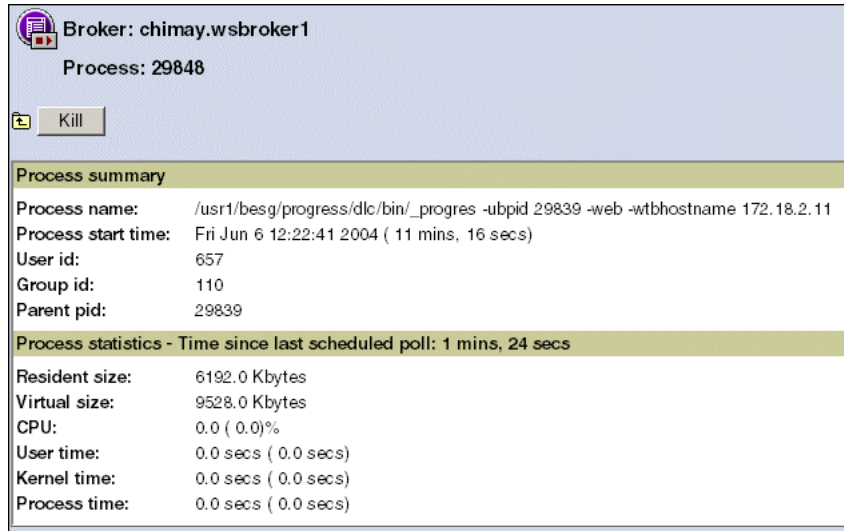


To access broker processing details:


1. Display the **WebSpeed** main page for the WebSpeed broker you want to review. See the [“Accessing OpenEdge resource information”](#) section on page 2–9 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 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 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
(1 of 2)

This field . . .	Identifies . . .
Resident size	The physical size of the process as defined by the host system.
Virtual size	The virtual size of the process as defined by the host system.
CPU%	The percentage of time spent using the CPU in either the user or kernel mode since the last scheduled poll.
User time	The amount of CPU time spent in the user mode since the last scheduled poll.
Kern time	The amount of CPU time spent in the kernel mode since the last scheduled poll.

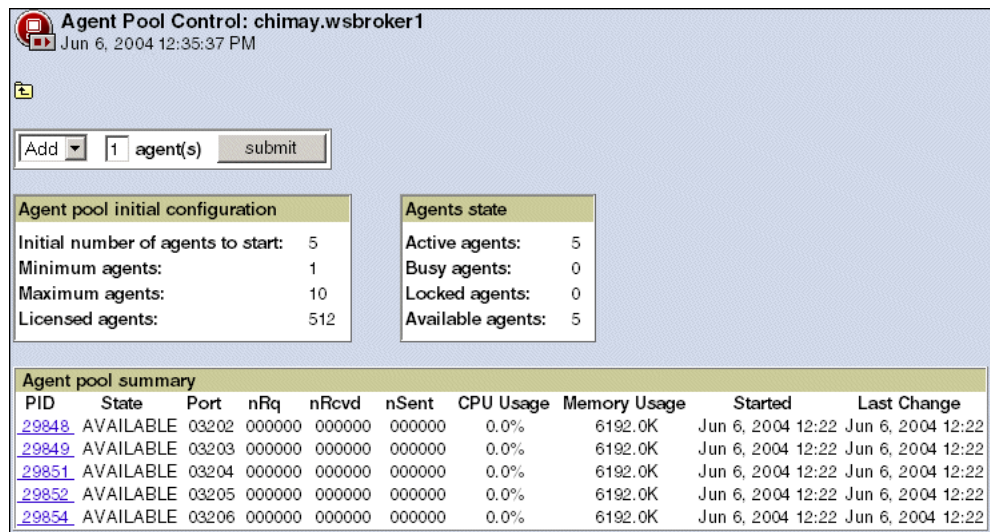
Table 3–3: Process statistics section real-time operational data
(2 of 2)

This field . . .	Identifies . . .
Process time	The sum of the values that display in the User time and Kernel time fields.
Weighted CPU	<p>The 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. The page displays data relevant to your current WebSpeed workload and allows you to add or reduce the number of WebSpeed agents currently running.

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, OpenEdge Management determines which agents to remove. See the “[Adding or trimming agents](#)” section on page 3–14 for detailed steps.
- The following 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 3–16 for the detailed steps.

See the “[Adding or trimming agents](#)” section on page 3–14 for details about how to add or trim agents.

Agent pool initial configuration section

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

[Table 3–4](#) identifies and describes each field that displays in the **Agent pool initial configuration** section.

Table 3–4: Agent pool initial configuration field definitions (1 of 2)

This property field . . .	Identifies . . .
Initial number of agents to start	The value OpenEdge Management 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 Licensed agents field.
Minimum agents	<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 the additional agents necessary to maintain this minimum.</p> <p>If you set a trim value that would require OpenEdge Management to trim the number of agents below the number specified for this field, OpenEdge Management displays a message.</p>

Table 3–4: Agent pool initial configuration field definitions (2 of 2)

This property field . . .	Identifies . . .
Maximum agents	The maximum number of WebSpeed processes that can be running simultaneously. OpenEdge Management will not fulfill add requests you initiate that exceed this specified maximum. OpenEdge Management will display a message to state this condition so that you can reconsider your request and, if necessary, initiate a new request.
Licensed agents	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 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](#) 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 agents	Running.
Busy agents	Serving ABL client requests.
Locked agents	Servicing a bound connection.
Available agents	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

This field . . .	Identifies. . .
PID	The process identifier for this agent. Click on the specific PID number to display a detail page that provides specific information about this agent process and, as necessary, kill the process. See the “ Killing a WebSpeed agent process ” section on page 3–16 for more information.
State	The current execution state of the agent process.
Port	The TCP/IP port number that the agent process uses.
nRq (Number of Requests)	The number of messages sent to the agent process.
nRcvd	The number of messages received by the agent process.
nSent	The number of requests sent by the agent process.
CPU Usage	The percentage of CPU user and system time consumed by a process.
Memory Usage	The amount of virtual memory (in Kbytes) consumed by a process.
Started	The time stamp that indicates when the agent process started. If the broker is restarted for any reason, the PID and the Last Change value might change.
Last Change	The 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 2–9 for the detailed steps.

- 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 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
29848	AVAILABLE	03202	000000	000000	000000	0.0%	6192.0K	Jun 6, 2004 12:22	Jun 6, 2004 12:22
29849	AVAILABLE	03203	000000	000000	000000	0.0%	6192.0K	Jun 6, 2004 12:22	Jun 6, 2004 12:22
29851	AVAILABLE	03204	000000	000000	000000	0.0%	6192.0K	Jun 6, 2004 12:22	Jun 6, 2004 12:22
29852	AVAILABLE	03205	000000	000000	000000	0.0%	6192.0K	Jun 6, 2004 12:22	Jun 6, 2004 12:22
29854	AVAILABLE	03206	000000	000000	000000	0.0%	6192.0K	Jun 6, 2004 12:22	Jun 6, 2004 12:22

- From the drop-down list box, select **Add** or **Trim**.
- 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, OpenEdge Management consults the following two sets of initial configuration details to determine if and how it can honor either request:

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

See the “[Agent pool initial configuration section](#)” section on page 3–12 for information about the configuration details.

- Click **Submit**. Depending on the changes you make and OpenEdge Management’s capability to implement them, you might notice updates to the numeric values displayed in the **Agents state** table. See the “[Agents state section](#)” section on page 3–13 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 necessary page to kill the offending agent's process.

Note: OpenEdge Management references the specific **PID** and its associated date and time start details to be sure of a process' identity 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

Process summary

Process name: /usr1/bscg/progress/dlo/bin/_progres -ubpid 42750 -web -webhostname 172.16.7.195 -webport 41304 -webmaxport 3502 -webname wsbroker1 -vntainstance 5 -ubprofile /usr1/bscg/progress/dlo/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 3–9 for details about this data.

2. Click **Kill** to terminate this process. OpenEdge Management will prompt you once again to verify you want to terminate this process. Click **OK**.

A final status page appears that identifies the status of your kill request and 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** — OpenEdge Management 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 the process.

Accessing and reviewing WebSpeed-related log file data

OpenEdge Management 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 OpenEdge Management log file monitor features and functionality, see the [OpenEdge Management: 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 OpenEdge Management discovers, OpenEdge Management supports monitoring its two associated log file monitors. OpenEdge Management 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 a log file monitor first starts monitoring either a WebSpeed broker or agents, it always starts at the end of the log file.

Naming conventions

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

- **Broker-related log file names** — Displays **vesta.wsbroker1BrokerLogFileMonitor** and **vesta.wsbroker1 WebSpeed Broker Log File Contents**.
- **Agent-related log file names** — Displays **vesta.wsbroker1AgentLogFileMonitor** and **vesta.wsbroker1 WebSpeed Server Log 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 helps 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. OpenEdge Management 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.)

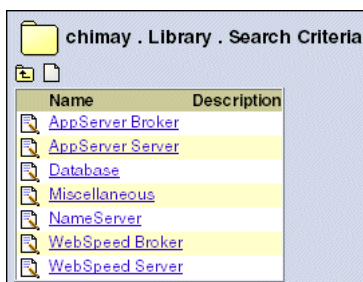


Figure 3–5: WebSpeed-related search criteria

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 cannot be shared at this level. See the [“Customizing a WebSpeed broker log file monitor”](#) section on page 3–20 for details.
- At the OpenEdge Management Component Library level under the appropriate WebSpeed subcategory. The search text and type can be shared at this level. See the [“Working with rule sets”](#) section on page 6–21 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, OpenEdge Management 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 3–20 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 the **On First Poll** property as they relate to log file monitors in general, see the [OpenEdge Management: Resource Monitoring Guide](#).

File Resource Defaults page

OpenEdge Management 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 OpenEdge Management-supplied default value set for the **Polling Interval** field at any time by clicking **Restore 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 you can 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 Server** 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:

Name	Description
Fatal_Memory_Error	Catches the following error: Fatal memory error.
Fatal_Transport_Failure	Catches the following error: Fatal transport failure (error_code).
Insufficient_Stack_Space	Catches the following error: Insufficient stack space. Increase the -s startup parameter. (number of bytes) bytes are needed, only (number of bytes) are available.
Transport_Resources_Unavailable	Catches the following error: Transport resources unavailable.
Unexpected_Transport_Error	Catches the following error: Unexpected transport error received (error_code).
Unknown_Transport_Error	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 3–20 for details.

Customizing a WebSpeed broker log file monitor

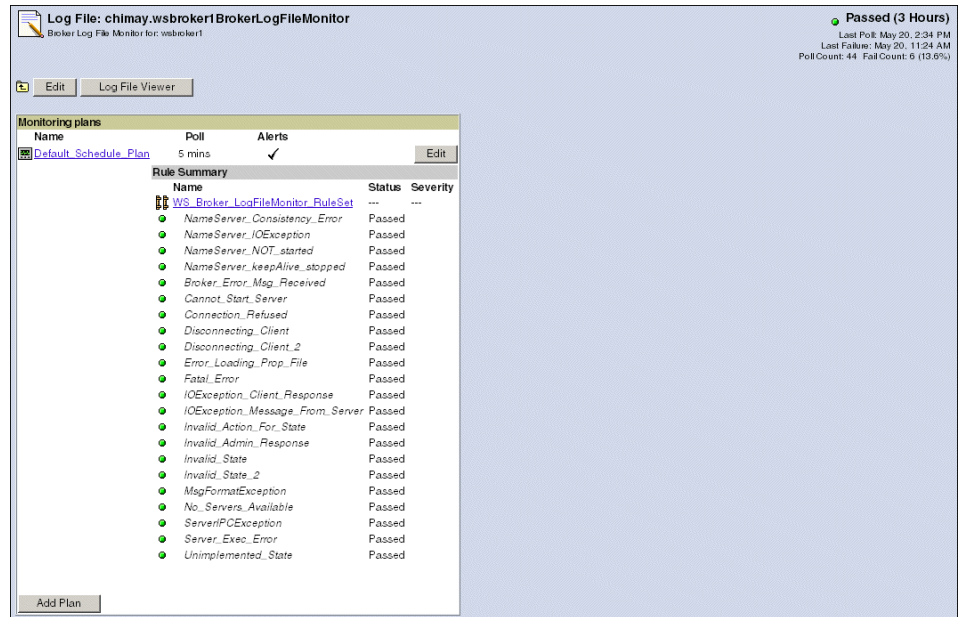
This procedure describes how to customize a WebSpeed broker log file monitor. Use these same general steps to customize a WebSpeed agent log file monitor.



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 2–9.

- 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:



- Customize or view the contents of a WebSpeed broker log file monitor as described in the following table:

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

Note: OpenEdge Management 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 shown in the following table:

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

- b. Select the appropriate values from the **Severity** and **On Alert Action Perform** fields to complete the alert severity and action definition 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 the 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 *OpenEdge Management: 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.

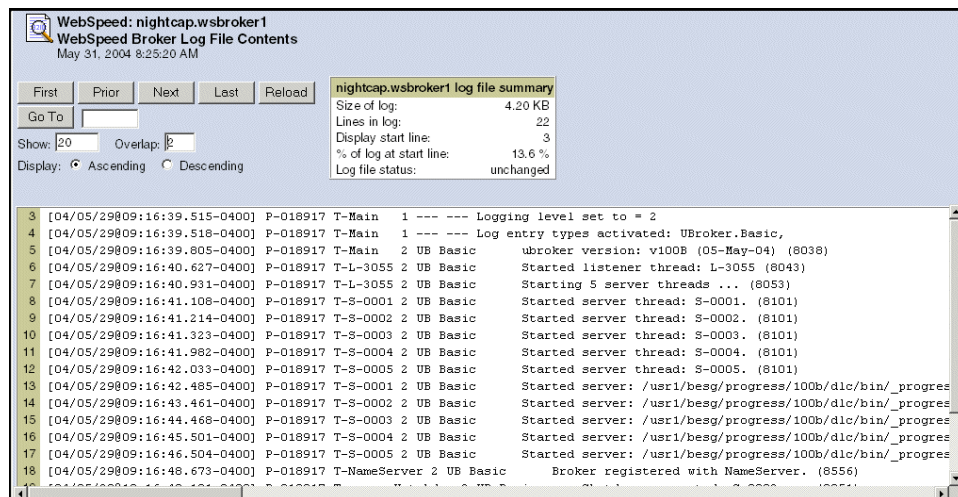


Figure 3–6: WebSpeed broker log file viewer example

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 must not 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 must not 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 OpenEdge Management 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 the 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 OpenEdge Management uses to automatically refresh the management console.



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

Refresh data to avoid the following situations:

- OpenEdge Management considers a viewer that has been inactive for more than four hours “stale.” Once a viewer becomes stale, OpenEdge Management releases ninety-five percent of any memory it holds. If you try to use a stale viewer, OpenEdge Management 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.
- OpenEdge Management considers a viewer that has been inactive for forty-eight hours “dead.” Once a viewer dies, OpenEdge Management 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.
- A 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 3–8 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.

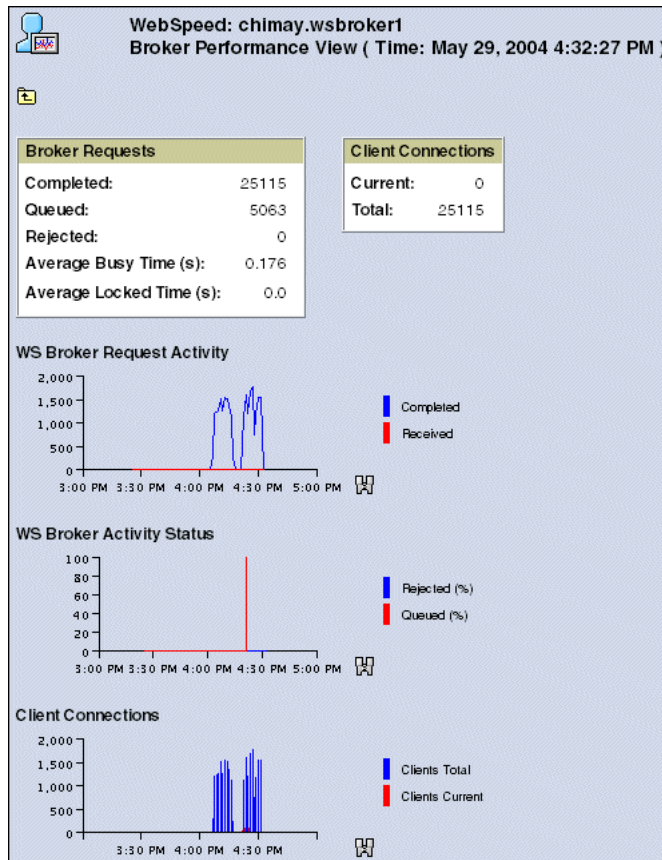
Accessing and reviewing the Broker Performance View

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.



To display and review this information:

1. Display the **WebSpeed** main page for the WebSpeed broker instance you want to review. See the [“Accessing OpenEdge resource information”](#) section on page 2–9 for the detailed procedure.
2. Click **Broker Performance View** in the **Operational views** section. OpenEdge Management 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 consists 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 . . .
Completed	Number of broker requests fulfilled.
Queued	Number of broker requests to be processed.
Rejected	Number of broker requests that could not be processed.
Average Busy Time (ms)	Average amount of time that the broker is busy servicing requests (expressed in milliseconds).
Average Locked Time (ms)	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 and 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**. If 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 3–8 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 briefly describes each of these graphs.

Table 3–8: WebSpeed Broker performance-related graphs

This WebSpeed Broker graph . . .	Displays two lines of . . .
WS Broker Request	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.
WS Broker Activity Status	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 OpenEdge Management 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.
Client Connections	<p>Client connections related to this broker over a specified time period. 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>Note: It is possible for this graph to accurately show that the number of current connections is higher than the total number of connections. The Clients Total reflects only new connections over the specified time period. In contrast, the Clients Current reflects all 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 2–19 for details about changing the data appearance of graphs.

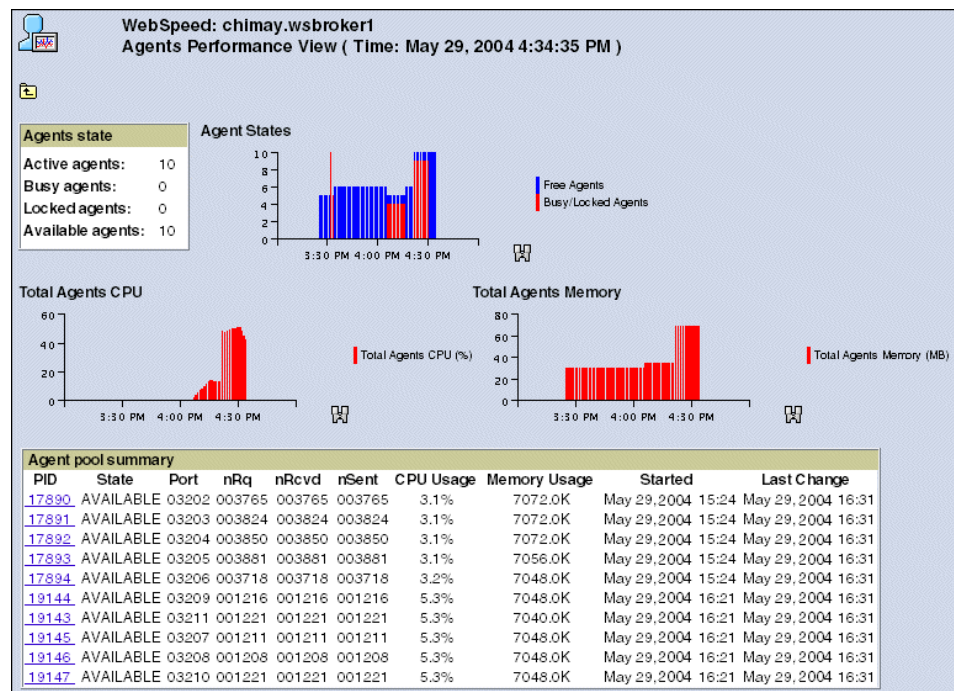
Accessing and reviewing the Agent Performance View

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 2–9 for the detailed procedure.
2. Click **Agent Performance View** in the **Operational views** section to display the **WebSpeed Agents Performance View** page, as shown:



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 3–16 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 3–8 for details.

One display format for these graphs is a line graph, as shown on the **Broker Performance View** page. 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 . . .
Agent States	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.
Total Agents CPU	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.
Total Agents Memory	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 2–19 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 steps are required 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 2–9 for the detailed procedure.
2. Click **Configuration Properties** in the **Informational views** section to display the **Configuration Properties** page:

WebSpeed: nbbhamelp.wsbroker1
Raw Configuration Properties
Jun 6, 2004 12:35:37 PM

Configuration Properties

PROPATH:	@ {WinChar Startup\PROPATH}; @ {WorkPath}
svrMaxPort:	3502
brkrLogThreshold:	0
brkrLogAppend:	1
brkrLoggingLevel:	2
controllingNameServer:	NS1
collectStatsData:	0
fileUploadDirectory:	
uuid:	932.99.000.ZZZ:1ee77e:cf3bbe04fd:-8000
autoTrimTimeout:	1800
hostName:	
applicationURL:	
portNumber:	3055
minSvrInstance:	1
defaultCookiePath:	
operatingMode:	Stateless
registrationMode:	Register-IP
wsRoot:	/webspeed31D
svrDebug:	Enabled
description:	A sample WebSpeed Transaction Server
autoStart:	0
brokerLogFile:	@ {WorkPath}\wsbroker1.broker.log
registrationRetry:	30
maxClientInstance:	512
svrStartupParam:	-p web\objects\web-disp.p -weblogerror
svrLogThreshold:	0
maxSvrInstance:	10
svrLoggingLevel:	2
svrAppMode:	Development
defaultService:	0
connectingTimeout:	60
userName:	
initialSvrInstance:	5
appserviceNameList:	wsbroker1
registerNameServer:	1
brkrLogEntries:	0
password:	
svrLogFile:	@ {WorkPath}\wsbroker1.server.log
infoVersion:	9010
defaultCookieDomain:	
classMain:	com.progress.ubroker.broker.ubroker
svrStartupTimeout:	3
priorityWeight:	0
requestTimeout:	15
svrLogAppend:	1

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

Managing AppServer Data

This chapter presents OpenEdge Management features and functionality related to the AppServer, as outlined in the following sections:

- [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

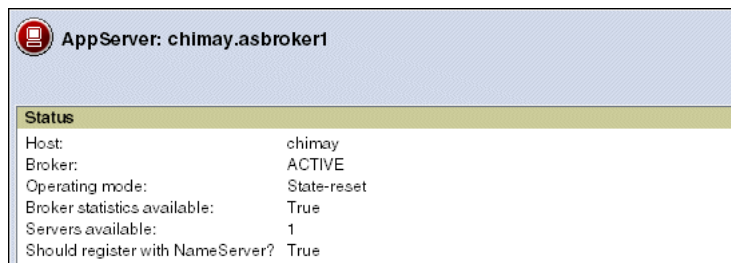
OpenEdge Management supports a variety of tasks that you can perform to manage an AppServer, including:

- 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 OpenEdge Management role authorization to perform several of these tasks. See the [“Role authorization and OpenEdge tasks”](#) section on page 1–6 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.



AppServer: chimay.asbroker1	
Status	
Host:	chimay
Broker:	ACTIVE
Operating mode:	State-reset
Broker statistics available:	True
Servers available:	1
Should register with NameServer?	True

Figure 4–1: AppServer Status section sample

Table 4–1 describes each of the AppServer broker details in the **Status** section of the AppServer main page.

Table 4–1: AppServer status details

This field . . .	Displays . . .
Host	The host machine's name.
Broker	<p>The running status of the broker. Possible values are:</p> <ul style="list-style-type: none"> ACTIVE (The broker is currently running.) Not Running (The broker is not currently running.) <p>The broker can also report Starting and Shutting Down values; however, depending on the speed of the machine on which your management console is running, you may not see these intermediary states.</p>
Operating mode	<p>The 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: Stateless, State-free, State-aware, or State-reset.</p>
Broker statistics available	<p>The status of the broker as it relates to data collection. The possible states are True or False.</p> <p>See the “Data collection details” section on page 4–8 for more information about data collection.</p>
Servers available	<p>The 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 ACTIVE.</p> <p>This value can change frequently, reporting the real-time changes in number of servers available.</p>
Should register with NameServer	The status of True or False to indicate whether or not the broker resource is registered with a NameServer.

The following points relate to the fields listed 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 the 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.

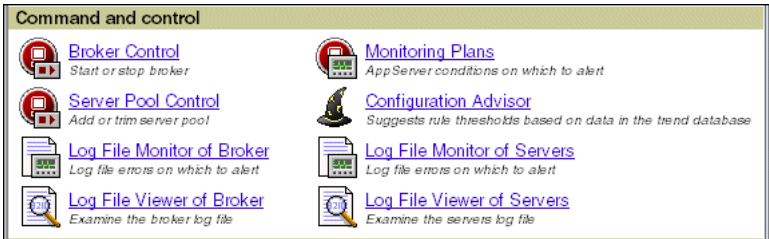


Figure 4–2: Command and control section example

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 . . .
Broker and server pool log file monitors and viewers.	The “ Accessing and reviewing AppServer-related log file data ” section on page 4–17.
Broker monitoring plans and rules.	Chapter 6, “ Monitoring Plans and Rules for OpenEdge Server Resources. ”
Broker rule sets.	The “ Working with rule sets ” section on page 6–21.

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 needed. [Figure 4–3](#) shows an example of the **Broker Control** page.

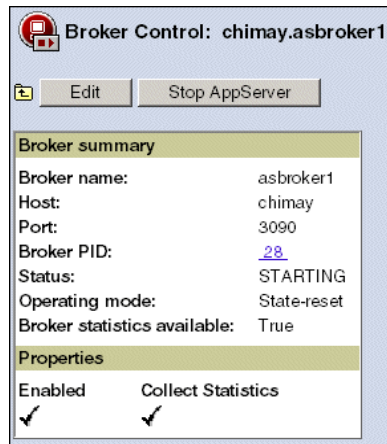


Figure 4–3: Broker Control page example

Note: The values associated with the **Broker statistics available** field and the **Collect Statistics** property are interdependent. See the procedure in the [“Data collection details”](#) section on page 4–8 for additional information.

The following describes 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 4–9 for additional information.
- 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 4–8 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 OpenEdge Management discovers and lists in the list frame under the AppServer category are *enabled* by default. Once a broker is enabled, OpenEdge Management uses the OpenEdge Management-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. OpenEdge Management 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), OpenEdge Management 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 4–8 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 4–8.

Changing AppServer Broker controls

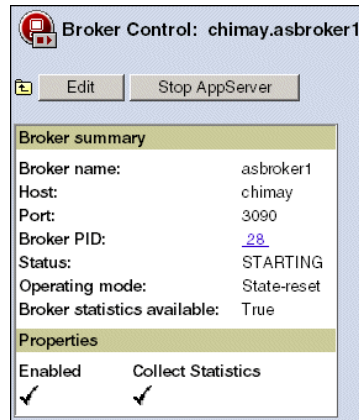
This section describes how to change AppServer broker controls.



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 2–9 for the detailed steps.

2. Click **Broker Control** in the **Command and control** section to display the **Broker Control** page, as shown:



This following table highlights the controls that you can change in the **Broker Control** page settings:

To . . .	Then . . .
<p>Change the current setting of the Enabled property.</p> <p>(A checkmark displays to indicate that the Enabled property is set. To clear this option, click the checkmark.)</p>	<p>Click Edit. Then select or clear the Enabled property.</p> <p>You must also restart the AppServer broker so that the property change is recognized.</p>
<p>Change the current setting of the Broker statistics available property displayed in the Broker summary section of the Broker Control page.</p>	<p>See the “Data collection details” section on page 4–8.</p>
<p>Toggle between stopping and starting the AppServer broker, depending on the current value displayed in the Status field. For example, if the broker status currently displays ACTIVE, the button label will read Stop AppServer. Click this button to stop the AppServer broker. OpenEdge Management stops the broker and updates the value in the Status field to display Not Running.</p>	<p>Click Stop AppServer or Start AppServer.</p>
<p>Exit this page without changing any values. The AppServer main page redisplay in the management console.</p>	<p>Click Cancel.</p>

Data collection details

Data collection ensures that broker-related performance statistics can be trended to the OpenEdge Management Trend Database. 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** option.
- 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. (OpenEdge Management automatically updates this field when it detects that the **Collect Statistics** option was enabled after you started, or stopped and restarted, the AppServer broker.)

On the AppServer broker resource monitoring plan, you must also select 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 OpenEdge Management Trend Database 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. The field must be selected 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 restarted 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 OpenEdge Management Trend Database.

3. Check the current setting of the **Trend** field in the **Properties** section of the **Broker Control** page. The field must be selected to indicate that this property is set. See the [“Default polling and trend values”](#) section on page 6–4 and the [“Default monitoring plan details”](#) section on page 6–7 for details about this field.

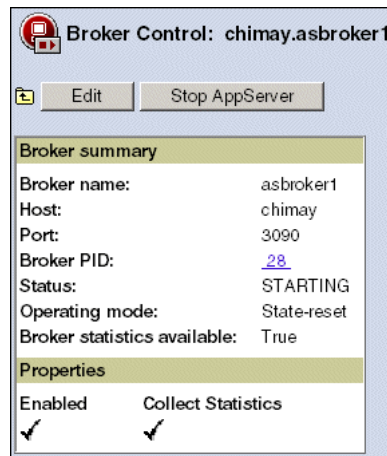
Viewing broker process details

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 2–9 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.asbroker1
Broker PID: 25554

Process summary

Process name: /usr1/besg/progress/dlc/jre/bin/./bin/sparc/native_threads/java-ms8m-mx128m
Process start time: Fri Jun 6 12:22:41 2004 (3 hours, 53 mins, 20 secs)
User id: 657
Group id: 110
Parent pid: 1

Process statistics - Time since last scheduled poll: 11 secs

Resident size: 29744.0 Kbytes
Virtual size: 73864.0 Kbytes
CPU: 0.0 (0.6)%
User time: 0.0 secs (71.0 secs)
Kernel time: 0.0 secs (8.0 secs)
Process time: 0.0 secs (79.0 secs)

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 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 (1 of 2)

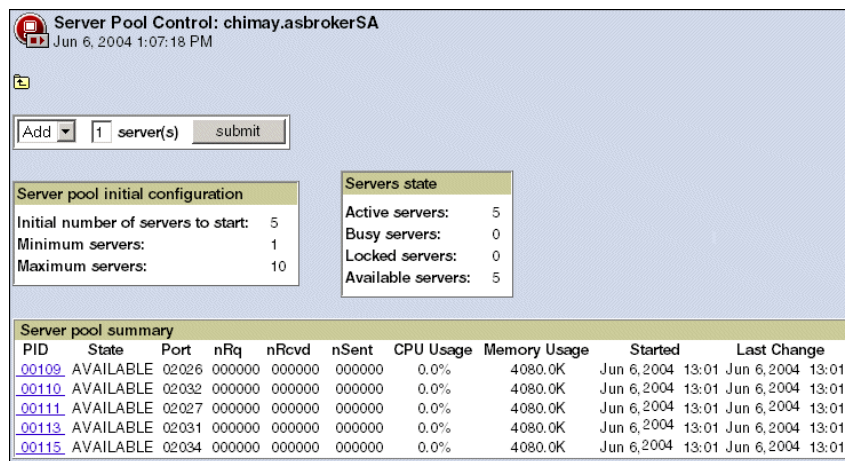
This field . . .	Identifies . . .
Resident size	The physical size of the process as defined by the host system.
Virtual size	The virtual size of the process as defined by the host system.
CPU	The percentage of time spent using the CPU in either the user or kernel mode since the last scheduled poll.
User time	The amount of CPU time spent in the user mode since the last scheduled poll.
Kernel time	The amount of CPU time spent in the kernel mode since the last scheduled poll.

Table 4–3: Process statistics section real-time operational data (2 of 2)

This field . . .	Identifies . . .
Process time	The sum of the values that display in the User time and Kernel time fields.
Weighted CPU	<p>The 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>

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 to 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 consists of the following:

- An add/trim selection control that you use to specify the activity you want to perform. When you initiate a manual trim request, OpenEdge Management determines which server(s) to actually remove. See the [“Adding or trimming AppServers”](#) section on page 4–14 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 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 4–15 for the detailed steps.

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 read-only mode.

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

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

This displayed property field . . .	Identifies . . .
Initial number of servers to start	The value OpenEdge Management references when the AppServer broker starts AppServers.
Minimum servers	<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 the additional servers needed to maintain the minimum.</p> <p>If you set a trim value that requires OpenEdge Management to trim the number of servers below the number specified for this field, OpenEdge Management displays a message.</p>
Maximum servers	The maximum number of AppServer processes that can be running simultaneously. OpenEdge Management will not fulfill add requests you initiate that will exceed the specified maximum. OpenEdge Management will display a message to state this condition so that you can reconsider 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 describes each field presented in the **Servers state** section.

Table 4–5: Servers state field definitions

This server state . . .	Identifies the number of AppServers currently . . .
Active	Running.
Busy	Serving ABL client requests.
Locked	Servicing a bound connection. (This state applies to a stateless AppServer.)
Available	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

(1 of 2)

This field identifies . . .	Identifies . . .
PID	The process identifier for this Appserver. Click the PID number to display a detail page that provides specific information about this server process and, as necessary, kill the process. See the “ Killing an AppServer process ” section on page 4–15 for more information.
State	The current execution state of the AppServer process.
Port	The TCP/IP port number that the AppServer process uses.
nRq (Number of Requests)	The number of messages sent to the AppServer process.

Table 4–6: Server pool summary field definitions

(2 of 2)

This field identifies . . .	Identifies . . .
nRcvd	The number of messages received by the AppServer process.
nSent	The number of requests sent by the AppServer process.
CPU Usage	The percentage of CPU user and system time consumed by a process.
Memory Usage	The amount of virtual memory (in Kbytes) consumed by a process.
Started	The time stamp that indicates when the AppServer process started. If the broker is restarted for any reason, the PID and the Last Change value might change.
Last Change	The time stamp that indicates when the AppServer process last changed execution state.

Adding or trimming AppServers

This section describes how to add or trim AppServers.



To initiate an AppServer add or trim request:

1. Display the **AppServer main** page for the broker you want to start. See the “[Accessing OpenEdge resource information](#)” section on page 2–9 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
00109	AVAILABLE	02026	000000	000000	000000	0.0%	4080.0K	Jun 6, 2004 13:01	Jun 6, 2004 13:01
00110	AVAILABLE	02032	000000	000000	000000	0.0%	4080.0K	Jun 6, 2004 13:01	Jun 6, 2004 13:01
00111	AVAILABLE	02027	000000	000000	000000	0.0%	4080.0K	Jun 6, 2004 13:01	Jun 6, 2004 13:01
00113	AVAILABLE	02031	000000	000000	000000	0.0%	4080.0K	Jun 6, 2004 13:01	Jun 6, 2004 13:01
00115	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, OpenEdge Management consults two sets of initial configuration details 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 4–12 for details about these configuration details.

5. Select **submit**. Depending on the changes you make and OpenEdge Management’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 4–13 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: OpenEdge Management references the specific **PID** and its associated date and time start details to be sure of a process’ identity before it attempts to kill the 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 4–9 for details about this data.

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

OpenEdge Management displays a final status page that identifies the status of your kill request. OpenEdge Management 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 any number of unknown reasons.
 - **Process number xxxxx has been reused** — OpenEdge Management 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 without terminating the process.

Accessing and reviewing AppServer-related log file data

OpenEdge Management supports log file monitors and associated 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 OpenEdge Management log file monitor features and functionality, see the *OpenEdge Management: 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 OpenEdge Management discovers, OpenEdge Management supports monitoring its two associated log file monitors. OpenEdge Management 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

OpenEdge Management prepends the broker's name to the name of the broker and server log file monitors and viewers. For example, OpenEdge Management 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 AppServer Broker Log File Contents**.
- **AppServer-related log file names** — Displays **vesta.asbroker1ServerLogFileMonitor** and **vesta.asbroker1 AppServer 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:

- 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. OpenEdge Management predefines search criteria to support the broker and server 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 the following 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 4–21 for details.
- At the OpenEdge Management 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 6–21 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, OpenEdge Management 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 4–21 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 *[OpenEdge Management: Resource Monitoring Guide](#)*.

File Resource Defaults page

OpenEdge Management 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 OpenEdge Management-supplied default value set for the **Polling Interval** field at any time by clicking **Restore Defaults**.

Reviewing predefined log file monitor search criteria

Each log file provides predefined search criteria that address common AppServer broker- or AppServer-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
Broker_Error_Msg_Received	Catches the following error: Admin error response received: (adminErrorMsg).
Cannot_Start_Server	Catches the following error: ERROR: cannot start server.
Connection_Refused	Catches the following error: ERROR: connection refused: maximum number of client connections has been reached.
Disconnecting_Client	Catches the following error: Error (errMsg) ... disconnecting client.
Error_Loading_Prop_File	Catches the following error: Error loading properties file (propertiesFileName) : (ProException_string) : (ProExceptionMsg_string).
Fatal_Error	Catches the following error: FATAL ERROR : (rspoode) (errMsg).
Invalid_Action_For_State	Catches the following error: Client FSM Error : Invalid ACTION for STATE : state= (state) (descState) event= (event) (descEvent) action= (action) nextState= (state) (descState).
Invalid_Admin_Response	Catches the following error: Invalid admin response received: (adminErrorMsg).
Invalid_State	Catches the following error: FSM Error : Invalid State : state= (currentState).
IOException_Client_Response	Catches the following error: IOException while sending clientResp : (IOException_string).
IOException_Message_From_Server	Catches the following error: IOException reading message from server : (IOException_string)
MsgFormatException	Catches the following error: MsgFormatException received on client connection : (MsgFormatException_string) (MsgFormatExceptionMsg_string).
NameServer_Consistency_Error	Catches the following error: Internal Consistency Error on NameServer (host):(portNum) : (InternalConsistencyException).
NameServer_IOException	Catches the following error: IOException on startup of NameServer (hostName):(portNum) : (IOException_string).
NameServer_keepAlive_stopped	Catches the following error: NameServer keepAlive to NameServer (host):(portNum) : stopped.
NameServer_NOT_started	Catches the following error: NameServer NOT started for services (appIServices).
No_Servers_Available	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 4–21 for details.

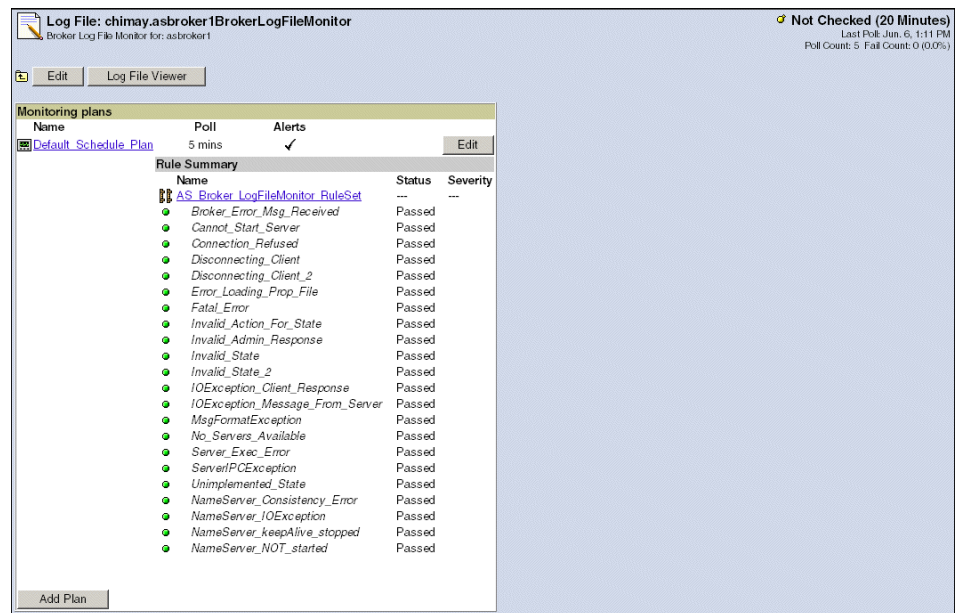
Customizing an AppServer Broker log file monitor

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 an AppServer broker 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 2–9.
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:



3. Customize or view the contents of an AppServer broker log file monitor as shown in the following table:

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

Note: OpenEdge Management 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.
5. Click **Add Rule** under the **Rules selected for this plan** section of the broker monitoring plan page. Proceed as indicated in the following table:

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

6. Select the appropriate values from the **Severity** and **On Alert Action Perform** fields to complete the alert severity and action definition that you want to associate with this rule.
7. To add another individual rule, repeat [Step 5](#) and [Step 6](#).

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 pick 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 AppServer broker's monitoring plan page showing the rules section updated with the new rules.

For more information about editing search criteria for rules, see the appropriate sections of the *OpenEdge Management: 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 the following 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 AppServer's 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.

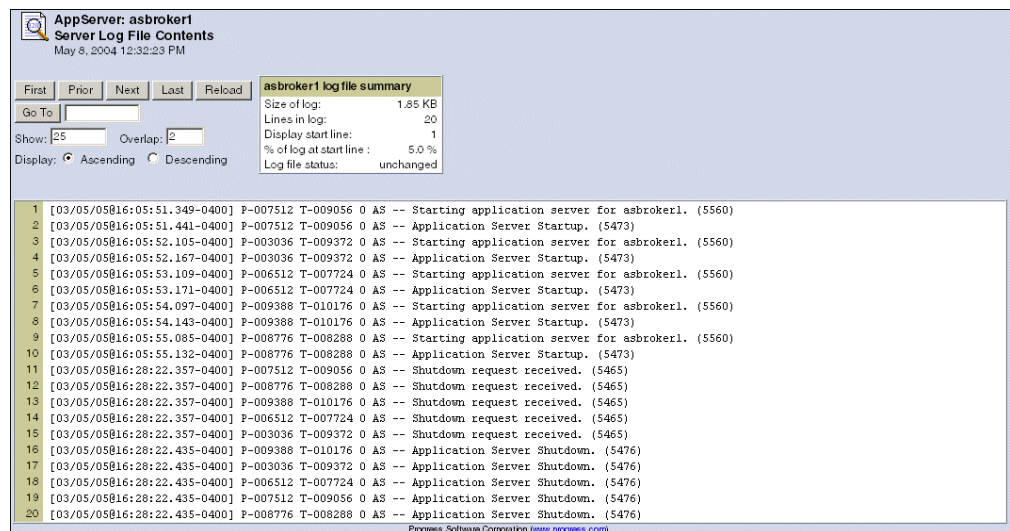


Figure 4–6: AppServer Broker log file viewer example

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 must not 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 OpenEdge Management 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 the 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. From the status bar, select the **Refresh** page icon for either the list or detail frame to repaint an existing page. You can also set a default value that OpenEdge Management uses to automatically refresh the management console.



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

Refresh data to avoid the following situations:

- OpenEdge Management considers a viewer that has been inactive for more than four hours “stale.” Once a viewer becomes stale, OpenEdge Management releases ninety-five percent of any memory it holds. If you try to use a stale viewer, OpenEdge Management 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.
- OpenEdge Management considers a viewer that has been inactive for forty-eight hours “dead.” Once a viewer dies, OpenEdge Management releases all of its memory. To return to the log file displayed in a dead view, you must renavigate to it, regardless of whether 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.
- A 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 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 4–8 for details.

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

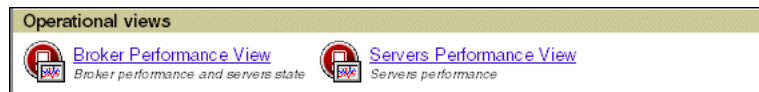


Figure 4–7: Operational views section example

The following information outlines how to access and review details associated with each of these views.

Accessing and reviewing the Broker Performance View

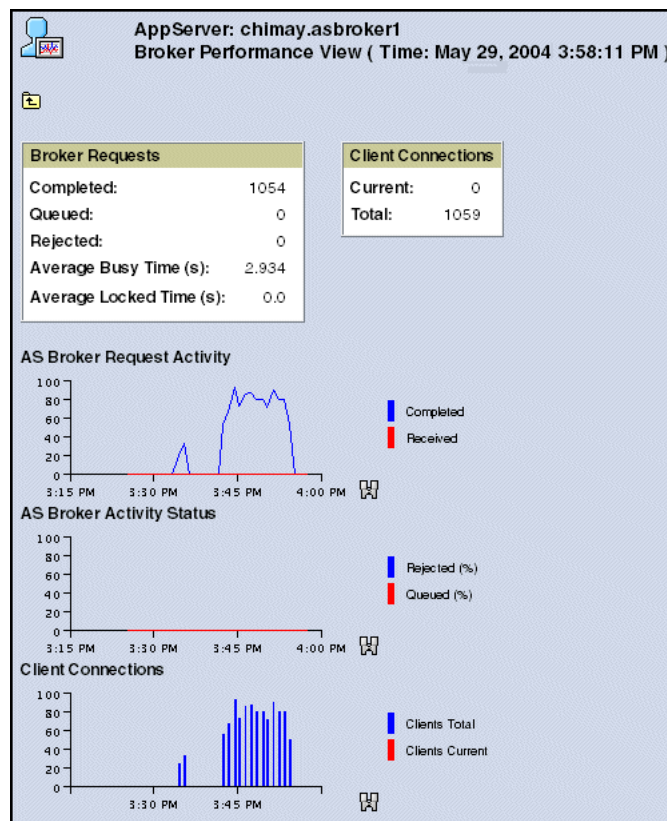
The AppServer **Operational views** section allows you to display information about the AppServer broker's performance and the state of the broker's associated servers. Review this data frequently, as it will help you make informed decisions about your use of the broker and server pool controls.



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 2–9 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 consists of two sections: **Broker Requests** and **Client Connections**. Data in both of these text boxes is determined when the page is initialized or refreshed.

The **Broker Requests** section provides 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 . . .
Completed	The number of broker requests fulfilled.
Queued	The number of broker requests to be processed.
Rejected	The number of broker requests that could not be processed.
Average Busy Time (s)	The average amount of time that the broker is busy servicing requests (expressed in milliseconds).
Average Locked Time (s)	The average amount of time that the broker is locked (expressed in milliseconds).

The **Client Connections** section identifies the number of client connections that the broker is currently handling, and 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 Activity**, **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 4-8 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 4-8](#) identifies and briefly describes each of these graphs.

Table 4-8: AppServer broker performance-related graphs (1 of 2)

This AppServer broker graph . . .	Displays two lines of . . .
AS Broker Request Activity	Broker-related performance data over a specified time period. 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.

Table 4–8: AppServer broker performance-related graphs

(2 of 2)

This AppServer broker graph . . .	Displays two lines of . . .
AS Broker Activity Status	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 OpenEdge Management 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.
Client Connections	<p>Client connections related to this broker over a specified time period. For example, data displayed might be related to the last polling activity. The blue line identifies the total number of client connections requested. The red line identifies the number of clients currently connected to this broker.</p> <p>Note: It is possible for this graph to accurately show that the number of current connections is higher than the total number of connections. The Clients Total reflects only new connections over the specified time period. In contrast, the Clients Current reflects all current connections in place when the graph is displayed, both newly connected and those that might still be connected from a previous polling period.</p>

See the “[Changing OpenEdge pinup graphical displays](#)” section on page 2–19 for details about changing the data appearance of graphs.

Accessing and reviewing the Server Performance View

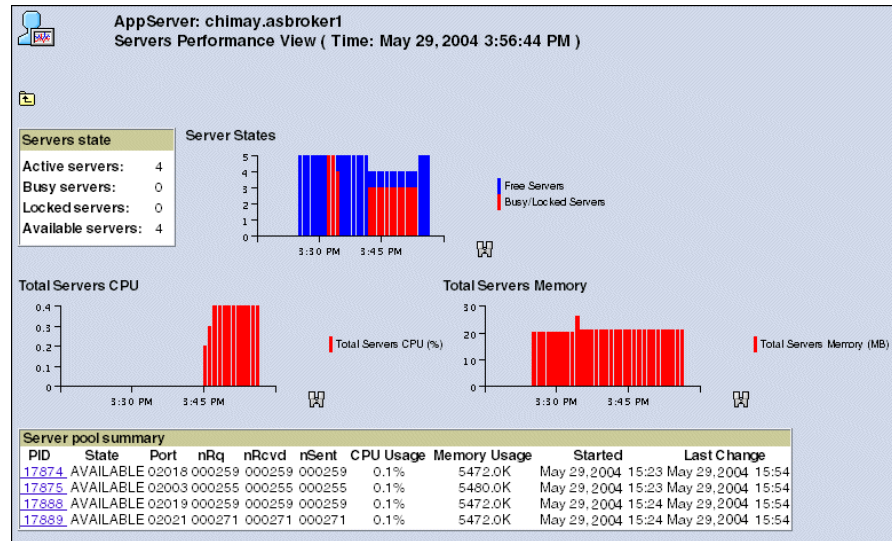
The AppServer **Operational views** section allows you to display information about servers’ status.



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 2–9 for the detailed procedure.

- 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 4–15 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 4–8 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 . . .
Server States	Two lines of server-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.
Total Servers CPU	One line of server-related performance data over a specified time period. This single data line indicates the total percent of the servers' CPU usage.
Total Servers Memory	One line of server-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 2–19 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.

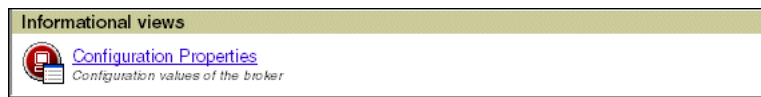


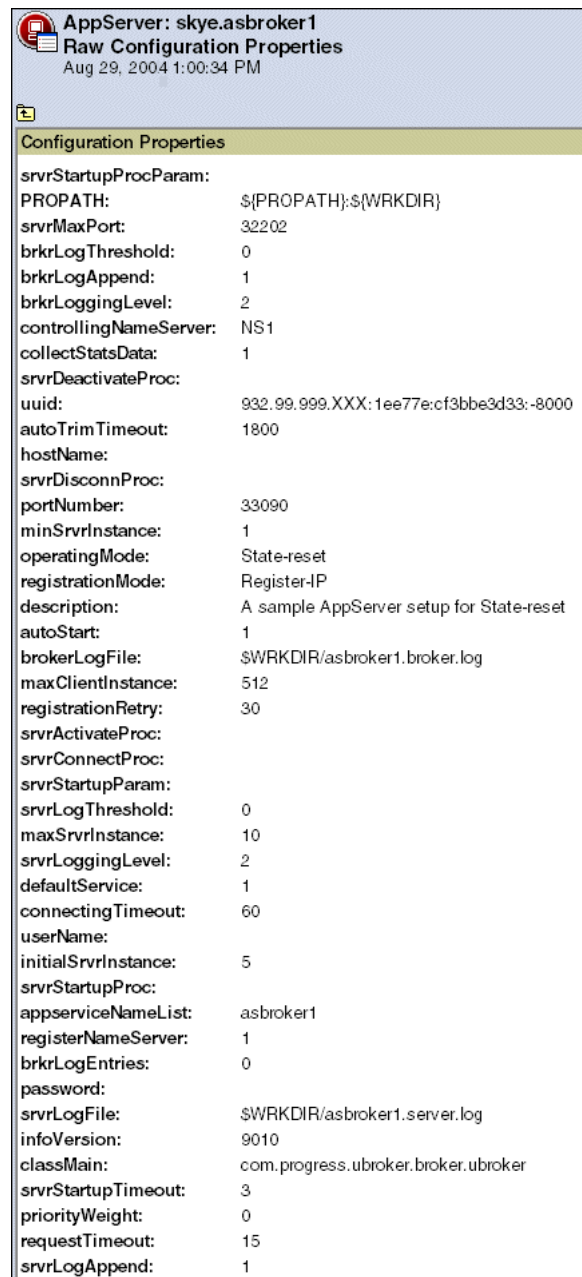
Figure 4–8: Informational views section example



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 2–9 for the detailed procedure.

2. Click **Configuration Properties** in the **Informational views** section to display the **AppServer Raw Configuration Properties** page, as shown:



AppServer: skye.asbroker1
Raw Configuration Properties
 Aug 29, 2004 1:00:34 PM

Configuration Properties

svrStartupProcParam:	
PROPATH:	\$(PROPATH);\$(WRKDIR)
svrMaxPort:	32202
brkrLogThreshold:	0
brkrLogAppend:	1
brkrLoggingLevel:	2
controllingNameServer:	NS1
collectStatsData:	1
svrDeactivateProc:	
uuid:	932.99.999.XXX:1ee77e:cf3bbe3d33:-8000
autoTrimTimeout:	1800
hostName:	
svrDisconnProc:	
portNumber:	33090
minSvrInstance:	1
operatingMode:	State-reset
registrationMode:	Register-IP
description:	A sample AppServer setup for State-reset
autoStart:	1
brokerLogFile:	\$WRKDIR/asbroker1.broker.log
maxClientInstance:	512
registrationRetry:	30
svrActivateProc:	
svrConnectProc:	
svrStartupParam:	
svrLogThreshold:	0
maxSvrInstance:	10
svrLoggingLevel:	2
defaultService:	1
connectingTimeout:	60
userName:	
initialSvrInstance:	5
svrStartupProc:	
appserviceNameList:	asbroker1
registerNameServer:	1
brkrLogEntries:	0
password:	
svrLogFile:	\$WRKDIR/asbroker1.server.log
infoVersion:	9010
classMain:	com.progress.ubroker.broker.ubroker
svrStartupTimeout:	3
priorityWeight:	0
requestTimeout:	15
svrLogAppend:	1

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

Managing NameServer Data

This chapter presents OpenEdge Management features and functionality related to NameServers, as described in the following sections:

- [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

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

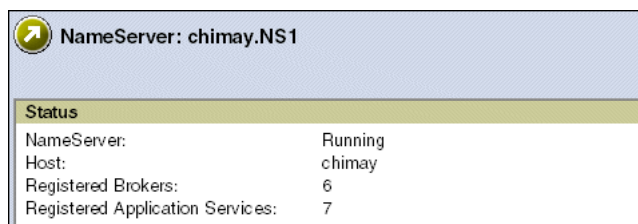
- Reviewing your current operating status and associated details.
- Reviewing property settings associated with a NameServer.
- Accessing and viewing data collected in a NameServer log file monitor.
- Working with NameServer resource-related details available through informational and operational views.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

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

You must have the appropriate OpenEdge Management role authorization to perform several of these tasks. For more information, see the [“Role authorization and OpenEdge tasks”](#) section on page 1–6.

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 title area of the page.



NameServer: chimay.NS1	
Status	
NameServer:	Running
Host:	chimay
Registered Brokers:	6
Registered Application Services:	7

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 . . .
NameServer	The running status of the NameServer. Possible values are: <ul style="list-style-type: none"> Running Not running
Host	The host machine's name.
Registered Brokers	The number of brokers currently registered with the NameServer.
Registered Application Services	The 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.

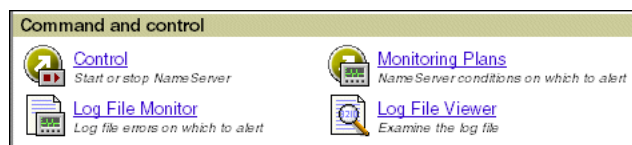


Figure 5–2: Command and control section example

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 . . .
Log file monitors and viewers.	The “ Accessing and reviewing NameServer-related log file data ” section on page 5–6.
Monitoring plans and rules.	Chapter 6, “Monitoring Plans and Rules for OpenEdge Server Resources.”
Rule sets.	The “ Working with rule sets ” section on page 6–21.

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 describes 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 **Status**. The NameServer’s name and port number are defined in the `ubroker.properties` file; the **Status** field reflects 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 OpenEdge Management 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, OpenEdge Management uses its 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 2–9 for the detailed steps.
2. Click **Control** in the **Command and control** section to display the **NameServer Control** page:



The following table highlights the controls that you can now change:

To . . .	Then . . .
Change the current setting of the Enabled property. (A checkmark displays to indicate that the Enabled property is set. To clear this option, click the checkmark.)	Click Edit . Then select or clear the Enabled property.
Toggle between stopping and starting the NameServer, depending on the current value displayed in the Status field. When the status displays Running , the button label will read Stop NameServer . Click this button to stop the NameServer. OpenEdge Management stops the NameServer and updates the value in the Status field to read Not Running .	Click Stop or Start .
Exit this page without changing any values. The NameServer main page redisplay in the management console.	Click Cancel .

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 OpenEdge Management log file monitor features and functionality, see the [OpenEdge Management: Resource Monitoring Guide](#).

Getting started with NameServer log files

OpenEdge Management supports monitoring the associated log file monitor for each local NameServer instance it discovers. OpenEdge Management 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 created is enabled. When the NameServer log file monitor first begins monitoring, it 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 helps you to:

- Ensure the integrity of NameServer log files by monitoring the 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 OpenEdge Management Component Library level under the NameServer subcategory. The search text and type are shareable at this level.

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, OpenEdge Management creates a NameServer log file monitor, using several default values, for that NameServer resource. Of the default NameServer log file monitor properties, you can modify only the 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.

The default values are:

- 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 the **On First Poll** property as they relate to log file monitors in general, see the appropriate section in the [OpenEdge Management: 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 OpenEdge Management-supplied default value, click **Restore Defaults**.

Reviewing predefined log file monitor search criteria

The NameServer log file monitor provides predefined search criteria that address common NameServer events. Use the search criteria as defined, or copy and customize it. 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 as shown:

The screenshot shows a web-based management console with a menu bar at the top containing 'Resources', 'Library', 'Reports', 'Jobs', 'Options', and 'Help'. Below the menu bar, a breadcrumb trail reads 'chlmay . Library . Search Criteria . NameServer'. The main content area displays a table of predefined search criteria for the NameServer.

Name	Description
Cannot_Return_Message	Catches the following error: Cannot return message to Client application at host (host).
Erroneous_UUID	Catches the following error: Erroneous UUID (UUID) received from Broker (name) (host) (port). UUID already registered by Broker (name) (host) (port).
Exceptions_Causing_NS_Shutdown	Catches the following error: Multiple I/O exception errors on port (port), the NameServer is shutting down.
Invalid_Message_Code	Catches the following error: Request received from (host) (port) contains invalid message code (code).
Listening_Error	Catches the following error: An error occurred while listening for network input requests on port (port).
Message_Error	Catches the following error: An error occurred while (marshalling/unmarshalling/sending) message to (client/broker/neighbor).
Socket_IO_Exception	Catches the following error: An exception occurred. Error message: (message).
Unhandled_Exception	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.

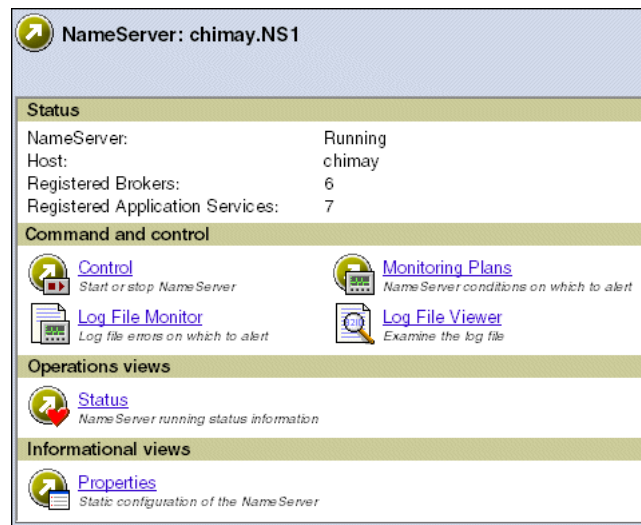
Customizing a NameServer log file monitor

You can make some custom changes to a NameServer log file monitor.

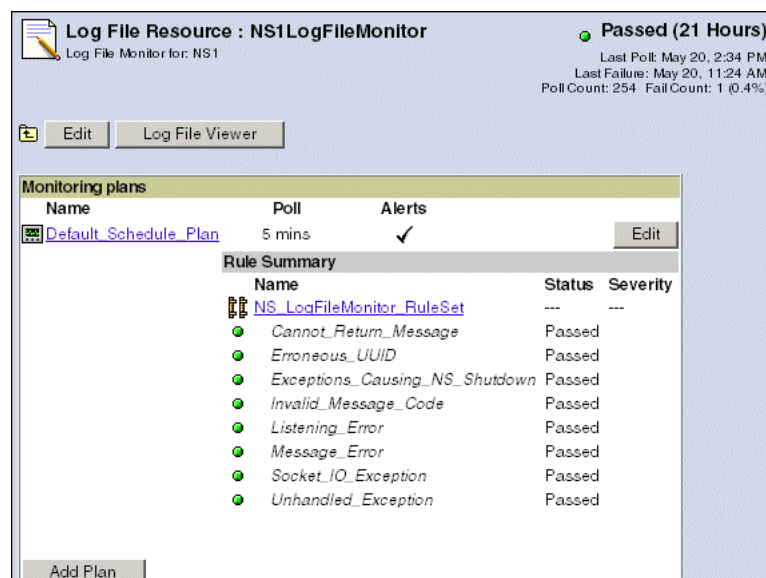


To customize a NameServer log file monitor:

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 shown in the following table:

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

Note: OpenEdge Management 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 11](#) in this procedure.
8. Click **Add Rule** under the **Rules selected for this plan** section of the NameServer monitoring plan page. Proceed as shown in the following table:

To ...	Then ...
Use a NameServer rule already defined in the library.	Select the: <ul style="list-style-type: none"> NameServer from the drop-down list associated with the Choose Criteria Category. The appropriate value from each drop-down list associated with Choose Search Criteria.

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

9. Select the appropriate values from the **Severity** and **On Alert Action Perform** fields to complete the alert severity and action definition that you want to associate with this rule.
10. To add another individual rule, repeat [Step 8](#) and [Step 9](#).
11. 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.
12. Click the detail page for the parent icon (the file folder with the up arrow on it) to redisplay the 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 *OpenEdge Management: 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:

- The **Log File Viewer** link in the **Command and control** section of the **NameServer main** page.
- 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.

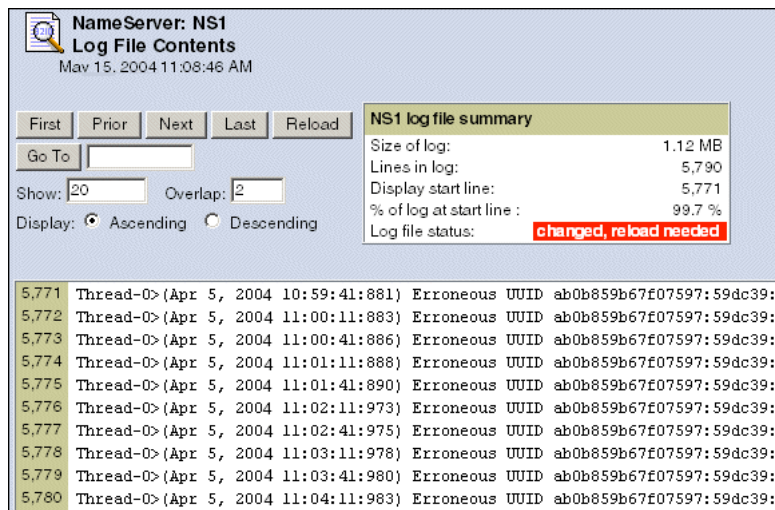


Figure 5–4: NameServer log file viewer example

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 must not 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 OpenEdge Management 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 in 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 OpenEdge Management uses to automatically refresh the management console.



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

Refresh data to avoid the following situations:

- OpenEdge Management considers a viewer that has been inactive for more than four hours “stale.” Once a viewer becomes stale, OpenEdge Management releases ninety-five percent of the memory it holds. If you try to use a stale viewer, OpenEdge Management 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.
- OpenEdge Management considers a viewer that has been inactive for forty-eight hours “dead.” Once a viewer dies, OpenEdge Management releases all of its memory. To return to the log file displayed in a dead view, you must renavigate to it, regardless of whether 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


Accessing and reviewing Operational views

The NameServer **Operational views** display 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 2–9 for the detailed procedure.
2. Click **Status** in the **Operational Views** section to display the **Operational Status** page:



NameServer: nightcap.NS1

Operational Status

May 29, 2004 9:33:04 AM

Summary

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

AppService:WS.wsbroker1

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

AppService:AS.asbroker1

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 an AppService section for each Application Service registered with the NameServer.

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 individual **AppService** sections. 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](#) and the **Requests Received** and **Requests Directed** fields in [Table 5-4](#) for details.

[Table 5-3](#) briefly describes each of the fields displayed in 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](#) describes 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 . . .
Host	The NameServer's host machine name.
Port	The number of the UDP Port that the NameServer uses to listen for client connection requests and registration messages from AppServers and Transaction Servers.
Timeout	<p>The 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>
Start time	The 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.
Number of AppServices	The 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 AppService sections in this view.

Table 5–3: Summary details on the Operational Status page

(2 of 2)

This field . . .	Displays. . .
Number of Brokers	<p>The 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 AppServices detail section. However, the broker will only count as 1 towards the total number of brokers recorded in this field.</p>
Total client requests received	<p>The total number of client requests received by the NameServer since it 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.</p>
Total client requests rejected	<p>The 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 (those 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 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** details associated with that broker will no longer display. If the broker reregisters, 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 . . .
Requests Received	The 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.
Broker	<p>The name of the broker that is capable of fulfilling the connection between the requesting client and the Application Service.</p> <p>When more than one broker is servicing the same Application Service, each new broker's data is appended to the individual AppService details section.</p>
Host	The broker's host machine name and numeric address.
Weight	The priority weight assigned to the Unified Broker instance for the purpose of load balancing.
Requests Directed	<p>The total number of client connection requests for the Application Service as directed by the NameServer to the broker instance.</p> <p>This count is maintained while the broker remains registered. If the broker is stopped or times out, 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.</p>
UUID	The unique number for the Unified Broker instance.
Port	The TCP/IP port number that the broker listens on to pick up client connection requests.
Timeout	The 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 2–9 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 . . .
Location	<p>The specific values that pertain to these fields:</p> <ul style="list-style-type: none"> • NameServer location — 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. • Host name — Identifies the name of the host machine. • Port number — 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.
General	<p>The specific values that pertain to these fields:</p> <ul style="list-style-type: none"> • Working directory — Identifies the NameServer working directory, including the pathname. • Broker keep alive timeout — 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. • Autostart — Indicates whether the NameServer will start automatically when the controlling AdminServer starts. If the value 1 displays, the Autostart option is set. If the value zero displays, then the Autostart option is not set.
Logging	<p>The specific values that pertain to these fields:</p> <ul style="list-style-type: none"> • Server log filename — Identifies the NameServer log filename, including the pathname. • Logging level — Displays one of three possible values to specify the amount of information to be written to the server log: Error only, Terse, or Verbose. • Append to log file — 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.

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

This field . . .	Displays . . .
Advanced	The specific value that pertains to this optional field: <ul style="list-style-type: none">• Neighboring NameServers — 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).
Environment	The specific NameServer environment variables that are defined for the process in which the NameServer executes.

Monitoring Plans and Rules for OpenEdge Server Resources

You use OpenEdge Management's monitoring capabilities to monitor OpenEdge server resources as you do other resource types. This chapter highlights OpenEdge-specific resource monitoring details, tasks, and related activities, as described in the following sections:

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

For additional details about OpenEdge Management resource monitoring and resource monitoring plans, see the *OpenEdge Management: Resource Monitoring Guide*. For complete details about alerts, see the *OpenEdge Management: Alerts Guide and Reference*.

OpenEdge Management resource monitoring overview

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

For example, if OpenEdge Management 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**. (You can edit or modify any OpenEdge monitoring plan and rules, setting your own values at any time.)

Other recognizable resource types—database, system, network, and file resources—also require monitoring plans and rules. All OpenEdge Management resources share standardized ways to perform monitoring operations and a common terminology with which to reference the resource activities.

Review the resource monitoring details provided in this section. This information will help orient you to the basics of resource monitoring. Then, follow the procedures outlined in the [“Maintaining monitoring plans”](#) section on page 6–9 and the [“Understanding and using resource monitor rules”](#) section on page 6–14 to use resource monitoring with 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 [OpenEdge Management: Resource Monitoring Guide](#).

OpenEdge Management resource monitoring terms include:

- **Resource** — A specific component of your configuration such as a WebSpeed broker, AppServer broker, or NameServer instance. Other OpenEdge Management 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 OpenEdge Management (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 OpenEdge Management checks to verify whether or not a resource complies with its performance criteria. Rule values, or settings, can be established by using either default or user-supplied 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. OpenEdge Management 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 OpenEdge Management 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. There are also 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 6–4 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. OpenEdge Management supports using, modifying, and copying predefined schedules to help you define them 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 they can indicate 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 when 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 OpenEdge Management must have one or more monitoring plans before OpenEdge Management 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 you specified.
- **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."](#)

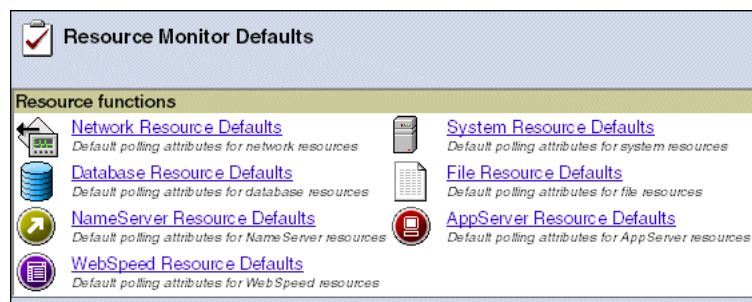
Default polling and trend values

During the OpenEdge server discovery process, OpenEdge Management 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.

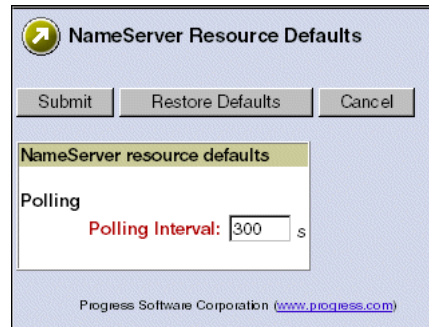


To display OpenEdge resource-specific default values:

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

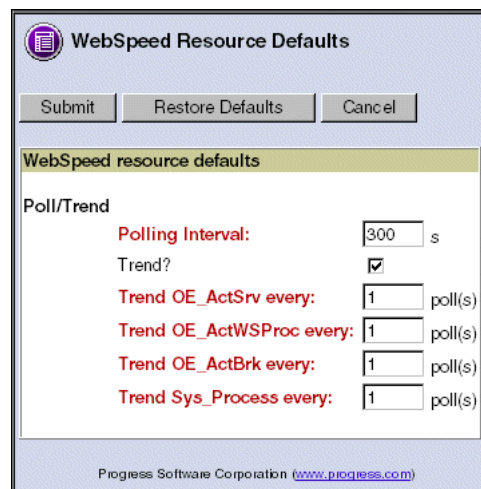


2. Click the link associated with the specific resource default values you want to review: **NameServer Resource Defaults**, **WebSpeed Resource Defaults**, or **AppServer Resource Defaults**. The associated **OpenEdge Resource Defaults** page appears, as shown in the following examples:
 - a. If you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



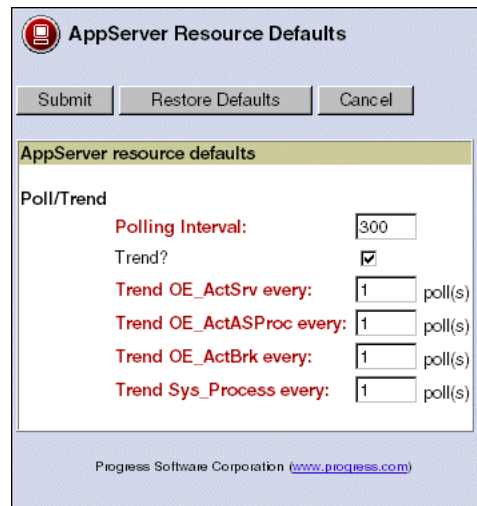
The screenshot shows the 'NameServer Resource Defaults' window. At the top, there is a title bar with a green arrow icon and the text 'NameServer Resource Defaults'. Below the title bar are three buttons: 'Submit', 'Restore Defaults', and 'Cancel'. The main content area has a header 'NameServer resource defaults' and a section titled 'Polling'. Under 'Polling', there is a label 'Polling Interval:' followed by a text input field containing '300' and a unit 's'. At the bottom of the window, it says 'Progress Software Corporation (www.progress.com)'.

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



The screenshot shows the 'WebSpeed Resource Defaults' window. At the top, there is a title bar with a purple icon and the text 'WebSpeed Resource Defaults'. Below the title bar are three buttons: 'Submit', 'Restore Defaults', and 'Cancel'. The main content area has a header 'WebSpeed resource defaults' and a section titled 'Poll/Trend'. Under 'Poll/Trend', there are several settings: 'Polling Interval:' with a text input field containing '300' and a unit 's'; 'Trend?' with a checked checkbox; 'Trend OE_ActSrv every:' with a text input field containing '1' and a unit 'poll(s)'; 'Trend OE_ActWSPProc every:' with a text input field containing '1' and a unit 'poll(s)'; 'Trend OE_ActBrk every:' with a text input field containing '1' and a unit 'poll(s)'; and 'Trend Sys_Process every:' with a text input field containing '1' and a unit 'poll(s)'. At the bottom of the window, it says 'Progress Software Corporation (www.progress.com)'.

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



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

- Polling Interval:** A text box containing the value "300".
- Trend?** A checkbox that is checked.
- Trend OE_ActSrv every:** A text box containing "1" followed by "poll(s)".
- Trend OE_ActASProc every:** A text box containing "1" followed by "poll(s)".
- Trend OE_ActBrk every:** A text box containing "1" followed by "poll(s)".
- Trend Sys_Process every:** A text box containing "1" followed by "poll(s)".

At the bottom of the page, there is a footer that reads "Progress Software Corporation (www.progress.com)".

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 OpenEdge Management-supplied default values at any time by clicking **Restore 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 OpenEdge Management Trend Database 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 3–8 for details about how to implement data collection with WebSpeed brokers. See the [“Data collection details”](#) section on page 4–8 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 so you can begin using many of the features of OpenEdge Management resource monitoring immediately.

This section:

- Identifies each resource monitoring plan's fields and the 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

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 OpenEdge Management 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 OpenEdge Management resource monitor.
Alerts	✓	Indicates whether alerts are active and will be generated when the plan is active.
Trend	✓	Indicates whether the statistical data monitored while the plan is active will be stored to the OpenEdge Management Trend Database. The Trend Performance Data field value is involved in setting data collection.
Rule Summary	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"> • AppServer brokers — AppServer Abnormal Shutdown. • WebSpeed brokers — WebSpeed Abnormal Shutdown. • NameServers — Abnormal Shutdown, Broker Timeout, and Duplicate Broker UUID.

Default Schedule details

OpenEdge Management provides one default monitoring plan per OpenEdge resource. This plan is called the **Default Schedule Plan**. However, when you update a monitoring plan, you can add different plans to monitor different resource activities.

Note: OpenEdge Management 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 other plans.

Each OpenEdge resource that OpenEdge Management 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 include a Rule Summary. The *Rule Summary* is a list of rules and rule sets that are applied to the particular monitoring plan. OpenEdge Management automatically applies the default rule set associated with a specific OpenEdge resource to a plan. Table 6–1 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 pages. 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 is an example of a NameServer default monitoring plan and associated default rules example. It shows the default plan and rule set for a NameServer named NS1.

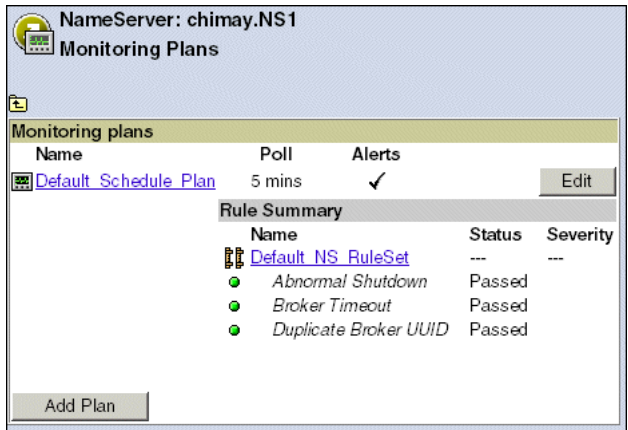


Figure 6–1: NameServer instance default monitoring example

AppServer default monitoring plan example

Figure 6–2 shows an example of a default monitoring plan and associated default rules, specifically for an AppServer broker. With the exception of the unique rule sets, the details displayed are identical for both the AppServer and WebSpeed broker resources.



Figure 6–2: AppServer broker default monitoring plan example

WebSpeed default monitoring plan example

Figure 6–3 shows an example of a default monitoring plan and associated default rules, specifically for a WebSpeed broker. With the exception of the unique rule sets, the details displayed are identical for both the AppServer and WebSpeed broker resources.

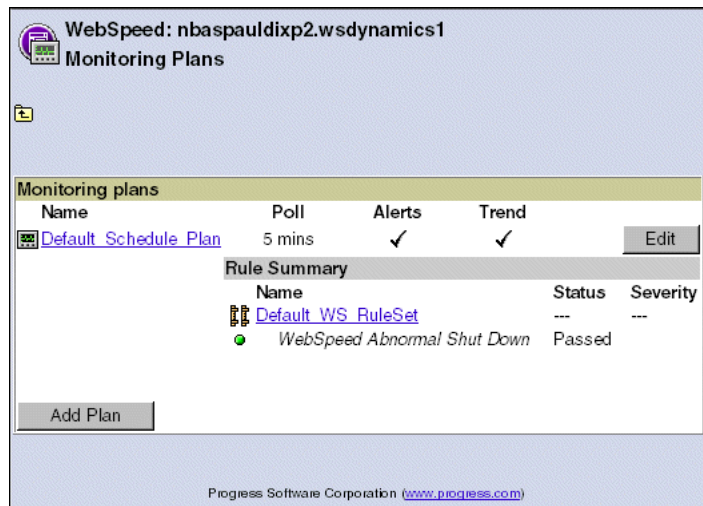


Figure 6–3: WebSpeed broker default monitoring plan example

Maintaining monitoring plans

The basic tasks you use to create resource monitoring plans for any OpenEdge Management 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.

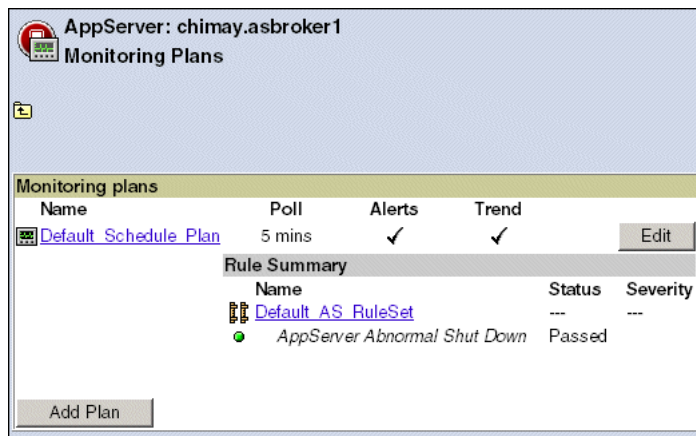
Updating monitoring plans

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 a WebSpeed broker and a NameServer instance; any unique requirements for a given OpenEdge resource type are noted.

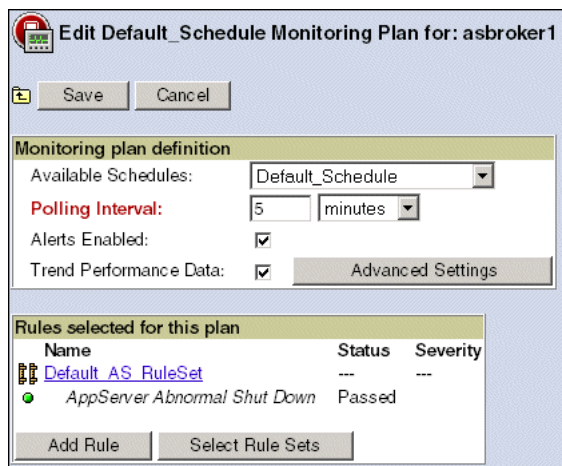


To update an OpenEdge resource monitoring plan:

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 2–9. In this example, the AppServer broker resource **asbroker1** is selected.
3. 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:



4. Select the specific schedule associated with the plan that you want to update. Click **Edit** associated with that plan. The following shows the page that appears in edit mode when the **Default Schedule Plan** is selected:



5. Update the monitoring plan values, as described here:
 - 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.)
 - c. Click **Advanced Settings**. (Available to WebSpeed and AppServer broker resources only.)
 - d. Select a specific rule or rule set to add, update, or remove from this plan. For details, go to [Step 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 OpenEdge Management Trend Database. For WebSpeed broker-related details, see the “[Properties section](#)” section on page 3–6. 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:

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
Default AS RuleSet	---	---
AppServer Abnormal Shut Down	Passed	

Add Rule Select Rule Sets

- 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** section of the **Default_Schedule Monitoring Plan** page. The **Available Rules** page for rules that are specific to the OpenEdge resource displays. In this example, the rules associated with an Appserver broker resource appear:



The **Available Rules** page contains a dynamic list that includes 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 6–14.

- Click the rule you want to add. For example, if you select **Process CPU High**, the detailed rule information shown in the following dialog box appears:

A screenshot of a configuration dialog box titled "Rule: Process CPU High". At the top, it says "Resource: asbroker1 Schedule: Default_Schedule". Below this are "Save" and "Cancel" buttons. The main area contains several settings: "Threshold:" with a text box containing "80" and "percent" to its right; "Alert severity:" with a dropdown menu showing "Error"; "Throw alert after:" with a text box containing "1" and "failed poll(s)" to its right; "Always throw new alert:" with an unchecked checkbox; "On alert perform action:" with a dropdown menu showing "Default_Action"; "Clear alert after:" with a text box containing "0" and "successful poll(s)" to its right; and "On clear perform action:" with a dropdown menu showing "None". At the bottom, there is a section titled "Rule description" with the text "A measure of how much CPU a single process is using." and the Progress Software Corporation logo and website URL "www.progress.com" at the very bottom.

Note the rule’s description on the bottom of the rule page.

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.

This rule 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 6–14. The remaining fields on this page are alert- and action-related fields. For details, see the *OpenEdge Management: 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 selected for this plan** section of the **Default_Schedule Monitoring Plan** page to choose the rule sets you want to add to the monitoring plan. OpenEdge Management displays the default rule set for the resource type you are updating, and any additional rule sets created using the **OpenEdge Management Component Library**. See the “[Working with rule sets](#)” section on page 6–21 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 *OpenEdge Management: Resource Monitoring Guide* for more resource status information.

Table 6–2: Resource status legend

This status . . .	Has a color indicator of . . .	To identify that . . .
Pass	Green	The resource monitor is currently working.
Fail	Red	<p>The most recent test involving the associated resource failed. For some resources, such as network, this includes statuses such as tardy, timeout, and unreachable.</p> <p>Check the Alert Summary page or the specific monitor for possible alert details. This status can also identify an internal error that prevents the resource from being monitored.</p>
Not Running	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.
Not Checked	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.
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 OpenEdge Management checks to verify whether a resource complies with an expected performance criterion. Certain rules specific to WebSpeed and AppServer 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 6–10, 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 type's **Available Rules list**.
- When you select any of the rules available in 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 **Add Rule** on the monitoring page when it is displayed in edit mode. See the procedure in the [“Maintaining monitoring plans”](#) section on page 6–9 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 OpenEdge Management 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 [OpenEdge Management: Alerts Guide and Reference](#) for detailed information about OpenEdge Management 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 NameServer 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 AppServer Rules** page as it displays the complete list of AppServer broker rules.

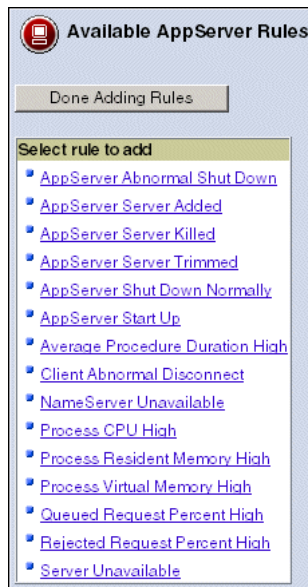


Figure 6–5: Available AppServer Rules page

WebSpeed rules

Figure 6–6 shows the **Available WebSpeed Rules** page as it displays the complete list of WebSpeed broker rules.

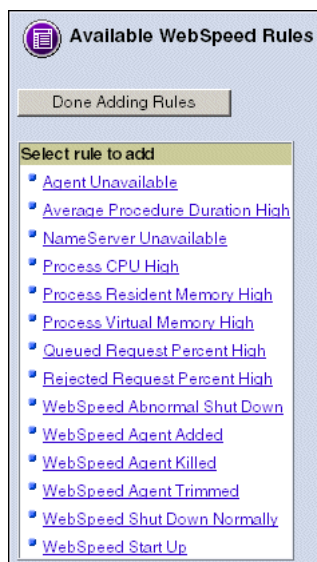


Figure 6–6: Available WebSpeed Rules page

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 an ABL 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** measures the execution time of the ABL procedure only 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 ABL procedures and WebSpeed Transaction Servers

For ABL 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 an ABL procedure, entering one procedure on one line. These procedures will generally be file types such as `.p`, `.w`, or `.html`. The following URL example shows the type of information required to measure a WebSpeed procedure:

`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 ABL procedures and AppServers

AppServer ABL procedures execute with the `RUN` statement based on an AppServer connection handle. The procedures 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 in the `RUN` statement.

Table 6–3 describes three examples.

Table 6–3: Examples of AppServer-related ABL procedure entries

This ABL 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 an 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. You can adjust your threshold setting to help minimize the impact these problems have on 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. The rule then determines the number of new rejected requests for the current poll period. This rule also 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. The number of requests rejected is then 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 as a percentage. The value identifies the number of rejected client requests during the polling interval that you will consider acceptable. Any rejected requests that exceed this value will cause the alert and action criteria that you set on this page to be triggered.

Queued Request Percent High rule

The WebSpeed and AppServer lists of available rules include **Queued Request Percent High**. This rule highlights the percentage of client requests queued during a polling interval set for either a WebSpeed agent or an 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. You can 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. The rule then determines the number of new queued requests for the current poll period. This rule also 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. The number of requests queued is then 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

Accessible from the **Available Rules** page is the **Queued Request Percent High** rule page. On this page, you specify the threshold value as a percentage. The value 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 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 OpenEdge Management rules, the **WS_Agent Unavailable** rule and the **WS_Server Unavailable** rule monitor the state of either an agent or a 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 initiates a customer order query in WebSpeed through a browser and 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 expects a quick display of a results set and 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. The user becomes impatient with the wait time and 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.

Unbeknownst to the user, each click of the **Submit** button 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 agents are unavailable to service new client requests. This can impact application performance and throughput. This performance degradation can easily be compounded by the drain 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 prevent performance problems from escalating.

AppServer server example

An AppServer server can 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 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

You associate a rule set with one or more resources through a monitoring plan. Rule sets are stored by resource type in the **OpenEdge Management Component Library**. The following links allow you to create OpenEdge-related rule sets:

- **Create AppServer Rule Set.**
- **Create NameServer Rule Set.**
- **Create WebSpeed Rule Set.**



To display the **OpenEdge Management Component Library** page where these links appear, click **Library**.

Note: The **Log File Rule Set** link on the **OpenEdge Management Component Library** page allows you to create rule sets that are 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 **OpenEdge Management Component Library**, making the rule set available for use and reuse 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. 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.

OpenEdge Management 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 OpenEdge Management, a default monitoring plan with a default rule set is assigned to it. OpenEdge Management also provides this default rule set assignment for WebSpeed and NameServer resources.

Benefits of using rule sets

Rule sets allow you to do the following:

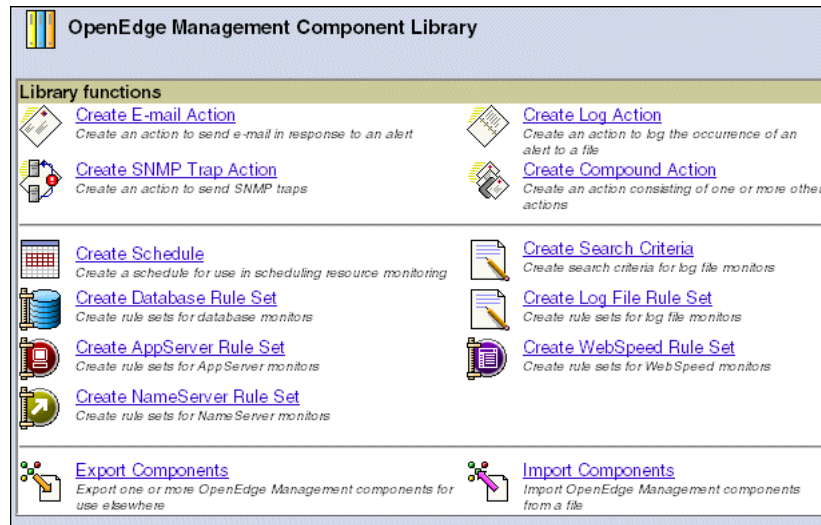
- Associate the rule set with a monitoring plan. The polled rules in the sets are evaluated when the monitoring plan is active and the resource is polled. (Asynchronous rules trigger immediately when these rules are violated.)
- Use an updated rule set. If you associate a rule set with a monitoring plan 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 a NameServer rule set as an example.



To create a NameServer rule set:

1. From the management console menu bar, click **Library**. The **OpenEdge Management 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 **OpenEdge Management Component Library** list frame under **Rule Sets** → **NameServer**. (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

Once you create a rule set, you can edit it later.



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 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 **OpenEdge Management Component Library** list frame. For example, click **Rule Sets** from the categories that display in the list frame. [Figure 6–7](#) shows the **Rule Sets** subcategories that display in the detail frame.

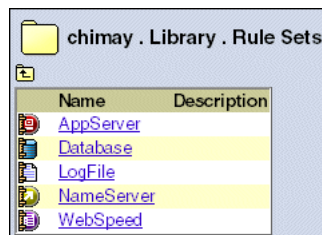


Figure 6–7: Accessing rule sets from the detail frame

Copying a rule set

You can copy a rule set and make whatever modifications you want. At a minimum, you must be sure to rename the copy.



To copy an AppServer rule set:

1. From the **AppServer Rule Set** page, click **Copy**. The **Copy AppServer Rule Set** page appears.
2. Rename the copy and (optionally) change the description.
3. Click **Save**.

From either the **Copy AppServer Rule Set** page or the **AppServer Rule Set** page, you can now add one or more rules to the copy.

Note that you can access the list of existing AppServer rule sets at any time from the **OpenEdge Management 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 **OpenEdge Management 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 illustrates this association for an AppServer rule set.



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 2–9 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, as outlined in the following sections:

- [Overview](#)
- [Setting rules-related criteria](#)
- [Understanding the recommended threshold settings](#)
- [Determining the effectiveness of your selections](#)

See the *OpenEdge Management: Database Management Guide* for Configuration Advisor details related to databases. See the *OpenEdge Management: Resource Monitoring Guide* for Configuration Advisor details related to CPU, disk, and file system resources.

Overview

The Configuration Advisor is an OpenEdge Management feature that helps you to determine optimum threshold settings for specific polled rules used. An alternative to using OpenEdge Management-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 OpenEdge Management Trend Database.

Note: You must have administrator privileges to use the Configuration Advisor.

The Configuration Advisor analyzes a rule's past performance for a specified period of time and, based on that 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, OpenEdge Management will use this setting the next time the rule is evaluated.

Recommendations are based on a representative sampling of data from the OpenEdge Management Trend Database. 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 OpenEdge Management requires to retrieve, evaluate, and generate baseline values, resources could be dedicated to this task for an unknown period of time. Allot a period of time to experiment with this feature to familiarize yourself with its benefits and processing requirements.

Rule details

The Configuration Advisor calculates recommended rule threshold settings for rules associated with a variety of OpenEdge Management resources. This section highlights the WebSpeed broker and AppServer broker rules. See the [OpenEdge Management: Database Management Guide](#) for details about the database rules. See the [OpenEdge Management: 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. You must maintain a one-to-one relationship between trending and polling data regardless of the time interval set for the polling. 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 3–8. For details about data collection and the AppServer broker, see the [“Data collection details”](#) section on page 4–8.

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 OpenEdge Management Trend Database 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 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. This determination is 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**.

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 OpenEdge Management Trend Database. Consider using the OpenEdge Management-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 in performing reads, writes, and updates to your system.

It is recommended that you use the OpenEdge Management-supplied Configuration Advisor-related default values for a set period of time (for example, one week) to capture data to the OpenEdge Management Trend Database for a rule. This initial step will provide you sufficient data 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 would select Monday through Friday from 8 AM to 6 PM 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 OpenEdge Management Trend Database. 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 not to be 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 OpenEdge Management Trend Database. 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. The range of values is adjusted as necessary, 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.

The screenshot shows the Configuration Advisor interface for user 'skye.asbroker1'. It includes buttons for 'Update Selected Rules' and 'Cancel'. A box displays the analysis time period: Start Date: 2004/7/14, End Date: 2004/7/21, Days: Sun Mon Tue Wed Thu Fri Sat, and Hours: From: 9:00 To: 17:00. Below this is a table of recommended values for three rules.

Rule	Recommend Values	Default	Schedule Select:	Update	Current Threshold	
Process CPU High	3.1 [0]* percent	<input checked="" type="checkbox"/>	All	<input checked="" type="checkbox"/>	80.0	Detail
Queued Request Percent High	4.0 [0]* Percent queued	<input checked="" type="checkbox"/>	None	<input checked="" type="checkbox"/>	20.0	Detail
Rejected Request Percent High	4.0 [0]* Percent rejected	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	5.0	Detail

Figure 7–2: Configuration Advisor recommended thresholds

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 Percent 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**, OpenEdge Management does not apply any of your selections.

If the data analysis calculation for a rule was unsuccessful, the Configuration Advisor cannot define a range. 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 for the Configuration Advisor to make a meaningful recommendation.

Generating and applying threshold rule settings

Table 7–1 highlights where to find additional information about using the Configuration Advisor.

Table 7–1: Configuration Advisor details

For information about . . .	See . . .
The procedure to set user-supplied criteria used in the data analysis process.	The “ Setting rules-related criteria ” section on page 7–7.
Understanding the Configuration Advisor’s recommended settings, including evaluating and applying these settings.	The “ Understanding the recommended threshold settings ” section on page 7–10.
Reviewing your selections.	The “ Determining the effectiveness of your selections ” section on page 7–14.

Setting rules-related criteria

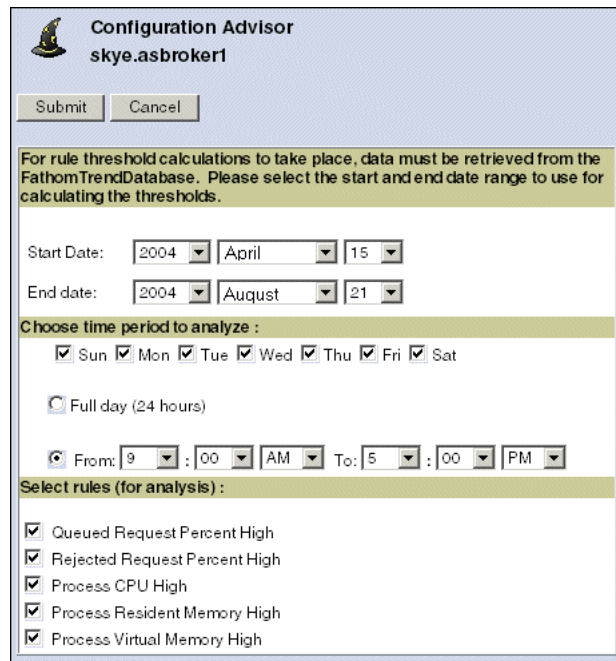
Once you have completed your specific rule analysis, you have the necessary information to use the Configuration Advisor. The following procedure shows how to use the Configuration Advisor to calculate AppServer threshold values. Use these same steps to calculate WebSpeed threshold values, substituting 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 2–9 for the detailed steps.

- 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

- In the **Start Date** and **End date** fields, define a date range that OpenEdge Management will use to collect data from the OpenEdge Management Trend Database. (The default date range 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 you specify. This requirement ensures that data was trended to the OpenEdge Management Trend Database 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 selected.
 - All rules associated with a given OpenEdge resource are individually evaluated against the rule-specific data retrieved from the OpenEdge Management Trend Database for a period of time you define.
- In the **Choose time period to analyze** section, identify the time frame that defines a representative period of time for which the rules are generally active, or being used. This time frame is the period against which OpenEdge Management calculates the baseline activity. (The default time period, as shown in [Step 2](#), is Sunday through Saturday, 9 AM to 5 PM.)

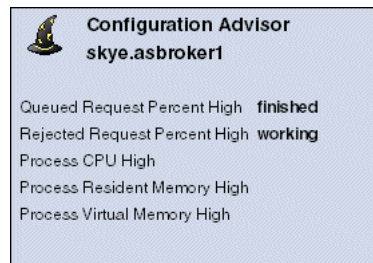
5. In the **Select rules (for analysis)** section, click the polled rules that you want the Configuration Advisor to use to calculate threshold settings.

Only those polled rules that are currently defined in existing monitoring plans for a broker resource can be candidates for processing by the Configuration Advisor. 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 sample shown in [Step 2](#), the Configuration Advisor determined that there are five 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 OpenEdge Management Trend Database 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:



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 a later time 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 7–10 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.

The screenshot shows the Configuration Advisor window for user 'skye.asbroker1'. It includes buttons for 'Update Selected Rules' and 'Cancel'. A box displays the analysis time period: Start Date: 2004/4/15, End Date: 2004/8/21, Days: Sun Mon Tue Wed Thu Fri Sat, and Hours: From: 9:00 To: 17:00. Below this is a table of recommended values for five rules, with columns for Rule, Recommend Values, Weekdays, and Weekends.

Rule	Recommend Values	Weekdays Select: All None	Update	Current Threshold	Weekends Select: All None	Update	Current Threshold	
Process CPU High	3.1 [0]* percent	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	80.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	80.0	Detail
Process Resident Memory High	3256.6 [0]* kilobytes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	10000.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	10000.0	Detail
Process Virtual Memory High	6650.5 [0]* kilobytes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	100.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	100.0	Detail
Queued Request Percent High	4.0 [0]* Percent queued	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	20.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	20.0	Detail
Rejected Request Percent High	4.0 [0]* Percent rejected	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5.0	Detail

Figure 7–3: Configuration Advisor recommended values

The Configuration Advisor page has multiple purposes. Table 7–2 identifies the key tasks you can perform from this page and the associated procedures.

Table 7–2: Tasks using the Configuration Advisor Calculations page

To ...	See ...
Review and evaluate the recommended threshold settings calculated for each rule processed.	The “ Evaluating recommended settings ” section on page 7–11.
Display and review the specific details about each individual rule’s analysis.	The “ Evaluating recommended settings ” section on page 7–11.
Compare current threshold settings defined for each of the rules processed with the recommended threshold calculations.	The “ Comparing and selecting threshold settings ” section on page 7–13.
Update the threshold values for the rules and the specific schedules that you have selected.	The “ Submitting your threshold setting selections ” section on page 7–14.

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 which contains 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 are not met to perform the data analysis successfully. See the [“Setting rules-related criteria”](#) section on page 7–7 for details.

Reviewing recommended values

The Configuration Advisor displays a range of possible values from which to select. [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.

The screenshot shows the Configuration Advisor window for 'skye.asbroker1'. It includes a time period analysis box and a table of rules with their recommended values and current thresholds.

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	Unit	Weekdays Select:	Update	Current Threshold	Weekends Select:	Update	Current Threshold	Action
Process CPU High	3.1 [0]*	percent	All	<input checked="" type="checkbox"/>	80.0	All	<input checked="" type="checkbox"/>	80.0	Detail
Process Resident Memory High	0.1 [32]	kilobytes	None	<input checked="" type="checkbox"/>	10000.0	None	<input checked="" type="checkbox"/>	100.0	Detail
Process Virtual Memory High	1.1 [0]	kilobytes	All	<input checked="" type="checkbox"/>	100.0	All	<input checked="" type="checkbox"/>	100.0	Detail
Queued Request Percent High	2.1 [0]	Percent queued	None	<input checked="" type="checkbox"/>	20.0	None	<input checked="" type="checkbox"/>	20.0	Detail
Rejected Request Percent High	3.1 [0]*	Percent rejected	All	<input checked="" type="checkbox"/>	5.0	All	<input checked="" type="checkbox"/>	5.0	Detail

Figure 7–4: Recommended Values field content

Each recommended value is expressed as a set of two numbers. The first number (in each row) specifies the recommended threshold setting. The second number, displayed in brackets, identifies the number of times the threshold value set at the associated setting would be exceeded and an alert 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.

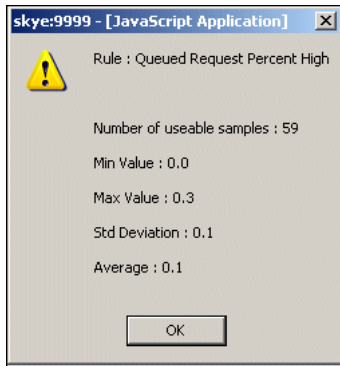


Figure 7–5: Detail page analysis content

Table 7–3 describes the contents of the **Detail** page.

Table 7–3: Detail page fields and descriptions

This field . . .	Describes . . .
Number of useable samples	The number of data samples extracted from the OpenEdge Management Trend Database.
Min Value	The minimum value derived from the data set.
Max Value	The maximum value derived from the data set.
Std Deviation	The root mean squared deviation.
Average	The 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 the time period to try to capture more samples and rerun 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. The Configuration Advisor selects the **Update** option for each rule. 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:

1. 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 **Weekdays** and **Weekends** monitoring plans:

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:	Update	Select:	Update	
Process CPU High	3.1 [0]* percent	All	<input checked="" type="checkbox"/>	All	<input checked="" type="checkbox"/>	
Process Resident Memory High	3256.6 [0]* kilobytes	None	<input checked="" type="checkbox"/>	None	<input checked="" type="checkbox"/>	
Process Virtual Memory High	6650.5 [0]* kilobytes		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	
Queued Request Percent High	4.0 [0]* Percent queued		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	
Rejected Request Percent High	4.0 [0]* Percent rejected		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	

2. Click **Recommended Values** to display the range of recommended values for the associated rule.
3. 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 as you compare the various values. See the [“Additional selection criteria”](#) section on page 7–14 for details.
4. Repeat [Step 1](#) through [Step 3](#) for each rule and its associated monitoring plan. If you know that you are going to select 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.
- Your knowledge of the system's operational needs and goals.

Submitting your threshold setting selections

When you click **Update Selected Rules**, OpenEdge Management 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, display 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 should further refine your threshold settings.

Analyzing OpenEdge Application Performance

This chapter describes how you can use OpenEdge Management to analyze OpenEdge server application performance, as detailed in the following sections:

- [Overview](#)
- [Investigating application performance issues](#)
- [OpenEdge Management in the workplace](#)
- [Planning an application performance review](#)
- [Responding to an application crisis](#)
- [For more 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 correct 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.

The following sections describe:

- [Investigating application performance issues.](#)

This section provides a model for administrators to use.

- [OpenEdge Management 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 proactive work practices using OpenEdge Management reports help to uncover clues about application performance changes and degradation.

- [Responding to an application crisis.](#)

This sample scenario highlights how the use of various OpenEdge Management features can help administrators quickly analyze and respond to a system or application problem.

Note: These scenarios are intentionally limited in scope. They are provided to help you understand some of the general principles by which OpenEdge Management 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 OpenEdge 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. With this fundamental understanding, you can use OpenEdge Management-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 concise manner.

- Reviewing OpenEdge Management-supplied data to investigate and analyze your problem or goal. Use your problem definition to review OpenEdge Management-generated information to better understand your problem. Through a process of elimination, you can evaluate the data and identify components that can potentially contribute 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.

OpenEdge Management in the workplace

This section describes the process of investigating application performance issues using a fictitious company, XYZ Corporation. At XYZ, the administrator has installed and configured OpenEdge Management.

OpenEdge Management at XYZ Corporation

The XYZ Corporation's system administrator has customized his OpenEdge Management resource monitoring capabilities and frequently consults the system's data as monitored by OpenEdge Management. 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, in this case the **Performance** and **Profile** reports.
- Establishes 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. The administrator 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.
- Sets up the AppServer brokers and servers on the **My Collections Home** page, along with other vital system operations such as memory and CPU consumption, so the data can be quickly referenced. Among other standard viewlets, he displays **resources running with alerts, active monitoring plans, and running reports viewlet options**. The administrator also monitors all those viewlets related to the AppServer brokers.
- Reviews the **System Activity** report frequently throughout the work day as it displays real-time system performance and resource usage details.
- Reviews the **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.

The administrator regularly reviews these pieces of data as they can provide clues about the system's application performance.

Consulting OpenEdge Management documentation

This administrator also frequently references the information in the OpenEdge Management documentation set and context-sensitive online help.

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 the application's performance. With a high volume of data entry taking place between 9 AM and 6 PM on the system, and most of the procedures distributed and run remotely on an AppServer, the users expect a consistently high level of application performance and availability. The administrator has learned over time how to deliver system availability that is consistent with this goal, and has come to learn that the 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, other technological elements might be considered, but these four remain of primary concern. The administrator is most concerned with OpenEdge Management performance indicators that relate to these elements so as to take action on any potential performance issue before it affects the 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 is a bit of a mystery. The administrator can consult the 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 is of concern because any indication of a performance weakness could become a real performance problem if the administrator does not determine the source of the problem as soon as possible.

Initial investigation

The first question the administrator asks is: “What’s changed in my production environment that is causing poor performance?” To begin solving this performance problem, the administrator starts to list the possibilities, as shown in [Table 8–1](#). Note the blank, first column in the table. As each possibility is reviewed, the administrator can use this table as a checklist to identify the items requiring further consideration.

Table 8–1: Initial investigative checklist

✓	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 period during which problems have been noticed and reported?
	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 payoff in performance if it is found to be a contributing factor.
	Data from the OpenEdge Management Trend Database from the troublesome time period.	By running reports at different time periods, is it possible to see any patterns in the data or reported application responsiveness that match experiences that the users have reported?

Drilling deeper into OpenEdge Management-supplied data

In addition to the considerations noted in [Table 8–1](#), the system administrator reviews the data gathered weekly through the AppServer Application Profile report. When the company installed and started to use OpenEdge Management, they began using the predefined report template feature to run report instances on a weekly basis. This report’s data provides the administrator with a high-level picture of the application’s health. The *OpenEdge Management: Reporting Guide* helps to set up and use the AppServer Profile report and all the other OpenEdge Management-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 the AppServer. Reviewing and routinely comparing reports from different time periods provides this administrator more insight into the AppServer’s performance.

Finding performance-related clues in the AppServer Profile report

The administrator knows that reviewing performance details about two of the ABL procedures might provide performance clues. Performance issues related to these high-level rate procedures—**zeta.p** and **zed.p**—might impact the application’s performance.

Figure 8–1 displays typical AppServer workload-related data that is consistent with an average weekday afternoon at the XYZ Corporation.

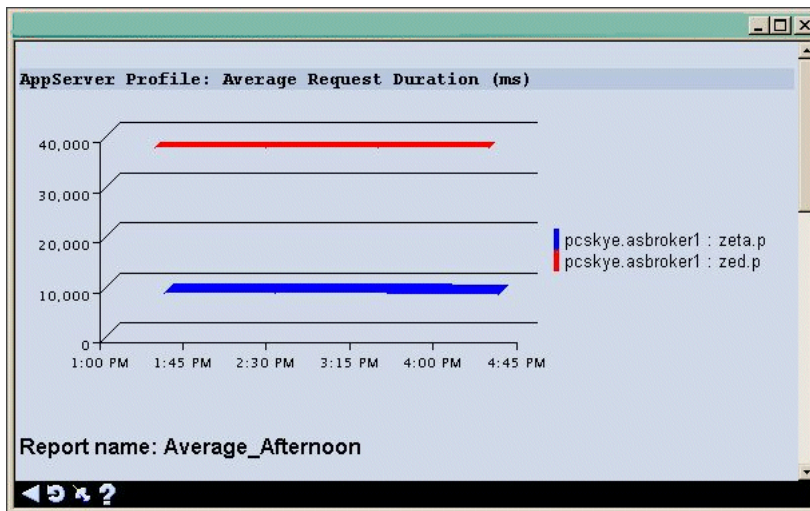


Figure 8–1: AppServer Profile Report for Average_Afternoon data

Note: In the figures presented in this section, the colors in the graphs are intended only 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 ABL 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 6–17. 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.

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 the users’ and the company’s business needs are being well met.

The administrator compares the report data results from previous weeks to the data results that appear in Figure 8-3. 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 to investigate.

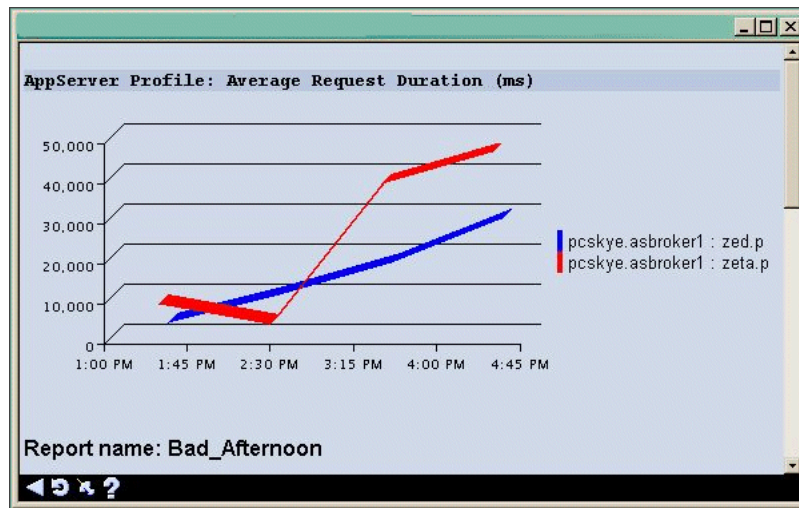


Figure 8-2: AppServer Profile Report for Bad_Afternoon data

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, the 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 rebalance the work load.

The 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 without diligence in routine review and investigation of OpenEdge Management report data.

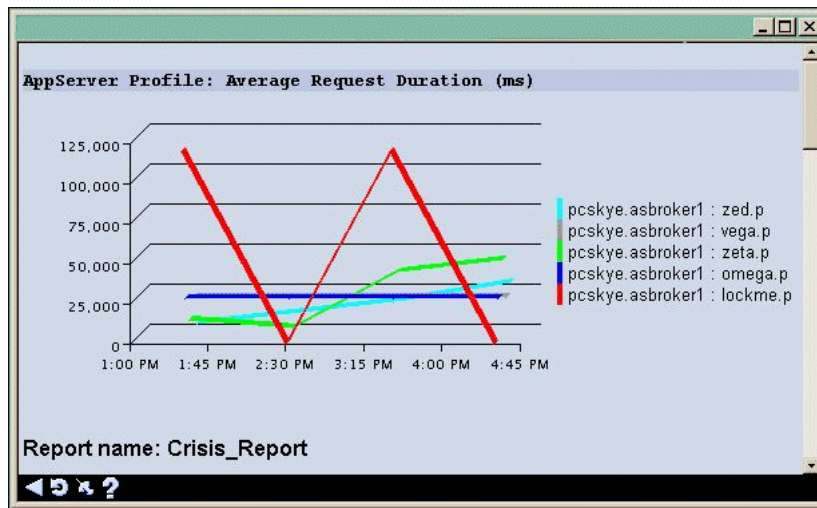


Figure 8–3: AppServer Profile Report for Crisis_Report data

Assuming the same 40,000 threshold for all of the procedures listed in Figure 8–3, it is very apparent that processing on 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 causing more problems 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 well-seasoned administrator who keeps a log of application and system problems, and consistently records the actions to correct difficulties.

To monitor this particular situation to ensure that the problem has been resolved satisfactorily, the administrator must:

- Work with the company's application group to ensure that they receive the time and records needed to address the application's performance problem.
- Monitor the impact of the fix closely to ensure that it did correct the problem and did not introduce any other application or system difficulties.
- Interview the application users to ensure that they experience an improvement in their application throughput.
- Document the problem and the efforts to correct the problem so that the information 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 offered in OpenEdge Management, a system administrator can arm himself with some fundamental informational tools. These tools help provide immediate data that is useful in understanding and addressing a crisis.

This section describes another problem that the system administrator from XYZ Corporation must face.

Note: The OpenEdge Management features outlined in the “[OpenEdge 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 running fine. Response time was good, and the users were very pleased.

Unexpectedly, the system’s performance began to decline rapidly. The system administrator began receiving 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 were failing.

Initial investigation

In an application crisis situation of this type, the administrator can leverage OpenEdge Management-supplied information to alert him to immediate problems and provide data related to the crisis.

Table 8–2 lists the possibilities the system administrator considers. Note the blank, first column in the table. As each possibility is reviewed, the administrator can use this table as a checklist, identifying the items requiring further consideration.

Table 8–2: Crisis review checklist

✓	Access and review . . .	To . . .
	Alert and other data indicators that have been set up to monitor and display data on the collections page.	Quickly examine issues that might be the reason for this dramatic change in performance. As previously noted in the “ OpenEdge Management at XYZ Corporation ” section on page 8–4, the administrator has several indicators set up, including viewlets related to the 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.
	Network- and server-related data details, using TCP resource monitors 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 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 remembered what the users said about the performance issue: Nothing was working. This could indicate there is a network problem to resolve, but where is the source? Since most of the transactions related to the procedures that were not currently functioning run on a remote AppServer, the administrator decides to follow this investigative path.

Drilling deeper into OpenEdge Management-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, the administrator needs to know that the information available to determine, resolve, and learn from the problem situation to minimize—if not eliminate—such a crisis of this kind from reoccurring is accurate and timely.

Accessing and examining AppServer data

The administrator accesses the OpenEdge resources in the OpenEdge Management console, browsing to the AppServer resources. The network uses only one AppServer, thus the administrator 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 the collection page and also on individual resources displayed in the list frame, the administrator notices that there are no new alerts displayed.

The administrator then accesses the **Database** page and scans for relevant information in the **Operational views** and **Informational views** sections. Finding no clues related to the issues, the **Server Performance View** details are displayed next. The server state and server pool summary details that display in this view, however, are not helpful. In this situation, the administrator considers where the most valuable information would be found, and 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 intended only to distinguish one data element from another.

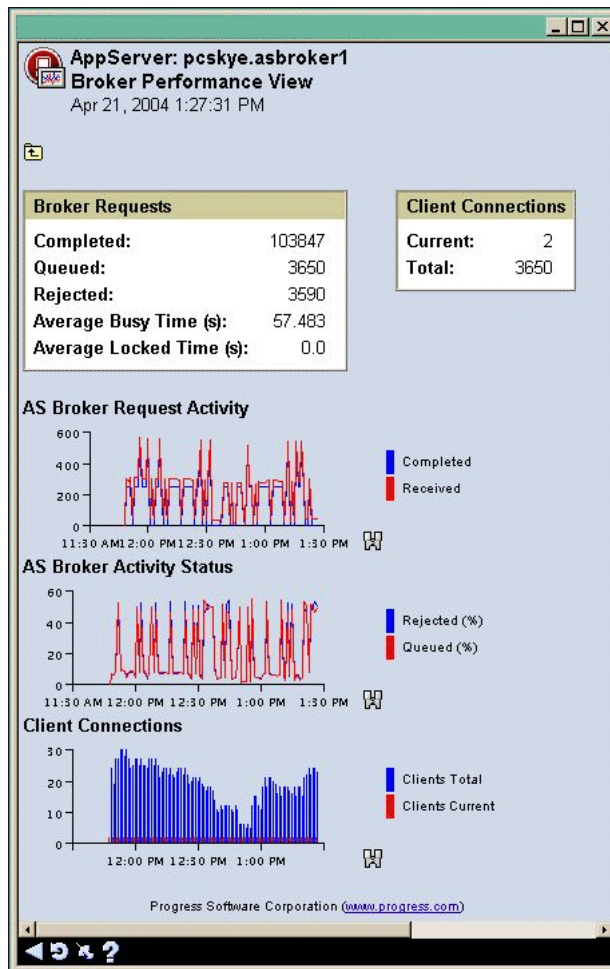


Figure 8–4: Broker Performance View for asbroker1

The administrator scans the summarized data in **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 is a problem in this area, but still needs to do more research. From the previous use of the data on the **Broker Performance View** page, the administrator knows that the **AS Broker Activity Status** graph is a representation of the **Queued** and **Rejected** values noted in **Broker Requests**.

The administrator 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](#).

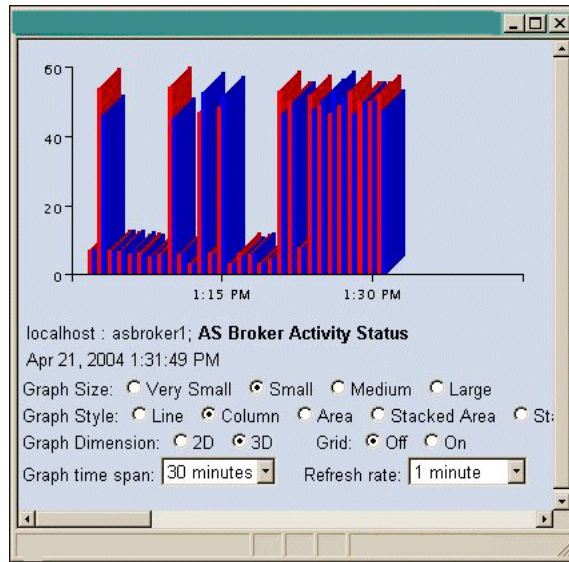


Figure 8-5: AS Broker Activity Status for asbroker1

The pinup graph in [Figure 8-5](#) focuses on 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 4–17.

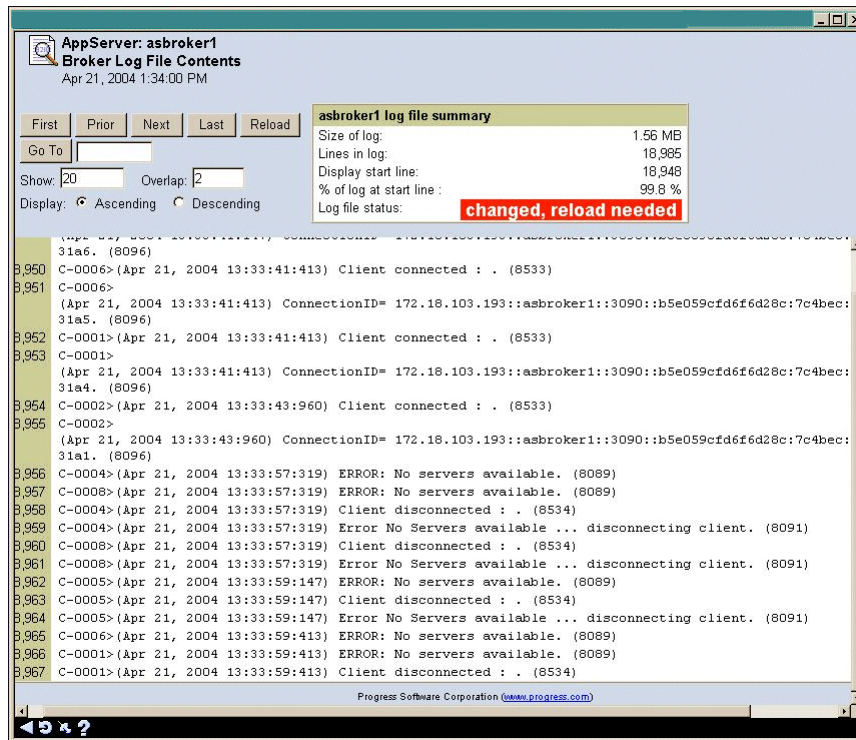


Figure 8–6: AppServer asbroker1 log file

At approximately the same time that 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.

The administrator redisplay the **Servers Performance View** page. All the investigative activities have confirmed that a runaway AppServer process has brought down the network, leaving the users unable to perform their application transaction-related tasks.

Figure 8–7 shows the suspicious data in the **CPU Use** column, indicating that no CPU consumption is occurring for the servers.

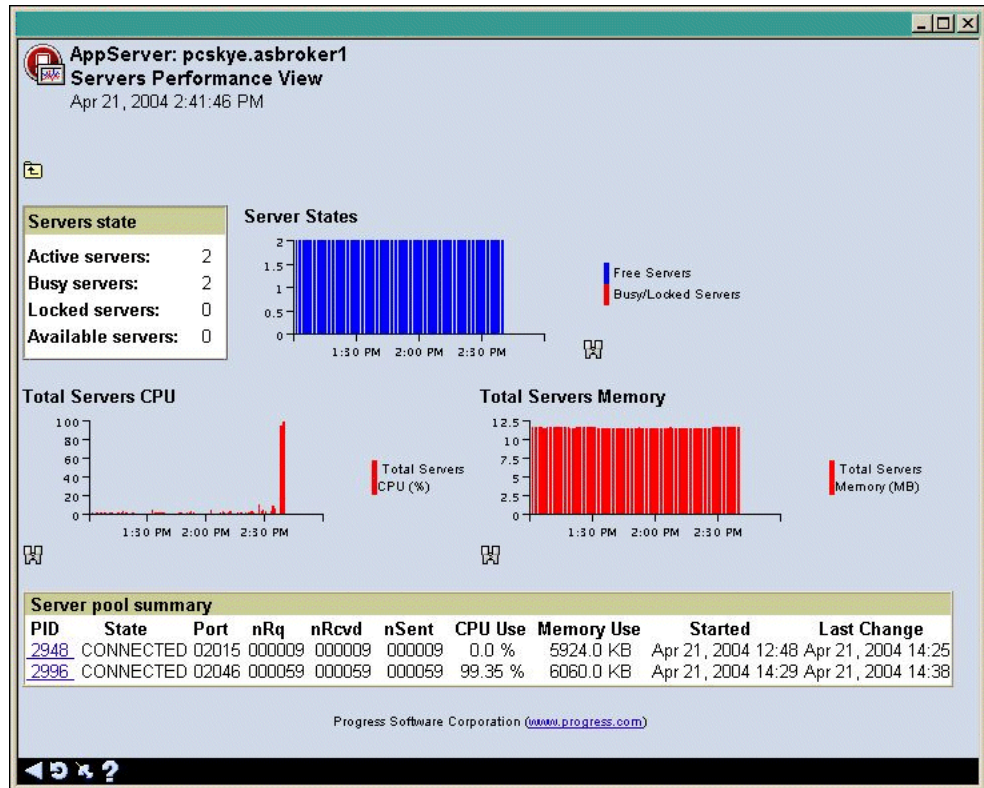


Figure 8–7: Servers Performance View page for asbroker1

Again, by clicking the binocular icon, the administrator can display this data in a pinup, as shown in Figure 8–8.

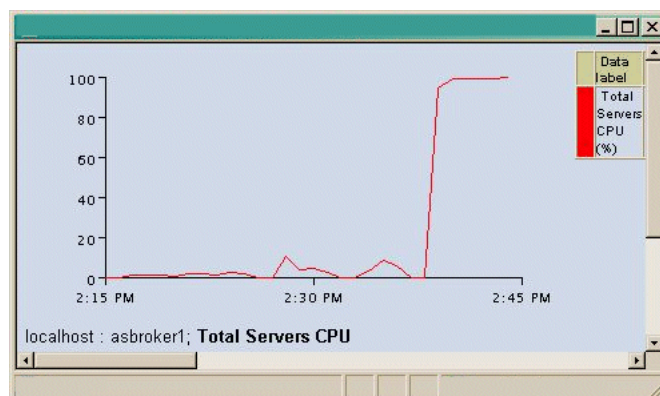


Figure 8–8: Total Servers CPU for asbroker1

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, the administrator can terminate this process, ending the network and application difficulties.

Testing and documenting your potential solutions

The administrator puts two plans in place to monitor this particular situation.

Adding new OpenEdge Management monitoring plans

The administrator determines there are a few additional setup options and controls to consider implementing. Using OpenEdge Management, the administrator can:

- Add a monitoring plan and rule for the CPU on the asbroker process so that the system will alert the administrator should processing not go according to expectations.
- Add a system level CPU monitor and associated rule also to alert the administrator 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 can:

- Look at a larger historical time period in OpenEdge Management using reports. Report data might show other instances in which there was a runaway process and what activities occurred to correct the problem.
- Review what has changed on the system to determine if a recent change has caused the issue.
- Check the issues and answers available in the KnowledgeBase (KBase) section of the Knowledge Center available by accessing:

<http://www.progress.com>

- Document the problem and the efforts to correct it so that the information will be available for future reference.

For more information

The application performance topic is a large one. [Table 8–3](#) identifies more information within the OpenEdge Management documentation set that complements and more fully explores performance-related topics.

Table 8–3: Additional OpenEdge Management 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 OpenEdge Management 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 OpenEdge Management.	<i>OpenEdge Revealed: Mastering the OpenEdge Database with Fathom Management</i> by Adam Backman.

Index

A

AdminServer warm start
 effects on OpenEdge Management 2–13,
 2–15
 implications 2–14
 stages 2–13

Agent Unavailable rule 6–19

Application crisis
 responding to 8–11
 review checklist 8–12

Application performance
 AppServer Operational views 8–13
 AppServer Profile report 8–7
 business requirements 8–3
 consulting documentation 8–4, 8–19
 crisis example 8–5
 planned review 8–5

Application performance issues
 at a fictitious company 8–3
 checklists 8–6, 8–12

Application review
 planned response 8–5

AppServer
 adding or trimming 4–14
 AppServer Operational views
 overview 4–25
 Broker Control page content 4–5
 Broker Performance View 4–26
 broker status details 4–2
 changing broker controls 4–6
 Informational views 4–30
 killing an AppServer process 4–15
 killing, a process 4–13
 modifying control settings 4–4
 Operational views
 troubleshooting 8–13
 overview 4–2
 See AppServer broker log file monitor
 See AppServer server log file monitor
 Server Performance View 4–28
 Server Pool Control page content 4–11
 viewing broker process details 4–9

AppServer broker log file monitor
 characteristics 4–18
 considerations 4–20
 customizing 4–21
 default values 4–19
 log file data 4–17
 naming conventions 4–17
 See AppServer log file viewer

AppServer log file viewer
troubleshooting 8–16
using 4–23

AppServer Profile report
troubleshooting 8–7

AppServer server log file monitor
characteristics 4–18
considerations 4–20
customizing 4–21
default values 4–19
log file data 4–17
naming conventions 4–17
See AppServer log file viewer

Average Procedure Duration High rule
6–17, 6–19

B

Baseline value *See* Configuration Advisor

C

Collection views
defining viewlets 2–17

Configuration Advisor
and data collection 7–3
effective selections 7–14
evaluating 7–11
overview 7–2
process details
 a rule definition 7–5
 data analysis 7–5
 recommendations 7–5
 user-supplied criteria 7–3
rules
 additional considerations 7–3
 WebSpeed and AppServer 7–2
setting rules-related criteria 7–7
submitting selections 7–14
understanding recommendations 7–10
using the Detail button 7–12

D

Data collection
and AppServer brokers 4–8
and Configuration Advisor 7–8
and trending data 3–8, 4–8
and WebSpeed brokers 3–8
fields related to
 Broker statistics available 3–8, 4–8
 Collect Statistics 3–8, 4–8
 Trend Performance Data 3–8, 4–8
options and conditions
 AppServer brokers 4–8
 WebSpeed brokers 3–8
setting for AppServer 4–8
setting for WebSpeed 3–8

Default monitoring plan details 6–7
examples 6–8

Default polling and trend values
reviewing 6–4

Detail menu bar 2–3

G

Graphs 2–15

I

Implementing business requirements
examples 8–4

M

Main menu bar 2–2
performing server-related tasks 2–4

Menu bars
detail menu 2–3
main menu 2–2

Monitoring plans
maintaining 6–9

N

NameServer
 changing controls for 5–5
 Control page content 5–4
 modifying control settings for 5–3
 overview 5–2
 status details 5–2

NameServer Informational views
 Properties details 5–18

NameServer log file monitor
 characteristics 5–7
 considerations 5–8
 default values 5–7
 how to customize 5–9
 log file data 5–6
 See NameServer log file viewer

NameServer log file viewer
 using 5–12

NameServer Operational views
 Status details 5–14

O

OpenEdge main page 2–6
 format and content 2–6
 polling and statistical details 2–8
 section details 2–7

OpenEdge resources
 deleting 2–11
 in list frame 2–9

OpenEdge servers 1–2
 features 1–4
 overview 1–2
 prerequisites 1–5
 using the main menu bar 2–4

P

Pinup charts 2–19

Q

Queued Request Percent High rule 6–19

R

Rejected Request Percent High rule 6–18

Resource monitoring
 overview 6–2
 terminology 6–2

Rule details
 AppServer rules 6–16
 common rule characteristics 6–15
 general rule conventions 6–13
 NameServer rules 6–15
 WebSpeed rules 6–16

Rule set
 and monitoring plans 6–25
 characteristics and benefits 6–21
 copying 6–24
 creating 6–22
 deleting 6–24
 editing 6–23
 evaluating when rules in common 6–24
 overview 6–21

S

Schedules
 Default Schedule 3–21, 4–21, 5–10

Server Unavailable rule 6–19

T

Transaction Server
 See WebSpeed Transaction Server

U

ubroker properties file 1–3

W

WebSpeed log file monitor

- characteristics 3–18
- considerations 3–20
- customizing 3–20
- default values 3–19
- log file data 3–17
- See* WebSpeed log file viewer

WebSpeed log file viewer

- considerations 3–25, 4–24, 5–13
- See* WebSpeed log file monitor
- using 3–23

WebSpeed Transaction Server

- adding or trimming agents 3–14

Agent Performance View 3–29

Agent Pool Control page 3–11

Broker Control page 3–5

Broker Performance View 3–26

changing WebSpeed broker controls 3–6

killing a WebSpeed agent process 3–13, 3–16

modifying WebSpeed control settings 3–4

naming conventions 3–18

overview 3–2

See WebSpeed log file monitor

viewing broker process details 3–9

WebSpeed broker status details 3–2

WebSpeed Informational views 3–30

WebSpeed Operational views overview 3–25